Informative Data Fusion: Beyond Canonical Correlation Analysis

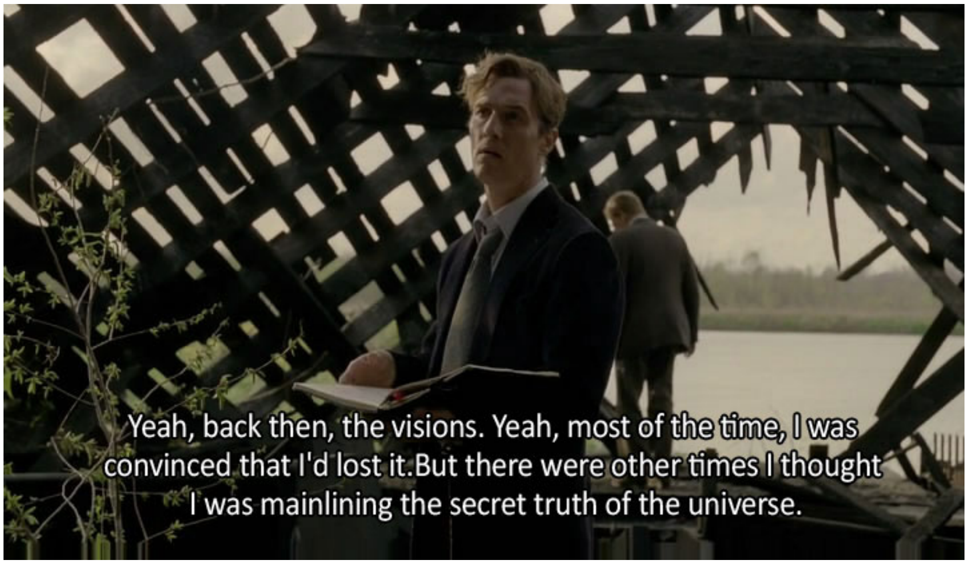
by

Nicholas A. Asendorf

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2015

Doctoral Committee:

Assistant Professor Raj Rao Nadakuditi, Chair
Assistant Professor Laura Balzano
Professor Alfred O. Hero III
Associate Professor Rada Mihalcea



Yeah, back then, the visions. Yeah, most of the time, I was convinced that I'd lost it.But there were other times I thought I was mainlining the secret truth of the universe.

For my grandmothers Josephine Asendorf and Doris Evashko, in memoriam.

ACKNOWLEDGEMENTS

I would first like to thank my adviser Prof. Raj Nadakuditi. Thank you for allowing me to be one of your very first students. Your ideas and passion for research truly do inspire. Thank you for teaching me how to pick the low hanging fruit, how to see the strucutre in complicated problems, and how to always "push it through".

I would specifically like to thank my committee members for their inspiring conversations and courses over the past 5 years. Prof. Rada Mihalcea, thank you for your information retrieval course. I enjoyed chatting with you about the possible applications of correlation analysis to such problems. Prof. Laura Balzano, thank you for coming to the University of Michigan; I consider myself lucky that your arrival was timed so perfectly with my dissertation. You have been a wonderful addition to EE:S and I am excited to learn from your work in the years to come. Prof. Alfred Hero, thank you for our numerous conversations over the years at conferences and MURI funding meetings. Your insights always helped to deepen my understanding of the problems that we solve. I would also like to thank my undergraduate research advisers at the University of Maryland, Prof. Jonathan Simon and Prof. Adam Hsieh, for first introducing me to the research process.

Thank you to all the members of the Nadakuditi group - Curtis Jin, Raj Tejas Suryaprakash, Himanshu Nayar, Brian Moore, Arvind Prasadan, and David Hiskens. Specifically, I would like to thank Raj Tejas for his help with the analysis of missing data and Brian Moore for his wonderfully elegant hacks of MATLAB.

Thank you to all the members of the Graduate Student Council and SPeecs organizing committee. Specifically I would like to thank Kevin Xu, Michael Allison, and Mads Almassalkhi for their leadership and passing down the needed graduate student lore. Thank you also to the advanced technology group at 3M. I thoroughly enjoyed my internship last summer and am looking forward to working with you all over the next years. I would also like to thank the National Basketball Association for providing an escape from research.

To my high school friends Alex McArthur, John Vaccacio, Allison Seyler, Andy Mellon, Matt Keel, Alec Brown, Chris Rowe, Mike Hyle, and Jason Pribble: thank you for always being there to remind me of my roots. While I have only sporadically been able to spend time with you in person over the last five years, I valued every minute of it. Alex - thank you for teaching me to always trust your decisions. Allison, Alec, Andy, Chris and Jason - thank you for humoring me by listening to my academic ramblings. Matt, Mike, and John - thank you for always keeping me guessing.

To my friends from the University of Maryland, Jaime Gomez, Austin Myers, Sarah Saslow, Kyle Smith, Andy Peters, and Nick Gagliolo: thank you for always being there and for appreciating the alma mater as much as I do - “steadfast in loyalty”. Jaime - I will never forget all of the memories at Courtyards. Thank you for teaching me how to play tennis, bridge, and spades. Thank you for teaching me about nuclear physics and for always providing logical advice when I needed it. Sarah - thank you for always providing a listening ear when I needed it. Thank you for all of the football weekends as we watched Michigan and Northwestern in graduate school. Kyle - thank you for all of the late night card games and video games, for teaching me about fire protection engineering, and all of the weekends that we were able to see each other since graduation. Austin - thank you for being an awesome roomate and colleague, for your goofy sense of humor, and all of the conversations that we had. Nick - thank you for keeping me sane through all of the courses, group projects, exams, and gemstone activities. Andy - thank you for all of the sports conversations and going to all of the Maryland games with me over the years.

To my friends and colleagues at the University of Michigan, Rob Vandermeulen, Madison McGaffin, Mitchell Thomas Hellman Young, Matthew Prelee, Steve Schmitt, Paul Ozog, Eric Uthoff, Pat O’Keefe, and Mike Henry: thank you for making graduate school fun. Paul, Eric, and Pat - thank you for all the first and second year hang outs that got us through our coursework. Steve - thank you for all the walk-and-talks, letting me borrow your sitting ball, and teaching me about the midwest. Rob - thank you for introducing me to your many hobbies; you have really broadened my world view. Matt - thank you for all of the walk-and-talks, football trips, sports arguments, board game nights, and long runs. Mike - thank you for being really good at trivia and the many fun nights at your place. Mitch - thank you for teaching me how to brew, being a Linux whisperer, teaching me about flux, and for the HoM. Madison - thank you for all of the walk-and-talks, entertaining my OCD Pathfinder ideas, teaching me how to make bread, being a word-o-mancer, and all of the memories contained at HoM.

To my girlfriend Christie VanTongeren - thank you for all of your support over the past three years. Thank you for your understanding for all of the late nights over the past months and always listening. Thank you teaching me all of your expert cooking techniques and for all of the movie, bowling, game, and trivia nights. Thank you for sharing my values but always challenging me to be better. I am so excited for Minneapolis.

I would like to thank all of the EECS support staff who made my job easier over the years at the University of Michigan. Special thanks go to our graduate coordinators Becky Turanski, Beth Stalnaker, and Rachel Young Antoun. Thank you for always being there to answer questions, keeping me on track, and brightening up my day. Thank you also to our wonderful 4th floor staff Beth Lawson, Anne Pace, and Shelly Feldkamp. You make our jobs so easy by hiding the many complicated intricacies of the funding process. I also greatly appreciated your kindness and support over the last four years. It was always a pleasure to catch up with you.

I would like to thank my piano teacher of eight years, Shirley Elsroad, and organ teacher Lee Pogue. Thank you for teaching the joy of the piano and organ and teaching me to express myself though them in my own voice. I will always fondly remember my lessons with the both of you. I would also like to thank Prof. Steven Ball for teaching me how to play the carillons at the University of Michigan. Thank you for teaching me how to perfume the Ann Arbor air with music.

Finally, I would like to thank my mother Nancy, father David, and sister Amy. I would not be where I am today without your support, encouragement, and amazing parenting. I know that not everyone is as lucky as I was to be born into such a supportive enviornment and for that, words cannot describe my gratitude. This thesis is a testament to your parenting skills.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Informative Data Fusion:
Beyond Canonical Correlation Analysis
by
Nicholas A. Asendorf

Chair: Raj Rao Nadakuditi

Multi-modal data fusion is a challenging but common problem arising in fields such as economics, statistical signal processing, medical imaging, and machine learning. In such applications, we have access to multiple datasets that use different data modalities to describe some system feature. Canonical correlation analysis (CCA) is a multidimensional joint dimensionality reduction algorithm for exactly two datasets. CCA finds a linear transformation for each feature vector set such that the correlation between the two transformed feature sets is maximized. These linear transformations are easily found by solving the SVD of a matrix that only involves the covariance and cross-covariance matrices of the feature vector sets. When these covariance matrices are unknown, an empirical version of CCA substitutes sample covariance estimates formed from training data. However, when the number of training samples is less than the combined dimension of the datasets, CCA fails to reliably detect correlation between the datasets.

This thesis explores the the problem of detecting correlations from data modeled by the ubiquitous signal-plus noise data model. We present a modification to CCA, which we call informative CCA (ICCA) that first projects each dataset onto a low-dimensional informative signal subspace. We verify the superior performance of ICCA on real-world datasets and argue the optimality of trim-then-fuse over fuse-then-trim correlation analysis strategies. We provide a significance test for the correlations returned by ICCA and derive improved estimates of the population canonical vectors using insights from random matrix theory. We then extend the analysis of CCA to regularized CCA (RCCA) and demonstrate that setting the regularization parameter to infinity results in the best performance and has the same solution as taking the SVD of the cross-covariance matrix of the two datasets. Finally, we apply the ideas learned from ICCA to multiset CCA (MCCA), which analyzes correlations for more than two datasets. There are multiple formulations of multiset CCA (MCCA), each using a different combination of objective function and constraint function to describe a notion of multiset correlation. We consider MAXVAR, provide an informative version of the algorithm, which we call informative MCCA (IMCCA), and demonstrate its superiority on a real-world dataset.

CHAPTER I

Introduction

Multi-modal data fusion is a ubiquitous problem in signal processing and machine learning. In many applications, we have access to multiple datasets, possibly of different modalities, each of which describe some feature of the system. This setup is becoming increasingly common today as data collection becomes cheaper and easier. We are no longer limited by the amount or variety of data that we can collect, but instead by how quickly and accurately we can process such a wide variety of data.

The underlying assumption in such settings is that each dataset contains signals that are correlated with signals of the other datasets. Correlation analysis algorithms hope to leverage this fact to extrude these correlated signals *jointly* from the datasets more accurately than from the individual datasets alone. Of course, every application has a different goal. Sometimes we want to detect the presence of the correlated signals. Other times, we may wish to predict one modality from the other. In other applications, we may desire to classify or cluster observations. Despite the differing objectives, all of these applications rely on the ability to accurately detect and extract the correlated signals between the datasets. This thesis focuses on developing theoretically justified, robust correlations analysis algorithms to use as a pre-processing step before learning algorithms that perform data fusion, as motivated by Figure 1.1.

1.1 Canonical Correlation Analysis (CCA)

1.1.1 What is it? What is it not?

Canonical correlation analysis (CCA) is a joint dimensionality reduction algorithm for exactly two datasets that finds a linear transformation for each dataset such that the correlation between the two transformed feature sets is maximized [4]. CCA, however, *is not* a data fusion algorithm. CCA returns two linear transformations and

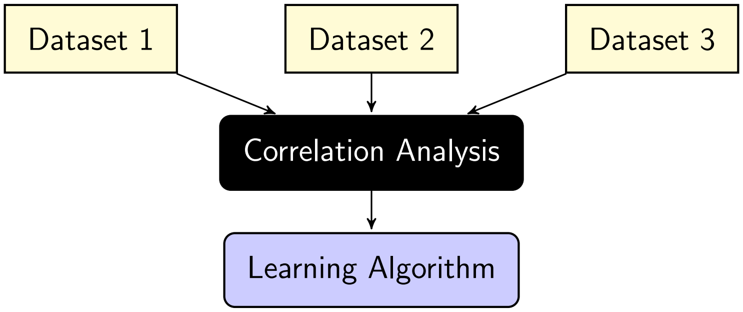


Figure 1.1: Illustration of multi-modal data fusion

a set of correlations. In this light, CCA is extremely similar to principle component analysis (PCA), which returns a linear transformation that accounts for the directions of largest possible variance in a dataset. These principle components are typically used as features vectors in a variety of machine learning algorithms. Just as PCA is a dimensionality reduction algorithm and not the final machine learning algorithm that uses the principle components, CCA is a joint dimensionality reduction algorithm whose dimensionality reduction ensures that datasets are maximally correlated in their reduced spaces. These maximally correlated features may then be used however a learning algorithm desires.

The solution to CCA is easily found by solving a quadratic optimization problem. This solution is a closed form expression relying on the singular value decomposition (SVD) of a matrix product involving the covariance matrices of each dataset and the cross-covariance between the two datasets. As these covariance matrices are rarely known *a priori*, practical uses of CCA rely on substituting sample covariance matrices formed from training data, which we call empirical CCA.

The performance of empirical CCA has been studied previously, but insufficiently. When the number of training samples is large compared to the dimensions of the datasets, the performance is well understood [5]. When the number of training samples is less than the sum of the dimension of each dataset (sample deficient regime), [6] proves that empirical CCA completely breaks down and always reports a perfect correlation between the datasets.

This extremely undesirable characteristic of empirical CCA has lead many to abandon CCA as a reliable statistical analysis technique. Pezeshki, L.L. Scharf et al. argue that in this sample deficient regime

> ... the empirical canonical correlations are defective and may not be used as estimates of canonical correlations between random variables.[6]

Similarly, Ge et al. conclude that

> ... CCA provide(s) reliable information about spatial correlations existing among pairs of data sets only when SNRs ... are reasonably high, and the sample support is significantly larger than the data dimensions.[7]

1.1.2 Variations on CCA

Due to this undesirable breakdown of CCA in the low-sample high-dimensionality regime, many researchers proposed variations of CCA to avoid this performance loss. Most notably, [8] used recent results from random matrix theory to demonstrate that this performance breakdown may be avoided by trimming the sample covariance matrix estimates to only include informative components. This algorithm is the crux of this thesis. We will study its performance and develop theoretical tools in order to use it for real-world applications. Throughout the thesis, we use the ubiquitous low-rank signal-plus-noise model for datasets

$$X = UV^H + Z,$$

where $X = [x_1, \dots, x_n]$ is our observed data matrix whose columns are individual multidimensional observations, U is a low-rank signal subspace, V is a low-rank signal matrix, and Z is a noise matrix. Surprisingly, correlation analysis for this classical low-rank signal-plus-noise model is not completely studied. This thesis seeks to complete the discussion. Here, we briefly touch on other variations based on CCA that do not assume the above linear low-rank signal-plus-noise model. Many of these algorithms are tuned for a specific application or seek to avoid the performance loss of CCA in a certain regime.

Regularized CCA (RCCA) [9] adds a penalty term to the magnitude of the canonical vectors. This results in adding a scaled copy of the identity matrix to the sample covariance matrix of each dataset, which allows each matrix to be inverted. Therefore, RCCA returns non-trivial results in the sample-deficient regime. However, this approach introduces a parameter to the algorithm; the effect of this parameter is not well studied. Other variations of RCCA, such as supervised RCCA [10], fast RCCA [11], and a multi-block RCCA [12], have also been proposed.

Kernel CCA (KCCA) [13] was proposed to deal with non-linear correlations existing between datasets. However, KCCA also introduces regularization parameters so as to not return trivial solutions (see [14] for an excellent derivation). Besides the choice of regularization parameter, there is also ambiguity in the choice of the kernel function, which is a common problem among kernel methods. Other variations of KCCA have also been proposed, such as penalized KCCA [15], alpha-beta divergence [16], and CCA based on kernel target alignment [17].

Sparse CCA [18] finds linear transformations such that the number of features used is minimized. This problem is often motivated by the need for interpretable canonical vectors that is often driven by the application, such as in brain imaging [19]. There are many variations on sparse CCA, typically motivated by application or mathematical intrigue. Sun and Keates [20] explore CCA in the context of censoring, Shin and Lee examine sparse functional data [21], Tao et al. consider joint sparse data in [22], Gao et al. explore efficient sparse CCA for high-dimensional data [23], and Zhang et al. extend the analysis to multi-class group sparse CCA [24]. Other formulations include a penalized decomposition [25], Bayesian CCA via group sparsity [26], and recursive sparse CCA [27].

1.1.3 Applications

CCA and its variants are widely used in a variety of fields where multiple datasets naturally arise, the most common of which is machine learning and computer vision. In [28], CCA is used to learn semantics of multimedia content by fusing image and text data. Related, [29] uses CCA to learn word embeddings for supervised natural language processing tasks. CCA has been widely applied to pose estimation [30, 31], as this is a natural examples where we have multiple views (image) of the same object. Other computer vision related tasks where correlation methods are natural fits include matching people across cameras [32], clustering social event images [33], automatic image annotation [34], and audio-visual speaker clustering [35].

Medical analysis is another field where there are ripe opportunities for correlation analysis due to the vast number of modalities (EEG, MRI, CT, fMRI, MEG, etc.). CCA is often used to determine interactions, or connectivities, between brain areas in fMRI data [36, 37, 38, 39] and used to fuse fMRI, sMRI, and EEG data [40]. CCA based methods have also been used to examine genetic connections [41, 42, 43], relying heavily on sparse methods due to the high dimensionality of gene data and need to interpret which genes are “on”. CCA is also a popular way to detect frequencies in steady-state visual evoked potential (SSVEP) in brain-computer interfaces (BCIs) [44, 45, 46]. Still further, CCA is used in de-noising and analysis of EEG, MEG and ECG data [47, 48, 49, 50].

CCA also has roots in classical signal processing applications. The authors of [51] apply CCA to to the common communications problem of blind equalization of single-input multiple-output (SIMO) channels. Pezeshki et al. [52] showed that the CCA coordinates are the correct coordinates for low-rank Gauss-Gauss detection and estimation. Scharf and Thomas [53] provide a wonderful exposition on using the canonical coordinates for Wiener filters, transform coding, filtering, and quantizing. CCA and multiset CCA have been used to achieve joint blind source separation (BSS) in [54]. CCA has also been applied to hyperspectral imaging [55], array processing [7], Gaussian channel capacity [56], and cognitive radio networks [57].

Other fun and interesting applications include climatology, finance, and music. Todros and Hero define a new measure transformed based CCA and show its utility on financial data in [58]. Torres et al. [59] use sparse CCA to label portions of musical songs with meaningful words or phrases. In the field of climatology, CCA has been used to study sea temperatures [60], forest planning [61], and tropical cyclones [62]. Finally, I would be remiss if I didn't share my personal favorite application of CCA to date: using CCA to analyze bovine growth [63].

1.2 Contributions of this thesis

In many of the application presented above, researchers either have access to many samples, or have designed an algorithm tuned specifically for a their particular application. This thesis considers the performance of empirical correlation algorithms in the low-sample, high dimensional setting. These algorithms are not geared toward a particular application but are general and may be applied to any application. We will demonstrate, both theoretically and empirically, that multi-modal correlation analysis in this regime is a possibility. We remark that statements labeled as theorems represent, to the best of our knowledge, new results while important results from literature are labeled as propositions or lemmas. Chapters II-III are self contained and may be read independently. Chapters IV-X consider the problem of correlation analysis.

Chapters II and III consider the classical problem of matched subspace detectors. We use insights from random matrix theory on the accuracy of subspace estimates to derive new, optimal detectors that demonstrate the sub-optimality of the classical plug-in detectors that simply substitute maximum likelihood estimates for unknown parameters. Under both a stochastic and deterministic data model, we argue that only the *informative* subspace components should be used in a detector. We extend this analysis to the case where our observations may contain missing data.

In Chapters IV and V, we explore the performance of CCA and re-derive informative CCA (ICCA). We demonstrate the extreme sub-optimality of CCA in the low-sample, high dimensionality regime. Specifically, we provide a statistical test for both CCA and ICCA that determines whether the correlations returned by the algorithms do indeed represent a true underlying correlation in the datasets. We prove when each of these statistics are consistent to showcase the superiority of ICCA. We also provide an analogous statistical test and consistency theorem to use when the datasets have missing data entries. We create 3 new real-world, multi-modal datasets involving video and audio to verify the performance of ICCA. We then showcase that the canonical vectors returned by ICCA are more accurate than the CCA vectors in the low-sample regime. Finally, we provide a new algorithm for estimating the canonical vectors that asymptotically optimal.

Next, we explore the performance of regularized CCA (RCCA) in Chapter VI. When the number of training samples is limited but correlation analysis is still desired, a common strategy is to regularize CCA by adding a penalty to the magnitude of the linear transformation. However, we demonstrate that setting the regularization parameter to infinity results in the best performance. In fact, in this setting, the solution to RCCA may be found by taking the SVD of the sample cross-covariance matrix between the two datasets. We then predict the behavior of the largest singular values of this cross-covariance matrix assuming a low-rank signal-plus-noise model on the individual datasets. We argue that using the top singular values of this cross-covariance matrix to detect correlations is sub-optimal because the correlation coefficients are coupled with the individual data signal strengths.

Using a similar proof technique, we predict the behavior of the largest singular values of the projection of low-rank signal plus noise matrices to a smaller dimension in Chapter VII. Specifically, we consider two types of projection matrices: one with standard complex Gaussian entries and one with orthonormal columns. We are able to provide a closed form expression for the largest singular values in the case where the projection matrix is unitary. Through numerical simulations, we demonstrate the superiority of the unitary projection matrix over the Gaussian projection matrix. The unitary projection matrix can reliably detect the signals at a lower signal-to-noise ratio than the Gaussian projection matrix.

In Chapter VIII we apply CCA and ICCA to the classical problems of detection and regression. First, we consider the low-rank signal-versus-noise subspace detection problem given two datasets. We prove that the standard likelihood ratio test (LRT) detector may be written using the canonical basis returned by CCA. We show that when using empirical parameter estimates, the CCA detector is extremely suboptimal but that the ICCA detector is equivalent to the plug-in LRT detector. We then show that the classical Gaussian regression problem may be written in terms of the CCA basis. However, similar to the detection problem, empirical CCA degrades the performance significantly while ICCA matches the classical plug-in detector. We show this via mean squared error prediction plots.

We then consider the joint problems of image retrieval and image annotation in Chapter IX. Correlation based methods are typically overlooked as solutions to such problems due to the problems with CCA outlined in this thesis. We show that using ICCA to solve these problems results in non-trivial solutions. We compare the performance of CCA and ICCA on four different image-text datasets and describe the capabilities and limitations of ICCA in this application. When the datasets contain multiple images of the same objects and meaningful captions, ICCA is able to capture correlations between images and text. However, ICCA fails to capture semantic meanings between documents and captions. We argue that with clever feature engineering and improved NLP techniques, correlation based methods may be relevant for image retrieval and image annotation.

Lastly, we consider multi-set CCA (MCCA) in Chapter X. Unfortunately, unlike CCA, there is no clear objective function to use in an optimization problem; Kettenring [64] proposes five such objective functions. Nielsen also provides a nice formulation of MCCA in [65] where he proposes four constraint functions. We provide derivations for these 20 formulations, in both a theoretical and empirical setting. We then choose to consider the MAXVAR problem as we are able to directly apply our insights from ICCA to create an informative version of it, which we call informative MCCA (IMCCA). We demonstrate the superior performance of IMCCA on a real-world video dataset that we created.

CHAPTER II

Performance of Matched Subspace Detectors Using Finite Training Data

2.1 Introduction

Many signal processing [66] and machine learning [67] applications involve the task of detecting a signal of interest buried in high dimensional noise. A matched subspace detector (MSD) is commonly used to solve this problem when the target signal is assumed to lie in a low-rank subspace. The low-rank signal buried in noise model is ubiquitous in signal processing. In array processing, [68] and [69] use multiple array snapshots to detect a low-rank signal in the presence of both interference and noise when the noise power is known and unknown, respectively. Similarly in adaptive radar detection, [70] and [71] adaptively detect distributed low-rank targets given multiple snapshots of primary (signal plus noise) and secondary (noise only) data under partially homogeneous and homogeneous noise assumptions, respectively. Low rank signal models are also used in electroencephalography (EEG) and magnetoencephalography (MEG) source localization as in [72] and [73], respectively. In [68, 69, 70, 71], the signal subspace is known. The performance of a MSD when the signal subspace is known was studied in [74] and [75] under deterministic signal assumptions and in [76] and [77] under stochastic signal assumptions. This chapter considers the performance of a MSD in the less studied setting where the signal subspace is unknown and must be estimated from finite, noisy, signal-bearing training data.

The setting we have in mind arises from machine learning related applications where the low-rank signal model is reasonable but the signal subspace is not parameterizable. This is in contrast to the array processing applications that motivated the original MSD work [74] where the signal subspace is explicitly parameterizable whenever the array geometry is known. The inferential problem is made tractable by the availability of a training dataset consisting of signal-bearing observations that have been collected in a variety of representative experimental (and thus noisy) conditions. In such a scenario, the truncated eigen-decomposition of the sample covariance matrix of this training data yields an estimate of the unknown low-rank signal subspace, which may then be used for signal versus noise discrimination.

An illustrating example of this is the classical problem of handwriting recognition [78, Chapter 10] where a MSD can be used to determine if an area of an image contains a digit $0-9$ or is pure noise. Here, a database [79], containing a large number of handwritten samples of each of the digits written by many different writers, is used to form a low-rank subspace estimate of each digit. The samples are noisy because of digitization effects and the inherent variation between writers. A nearest-subspace classifier based on retaining only the first few ($10-12$, in this example) principal components (or leading eigenvectors of the digit's training data sample covariance matrix) associated with each digit yields greater than 93% classification performance [78, Table 10.1, pp. 121], indicating that the low-rank signal buried in noise model is appropriate. The motivating setting described also arises in the context of image or wavefront recognition applications (e.g. license plate character recognition) where the target and the camera are separated by a dynamic random medium and in hyperspectral imaging based anomaly detection [80, 81, 82] relative to a statistically stationary scene (e.g. toxic gas detection). Here too, a practitioner might have access to training samples collected over a variety of experimental conditions and might employ the MSD in a similar manner.

In these applications, the standard plug-in detector, which substitutes an estimate of the signal subspace into the expression for the oracle MSD that was derived assuming the subspace is perfectly known, realizes a performance loss because additive noise and finite training data decrease the accuracy of the estimated subspace. This motivates questions such as: What is the expected plug-in detector performance? Is it possible to avoid some of this performance loss? How does the estimation of the signal subspace dimension influence detector performance? Is the "play-it-safe" over-estimation of subspace dimension, to compensate for the potential underestimation of schemes discussed in [83] , a good idea?

Our performance analysis, which relies on insights from random matrix theory (RMT), highlights the importance of using no more than k_{eff} *informative* signal subspace components, where k_{eff} is a number that depends on the system dimensionality, number of training samples, and eigen-SNR (signal-to-noise-ratio). We derive a new

RMT detector that only utilizes the k_{eff} *informative* signal subspace components, thereby avoiding some of the possible performance loss suffered by the plug-in detector. Given the number and quality (i.e. SNR) of the training samples, our analysis also allows a practitoner to predict the expected receiver operating characteristic (ROC) performance of a general class of detectors. An outcome of this analyis is that we can accurately predict how many training samples are needed to get to within ϵ of the oracle MSD's performance (see Figures 2.2, 2.6(a), and 2.6(b)). This performance characterization can provide the practitioner with experimental guidance and might be a starting point for the formulation of achievable system performance specifications.

This chapter differs from previous works in several aspects. The focus and main contribution is analytically quantifying the performance of a general class of MSD's as a function of the system dimensionality, number of training samples, and eigen-SNR. Theorem 2.5.1 and Corollary 2.5.1 extend recent results from RMT [84, 85, 86] to precisely quantify the accuracy of the subspace estimate. This quantification yields approximations that appear to hold for moderate system dimensions even though the theory is asymptotic, in the limit of large dimensionality and relatively large training sample size. We provide a first-principles derivation of a new RMT detector that incorporates this knowledge of the accuracy of the estimated subspace, thereby illuminating the asymptotic form of a detector that mitigates some of the potential performance loss suffered by the plug-in detector. These RMT insights also allow us to characterize the ROC performance of a MSD under both a deterministic and stochastic model for the test vector. This work builds on [87] by providing the proofs of Theorem 2.5.1 and Corollary 2.5.1, analyzing the performance of the general class of detectors given in (2.14), considering the deterministic test vector setting, and unifying the performance analysis of the stochastic and deterministic MSD's.

This chapter is organized as follows. We describe the generative models for the training data and test vector and also estimate unknown parameters in Section 2.2. In Section 2.3, we derive standard oracle and plug-in detectors for each testing setting and highlight how finite training data causes subspace estimation errors and subsequent performance loss. We formally pose the questions addressed herein in Section 2.4. Section 2.5 contains pertinent results from RMT and our definition in (2.16) of k_{eff}. In Section 2.6 we derive RMT detectors for the stochastic and deterministic test vector models. Aided by RMT and a saddlepoint approximation of the CDF of a weighted sum of chi-square random variables, we predict ROC performance curves for a general detector in Section 2.7. We validate our asymptotic ROC predictions and demonstrate the importance of using the k_{eff} informative subspace components in Section 2.8. We provide concluding remarks in Section 2.9.

2.2 Data Models and Parameter Estimation

Given an observation, we wish to discriminate between the H_0 hypothesis that the observation is purely noise and the H_1 hypothesis that the observation contains a target signal. We assume that the signal of interest lies in a low dimensional subspace as in [68, 69, 70, 71, 72, 73, 80, 81, 82]. However, this low-rank subspace and the SNR governing the subspace components are unknown. To design a detector to distinguish between the H_0 and H_1 hypotheses, we have access to a training dataset, recorded under similar noisy conditions, whose observations are known to contain the signal of interest (see, for example, [81, 82]). We use this training data to form estimates of the unknown low-rank subspace and each component's SNR. This section will mathematically describe the training data models, how we estimate any unknown parameters, and a stochastic and deterministic model for the testing data. Both testing models share the same training data model.

2.2.1 Training Data Model

We model our unknown subspace with the complex matrix $U = [u_1, \ldots, u_k]$ such that $\dim u_i = n$ and $\langle u_i, u_j \rangle = u_i^H u_j = \delta_{ij}$ for $i, j = 1, \ldots, k$. Here δ_{ij} is the delta function such that $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ij} = 1$ for $i = j$. We are given m signal-bearing training vectors $y_i \in \mathbb{C}^{n \times 1}$, $i = 1, \ldots, m$, modeled[1] as $y_i = Ux_i + z_i$ where $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_n)$ and $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, \Sigma)$ where $\Sigma = \mathbf{diag}(\sigma_1^2, \ldots, \sigma_k^2)$ with $\sigma_1 > \sigma_2 > \cdots > \sigma_k > 0$ unknown. Similar gaussian priors appear in [70, 71, 80, 81]. Σ models the SNR of each subspace component and z_i models the additive noise. For each observation, x_i and z_i are independent. The dimension, k, of our subspace is unknown and we assume throughout that $k \ll n$ so that we have a low-rank signal embedded in a high-dimensional observation vector.

2.2.2 Parameter Estimation

The parameters k, U, and Σ are all unknown in our training model. For the rest of the paper, we assume that we are given a dimension estimate, $\widehat{k}$; this may have been estimated from the training data or provided by a domain expert. Typically, $\widehat{k}$

[1] For expositional simplicity, we have assumed that all our matrices and vectors are complex-valued; our results also hold for real-valued matrices and vectors.

is an overestimation of a dimension estimate provided by percent variance, scree plots [88], or robust techniques [89, 90, 91]. This overestimation, or "play-it-safe" strategy, strives to include all signal subspace components at the expense of possibly including non-signal subspace components.

Given $\widehat{k}$ and the signal bearing training data $Y = \begin{bmatrix} y_1 & \dots & y_m \end{bmatrix}$, we form the sample covariance matrix $S = \frac{1}{m}YY^H$. The covariance matrix of y_i is $U\Sigma U^H + I_n$ and it follows that the (classical) ML estimates (in the many-sample, small matrix setting) for U and Σ are given by [92]

$$\begin{aligned} &\widehat{U} = [\widehat{u}_1 \dots \widehat{u}_{\widehat{k}}] \\ &\widehat{\sigma}_i^2 = \max(0, \widehat{\lambda}_i - 1) \text{ for } i = 1, \dots, \widehat{k} \end{aligned} \tag{2.1}$$

where $\widehat{\lambda}_1, \dots, \widehat{\lambda}_{\widehat{k}}$ are the $\widehat{k}$ largest eigenvalues of the sample covariance matrix, S, and $\widehat{u}_1, \dots, \widehat{u}_{\widehat{k}}$ are the corresponding eigenvectors. Define the signal covariance matrix estimate as $\widehat{\Sigma} = \mathbf{diag}(\widehat{\sigma}_1^2, \dots, \widehat{\sigma}_{\widehat{k}}^2)$. We are now able to use the parameter estimates $\widehat{U}$ and $\widehat{\Sigma}$ in detectors where necessary.

2.2.3 Testing Data Model

We will consider both a stochastic and deterministic model for a test vector. In both settings, parameter estimates are formed as described in (2.1) from training data modeled in Section 2.2.1.

In the stochastic setting, the test vector $y \in \mathbb{C}^{n \times 1}$ is modeled as

$$\text{Stochastic Model: } y = \begin{cases} z & y \in H_0 : \text{ Noise only} \\ Ux + z & y \in H_1 : \text{ Signal-plus noise} \end{cases}, \tag{2.2}$$

where U, z, and x are modeled as described in Section 2.2.1. This assumes that the signal, Ux, may lie anywhere in the subspace and whose position in the subspace is governed by the signal covariance matrix Σ.

In the deterministic setting, the test vector $y \in \mathbb{C}^{n \times 1}$ is modeled as

$$\text{Deterministic Model: } y = \begin{cases} z & y \in H_0 : \text{ Noise only} \\ U\Sigma^{1/2}x + z & y \in H_1 : \text{ Signal-plus noise} \end{cases}, \tag{2.3}$$

where U, Σ, and z are modeled as described in Section 2.2.1. Here, in contrast to the stochastic setting, x is a non-random deterministic vector. Thus the signal, $U\Sigma^{1/2}x$, lies at a fixed point in the unknown subspace. Note that Σ still controls the SNR of each subspace component and that placing a mean zero, identity covariance Gaussian prior on x in (2.3) yields the stochastic model described in (2.2).

2.3 Standard Detector Derivations

In this chapter, we focus on the Neyman-Pearson setting (see [93]) where, given a test observation from (2.2) or (2.3), a MSD is a likelihood ratio test (LRT) taking the form

$$\Lambda(y) := \frac{f(y|H_1)}{f(y|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{2.4}$$

where $\Lambda(y)$ is the test statistic, η is the threshold set to achieve a given false alarm rate, and f is the appropriate conditional density of the test observation. In the following section, for both testing data models we derive the standard oracle detector (assuming all parameters are known) and plug-in detector (formed by substituting the parameter estimates of (2.1) in the oracle detector). The oracle detectors, while unrealizable, give an upper bound for the performance of a MSD. We will see that when only finite training data is available (as is the case in real applications), the plug-in detector will realize a performance loss relative to this bound.

2.3.1 Stochastic Testing Model

The LRT in (2.4) depends on the conditional distribution of the test vector, y. By properties of Gaussian random variables, when using the stochastic test model in (2.2), these distributions are $y|H_0 \sim \mathcal{N}(0, I_n)$ and $y|H_1 \sim \mathcal{N}\left(0, U\Sigma U^H + I_n\right)$. The resulting LRT statistic is

$$\Lambda(y) = \frac{\mathcal{N}(0, U\Sigma U^H + I_n)}{\mathcal{N}(0, I_n)}. \tag{2.5}$$

We derive an oracle detector by assuming that k, Σ, and U are all known in (2.5). After simplification of this expression (see Section 4.14 of [66]), the oracle statistic becomes

$$\Lambda_{\text{oracle}}(y) = y^H U \left(\Sigma^{-1} + I_k\right)^{-1} U^H y. \tag{2.6}$$

Note that the oracle statistic depends on the sufficient statistic $w := U^H y$. Using this notation, the oracle statistic is

$$\boxed{\Lambda_{\text{oracle}}(w) = w^H \left(\Sigma^{-1} + I_k\right)^{-1} w = \sum_{i=1}^{k} \left(\frac{\sigma_i^2}{\sigma_i^2 + 1}\right) w_i^2} \tag{2.7}$$

and the oracle detector is $\Lambda_{\text{oracle}}(w) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{oracle}}$ where the threshold γ_{oracle} is chosen in the usual manner, *i.e.*, so that it satisfies $P(\Lambda_{\text{oracle}}(w) > \gamma_{\text{oracle}}|H_0) = \alpha$ with α a desired false alarm rate.

However, as the parameters U and Σ are unknown, the oracle statistic in (2.7) cannot be computed. Given a dimension estimate $\widehat{k}$, we substitute the ML estimates of U and Σ given in (2.1) for the unknown parameters in (2.6) as similarly done in [76] and [77]. This results in the plug-in detector's LRT statistic: $\Lambda_{\text{plugin}}(y) = y^H\widehat{U}\left(\widehat{\Sigma}^{-1} + I_{\widehat{k}}\right)^{-1}\widehat{U}^Hy$. Simplifying this expression using the statistic $\widehat{w} = \widehat{U}^Hy$, yields the plug-in statistic

$$\boxed{\Lambda_{\text{plugin}}(\widehat{w}) = \widehat{w}^H\,\mathbf{diag}\left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2+1}\right)\widehat{w} = \sum_{i=1}^{\widehat{k}}\left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2+1}\right)\widehat{w}_i^2} \tag{2.8}$$

and the plug-in detector takes the form $\Lambda_{\text{plugin}}(w) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{plugin}}$ where the threshold γ_{plugin} is chosen in the usual manner.

The plug-in detector assumes that the estimated signal subspace, $\widehat{U}$, is equal to the true signal subspace, U, and that the estimated signal covariance, $\widehat{\Sigma}$, is equal to the true signal covariance, Σ. In other words, the plug-in detector derivation assumes that $\widehat{U}^HU = I_{\widehat{k}}$, $\widehat{\sigma}_i^2 = \sigma_i^2$ for $i = 1,\ldots,\widehat{k}$, and the provided subspace dimension estimate, $\widehat{k}$, is equal to the true underlying dimension of our signal subspace, k. Perhaps unsurprisingly, (as discussed in Section 2.5) incorrectly choosing $\widehat{k}$ degrades the performance of the plug-in detector.

2.3.2 Deterministic Testing Model

We now consider the alternative deterministic test vector model (2.3), which results in the following conditional distributions of the test vector $y|H_0 \sim \mathcal{N}(0, I_n)$ and $y|H_1 \sim \mathcal{N}(U\Sigma^{1/2}x, I_n)$. We begin by deriving an oracle detector, which assumes that U, Σ, x, and k are all known. The LRT statistic for such a scenario is $\Lambda(y) = \frac{\mathcal{N}(U\Sigma^{1/2}x, I_n)}{\mathcal{N}(0, I_n)}$. Simplifying this expression leads to the oracle statistic

$$\Lambda_{\text{oracle}}(y) = x^H\Sigma^{1/2}U^Hy. \tag{2.9}$$

As in the stochastic setting, $w = U^H y$ is a sufficient statistic and the oracle statistic simplifies to

$$\boxed{\Lambda_{\text{oracle}}(w) = x^H \Sigma^{1/2} w = \sum_{i=1}^{k} x_i \sigma_i w_i.} \tag{2.10}$$

However, as the parameters U, Σ, and x are unknown, the oracle statistic in (2.10) cannot be computed. Since we must estimate x from the test vector, we employ the generalized likelihood ratio test (GLRT) where $\Lambda(y) = \frac{\max_x f(y|H_1)}{f(y|H_0)}$, resulting in the GLRT statistic

$$\Lambda(y) = \frac{\max_x \mathcal{N}(U\Sigma^{1/2}x, I_n)}{\mathcal{N}(0, I_n)}. \tag{2.11}$$

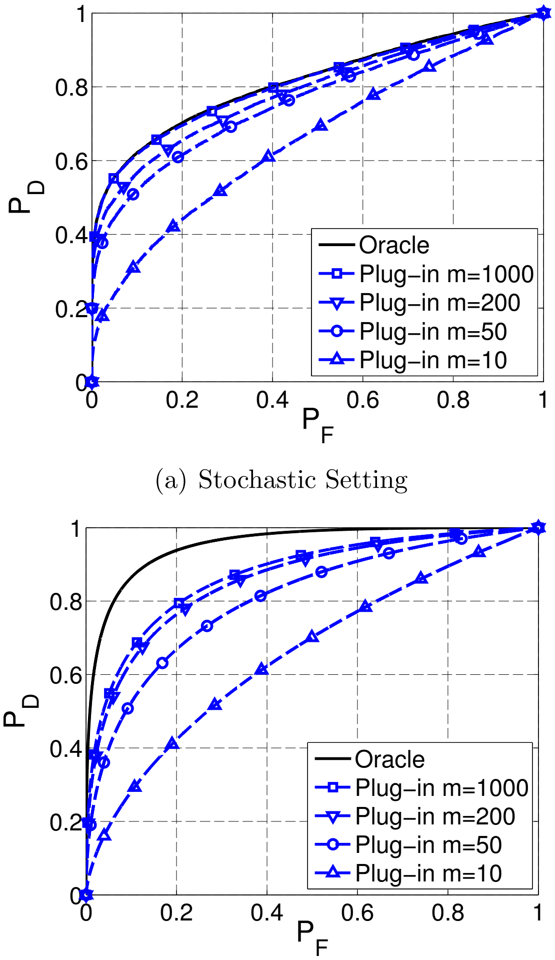
Employing maximum likelihood estimation on x in (2.11) yields the estimate $\widehat{x} = \Sigma^{-1/2} U^H y$. Proceeding as in the stochastic setting, we substitute $\widehat{x}$ for the unknown x in (2.9) and then substitute the ML estimates of U and Σ given in (2.1) for the unknown U and Σ (see Section 4.11 of [66] for a similar treatment). This results in the plug-in statistic $\Lambda_{\text{plugin}}(y) = y^H \widehat{U}\widehat{U}^H y$. Again, $\widehat{w} = \widehat{U}^H y$ is a statistic that can be used to write the plug-in statistic as

$$\boxed{\Lambda_{\text{plugin}}(\widehat{w}) = \widehat{w}^H \widehat{w} = \sum_{i=1}^{\widehat{k}} \widehat{w}_i^2}, \tag{2.12}$$

resulting in the detector $\Lambda_{\text{plugin}}(\widehat{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{plugin}}$, where the threshold γ_{plugin} is chosen in the usual manner. The deterministic plug-in detector is an 'energy detector', which sums the energy of the test observation lying in the subspace $\widehat{U}$.

2.3.3 Effect of the Number of Training Samples

In both the stochastic and deterministic testing settings, $\widehat{w} = \widehat{U}^H y$ is a statistic used in the plug-in statistics (2.8) and (2.12). This statistic relies on the estimated subspace $\widehat{U}$ formed from the top $\widehat{k}$ eigenvectors of the sample covariance matrix, S, of the training data. The stochastic detector also relies on the subspace-SNR estimate $\widehat{\Sigma}$ formed from the top $\widehat{k}$ eigenvalues of S. For a fixed Σ, the accuracy of these estimates depends on the number of training data samples, m; we will mathematically show this in Section 2.5. If we had access to an infinite amount of training data, the parameter estimates would be exact ($\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$). However, when we have access to only a finite amount of training data, $\widehat{U}$ and $\widehat{\Sigma}$ are inaccurate and will degrade the performance of the plug-in detectors with respect to the oracle detector, which



(a) Stochastic Setting

(b) Deterministic Setting

Figure 2.1: Empirical ROC curves for the plug-in and oracle detectors. Empirical ROC curves were simulated with $n = 200$, $\widehat{k} = k = 2$, and $\Sigma = \mathbf{diag}\,(10, 0.1)$. The empirical ROC curves were computed using 10000 test samples and averaged over 100 trials using algorithms 2 and 4 of [1]. (a) Shows results for the stochastic MSD. (b) Shows results for the deterministic MSD when $x = [0.75, 0.75]^T$. For both settings, as m decreases, the performance of the plug-in detector degrades.

provides an upper bound on detector performance.

To illustrate this performance loss, we consider a moderately sized system where $n = 200$ and $\Sigma = \mathbf{diag}(10, 0.1)$. We consider five detectors: the oracle detector and four plug-in detectors each using parameter estimates formed from varying amounts of training data. Figures 2.1(a) and 2.1(b) plot the empirical ROC curves for the stochastic and deterministic testing settings, respectively. The amount of training data drastically affects the performance of the plug-in detector. As m decreases, the plug-in detectors realize a significance performance loss. However, as $m \to \infty$, the plug-in detectors realize improved performance, closer to that of the oracle detectors.

For the stochastic detector, as $m \to \infty$, the plug-in detector achieves the same performance as that of the oracle detector. Examination of the statistics (2.7) and (2.8) shows that these statistics will be identical when $\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$, which is the case when infinite training data is available. However, this is not the case for the deterministic plug-in detector. Even with an infinite amount of training data, the plug-in detector will not achieve the oracle detector's performance. The deterministic plug-in detector must estimate x given a noisy test observation y, which is independent from the training data. Even with infinite training data causing $\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$, $\widehat{x}$ does not converge to x. Therefore, the deterministic plug-in detector cannot achieve the performance bound of the oracle detector, which assumes that x is known.

For a fixed probability of false alarm (P_F), we can explore this performance loss by comparing the achieved probability of detection (P_D) of the plug-in detector to that of the oracle detector. Let

$$\epsilon = 1 - \frac{P_D^{\text{plugin}}}{P_D^{\text{oracle}}} \tag{2.13}$$

be the performance loss of the plug-in detector. Figure 2.2 empirically plots the number of training samples needed to achieve a desired performance loss ϵ for the stochastic plug-in detector. There is an exponential relationship between ϵ and m indicating that we need infinite training samples to achieve zero performance loss ($\epsilon = 0$). However, in any practical application we will never have an infinite amount of training data and so the plug-in detector will realize some non-zero performance loss. The rest of the paper will mathematically predict how finite training data affects detector performance and will derive new detectors to avoid some of this performance loss.

2.4 Problem Statements

We saw in Section 2.3 that the plug-in detectors rely on the statistic $\widehat{w} = \widehat{U}^H y$. When only finite training data is available, the subspace estimate $\widehat{U}$ is inaccurate and subsequently degrades the performance of the plug-in detector. Motivated by this observation, we formulate the problems addressed in this paper.

2.4.1 Problem 1: Derive a New Detector that Exploits Predictions of Subspace Accuracy

We know that subspace estimation errors degrade the performance of the plug-in detector. Recent results from RMT specifically quantify the accuracy of $\widehat{U}$ relative

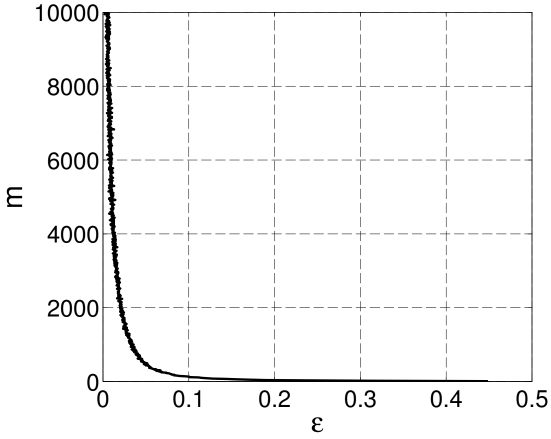


Figure 2.2: Empirically determined number of training samples, m, needed for the stochastic plug-in detector to achieve a desired performance loss, ϵ, as defined in (2.13). The required false alarm rate is $P_F = 0.1$. Empirical ROC curves were generated for $n = 200$, $\Sigma = \textbf{diag}(10, 0.1)$, $\widehat{k} = k = 2$ using 10000 testing samples and averaged over 100 trials using algorithms 2 and 4 of [1].

to U. By deriving a new detector that accounts for this accuracy of the estimated subspace, we hope to avoid some of the performance loss associated with the plug-in detector. For both the stochastic and deterministic testing settings our goal is to

> Design a new detector that exploits RMT predictions of subspace estimation accuracy.

The detector derivations in Section 2.6 will provide insights on when, if, and how the performance of plug-in detectors that do not exploit the knowledge of subspace estimation accuracy can be improved.

2.4.2 Problem 2: Characterize ROC Performance Curves

We saw in Section 2.3 that both plug-in detectors took the form

$$\widehat{w}^H D \widehat{w} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{2.14}$$

where D is the appropriate diagonal matrix and the test statistic $\Lambda(\widehat{w}) = \widehat{w}^H D \widehat{w}$ is compared against a threshold, η, set to achieve a prescribed false alarm rate. After solving Problem 1, we will see that the RMT detectors derived in Section 2.6 also take the form of (2.14). In order to compare detectors of this form without training data or empirically generated test samples, we wish to analytically predict their ROC performance. Formally, for detectors with the form of (2.14) and for test vectors modeled as (2.2) or (2.3), our goal is to

Predict $P_D := \mathbb{P}(\text{Detection})$, for every $P_F := \mathbb{P}(\text{False Alarm}) = \alpha \in (0,1)$ given n, m, $\widehat{k}$, D and Σ.

For this problem, we assume that we are given Σ. We derive this theoretical prediction of ROC performance curves in Section 2.7 and show that this performance prediction also relies on RMT results quantifying the accuracy of the subspace estimate $\widehat{U}$, specifically the entries of the matrix $\widehat{U}^H U$. In Section 2.5 we provide an asymptotic diagonal approximation to this matrix that makes the ROC prediction possible.

2.5 Pertinent Results from Random Matrix Theory

In Section 2.2.2 we formed estimates $\widehat{U}$ and $\widehat{\Sigma}$ of the unknown U and Σ by taking the eigen-decomposition of the sample covariance matrix S of the training data matrix Y. These estimates are inaccurate because the training data is noisy and contains only a finite number of observations. The following analysis specifically quantifies the accuracy of these estimates and is necessary to derive a new detector and predict ROC performance curves of detectors with the form of (2.14).

2.5.1 Eigenvector Aspects

The subspace estimate $\widehat{U}$ is formed from the eigenvectors corresponding to the $\widehat{k}$ largest eigenvalues of S. For an arbitrary non-random diagonal matrix D, we will be particularly interested in the matrix $\widehat{U}^H U D U^H \widehat{U}$ that appears in detector derivations and the ROC performance analysis in Sections 2.6 and 2.7. The following proposition characterizes the limiting behavior (up to an arbitrary phase) of the diagonal entries of the matrix $\widehat{U}^H U$.

Proposition 2.5.1. *Assume that the columns of the training data matrix Y were generated as described in Section 2.2.1. Let $\widehat{u}_i$ denote the eigenvector associated with the i-th largest eigenvalue of S. Then for $i = 1, \ldots, k$ and $n, m \longrightarrow \infty$ with $n/m \to c$, we have that*

$$|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{a.s.} \begin{cases} \dfrac{\sigma_i^4 - c}{\sigma_i^4 + \sigma_i^2 c} & \text{if } \sigma_i^2 > \sqrt{c} \\ 0 & \text{otherwise} \end{cases}. \tag{2.15}$$

Proof. This follows from Theorem 4 of [84] when $\gamma = c$, $\ell_\nu - 1 = \sigma_\nu^2$, $\widetilde{e}_\nu = u_v$, and $p_\nu = \widehat{u}_\nu$. This result also appears in Theorem 2.2 of [85]. □

We note that $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence. The key insight from Proposition 2.5.1 is that only the eigenvectors corresponding to the signal variances, σ_i^2, lying above the phase transition $\sqrt{c}$ are *informative*. When a signal variance drops below this critical threshold, the corresponding eigenvector estimate is essentially noise-like (i.e. $|\langle u_i, \widehat{u}_i \rangle|^2 = o_p(1)$ meaning $|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{p} 0$ as $n \to \infty$, denoting convergence in probability) and thus *uninformative*. Decreasing the amount of training data, m, increases c, thereby decreasing the value of $|\langle u_i, \widehat{u}_i \rangle|^2$; if this quantity became 0, the associated subspace component would become uninformative.

The term $|\langle u_i, \widehat{u}_i \rangle|^2$ quantifies mismatch between the estimated and underlying eigenvectors and will play an important role in deriving a new RMT detector and in characterizing detector performance; a similar term also appears in the analysis of the resolving power of arrays due to model mismatch such as in [94].

Following [83], we define the effective number of (asymptotically) identifiable subspace components k_{eff} as:

$$\boxed{k_{\text{eff}} = \text{Number of } \sigma_i^2 > \sqrt{c}}. \tag{2.16}$$

We can form an estimate of k_{eff}, $\widehat{k}_{\text{eff}}$, using 'Algorithm 2' of [89]. This algorithm assumes the same model of a low-rank signal buried in high dimensional noise as our training data. Given a desired significance level, the algorithm estimates the number of signals present in a finite number of samples. When the noise covariance matrix is not known a priori, we would instead use 'Algorithm 1' of [89]. Both algorithms rely on the Tracy-Widom distribution. Note that $\widehat{k}_{\text{eff}} \leq k$ but that we allow $\widehat{k} \geq \widehat{k}_{\text{eff}}$ so we may understand the impact of a play-it-safe overestimation of the signal subspace dimension estimate $\widehat{k}_{\text{eff}}$ returned using RMT based detectors [89, 90, 91].

Proposition 2.5.1 only characterizes the limiting behavior (up to an arbitrary phase) of the diagonal entries of the matrix $\widehat{U}^H U$. We now state a new theorem characterizing the limiting behavior of the off-diagonal entries in $\widehat{U}^H U$.

Theorem 2.5.1. *Assume the same hypothesis as in Proposition 2.5.1. Let* $\widehat{k} = k_{\mathit{eff}} = k$. *For* $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, *and* $i \neq j$, *as* $n, m \to \infty$ *with* $n/m \to c$, $\langle u_j, \widehat{u}_i \rangle \xrightarrow{a.s.} 0$.

Proof. This is a new result. See Appendix A for proof. □

Conjecture 2.5.1. *Assume the same hypothesis as in Proposition 2.5.1. For* $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, *and* $i \neq j$, *as* $n, m \to \infty$ *with* $n/m \to c$, $\langle u_j, \widehat{u}_i \rangle \xrightarrow{a.s.} 0$.

Remark 2.5.1. *See Appendix for a brief discussion of this conjecture.*

Together, Proposition 2.5.1 and Conjecture 2.5.1 characterize the limiting behavior of the entries of $\widehat{U}^H U$. This permits approximation, in the large matrix limit, of $\widehat{U}^H U D U^H \widehat{U}$ by a suitable diagonal matrix.

Corollary 2.5.1. *Suppose $\widehat{k} \leq k$ and let D be a $k \times k$ (non-random) diagonal matrix such that $D = \mathbf{diag}(d_1, \ldots, d_k)$, independent of $\widehat{U}$. Then as $n, m \longrightarrow \infty$ with $n/m \to c$, we have that*

$$\widehat{U}^H U D U^H \widehat{U} \xrightarrow{a.s.} \mathbf{diag}(d_1 |\langle u_1, \widehat{u}_1 \rangle|^2, \ldots, d_{\widehat{k}} |\langle u_{\widehat{k}}, \widehat{u}_{\widehat{k}} \rangle|^2)$$

where for $i = 1, \ldots, \widehat{k}$ the quantity $|\langle u_i, \widehat{u}_i \rangle|^2$ is given in Proposition 2.5.1.

Proof. This follows directly by applying Proposition 2.5.1 and Conjecture 2.5.1 to the entries of the matrix $U^H \widehat{U}$. □

This diagonal approximation of $\widehat{U}^H U D U^H \widehat{U}$ will be used in detector derivations and ROC performance analyses in Sections 2.6 and 2.7.

2.5.2 Eigenvalue Aspects

The signal covariance estimate $\widehat{\Sigma}$ is formed from the largest $\widehat{k}$ eigenvalues of S. To characterize the ROC performance curves of plug-in detectors that use $\widehat{\Sigma}$ as the signal covariance estimate, we will also need to characterize the limiting behavior of $\widehat{\Sigma}$. The following proposition gives the limiting behavior of these signal variance estimates.

Proposition 2.5.2. *As $n, m \longrightarrow \infty$ with $n/m \to c$ we have that:*

$$\widehat{\sigma}_i^2 \xrightarrow{a.s.} \begin{cases} \sigma_i^2 + c + \frac{c}{\sigma_i^2} & \text{if } \sigma_i^2 > \sqrt{c} \\ c + 2\sqrt{c} & \text{if } \sigma_i^2 \leq \sqrt{c} \end{cases}.$$

Proof. This follows from Theorems 1 and 2 in [84] for the real setting for $c < 1$ when $\gamma = c$, $\ell_\nu - 1 = \sigma_\nu^2$, and $\widehat{\ell}_\nu - 1 = \widehat{\sigma}_\nu^2$. See Theorem 2.6 in [86] for the complete result. □

These limiting values will be used in Section 2.7 when deriving the ROC performance of the plug-in detectors.

When only finite training data is available, c is non-zero and Proposition 2.5.2 shows that $\widehat{\sigma}_i^2$ is biased. We wish to derive an improved signal variance estimate to use in a new RMT detector and to estimate $|\langle u_i, \widehat{u}_i \rangle|^2$ in (2.15). As seen in Proposition 2.5.1, when $\sigma_i^2 \leq \sqrt{c}$ the eigenvector estimate is uninformative and we would not want to include that subspace component in a detector; the associated signal variance estimate is therefore unnecessary. For the $\widehat{k}_{\text{eff}}$ subspace components that are informative (i.e. when $\sigma_i^2 > \sqrt{c}$) we form an improved signal variance estimate using the following proposition that characterizes the fluctuations of these signal variance estimates.

Proposition 2.5.3. *As $n, m \longrightarrow \infty$ with $n/m \to c$, we have that for $i = 1, \ldots, k_{\mathit{eff}}$*

$$\sqrt{n}\left(\widehat{\sigma}_i^2 - \left(\sigma_i^2 + c + \frac{c}{\sigma_i^2}\right)\right) \Rightarrow \mathcal{N}\left(0, \frac{2\left(\sigma_i^2+1\right)^2}{\beta}\left(1 - \frac{c}{\sigma_i^4}\right)\right),$$

where $\beta = 1$ when the data is real-valued and $\beta = 2$ when the data is complex-valued.

Proof. This follows from Theorem 3 in [84] for the real setting for $c < 1$ when $\gamma = c$, $\ell_\nu - 1 = \sigma_\nu^2$, $\widehat{\ell}_\nu - 1 = \widehat{\sigma}_\nu^2$, and p_ν is the limit of Theorem 2 of [84]. See Theorem 2.15 in [86] for the complete result. □

For the $\widehat{k}_{\text{eff}}$ informative subspace components we form an improved estimate, $\widehat{\sigma}_{i_{\text{rmt}}}^2$, of the unknown signal variance, σ_i^2, by employing maximum-likelihood (ML) estimation on the distribution in Proposition 2.5.3. Specifically, for only the $\widehat{k}_{\text{eff}}$ signal eigenvalues, we form the RMT estimate:

$$\widehat{\sigma}_{i_{\text{rmt}}}^2 = \underset{\sigma_i^2}{\operatorname{argmax}} \log\left(f_{\widehat{\sigma}_i^2}(\sigma_i^2)\right) \tag{2.17}$$

where

$$f_{\widehat{\sigma}_i^2}(\sigma_i^2) := \mathcal{N}\left(\left(\sigma_i^2 + c + \frac{c}{\sigma_i^2}\right), \frac{2\left(\sigma_i^2+1\right)^2}{n\beta}\left(1 - \frac{c}{\sigma_i^4}\right)\right).$$

We may then estimate $|\langle u_i, \widehat{u}_i \rangle|^2$ in (2.15) by substituting the improved signal variance estimates, $\widehat{\sigma}_{i_{\text{rmt}}}^2$, for the unknown σ_i^2 in Proposition 2.5.1. We refer to this estimate as $|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2$. For the $\widehat{k} - \widehat{k}_{\text{eff}}$ uninformative subspace components, we set $|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 = 0$.

2.6 Derivation of New RMT Matched Subspace Detectors

We saw in Section 2.3 that the plug-in detectors rely on the statistic $\widehat{w} = \widehat{U}^H y$. Instead of deriving the LRT statistic using the conditional distributions of y, we will instead use the conditional distributions of $\widehat{w}$; this will reveal the importance of the matrix $\widehat{U}^H U$. The plug-in detectors assume that $\widehat{U}^H U = I_{\widehat{k}}$, however, the analysis in Section 2.5.1 shows that this assumption is incorrect. Knowing the importance of only using k_{eff} subspace components and armed with the asymptotic diagonal approximation of Corollary 2.5.1 and the improved signal variance estimates in (2.17), we are now in position to answer Problem 1 and derive a new RMT detector for both testing settings.

2.6.1 Stochastic RMT Detector

We begin with the stochastic test setting and form the test vector $\widehat{w} = \widehat{U}^H y$ where $\widehat{U}$ is the subspace estimated from (2.1) and y is generated from (2.2). The LRT statistic using $\widehat{w}$ depends on the conditional distributions under each hypothesis, which by properties of Gaussian random variables are simply

$$\widehat{w}|H_0 \sim \mathcal{N}\left(0, I_{\widehat{k}}\right) \quad \text{and} \quad \widehat{w}|H_1 \sim \mathcal{N}\left(0, \widehat{U}^H U \Sigma U^H \widehat{U} + I_{\widehat{k}}\right). \tag{2.18}$$

We immediately see the matrix of interest, $\widehat{U}^H U \Sigma U^H \widehat{U}$. The plug-in detector substitutes $\widehat{U}$ for U and $\widehat{\Sigma}$ for Σ; this results in $\widehat{w}|H_1 \sim \mathcal{N}(0, \widehat{\Sigma} + I_{\widehat{k}})$. However, Corollary 2.5.1 shows that this is incorrect by providing the asymptotic limit of the covariance matrix in (2.18):

$$\widehat{U}^H U \Sigma U^H \widehat{U} + I_{\widehat{k}} \xrightarrow{\text{a.s.}} \mathbf{diag}\left(|\langle u_i, \widehat{u}_i \rangle|^2 \sigma_i^2 + 1\right). \tag{2.19}$$

If σ_i^2 were assumed known, this limit would suffice because we could plug in the results in Proposition 2.5.1 to get the desired statistic. However, the signal variances are unknown so σ_i^2 and subsequently $|\langle u_i, \widehat{u}_i \rangle|^2$ must be estimated from data. For the $\widehat{k}_{\text{eff}}$ subspace components estimated from 'Algorithm 2' of [89], we form an improved signal variance estimate, $\widehat{\sigma}^2_{i_{\text{rmt}}}$, obtained via (2.17) and use it to estimate $|\langle u_i, \widehat{u}_i \rangle|^2$, denoted by $|\langle u_i, \widehat{u}_i \rangle|^2_{\text{rmt}}$. Of course, there are correction terms due to finite system size effects, which we ignore, that affect the convergence properties but not the asymptotic form of the detector.

We obtain the RMT detector by computing the LRT statistic using the conditional distributions of (2.18). The covariance matrix of $\widehat{w}|H_1$ is computed by substituting $|\langle u_i, \widehat{u}_i \rangle|^2_{\text{rmt}}$ and $\widehat{\sigma}^2_{i_{\text{rmt}}}$ into the diagonal covariance matrix (2.19). After some straightforward algebra we obtain the desired RMT statistic

$$\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{\widehat{k}} \left(\frac{|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2}{|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2 + 1} \right) \widehat{w}_i^2 .$$

As seen in Proposition 2.5.1, when $i > k_{\text{eff}}$, $|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{\text{a.s.}} 0$. The sum on the right hand side (asymptotically) discards the uninformative subspace components. Thus the RMT detector only uses the $\widehat{k}_{\text{eff}}$ informative components given by (2.16). Consequently, we obtain the test statistic

$$\boxed{\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\frac{|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2}{|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2 + 1} \right) \widehat{w}_i^2} \tag{2.20}$$

and the RMT detector becomes $\Lambda_{\text{rmt}}(\widehat{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{rmt}}$ where the threshold γ_{rmt} is chosen in the usual manner. Note that the stochastic RMT detector also takes the form of (2.14). The principal difference between the RMT test statistic in (2.20) and the plug-in test statistic in (2.8) is the role of $\widehat{k}_{\text{eff}}$ in the former. The scaling factors associated with each $\widehat{w}_i^2$ for both detectors are about the same; this is why the plug-in detector that uses $\widehat{k}_{\text{eff}}$ components exhibits the same (asymptotic) performance as the RMT detector, which incorporates knowledge of the subspace estimate accuracy. However, our analysis shows that overcompensating and "playing-it-safe" by setting $\widehat{k} > \widehat{k}_{\text{eff}}$ can lead to performance loss.

Detector	Detector Statistic $\Lambda(\widehat{w})$
Plug-in	$\sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2+1} \right) \widehat{w}_i^2$
RMT	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\frac{|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2}{|\langle u_i, \widehat{u}_i \rangle|_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2 + 1} \right) \widehat{w}_i^2$

Table 2.1: Summary of the plug-in and RMT stochastic MSDs. See Sections 2.3.1 and 2.6.1 for derivations.

2.6.2 Deterministic RMT Detector

When forming $\widehat{w}$ with y generated from (2.3), the conditional distributions of $\widehat{w}$ under each hypothesis are $\widehat{w}|H_0 \sim \mathcal{N}(0, I_{\widehat{k}})$ and $\widehat{w}|H_1 \sim \mathcal{N}(\widehat{U}^H U \Sigma^{1/2} x, I_{\widehat{k}})$. Again, as x is unknown, we use a GLRT. Employing maximum likelihood estimation on x yields the estimate $\widehat{x} = \left(\Sigma^{1/2} U^H \widehat{U} \widehat{U}^H U \Sigma^{1/2} \right)^{\dagger} \Sigma^{1/2} U^H \widehat{U} \widehat{w}$ where † denotes the Moore-

Detector	Distribution of $\Lambda \mid H_0$	Distribution of $\Lambda \mid H_1$
Plug-in	$\sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2+1}\right) \chi_{1i}^2$	$\sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2\left(\sigma_i^2 \lvert\langle u_i,\widehat{u}_i\rangle\rvert^2+1\right)}{\widehat{\sigma}_i^2+1}\right) \chi_{1i}^2$
RMT	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\frac{\widehat{\sigma}_{i_{\text{rmt}}}^2 \lvert\langle u_i,\widehat{u}_i\rangle\rvert_{\text{rmt}}^2}{\widehat{\sigma}_{i_{\text{rmt}}}^2 \lvert\langle u_i,\widehat{u}_i\rangle\rvert_{\text{rmt}}^2+1}\right) \chi_{1i}^2$	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\widehat{\sigma}_{i_{\text{rmt}}}^2 \lvert\langle u_i,\widehat{u}_i\rangle\rvert_{\text{rmt}}^2\right) \chi_{1i}^2$

Table 2.2: Summary of the conditional distributions of the plug-in and RMT stochastic MSDs.

Penrose pseudoinverse. After simplifying using $\widehat{x}$ and using the natural logarithm operator as a monotonic operation, the GLRT statistic becomes

$$\Lambda(\widehat{w}) = \widehat{w}^H \left(\widehat{U}^H U \Sigma^{1/2} \left(\Sigma^{1/2} U^H \widehat{U} \widehat{U}^H U \Sigma^{1/2}\right)^{\dagger} \Sigma^{1/2} U^H \widehat{U}\right) \widehat{w}.$$

Consider the term $\widehat{U}^H U$. By Proposition 2.5.1 and Conjecture 2.5.1 and by noting that the eigenvectors are unique up to a phase, we have that $\widehat{U}^H U \xrightarrow{\text{a.s.}} BA$ where B is a $\widehat{k} \times k$ matrix and A is a $k \times k$ matrix defined as

$$B_{i\ell} := \begin{cases} b_i = \exp(j\psi_i) & i = \ell \\ 0 & \text{otherwise} \end{cases}, \; A_{i\ell} := \begin{cases} a_i = |\langle u_i, \widehat{u}_i\rangle| & i = \ell \\ 0 & \text{otherwise} \end{cases}.$$

For some ψ_i, b_i denotes the random phase ambiguity in the eigenvector computation (since eigenvectors are unique up to a phase).

The plug-in detector assumes that $A = B = I_{\widehat{k}}$, that is $b_i = 1$ and $|\langle u_i, \widehat{u}_i\rangle| = 1$. However, as seen in Section 2.5, we have knowledge of $|\langle u_i, \widehat{u}_i\rangle|$ which we may exploit in deriving a new detector. Using the notation just developed, the GLRT statistic may be written as

$$\Lambda(\widehat{w}) = \widehat{w}^H BA\Sigma^{1/2}(\Sigma^{1/2}A^H B^H BA\Sigma^{1/2})^{\dagger}\Sigma^{1/2}A^H B^H \widehat{w}.$$

We use (2.17) and Proposition 2.5.1 to estimate $a_i = \sqrt{|\langle u_i, \widehat{u}_i\rangle|_{\text{rmt}}^2}$. Recall that $\widehat{k}_{\text{eff}}$ is an estimate for the number of σ_i^2 above the phase transition and note that $a_i = 0$ when $\sigma_i^2 \leq \sqrt{c}$. Incorporating this into the detector, and noting that A, B, and Σ contain only diagonal elements, the GLRT simplifies to

$$\boxed{\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{\widehat{k}_{\text{eff}}} \widehat{w}_i^2} \tag{2.21}$$

and the deterministic RMT detector is $\Lambda_{\text{rmt}}(\widehat{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{rmt}}$ where the threshold γ_{rmt} is chosen in the usual manner. This addresses the problem posed in Section 2.4.1 for the deterministic test vector setting. We note that this deterministic RMT detector also takes on the form of (2.14). In fact, in the deterministic setting, the plug-in and RMT detectors are both 'energy detectors' and have the same statistic except for the upper bound in the summation. As in the stochastic setting, the principal difference between the RMT test statistic in (2.21) and the plug-in test statistic in (2.12) is the role of $\widehat{k}_{\text{eff}}$ in the former. This is also why the plug-in detector that uses $\widehat{k}_{\text{eff}}$ components exhibits the same performance as the RMT detector, which incorporates knowledge of the subspace estimates.

Detector	Detector Statistic $\Lambda(\widehat{w})$
Plug-in	$\sum_{i=1}^{\widehat{k}} \widehat{w}_i^2$
RMT	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \widehat{w}_i^2$

Table 2.3: Summary of the plug-in and RMT deterministic MSDs. See Sections 2.3.2 and 2.6.2 for derivations.

Detector	Distribution of $\Lambda \vert H_0$	Distribution of $\Lambda \vert H_1$
Plug-in	$\chi^2_{\widehat{k}}$	$\chi^2_{\widehat{k}}\left(\sum_{i=1}^{\widehat{k}_{\text{eff}}} \sigma_i^2 \lvert\langle u_i, \widehat{u}_i\rangle\rvert^2 x_i^2\right)$
RMT	$\chi^2_{\widehat{k}_{\text{eff}}}$	$\chi^2_{\widehat{k}_{\text{eff}}}\left(\sum_{i=1}^{\widehat{k}_{\text{eff}}} \sigma_i^2 \lvert\langle u_i, \widehat{u}_i\rangle\rvert^2 x_i^2\right)$

Table 2.4: Summary of the conditional distributions of the plug-in and RMT deterministic MSDs.

2.7 Theoretical ROC Curve Predictions

We saw in Sections 2.3 and 2.6 that the plug-in and RMT detectors under both testing settings are (exactly or asymptotically) of the form given by (2.14). Thus by answering the ROC curve prediction problem posed in Section 2.4.2, we have characterized the asymptotic (or large system) performance of the detectors considered herein. For the following analysis, we are given n, m, $\widehat{k}$, D, Σ, and x (in the deterministic setting).

We first note that each previously derived detector corresponds to a specific choice of the diagonal matrix D in (2.14), which can be discerned by inspection of Tables 2.1 and 2.3. In what follows, we solve the ROC prediction problem for general D;

direct substitution of the relevant parameters for D will yield the performance curves for individual detectors.

Recall that the ROC curve [1] for a test statistic $\Lambda(\widehat{w})$ is obtained by computing

$$P_D = P(\Lambda(\widehat{w}) \geq \gamma | \widehat{w} \in H_1),\ P_F = P(\Lambda(\widehat{w}) \geq \gamma | \widehat{w} \in H_0) \tag{2.22}$$

for $-\infty < \gamma < \infty$ and plotting P_D versus P_F. To compute these expressions in (2.22) for the deterministic and stochastic test vector setting, we need to characterize the conditional cumulative distribution function (c.d.f.) under H_0 and H_1 for a detector with a test statistic of the form (2.14). The results in Section 2.5, especially an application of Corollary 2.5.1, simplify this analysis in the large system limit. The following analysis shows that the conditional distributions are a weighted sum of chi-square random variables. For general D, we use a previous algorithm to compute the c.d.f. of this weighted sum of chi-square random variables necessary in the ROC derivation. However, for the deterministic plug-in and RMT detectors, the theoretical ROC curves may be computed in closed form.

2.7.1 Stochastic Testing Setting

In the stochastic setting, the conditional distributions of our test samples under each hypothesis are $\widehat{w}|H_0 \sim \mathcal{N}(0, I_{\widehat{k}})$ and $\widehat{w}|H_1 \sim \mathcal{N}(0, \widehat{U}^H U \Sigma U^H \widehat{U} + I_{\widehat{k}})$. Because the covariance matrix of $\widehat{w}|H_0$ is diagonal, for $i = 1, \ldots, \widehat{k}$, $\widehat{w}_i|H_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, which implies that $\widehat{w}_i^2|H_0 \overset{\text{i.i.d.}}{\sim} \chi_1^2$. By Corollary 2.5.1, the covariance matrix of $\widehat{w}|H_1$ is asymptotically diagonal. Therefore for $i = 1, \ldots, \widehat{k}$, $\widehat{w}_i|H_1 \overset{\text{i.i.d.}}{\approx} \mathcal{N}(0, \sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 + 1)$ and

$$\frac{w_i^2|H_1}{\sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 + 1} \sim \chi_1^2.$$

Using this analysis, for a stochastic detector with the form of (2.14), the conditional distributions of its test statistic under each hypothesis are

$$\begin{aligned} \Lambda(\widehat{w})|H_0 &\sim \sum_{i=1}^{\widehat{k}} d_i \chi_{1i}^2 \\ \Lambda(\widehat{w})|H_1 &\sim \sum_{i=1}^{\widehat{k}} d_i (\sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 + 1) \chi_{1i}^2 \end{aligned} \tag{2.23}$$

where χ_{1i}^2 are independent chi-square random variables. Table 2.2 uses this general analysis to summarize the sample conditional distributions of $\Lambda(\widehat{w})$ under each hypothesis for the stochastic plug-in and RMT detectors. An analytical expression for the asymptotic performance in the large matrix limit is obtained by substituting expressions from (2.17) and Propositions 2.5.1 and 2.5.2 for the pertinent quantities in these distributions.

Note that the conditional distributions in (2.23) are a weighted sum of independent chi-square random variables with one degree of freedom. The c.d.f. of a chi-square random variable is known in closed form. However, the c.d.f. of a weighted sum of independent chi-square random variables is not known in closed form. To evaluate (2.22), we use a saddlepoint approximation of the conditional c.d.f. of $\Lambda(\widehat{w})$ by employing the generalized Lugannani-Rice formula proposed in [95]. To then compute a theoretical ROC curve, we sweep γ over $(0,\infty)$ and for each value of γ, we compute the saddlepoint approximation of the conditional c.d.f. under each hypothesis using this method. This generates a set of points (P_F, P_D) which approximate the (asymptotic) theoretical ROC curve.

2.7.2 Deterministic Testing Setting

In the deterministic setting, the conditional distribution of a test sample under H_0 is $\widehat{w}|H_0 \sim \mathcal{N}(0, I_{\widehat{k}})$. The conditional distribution under H_1 is $\widehat{w}|H_1 \sim \mathcal{N}(\widehat{U}^H U\Sigma^{1/2}x, I_{\widehat{k}})$. By Proposition 2.5.1 and Conjecture 2.5.1, $\widehat{U}^H U \xrightarrow{\text{a.s.}} BA$ is asymptotically diagonal with B and A defined in Section 2.6.2. Therefore, $\widehat{w}_i|H_1 \overset{\text{i.i.d.}}{\approx} \mathcal{N}(a_i b_i \sigma_i x_i, 1)$ for $i = 1, \ldots, \widehat{k}$. Using this approximation, for a detector with the form of (2.14), the conditional distributions of its test statistic are

$$\Lambda(\widehat{w})|H_0 \sim \sum_{i=1}^{\widehat{k}} d_i \chi^2_{1i} \quad \text{and} \quad \Lambda(\widehat{w})|H_1 \sim \sum_{i=1}^{\widehat{k}} d_i \chi^2_{1i}(\delta_i) \tag{2.24}$$

where $\delta_i = \sigma_i^2 |\langle u_i, \widehat{u}_i\rangle|^2 x_i^2$ is the non-centrality parameter for the noncentral chi-square distribution. The deterministic plug-in and RMT detectors are a special case of these conditional distributions. For the plug-in detector, $d_i = 1$ for $i = 1, \ldots, \widehat{k}$. For the RMT detector $d_i = 1$ for $i = 1, \ldots, \widehat{k}_{\text{eff}}$ and $d_i = 0$ for $i = \widehat{k}_{\text{eff}} + 1, \ldots, \widehat{k}$.

For the plug-in and RMT detectors, $\Lambda_{\text{plugin}}(\widehat{w})|H_0 \sim \chi^2_{\widehat{k}}$ and $\Lambda_{\text{rmt}}(\widehat{w})|H_0 \sim \chi^2_{\widehat{k}_{\text{eff}}}$. Similarly, $\Lambda_{\text{plugin}}(\widehat{w})|H_1 \sim \chi^2_{\widehat{k}}(\delta)$ and $\Lambda_{\text{rmt}}(\widehat{w})|H_1 \sim \chi^2_{\widehat{k}_{\text{eff}}}(\delta)$ where

$$\delta = \sum_{i=1}^{\widehat{k}} \sigma_i^2 |\langle u_i, \widehat{u}_i\rangle|^2 x_i^2 = \sum_{i=1}^{\widehat{k}_{\text{eff}}} \sigma_i^2 |\langle u_i, \widehat{u}_i\rangle|^2 x_i^2. \tag{2.25}$$

Because $d_i = 1$ for $i = 1, \ldots, \widehat{k}_{\text{eff}}$ for both the plug-in and RMT detectors, the resulting non-centrality parameter is the sum of all the individual non-centrality parameters. An analytical expression for the asymptotic performance in the large matrix limit is obtained by substituting expressions from Proposition 2.5.1 in (2.25). Unlike the stochastic setting, we can obtain a closed form expression for the deterministic plug-in and RMT ROC curves by solving for γ in terms of P_F and substituting this into the expression for P_D in (2.22). Doing so yields

$$\begin{aligned} P_{D_{\text{plugin}}} &= 1 - Q_{\chi^2_{\widehat{k}}(\delta)}\left(Q^{-1}_{\chi^2_{\widehat{k}}}\left(1 - P_F\right)\right) \\ P_{D_{\text{rmt}}} &= 1 - Q_{\chi^2_{\widehat{k}_{\text{eff}}}}(\delta)\left(Q^{-1}_{\chi^2_{\widehat{k}_{\text{eff}}}}\left(1 - P_F\right)\right) \end{aligned} \tag{2.26}$$

where Q is the appropriate c.d.f. function.

2.8 Discussion and Insights

We use numerical simulations to verify our theoretical ROC curve predictions from Section 2.7 that rely on RMT approximations presented in Section 2.5. We also demonstrate properties of the new RMT detectors that we derived in Section 2.6, as described next.

2.8.1 Simulation Protocol

To compute an empirical ROC curve, we first generate a random subspace, U, by taking the first k left singular vectors of a random matrix with i.i.d. $\mathcal{N}(0, 1)$ entries. Using this U, we generate training samples as described in Section 2.2.1 from which we form estimates $\widehat{U}$ and $\widehat{\Sigma}$ from the eigenvalue decomposition of the sample covariance matrix as described in (2.1).

We then generate a desired number of test samples from each hypothesis using either (2.2) or (2.3). For each test sample, we compute the test statistic for each detector. Using Fawcett's [1] 'Algorithm 2', we compute an empirical ROC curve by first sorting the test statistics. At each statistic, we log a (PF, PD) pair by counting the number of lower scores generated from each hypothesis. This is repeated for multiple realizations of U, generating multiple empirical ROC curves. We refer to a single empirical ROC curve corresponding to a realization of U as a trial. We then average the empirical ROC curves over multiple trials using Fawcett's [1] 'Algorithm 4'. This performs threshold averaging by first uniformly sampling the sorted list of all test scores of ROC curves and then computing (PF, PD) pairs in the same way as 'Algorithm 2'.

2.8.2 Convergence and Accuracy of ROC Curve Predictions

The theoretical ROC curve predictions for the plug-in and RMT detectors rely on the asymptotic approximations that ignore finite n and m correction terms. To examine the validity of the asymptotic approximations (Propositions 2.5.1 and 2.5.2, Theorem 2.5.1, and Corollary 2.5.1) and the rate of convergence, we consider two different settings for the stochastic plug-in detector. Figures 2.3(a)-2.3(b) plot three empirical ROC curves for $n = 50, 200, 1000$ as well as the theoretically predicted plug-in ROC curve. Each figure uses different values of k and c but in each case, $\widehat{k} = k$.

For both figures, as n increases, the empirical ROC curves approach the theoretical prediction, attesting to the asymptotic convergence of the RMT approximations. Analyzing the rate of convergence (which we conjecture to be $n^{1/2}$ for fixed k and c) is an important open problem that we shall tackle in future work. As evident in Figures 2.3(a)-2.3(b) the values of k and c play an important roll in the convergence of the empirical ROC curves. For the larger value of k and c (corresponding to the sample starved regime where the amount of training data is smaller than the system dimensionality i.e. $n > m$) the convergence is also slower. We see that for larger k and c, when n is small the empirical ROC curve is not well approximated by the asymptotic theoretical predictions. However, as n increases, the deviation of the empirically generated ROC curve from the theoretically predicted one decreases. Conjecture 2.5.1 suggests that the off diagonal terms of $\widehat{U}^H U$ asymptotically tend to zero. However, in the finite n and m case these terms are $O(1/\sqrt{n})$ and thus not identically zero. For larger rank systems (increased k), there are more of these non-identically-zero terms that worsen the approximation quality for fixed, relatively small n. As n increases, this bias vanishes.

The ROC predictions developed in Section 2.7 also depend on parameters such as Σ and the deterministic vector x. To test the accuracy of the ROC predictions with respect to these parameters, we consider a setting where $\widehat{k} = k = 2$. Figure 2.4(a) plots empirical and theoretical ROC curves for the plug-in and RMT stochastic detectors for $\Sigma = \alpha\,\mathbf{diag}(10, 5)$ for three choices of α. As intuition suggests, smaller values of Σ decrease the performance for both the plug-in and RMT detectors. For each choice of α, the empirical ROC curves match the ROC predictions that rely on random matrix theoretic approximations presented in Section 2.5. Using $\alpha = 1$ or $\alpha = 0.5$ results in $k_{\text{eff}} = k = \widehat{k} = 2$ but using $\alpha = 0.25$ results in $k_{\text{eff}} = 1$. As $\widehat{k} > k_{\text{eff}}$ for this last case, the plug-in detector realizes a performance loss compared to the RMT detector.

In the deterministic setting, x is an additional parameter that affects detector performance. Figure 2.4(b) plots empirical and theoretical ROC curves for the plug-in and RMT deterministic detectors for $\Sigma = \mathbf{diag}(10, 5)$ for three choices of the deterministic test vector x. Larger values of $|x|$ result in better detector performance but for each choice of x, the theoretically predicted ROC curves match their empirical counterparts. As x does not affect the value of $k_{\text{eff}} = \widehat{k} = k = 2$, the plug-in and RMT detectors achieve the same performance because they have identical statistics. For both test vector models, the theoretical ROC curves match the empirical ROC curves thereby validating the accuracy of the random matrix theoretic approximations employed and the accuracy of the saddlepoint approximation to the c.d.f. used in the stochastic derivation.

2.8.3 Effect of the Number of Training Samples

We saw in Section 2.3.3 that finite training data degraded the performance of the plug-in detector relative to that of the oracle detector. The analysis of Section 2.5 mathematically justifies this observation showing that, for a fixed Σ, the number of training samples, m, directly affects k_{eff} via (2.16). While the plug-in detector ignores this analysis, we derived a new RMT detector that accounts for subspace estimation errors due to finite training data. By only using the k_{eff} informative signal subspace components, we hope that the RMT detector will avoid some of the performance loss associated with the plug-in detector. To explore how the number of training samples affects the relative performances of the plug-in and RMT detectors, we first consider the setting where $\widehat{k} = k = 4$ with $\Sigma = \mathbf{diag}(10, 3, 2.5, 2)$.

Figure 2.5(a) investigates the performance when $m = n$ so that $c = 1$ for the stochastic setting. This choice of m results in $k_{\text{eff}} = \widehat{k} = 4$. As expected, the plug-in and RMT detectors achieve relatively the same performance because $\widehat{k} = k_{\text{eff}}$. A similar phenomenon occurs in the deterministic setting. Figure 2.5(b) chooses $20m = n$ so that $c = 20$ and $k_{\text{eff}} = 1$ for the stochastic settings. This corresponds to the sample starved regime where $m < n$. In this second experiment, the plug-in detector becomes suboptimal because it uses $4 = \widehat{k} > k_{\text{eff}} = 1$ subspace components. A similar phenomenon occurs in the deterministic setting. Whenever $k_{\text{eff}} < \widehat{k}$ the RMT detectors avoid some of the performance loss (compared to the oracle detectors) realized by the plug-in detectors. We could have observed this same effect by instead varying Σ as both of these quantities drive the value of k_{eff}. The disagreement between the theoretical and empirical stochastic ROC curves for the plug-in detector is attributed to the finite n and m correction terms, which we have discussed previously.

Figure 2.5 shows that the number of training samples helps to drive the performance of matched subspace detectors. In Section 2.3, we mathematically defined the performance loss of a detector relative to its oracle detector as ϵ in (2.13) and empirically plotted the number of training samples needed to achieve a desired performance loss for the stochastic plug-in detector in Figure 2.2. Figures 2.6(a) and 2.6(b) theoretically plot this same curve for the plug-in and RMT detectors for each testing setting, respectively.

These figures show that when $k_{\text{eff}} < \widehat{k}$, the RMT detector achieves a much smaller performance loss for a fixed number of training samples. Put another way, to achieve the same performance loss, the RMT detectors need significantly fewer training samples when $k_{\text{eff}} < \widehat{k}$. Figure 2.6(a) shows that the stochastic detectors can achieve an arbitrarily small performance loss given a particularly large number of training samples. However, Figure 2.6(b) shows that there is a performance loss limit for the deterministic detectors. As discussed in Section 2.3, this arises because the oracle deterministic detector assumes that x is known. As $m \to \infty$, $\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$, however, the plug-in detector's estimate of $\widehat{x}$ still depends on the noisy observed data y. Therefore, unlike the stochastic detectors that can achieve an arbitrarily small performance loss, the deterministic plug-in and RMT detectors can never achieve the same performance as the deterministic oracle detector.

2.8.4 Effect of $\widehat{k}$

We discussed in Section 2.2.2 that we are given a dimension estimate $\widehat{k}$ when deriving our detector. From our perspective, we don't know how $\widehat{k}$ was estimated (possibly from the training data or by a domain expert) but simply use it when forming our subspace and signal covariance estimates. Figure 2.7 empirically examines the performance of the plug-in and RMT detectors as a function of $\widehat{k}$ for the stochastic setting. A similar phenomenon arises in the deterministic setting. Here, we relax the constraint that $\widehat{k} \geq k$. The figures plot the achieved probability of detection for a constant false alarm rate of 0.01. The result confirms that k_{eff} is the optimal choice for $\widehat{k}$. When the plug-in detectors use $\widehat{k} = k_{\text{eff}}$ they achieve an equivalent performance as that of the RMT detector.

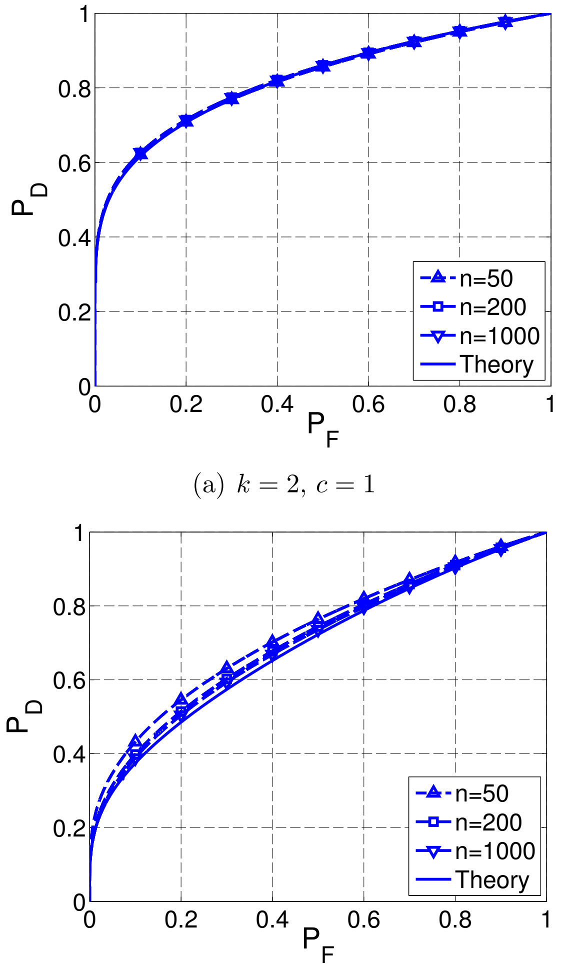
Setting $\widehat{k} < k_{\text{eff}}$ drastically degrades performance for all detectors. In this regime, the plug-in and RMT detectors realize the same ROC performance, demonstrating that quantification and exploitation of the subspace estimation accuracy ($|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}}$ and $\sigma^2_{i_{\text{rmt}}}$), while useful in ROC performance prediction, does *not* noticeably enhance detection performance. When $\widehat{k} > k_{\text{eff}}$, the performances of the plug-in detectors degrade while those of the RMT detectors are stable as if $\widehat{k} = k_{\text{eff}}$. In other words, we do not pay a price for overestimating the subspace dimension with the RMT detectors. This makes sense (and is slightly contrived) because the RMT detectors will only sum to a maximum of k_{eff} indices as evident in (2.20) and (2.21). In many applications, practitioners might employ the "play-it-safe" approach and set $\widehat{k}$ to be significantly greater than k_{eff}. The performance loss caused by adding each uninformative subspace, as seen in Figure 2.7, constitutes evidence to the assertion that overestimating the signal subspace dimension is a bad idea. When $k_{\text{eff}} < k$, even perfectly estimating the subspace dimension (i.e. setting $\widehat{k} = k$) is suboptimal.

2.9 Conclusion

In this chapter, we considered a matched subspace detection problem where the low-rank signal subspace is unknown and must be estimated from finite, noisy, signal-bearing training data. We considered both a stochastic and deterministic model for the testing data. The subspace estimate is inaccurate due to finite and noisy training samples and therefore degrades the performance of plug-in detectors compared to an oracle detector. We showed how the ROC performance curve can be derived from the RMT-aided quantification of the subspace estimation accuracy.

Armed with this RMT knowledge, we derived a new RMT detector that only uses the effective number of informative subspace components, k_{eff}. Plug-in detectors that use the uninformative components will thus incur a performance degradation, relative to the RMT detector. In settings where a practitioner might play-it-safe and set $\widehat{k} > \widehat{k}_{\text{eff}}$, the performance loss in significant (see Figures 2.6(a) and 2.6(b) for a demonstration of how much training data such a play-it-safe plug-in detector would need to match the performance of a k_{eff}-tuned RMT detector). This highlights the importance of robust techniques [89, 90, 91] for estimating k_{eff} in subspace based detection schemes as opposed to estimating k, particularly in the regime where $k_{\text{eff}} < k$. We showed in Tables 2.2 and 2.4 that the distributions of the test statistics could be expressed as a weighted sum of independent chi-squared random variables. The associated ROC curves can then be computed using a saddlepoint approximation.

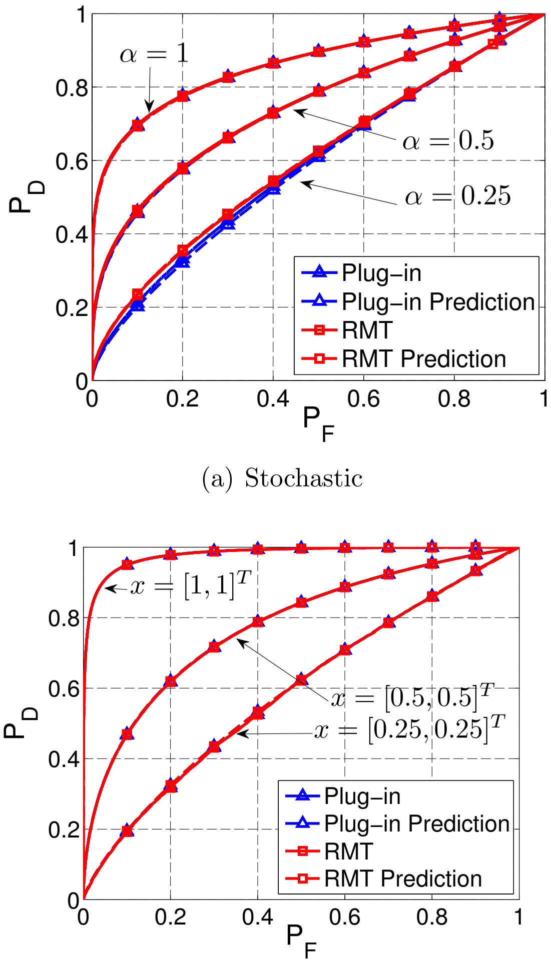
The results in this chapter can be extended in several directions. We note that the stochastic detector setting assumed normally distributed training and test data. We can extend the analysis to the Gaussian training data but non-Gaussian test vector setting by 'integrating-out' the deterministic detector performance curves with respect to the non-Gaussian distribution of the test-vector. Our results relied on characterization of the quantity $\langle u_j, \widehat{u}_i \rangle$. Thus analogous performance curves can be obtained for any alternate training data models for which this quantity can be analytically quantified. To that end, the results in [86] facilitate such an analysis for a broader class of models including the correlatted Gaussians training data setting. An extension to the missing data setting might follow a similar approach and appears within reach. Aspects related to rate of convergence are open and will be the subject of future work.



(a) $k = 2$, $c = 1$

(b) $k = 4$, $c = 10$

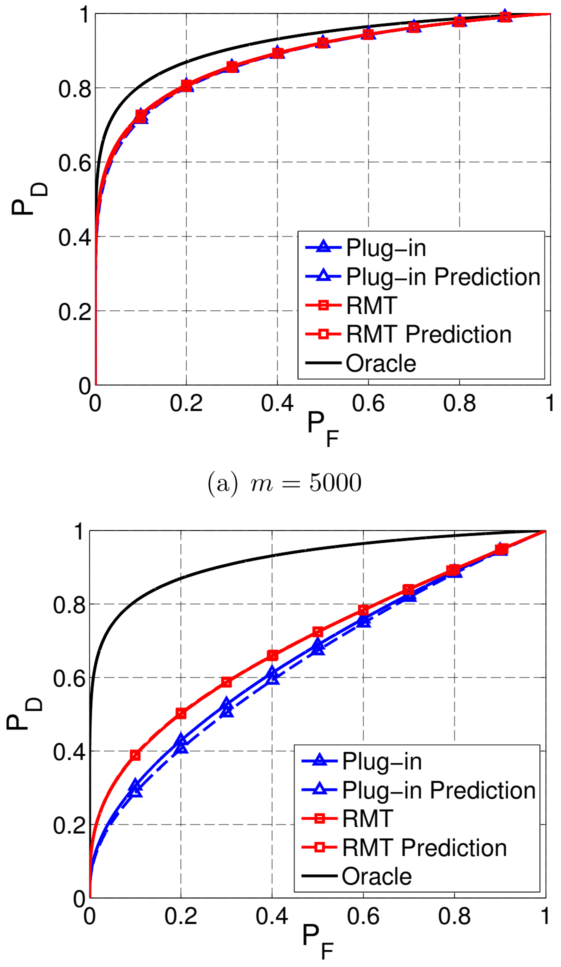
Figure 2.3: Empirical and theoretical ROC curves for the stochastic plug-in detector. Empirical ROC curves were simulated using 10000 test samples and averaged over 50 trials using algorithms 2 and 4 of [1]. (a) $\Sigma = \textbf{diag}(10, 2)$, $c = 1$, $\widehat{k} = k = 2$ so that $k_{\text{eff}} = 2$. (b) $\Sigma = \textbf{diag}(10, 2, 0.5, 0.1)$, $c = 10$, $\widehat{k} = k = 4$ so that $k_{\text{eff}} = 1$. Each figure plots empirical ROC curves for $n = 50, 200, 1000$. Theoretical ROC curves were computed as described in Section 2.7. As n increases, the empirical ROC curves approach the theoretically predicted one. However, this convergence is slower for larger k and c.



(a) Stochastic

(b) Deterministic

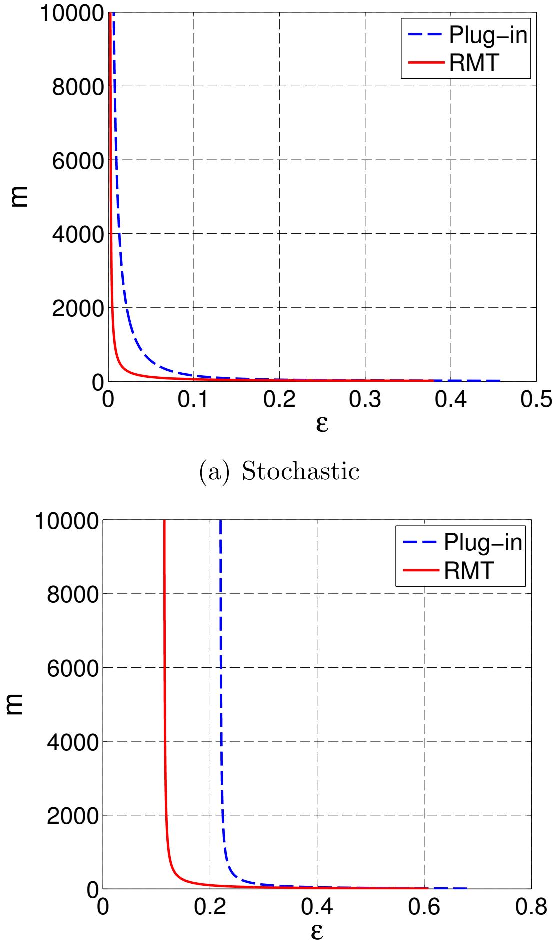
Figure 2.4: Empirical and theoretical ROC curves for the plug-in and RMT detectors. Empirical ROC curves were simulated using 10000 test vectors and averaged over 100 trials with $n = 1000$, $m = 500$, and $\Sigma = \alpha\,\mathbf{diag}\,(10, 5)$. The theoretical ROC curves were computed as described in Section 2.7. (a) Stochastic testing setting. Results are plotted for $\alpha = 1, 0.5, 0.25$. For $\alpha = 1$ and $\alpha = 0.5$, $\widehat{k} = k = k_{\text{eff}} = 2$ by (2.16). For $\alpha = 0.25$, $k_{\text{eff}} = 1$.Since $\widehat{k} > k_{\text{eff}}$ when $\alpha = 0.25$, we observe a performance gain when using the RMT detector. (b) Deterministic testing setting. Results are plotted for $\alpha = 1$ so that $k_{\text{eff}} = 2$. Three values of the deterministic signal vector were used: $x = [1, 1]^T$, $x = [0.5, 0.5]^T$, and $x = [0.25, 0.25]^T$. The resulting ROC curves depend on the choice of x, however, since $\widehat{k} = k_{\text{eff}}$, the plug-in and RMT detector achieve the same performance for all x. For both the stochastic and deterministic detectors, the theoretically predicted ROC curves match the empirical ROC curves, reflecting the accuracy of Corollary 2.5.1 and the Lugannani-Rice formula.



(a) $m = 5000$

(b) $m = 250$

Figure 2.5: Empirical and theoretical ROC curves for the plug-in and RMT stochastic detectors. Empirical ROC curves were computed with 10000 test samples and averaged over 100 trials. Here, $n = 5000$, $\widehat{k} = k = 4$ and $\Sigma = \mathbf{diag}(\mathbf{10}, \mathbf{3}, \mathbf{2.5}, \mathbf{2})$. The empirical oracle ROC curve is provided for relative comparison purposes. (a) $m = 5000$ so that $c = 1$ and $k_{\text{eff}} = \widehat{k} = 4$. The plug-in and RMT detectors achieve relatively the same performance. (b) $m = 250$ so that $c = 20$ and $k_{\text{eff}} = 1 < \widehat{k} = 4$. The RMT detector avoids some of the performance loss realized by the plug-in detector. As seen in Section 2.3, limited training samples degrades detector performance. However, the new RMT detector does not suffer as badly as the plug-in detector because it accounts for subspace estimation errors due to finite training data. The disagreement between the theoretical and empirical ROC curves is attributed to finite dimensionality.



(a) Stochastic

(b) Deterministic

Figure 2.6: Theoretically determined number of training samples, m, needed to achieve a desired performance loss, ϵ, as defined in (2.13). The required false alarm rate is $P_F = 0.1$ with $n = 200$, $\Sigma = \mathbf{diag}(10, 0.1)$, and $\widehat{k} = k = 2$. (a) Results for the stochastic detectors. We see that for a given ϵ, the new RMT detector requires less training samples. (b) Results for the deterministic detectors when $x = [0.75, 0.75]^T$. Again, for a given ϵ, the new RMT detector requires less training samples. In the deterministic setting, the limiting performance loss is different (and non-zero) for the plug-in and RMT detectors. This arises in estimation errors of x in the GLRT.

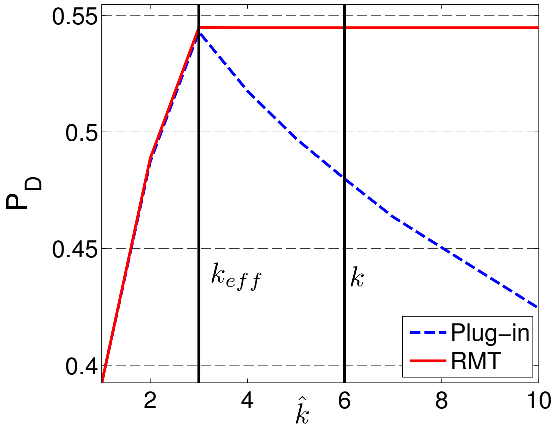


Figure 2.7: Empirical exploration of the achieved probability of detection, P_D, for a fixed probability of false alarm, $P_F = 0.01$, for various $\widehat{k}$. Empirical ROC curves were computed using 10000 test samples and averaged over 100 trials with $n = 1000$, $m = 500$, and $\Sigma = \mathbf{diag}(\mathbf{10}, \mathbf{5}, \mathbf{4}, 0.75, 0.5, 0.25)$ so that $k_{\text{eff}} = 3$. Results for the stochastic detectors. The optimal $\widehat{k}$ resulting in the largest P_D is not the true k, but rather k_{eff}.

CHAPTER III

Extensions of Deterministic Matched Subspace Detectors: Missing Data and Useful Subspace Components

3.1 Introduction

A ubiquitous problem in signal and array processing is designing multi-dimensional signal-plus-noise versus noise detectors. In such applications, an observation w may belong to either the noise only hypothesis (H_0) or the signal-plus-noise hypothesis (H_1), via the model

$$w = \begin{cases} z & w \in H_0 \\ \delta + z & w \in H_1, \end{cases} \tag{3.1}$$

where δ is the unknown signal vector and z is additive noise. When modeling δ as a fixed deterministic vector and z as Gaussian noise, the standard detector statistic is $\|w\|^2$, the squared norm (magnitude) of the observed vector w. This detector is commonly referred to as an energy detector because the squared norm measures the amount of energy contained in the observation. Energy detectors arise in applications such as incoherent radar detection [96], Global Navigation Satellite Systems (GNSS) [97], and MIMO radar [98, 99].

In this chapter, we analyze the performance of the energy detector, starting from first principles. Using a receiver operating characteristic (ROC) performance analysis, we investigate the conditional distributions of the energy detector's test statistic and showcase how these distributions shift depending on the number of signal components that the energy detector uses. We saw in the previous chapter that including more than the k_{eff} number of subspace components degrades detector performance. In this chapter we show that, surprisingly, even if a signal component is one of the k_{eff}

informative components, if its signal strength is too small, including it in an energy detector actually degrades detector performance. Using this observation, we define the number of signal components that maximize detector performance as k_{useful}, which is dependent on the desired false alarm rate of the energy detector. Our goal is to bring this phenomenon into focus so that effort can be spent on designing better real world detectors.

We are motivated by the more specific problem of deterministic matched subspace detection. A matched subspace detector (MSD) is commonly used to detect a signal buried in high dimensional noise under the assumption that the signal lies in a low-rank signal subspace. Many applications in signal and array processing use such low-rank signal-plus-noise models, including incoherent radar detectors [96], direction detection [100, 101, 102], GNSS [97], MIMO radar [103, 98, 99] and target detection [104]. A deterministic signal model, which assumes that the target signal lies at an unknown but fixed point in the signal subspace, occurs in array processing [68, 71, 70], MIMO radar [105], and cognitive radio [106]. When the signal subspace is known *a priori*, the performance of such deterministic MSDs has been extensively studied (see, for example, [74, 75, 69]). In a recent paper [107], we considered the performance of a MSD in the alternative setting where the signal subspace is unknown and estimated from finite, noisy, signal-bearing training data.

Under a deterministic signal model and appropriate noise assumptions, a MSD is an energy detector that projects a observation onto this estimated signal subspace and uses the squared norm of the projection as the detector's statistic. In [107], we used random matrix theory (RMT) to showcase that using more than the k_{eff} *informative* subspace components decreases detector performance. In this chapter, we show that even though a subspace component may be informative (as defined by k_{eff}), including it in a detector may degrade performance. Using exactly the k_{useful} subspace components results in the best detector performance. However, as k_{useful} is computed assuming knowledge of the unknown deterministic vector, k_{eff} provides a realizable upper bound for k_{useful}.

Finally, we consider the deterministic MSD setting where the training data is noisy *and* has missing entries. The missing entry context is motivated in [108] by distributed detection scenarios where it might be prohibitive to collect and transmit only a (randomly chosen) fraction p of the training data entries. Alternately one might think of $1 - p \in (0, 1)$ as a compression factor as in compressed sensing. We precisely quantify the performance of the MSD with missing data. We uncover a phase transition phenomenon by showing that there is a critical fraction, p_{crit}, which is a simple function of the eigen-SNR, the number of training samples, and the number of sensors, below which detection performance deteriorates to random guessing. Compressing the training dataset below this critical fraction is undesirable.

The chapter is organized as follows. In Section 3.2, we formulate the standard signal versus noise detection problem and derive the standard energy detector. We discuss the energy detector's conditional distributions, define k_{useful}, and discuss its properties in Section 3.3. In Section 3.4, we apply these insights to deterministic MSDs and highlight the relationship between k_{eff} and k_{useful} through numerical simulations. In Section 3.5, we discuss the weighted energy detector as a natural extension to this work. We extend the results to the setting where our original data matrix may have missing data in Section 3.6. Finally, we provide concluding remarks in Section 3.8.

3.2 Problem Formulation

We wish to design a detector that discriminates between the H_0 hypothesis that an observation is purely noise and the H_1 hypothesis that the observation contains an unknown signal. We model the observation $w \in \mathbb{R}^{k\times 1}$ as in (3.1) where $\delta = [\delta_1, \ldots, \delta_k]^T \in \mathbb{R}^{k\times 1}$, with $\delta_i \neq 0$, is an unknown deterministic vector, $z \sim \mathcal{N}(0, I_k)$ is additive white Gaussian noise (AWGN), and k is known. See [96, 100, 97, 103, 98, 99, 104, 101, 102] for similar signal-plus-noise models in signal and array processing. In the Neyman-Pearson detection setting (see [93]), the detector for this data model is the likelihood ratio test (LRT)

$$\Lambda(w) = \frac{f(w \,|\, H_1)}{f(w \,|\, H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{3.2}$$

Here $f(\cdot)$ is the appropriate conditional probability density function (p.d.f.) of the observation and η is a scalar threshold set so that $\mathbb{P}(\Lambda(w) > \eta \,|\, w \in H_0) = \alpha$ where $\alpha \in [0, 1]$ is a desired false alarm rate.

The conditional distributions of w modeled as in (3.1) are $w|H_0 \sim \mathcal{N}(0, I_k)$ and $w|H_1 \sim \mathcal{N}(\delta, I_k)$. However, as δ is unknown, we cannot substitute the p.d.f. of $w|H_1$ into (3.2). Instead, we use the generalized LRT (GLRT), which maximizes $f(w|H_1)$ with respect to any unknown parameters. The GLRT for our problem is

$$\Lambda(w) = \frac{\max_\delta f(w \,|\, H_1)}{f(w \,|\, H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta.$$

The conditional p.d.f. of w under the H_1 hypothesis is

$$f\left(w \,|\, H_1\right) = \left(2\pi\right)^{-k/2} \exp\left\{-\frac{1}{2}\left(w-\delta\right)^T\left(w-\delta\right)\right\}.$$

This p.d.f. is maximized when $\delta = w$ with the maximum value of $(2\pi)^{-k/2}$. Substituting this into the GLRT yields

$$\Lambda(w) = \exp\{\frac{1}{2}w^T w\}$$

Taking the natural logarithm results in the test statistic

$$\Lambda_{\text{energy}}(w) = w^T w = \sum_{i=1}^{k} w_i^2 \tag{3.3}$$

where $w = [w_1, \ldots, w_k]^T$. This is an energy detector as its test statistic sums the energy residing in each component (or dimension) of the given observation.

3.2.1 ROC Curve Analysis

To compare the performance of multiple detectors, we will compare their receiver operating characteristic (ROC) curves. A ROC curve is a collection of points (P_F, P_D) where for $-\infty < \eta < \infty$,

$$\begin{aligned} P_F &= \mathbb{P}\left(\Lambda(w) > \eta \,|\, w \in H_0\right), \\ P_D &= \mathbb{P}\left(\Lambda(w) > \eta \,|\, w \in H_1\right). \end{aligned} \tag{3.4}$$

For $0 \leq P_F \leq 1$ we want to express the probability of detection P_D as a function of the false alarm rate, P_F, while noting that P_F is a function of η. To make analytical progress, we assume that δ is known for ROC derivations. First, we compute the conditional distributions of the statistic in (3.3). The conditional distributions of the components in w are simply $w_i|H_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0,1\right)$ and $w_i|H_1 \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\delta_i,1\right)$. Therefore, $w_i^2|H_0 \overset{\text{i.i.d.}}{\sim} \chi_1^2$ and $w_i^2|H_1 \overset{\text{i.i.d.}}{\sim} \chi_1^2\left(\delta_i^2\right)$ where χ_1^2 is a chi-square random variable with one degree of freedom and $\chi_1^2(\delta_i^2)$ is a non-central chi-square random variable with one degree of freedom and non-centrality parameter δ_i^2. As each component w_i is independent,

$$\begin{aligned} \Lambda(w)|H_0 &\sim \chi_k^2, \\ \Lambda(w)|H_1 &\sim \chi_k^2\left(\delta^T\delta\right), \end{aligned} \tag{3.5}$$

where χ_k^2 is a chi-square random variable with k degrees of freedom and $\chi_k^2(\delta^T\delta)$ is a non-central chi-square random variable with k degrees of freedom and non-centrality parameter $\delta^T\delta = \sum_{i=1}^k \delta_i^2$. Armed with these characterizations in (3.5) and solving for η in (3.4), we can relate P_D to P_F using the expression

$$P_{D_{\text{energy}}}(P_F, k) = 1 - Q_{\chi_k^2(\lambda_k)}\left(Q^{-1}_{\chi_k^2}\left(1 - P_F\right)\right). \tag{3.6}$$

In (3.6), $Q_{\chi_k^2}(\lambda_k)$ is the cumulative distribution function (c.d.f.) of a non-central chi-square random variable with k degrees of freedom and non-centrality parameter $\lambda_k = \sum_{i=1}^k \delta_i^2$ and $Q_{\chi_k^2}$ is the c.d.f. of a chi-square random variable with k degrees of freedom. See [96, 97, 99] for similar ROC performance curve derivations.

3.2.2 Problem Statement

As practitioners, we can control which signal components that the energy detector in (3.3) uses. Without loss of generality, we assume that the entries of δ are ordered (i.e. $|\delta_1| \geq |\delta_2| \geq \ldots |\delta_k|$). With this assumption, we can decide how many signal components, d, to use in the energy detector

$$\Lambda_d(w) = \sum_{i=1}^{d} w_i^2. \tag{3.7}$$

Specifically, we wish to answer the following question:

> Given a signal vector δ and a desired false alarm rate P_F, how many signal components, d, maximize $P_{D_{\text{energy}}}(P_F, d)$ in (3.6) for an energy detector with the form of (3.7) derived from observations as in (3.1)?

Answering this question will provide some surprising results. We will show that if the components δ_i are too small in magnitude, including them in a detector actually degrades performance. The setting where δ_i equals zero is a special case where not including it will always yield a performance gain.

3.3 Useful Components In Energy Detectors

In this section, we answer the question posed at the end of Section 3.2 by defining k_{useful}, the number of useful signal components. We show that k_{useful} is dependent on δ and the desired false alarm rate P_F. We provide some intuition behind our definition by discussing how the conditional distributions of the energy detector's test statistic shift when adding additional components. If an additional signal component further separates the conditional distributions, it is one of the k_{useful} components in detection; otherwise, including that component would degrade detector performance. Of particular importance, k_{useful} may be less than the inherent dimension, k, of the observed data, even when $\delta_i \neq 0$.

3.3.1 Definition and Computation of k_{useful}

We define the number of useful detection components at a false alarm rate P_F as the solution to the following optimization problem

$$k_{\text{useful}} = \operatorname*{argmax}_{d \in \{1,\ldots,k\}} P_{D_{\text{energy}}}(P_F, d) \tag{3.8}$$

where $P_{D_{\text{energy}}}(P_F, d)$ is defined in (3.6). This is the optimal number of components to include in an energy detector in (3.7) to maximize detector performance. Using exactly k_{useful} components includes all components that improve detection ability and excludes all components that degrade detection ability.

To determine k_{useful}, we propose the greedy "algorithm" in Figure 3.1. The algorithm relies on the fact that the components of δ are ordered (i.e. $|\delta_1| \geq \ldots |\delta_k|$). It adds one component at a time and searches for the last component that resulted in an increase in detection ability. This algorithm relies on knowledge of δ and so by definition k_{useful} is an oracle quantity. Therefore, a realizable detector using exactly k_{useful} components is currently beyond reach. Estimating k_{useful} is a topic for future work and may involve placing a prior distribution on the test vector. In Section 3.4, we discuss using the effective number of subspace components, k_{eff}, as an estimate for k_{useful}.

3.3.2 Discussion of Test Statistic Distributions

In order to provide intuition behind the definition of k_{useful}, we examine the conditional distribution of the test statistic in (3.7):

$$\begin{aligned} \Lambda_d(w) \,|\, H_0 &\sim \chi^2_d, \\ \Lambda_d(w) \,|\, H_1 &\sim \chi^2_d(\lambda_d) \end{aligned} \tag{3.9}$$

where

$$\lambda_d = \sum_{i=1}^{d} \delta_i^2. \tag{3.10}$$

```
Input: $P_F$, $\delta$
Compute $P_D(P_F, 1)$ from (3.6)
for $h = 2, \ldots, k$ do
    Compute $P_D(P_F, h)$ from (3.6)
    if $P_D(P_F, h) < P_D(P_F, h-1)$ then
        $k_{\text{useful}} = h - 1$
        Return: $k_{\text{useful}}$
$k_{\text{useful}} = k$
Output: $k_{\text{useful}}$
```

Figure 3.1: Algorithm to determine k_{useful}. This is computable in an oracle setting where δ is known.

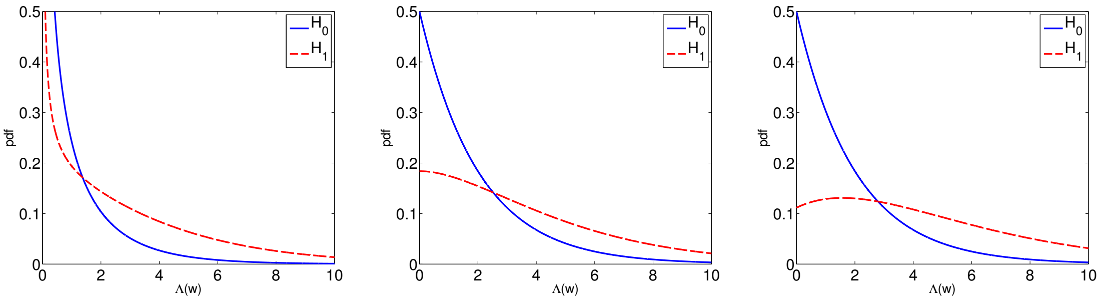
Clearly, both distributions and the non-centrality parameter depend on d. Therefore, a closed form expression for k_{useful} is not possible and we rely on the greedy algorithm in Figure 3.1. Adding an additional component presents a tradeoff between adding δ_i^2 to the non-centrality parameter and adding 1 to the degrees of freedom in the c.d.f's in (3.9).

This tradeoff becomes more evident when using (3.6) to rewrite the optimization problem in (3.8) as

$$k_{\text{useful}} = \underset{d \in \{1,\ldots,k\}}{\operatorname{argmin}} \, Q_{\chi_d^2(\lambda_d)} \left(Q_{\chi_d^2}^{-1} \left(1 - P_F\right) \right). \tag{3.11}$$

By fixing the signal distribution $Q_{\chi_d^2(\lambda_d)}$, solving (3.11) is equivalent to minimizing $Q_{\chi_d^2}^{-1}(1 - P_F)$, which is achieved when $d = 1$. This minimizes the variance contribution from the noise distribution. However, by fixing the noise distribution $Q_{\chi_d^2}$, solving (3.11) is equivalent to minimizing $Q_{\chi_d^2(\lambda_d)}(\cdot)$, which is achieved when $d = k$. This maximizes the variance contribution from the signal distribution. The solution to (3.11) is dependent on how much each additional component contributes to the overall non-centrality parameter. If the contribution is large enough, the added variance in the noise distribution from the extra degree of freedom is overcome by the distribution shift induced by the increase in non-centrality parameter.

To illustrate how the conditional distributions shift when adding components to the energy detector, Figure 3.2 plots the distributions of $\Lambda_d(w)|H_0$ and $\Lambda_d(w)|H_1$ for three choices of d and λ_d. Figure 3.2(a) sets $d = 1$ and $\lambda_d = 2$ and is used as a baseline. Figure 3.2(b) increases the number of components to $d = 2$ while keeping the non-centrality parameter fixed at $\lambda_d = 2$. The added component increases



(a) $d = 1$, $\lambda_d = 2$ (b) $d = 2$, $\lambda_d = 2$ (c) $d = 2$, $\lambda_d = 3$

Figure 3.2: Probability density function (p.d.f.) of $\Lambda(w) \,|\, H_0$ and $\Lambda(w) \,|\, H_1$ for three combinations of the number of components d and non-centrality parameter λ_d. (a) Baseline: $d = 1$, $\lambda_d = 2$ (b) Increases d but keeps λ_d fixed. The distributions are less separable. (c) Increases both d and λ_d. The distributions are more separable.

the noise variance by 2. However, as there is no increase in the non-centrality parameter, the signal variance also increases by 2. Thus, the signal-to-noise ratio is effectively decreased. Therefore, the second component causes the conditional distributions to become more similar and thus degrades detector performance; therefore, $k_{\text{useful}} = 1$. Figure 3.2(c) keeps the number of components at $d = 2$ but increases the non-centrality parameter to $\lambda_d = 3$. In this setting, the increase in non-centrality parameter increases the signal variance, which overcomes the resulting increase in noise variance. The second component further separates the conditional distributions and improves detection performance; therefore, $k_{\text{useful}} = 2$. Figure 3.3 plots the corresponding ROC curves for the three choices of parameters in Figure 3.2. When adding a component causes the conditional distributions to better separate as in Figure 3.2(c), the resulting ROC curve shows an improvement in detection. *For an additional component to be one of the k_{useful} components, the resulting increase in noise variance must be overcome by a sufficiently large enough increase in non-centrality parameter.*

Finally, we explore the minimum increase in non-centrality parameter needed to improve detection ability. Consider a setting with $d = 1$ component and corresponding non-centrality parameter λ_1. Let λ_2 be the resulting non-centrality parameter by adding a second component, $d = 2$, and let $\Delta\lambda = \lambda_2 - \lambda_1$ be the resulting increase in non-centrality parameter. Figure 3.4 plots the minimum increase in non-centrality parameter needed to improve detection as a function of λ_1 for a few choices of P_F. If the increase in non-centrality parameter exceeds this minimum threshold, that component is one of the k_{useful} components.

We observe that the minimum increase in non-centrality parameter is dependent both on the desired false alarm rate, P_F, and the first non-centrality parameter, λ_1.

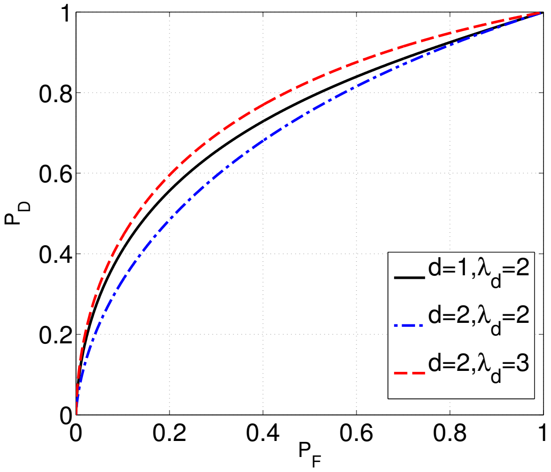


Figure 3.3: The corresponding ROC curves to the three choices of d and λ_d in Figure 3.2. ROC curves were generated from (3.4). When adding an additional subspace component, the non-centrality parameter must increase sufficiently in order to achieve improved detection.

The minimum increase in non-centrality parameter is larger for smaller false alarm rates and is larger for larger λ_1. This is intuitive because larger values of λ_1 separate the conditional distributions very well, indicating that the first component is an excellent discriminant between the two hypotheses H_0 and H_1. For the second component to improve detection ability, its contribution to the non-centrality parameter must be larger for larger λ_1. Otherwise, the second component only adds more noise to the detector. More generally, for the ith component to be one of the k_{useful} components, δ_i^2 must exceed a critical threshold that is dependent on $\sum_{j=1}^{i-1} \delta_j^2$.

3.4 Useful Components in Deterministic Matched Subspace Detectors

This section will apply the results in Section 3.3 about useful components to deterministic matched subspace detection. In this detection setting, we are given a high dimensional test observation and wish to discriminate between the H_0 hypothesis that the observation is purely noise and the H_1 hypothesis that the observation contains a low-rank-k signal that lies at a fixed point in an unknown subspace. To design a detector, we have access to a training dataset of signal bearing observations. We assume that the training data was collected in a variety of representative experimental conditions, allowing each observation's signal component to lie at a different location in the signal subspace. This setup is the similar to that in [107] and the

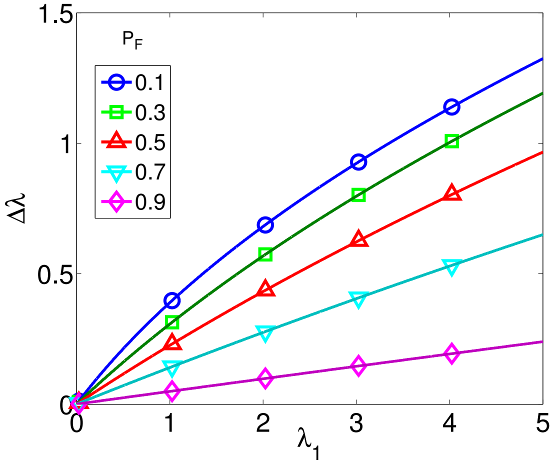


Figure 3.4: Minimum increase in non-centrality parameter necessary for increased detector performance. Results are shown for multiple choices of P_F. λ_1 indicates the non-centrality parameter when $d = 1$ and $\Delta\lambda$ indicates the increase in non-centrality parameter when increasing the number of components to $d = 2$.

resulting standard matched subspace detector is an energy detector with the same form as (3.7). We use random matrix theory to determine the number of informative subspace components, k_{eff}, which is an upper bound for k_{useful}. Through a numerical example, we demonstrate the relationship between the standard plug-in detector using exactly k subspace components, a detector using k_{eff} subspace components, and a detector using exactly k_{useful} subspace components.

3.4.1 Training Data Model

Let $U = [u_1, \dots, u_k] \in \mathbb{R}^{n \times k}$ be an unknown signal subspace matrix with pairwise orthonormal columns $u_i \in \mathbb{R}^{n \times 1}$. To estimate U, we are provided a dataset containing m signal-bearing training vectors $y_i \in \mathbb{R}^{n \times 1}$, $i = 1, \dots, m$, modeled as

$$y_i = Ux_i + z_i \tag{3.12}$$

where $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n)$ and $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$ where $\Sigma = \textbf{diag}(\sigma_1^2, \dots, \sigma_k^2) \in \mathbb{R}^{k \times k}$ with $\sigma_1 > \sigma_2 > \cdots > \sigma_k > 0$ known. For each observation, x_i and z_i are independent. In the training data, x_i is modeled stochastically to represent the variety of conditions under which the training data may be collected. We assume that the dimension, k, of our subspace is known and that $k \ll n$ so that we have a low-rank signal embedded in a high-dimensional observation vector. Applications in which training datasets arise include MIMO radar [103], GNSS receivers [97], source localization [109], DOA

[102], and target detection [104]. In such applications, we may think of the entries of y_i received data from an antenna array, U as the channel response matrix, x_i as the transmitted waveform, Σ as the signal-to-noise ratio (SNR) matrix, and z_i as additive noise.

3.4.2 Testing Data Model

In the testing setting, we are given an unlabeled observation $y \in \mathbb{R}^{n\times 1}$ modeled as

$$y = \begin{cases} z & y \in H_0 : \text{ Noise only} \\ U\Sigma^{1/2}x + z & y \in H_1 : \text{ Signal-plus noise} \end{cases}, \tag{3.13}$$

where U, Σ, and z are modeled the same as the training data as described in Section 3.4.1. However, for the test observations, $x = [x_1, \dots, x_k]^T$ is a non-random, unknown deterministic vector. Thus the signal, $U\Sigma^{1/2}x$, lies at a fixed point in the unknown subspace. Note that Σ controls the SNR of each subspace component.

3.4.3 Subspace Estimation and Accuracy

In the testing model, the signal subspace U is unknown and must be estimated from the provided training data. Given the signal bearing training data

$$Y = [y_1, \dots, y_m] \in \mathbb{R}^{n\times m},$$

we form the sample covariance matrix $S = \frac{1}{m}YY^T$. The covariance matrix of a training observation is $\mathbb{E}\left[y_i y_i^T\right] = U\Sigma U^T + I_n$ and it follows that the (classical) maximum likelihood estimates (in the many-sample, small matrix setting) for U is given by

$$\widehat{U} = [\widehat{u}_1 \dots \widehat{u}_k] \tag{3.14}$$

where $\widehat{u}_1, \dots, \widehat{u}_k$ are the eigenvectors of S corresponding to the largest k eigenvalues [92] .

In any real world setting, we have finite training data and finite SNR. Therefore, $\widehat{U}$ is inaccurate and degrades the performance of any detector that relies on it. Proposition 5.1 of [107] characterized the asymptotic accuracy of the eigenvectors of the sample covariance matrix S stating that as $n, m \to \infty$ with $c = n/m$

$$|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{\text{a.s.}} \begin{cases} \dfrac{\sigma_i^4 - c}{\sigma_i^4 + \sigma_i^2 c} & \text{if } \sigma_i^2 > \sqrt{c} \\ 0 & \text{otherwise} \end{cases}. \tag{3.15}$$

We note that $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence. The key insight to (3.15) is that only the eigenvectors corresponding to the signal variances, σ_i^2, lying above the phase transition $\sqrt{c}$ are *informative*. Following [107, 83], we define the effective number of (asymptotically) identifiable subspace components k_{eff} as:

$$k_{\text{eff}} = \text{Number of } \sigma_i^2 > \sqrt{c}. \tag{3.16}$$

3.4.4 Plug-in and RMT Detectors

If U was known, the matched subspace detector is the GLRT using the test statistic (see [74, 75, 110])

$$\Lambda(w) = y^T U U^T y = w^T w$$

where $w = U^T y \in \mathbb{R}^{k \times 1}$. This is clearly an energy detector of the same form as (3.7) where each component of w is the energy of y residing in that direction of the subspace. However, this detector is not realizable as U is unknown and so we substitute $\widehat{U}$ for the unknown U, resulting in the plug-in detector [107]

$$\Lambda_{\text{plugin}}(\widehat{w}) = \widehat{w}^T \widehat{w} = \sum_{i=1}^{k} \widehat{w}_i^2 \tag{3.17}$$

where $\widehat{w} = \widehat{U}^T y$ is the projection of the test observation onto the estimated subspace. Similar plug-in techniques using sample covariance matrices occur in direction detection [100] and GNSS receivers [97]. The plug-in detector incorrectly assumes that $\widehat{U} = U$ and consequently that all k subspace components are informative. To avoid some of the performance loss of the plug-in detector associated with including uninformative subspace components, we derived a RMT detector that only includes the informative subspace components (see [107] for a derivation). The RMT detector statistic is

$$\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{k_{\text{eff}}} \widehat{w}_i^2. \tag{3.18}$$

Clearly, both the plug-in and RMT detectors are energy detectors of the form in

(3.7) and so we may use (3.6) to analyze the performance of each detector. In the MSD application, $\delta_i = \sigma_i|\langle u_i, \widehat{u}_i\rangle|s_i x_i$ where $s_i \in \{1, -1\}$ represents the random phase ambiguity in the eigenvector computation. Therefore, the non-centrality parameter for this problem is

$$\lambda_d = \sum_{i=1}^{d} \sigma_i^2 |\langle u_i, \widehat{u}_i\rangle|^2 x_i^2 \tag{3.19}$$

where the plug-in detector uses $d = k$ subspace components and the RMT detector uses $d = k_{\text{eff}}$ subspace components. In [107], we demonstrated that the plug-in detector is suboptimal and that the RMT detector will always achieve the same or better performance.

3.4.5 Relationship between k_{useful} and k_{eff}

We first note that $k_{\text{useful}} \leq k_{\text{eff}}$. If a subspace component is uninformative ($|\langle u_i, \widehat{u}_i\rangle|^2 = 0$ as determined by (3.16)), that component contributes nothing to the non-centrality parameter as defined in (3.19). From the analysis in Section 3.3, including this subspace component in a detector would degrade detector performance. Therefore, a subspace component must be informative to be one of the k_{useful} subspace components.

However, the number of useful subspace components may be strictly less than the number of informative subspace components. As demonstrated in Figure 3.4, when adding an additional subspace component, the increase in non-centrality parameter must exceed a minimum value. Examining (3.19), the non-centrality parameter depends on Σ, x, and the accuracy of the eigenvectors of the sample covariance matrix ($|\langle u_i, \widehat{u}_i\rangle|^2$). Depending on these values, adding the i-th component may not increase the non-centrality parameter enough to improve detection, even when the subspace component is informative ($|\langle u_i, \widehat{u}_i\rangle|^2 > 0$). Thus, it is possible for informative subspace components to not be useful in detection.

Besides the desired false alarm rate, P_F, k_{useful} also depends on Σ and x for the matched subspace detector. Larger values of $|x_i|$ and σ_i lead to larger non-centrality parameters as defined in (3.19), making it more likely for that component to be useful. This is intuitive because the larger $|x_i|$ and σ_i force the mean of the conditional distribution of $\widehat{w}_i|H_1$ further from 0, which is the mean of the conditional distribution of $\widehat{w}_i|H_0$. If we instead fix Σ, n , and x and allow m to change, we observe that more training data increases the accuracy the subspace estimate as seen in (3.16). Therefore, increasing m increases δ_i, which may make subspace components useful.

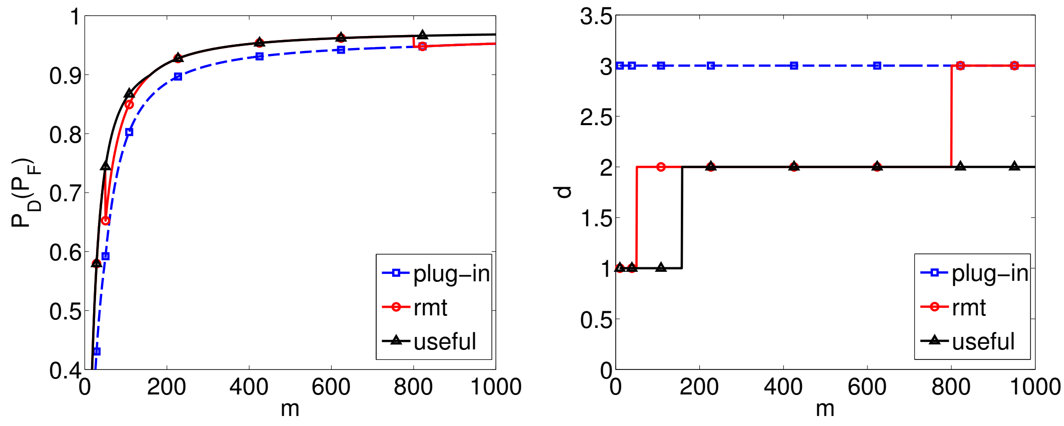
The number of informative subspace components, k_{eff}, is an upper bound for the number of useful subspace components, k_{useful}. As mentioned earlier, we cannot compute k_{useful} in closed form because the deterministic vector x, which drives the non-centrality parameters δ_i, is unknown. Therefore, k_{useful} is an oracle statistic as so we use k_{eff} as a proxy for k_{useful} in a realizable detector. However, as k_{eff} does not depend on x, whenever $k_{\text{eff}} \neq k_{\text{useful}}$, detectors using k_{eff} subspace components will be suboptimal.

Finally, we note that the derivation and computation of k_{useful} for the matched subspace detection application relies on random matrix theory. Without these insights, we would have no expression for $|\langle u_i, \widehat{u}_i \rangle|^2$ and subsequently could not compute the non-centrality parameter in (3.19) to use in the algorithm in Figure 3.1.

3.4.6 Numerical Example

In Figure 3.5 we compare the performance of the plug-in and RMT detectors to the performance of a detector that uses $d = k_{\text{useful}}$ subspace components. We consider the setting when $k = 3$, $n = 200$, $\Sigma = \textbf{diag}(5, 2, 0.5)$, and $x = [1.5, 1.5, 1.5]^T$. For a fixed $P_F = 0.1$, Figure 3.5(a) plots the theoretical detection probability (as computed in (3.6) using (3.16) and (3.19)) given various amounts of training data. Results are shown for the plug-in ($d = k$), RMT ($d = k_{\text{eff}}$), and useful ($d = k_{\text{useful}}$) detectors. Figure 3.5(b) plots the corresponding number of subspace components each uses given various amounts of training data.

Evident in Figure 3.5(a), the detector using k_{useful} subspace components achieves the maximum detection ability of all detectors for every amount of training samples. This is slightly contrived because k_{useful} is optimized to do just this. More importantly, we empirically see that using k_{eff} subspace components is not always optimal. However, examination of Figure 3.5(b) reveals why this occurs. For $50 \leq m \leq 160$, $k_{\text{eff}} = 2 > k_{\text{useful}} = 1$. Therefore, even though the second subspace component is informative by definition, it is not *useful* in detection. Including it in an energy detector decreases detector performance. A similar phenomenon occurs at $m = 800$ when k_{eff} increases to 3 but k_{useful} remains constant at 2. Unlike the RMT detector, the detection performance of the useful detector increases monotonically with an increase in training samples. Both the RMT and useful detectors outperform the standard plug-in detector which uses all k subspace components.



(a) Detector Performance (b) Number of Subspace Components

Figure 3.5: Deterministic energy detector performance as a function of the number of training samples. In this experiment $n = 200$, $\Sigma = \mathbf{diag}(5, 2, 0.5)$, $x = [1.5, 1.5, 1.5]^T$, and the required false alarm rate is $P_F = 0.1$. (a) The theoretical probability of detection achieved by the plug-in, RMT, and useful detectors. $P_D(P_F)$ is calculated in (3.4). The plug-in detector sets $d = k$, the RMT detector sets $k = k_{\text{eff}}$ as defined in (3.16), and the useful detector sets $d = k_{\text{useful}}$ as calculated in Figure 3.1 using the non-centrality parameter defined in (3.19). The useful detector achieves the optimal performance. (b) The number of subspace components used by the plug-in, RMT, and useful detectors. Whenever $k_{\text{eff}} \neq k_{\text{useful}}$, the RMT detector realizes a suboptimal detector performance. Even though these subspace components are *informative*, there is not enough training data to make them *useful* in detection.

3.5 Extension - Weighted Energy Detector

The energy detector in (3.3) may be generalized by adding a non-negative weight to each component in the sum. The statistic for the weighted energy detector is

$$\Lambda_{\text{weighted}}(w) = w^T A w = \sum_{i=1}^{k} a_i w_i^2. \tag{3.20}$$

where $A = \mathbf{diag}(a_1, \ldots, a_k) \in \mathbb{R}^{k \times k}$ and $a_i \geq 0$. We constrain $\sum_{i=1}^{k} a_i = 1$ so that the weights reside on the $(k-1)$-simplex. This reduces the set of possible weights by eliminating those that are multiples of each other, which results in equivalent detectors. The weighted energy detector gives practitioners additional design freedom to maximize detector performance. Using a similar analysis as in Section 3.2, the conditional distributions of the weighted energy detector's statistic in (3.20) are

$$\begin{aligned}\Lambda(w)|H_0 &\sim \sum_{i=1}^{k} a_i \chi^2_{1i},\\ \Lambda(w)|H_1 &\sim \sum_{i=1}^{k} a_i \chi^2_{1i}\left(\delta_i^2\right),\end{aligned} \tag{3.21}$$

where χ^2_{1i} are independent chi-square random variables with one degree of freedom and $\chi^2_{1i}\left(\delta_i^2\right)$ are independent non-central chi-square random variable with one degree of freedom and non-centrality parameter δ_i^2. We can relate P_D to P_F using the expression

$$P_{D_{\text{weighted}}}(P_F, A) = 1 - Q_{\Lambda|H_1}\left(Q^{-1}_{\Lambda|H_0}(1-P_F)\right) \tag{3.22}$$

where $Q_{\Lambda|H_1}$ is the c.d.f of $\Lambda(w)|H_1$ in (3.21) and $Q_{\Lambda|H_0}$ is the c.d.f. of $\Lambda(w)|H_0$ in (3.21).

The definition of $\Lambda_{\text{weighted}}(w)$ in (3.20) raises the natural question

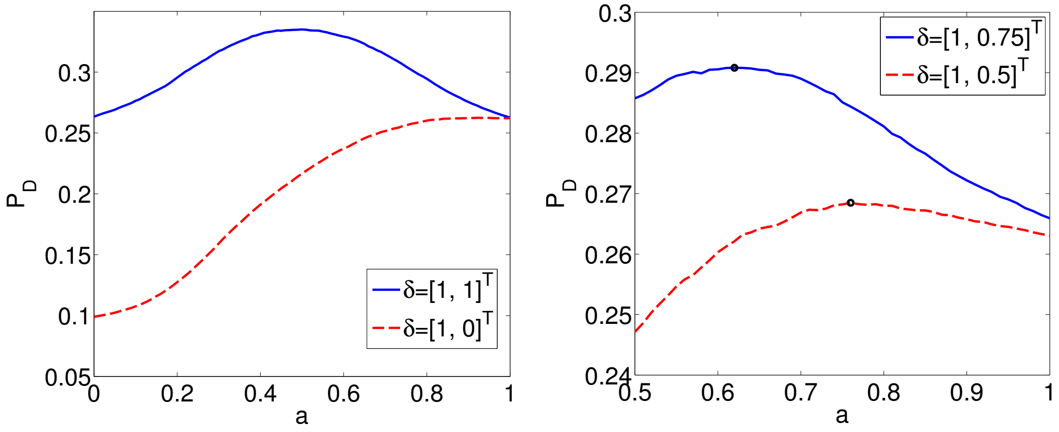
> Given δ and a desired P_F, what is the optimal choice of weighting matrix, A, that maximizes $P_{D_{\text{weighted}}}(P_F)$ for a weighted energy detector with the form of (3.20) using observations generated from (3.1)?

While the c.d.f. of chi-square and non-central chi-square random variables are known in closed form, the c.d.f. of a weighted sum of chi-square random variables is not known in closed form and therefore (3.22) cannot be computed analytically. It is common to use saddlepoint approximation techniques [95] to compute the c.d.f. of such sums in (3.21), however, such techniques must be computed for many thresholds, η, to generate a ROC curve for one weighting matrix A. To optimize over A in (3.22), this process would need to be repeated over a discretization of the $(k-1)$-simplex. Developing a more efficient algorithm to optimize over the weighting matrix, A, is an important topic for future work.

To illustrate how weighted energy detectors can improve detection performance, consider a rank-2 setting where the desired false alarm rate is $P_F = 0.1$. Optimizing $A = \mathbf{diag}(a_1, a_2)$ on the simplex $a_1 + a_2 = 1$ results in one degree of freedom and so

$$\Lambda(w)_{\text{weighted}} = a w_1^2 + (1-a) w_2^2$$

where $a \in [0,1]$. Figure 3.6 plots the empirically achieved (see [1]) probability of detection as a function of the weighting parameter a for four detectors each with a



(a) Easy Optimal Weights

(b) Difficult Optimal Weights

Figure 3.6: Empirically achieved probability of detection (P_D) as a function of the weighting coefficient a for a fixed false alarm rate of $P_F = 0.1$. (a) Two detectors, one using the deterministic vector $\delta = [1, 1]^T$ and the second using $\delta = [1, 0]^T$. The first detector achieves its maximum performance around $a = 0.5$ indicating that both components are equally informative. The second detector achieves its maximum performance at $a = 1$ indicating the second subspace component is not useful in detection. (b) Two detectors, one using $\delta = [1, 0.75]^T$ and the other using $\delta = [1, 0.5]^T$. The maximum performance of each detector is no longer achieved at $a = 0.5$ or $a = 1$ as the entries of δ are non-zero and are not equal. The maximum performance is indicated by a black circle.

different signal vector δ. Figure 3.6(a) shows results for detectors using $\delta = [1, 1]^T$ and $\delta = [1, 0]^T$. The detector with $\delta = [1, 1]^T$ achieves maximum performance when $a = 0.5$, which weights both components equally. As $\delta_1 = \delta_2 = 1$ both w_1 and w_2 have the same conditional distributions and it is intuitive that we weight both components equally. However, the detector using $\delta = [1, 0]^T$ achieves maximum performance when $a = 1$ indicating that the second component is not useful in detection. As $\delta_2 = 0$, w_2 has the same distribution under both the H_1 and H_0 hypotheses, giving it no discriminatory power. For these values of δ, the optimal a is obvious and the performance of the weighted energy detector is the same as that of the standard energy detector.

Figure 3.6(b) considers detectors using $\delta = [1, 0.75]^T$ and $\delta = [1, 0.5]^T$. Both choices place $\delta_1 > \delta_2$ so we only consider the regime $a \in [0.5, 1]$, which weights the first component stronger than the second. The maximum performance of each detector is indicated by a black circle. Unlike the detectors in Figure 3.6(a), the maximum P_D is not achieved at $a = 0.5$ or $a = 1$; both components are needed to achieve optimal performance. For these choices of δ, the weighted energy detector is able to achieve a better performance than a standard energy detector using either one

$(a = 1)$ or both $(a = 0.5)$ components. Developing an efficient algorithm to compute these optimal weights is an important extension of the work in this chapter.

3.6 Deterministic Matched Subspace Detectors with Missing Data

We consider the same detection setting as described in Section 3.4 using the training data model in (3.12). However, we only observe a fraction $p \in (0, 1)$ of the entries of our training matrix $Y = [y_1, \ldots, y_m]$; p is independent of n and m. Define our observed training data matrix, $\widetilde{Y}$, as

$$\widetilde{Y} = Y \odot M \tag{3.23}$$

where

$$M_{ij} = \begin{cases} 1 & \text{with probability } \gamma_y \\ 0 & \text{with probability } 1 - \gamma_y \end{cases}$$

and $\odot$ denotes the Hadamard or element-wise product. Finally we make the following assumption about our signal subspace, U.

Assumption 3.6.1. *In the missing data setting, assume that the columns of U satisfy a 'low-coherence' condition in the following sense: we suppose that there exist non-negative constants η, C independent of n, such that for $i = 1, \ldots, k$*

$$\max_i \|u_i\|_\infty \leq \eta \frac{\log^C n}{\sqrt{n}}.$$

We form a signal subspace estimate as in (3.14), except that we use our partially observed training matrix $\widetilde{Y}$ to form the sample covariance matrix S . Call this signal subspace estimate $\widetilde{U}$.

3.6.1 Pertinent Results from RMT

By modifying an argument in [86], we obtain the following result.

Theorem 3.6.1. *Assume that $x_i \sim \mathcal{CN}(0, \Sigma^2)$ as in (3.12) and that U in (3.12) obeys the low coherence condition in Assumption 3.6.1. Then as $n, m \to \infty$ with $n/m \to c$*

we have that for $i, j = 1, \ldots, k$:

$$|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{a.s.} \begin{cases} 1 - \dfrac{c\,(1 + p\sigma_i^2)}{p\sigma_i^2\,(p\sigma_i^2 + c)} & \text{if } \sigma_i > \dfrac{c^{1/4}}{\sqrt{p}} \\ 0 & \text{otherwise} \end{cases} .$$

$$|\langle u_i, \widehat{u}_j \rangle|^2 \xrightarrow{a.s.} 0 \qquad \text{for } i \neq j.$$

where $\widehat{u}_i$ are the left singular vectors of $\widetilde{Y}$.

The low coherence condition appears in, for example, [108] with the idea being that the matrix $U \begin{bmatrix} x_1 & \ldots & x_m \end{bmatrix}$ has entries of about the same magnitude. With the Gaussianity assumption for x, all we need is U to have low coherence. Recall that the coherence of a matrix with orthonormal columns is $\max_{i,j} |U_{i,j}|$. When a matrix is spiky, random sampling of its entries may result in a loss of information; matrices with low coherence behave better under random sampling and it is this setting that we focus on in this chapter.

The key insight from Theorem 3.6.1 is that only the singular vectors corresponding to signal singular values above the phase transition $\frac{c^{1/4}}{\sqrt{p}}$ are *informative*. The fraction of missing entries p regulates this phase transition point as $O(1/\sqrt{p})$. When a signal singular value drops below this critical threshold, the corresponding singular vector estimate is essentially noise-like (i.e. $|\langle u_i, \widehat{u}_i \rangle|^2 = o_p(1)$) and thus *uninformative*. The term $|\langle u_i, \widehat{u}_i \rangle|^2$ quantifies mismatch between the estimated and underlying singular vectors; when $p < p_{\text{crit.}} := \sqrt{c}/\max_i(\sigma_i^2)$ then *all* singular vectors are uninformative. Intuitively we expect a degradation in the performance of detectors that utilize subspace components for which $|\langle u_i, \widehat{u}_i \rangle|^2 = o_p(1)$. We refer to the estimate in Theorem 3.6.1 as $|\langle u_i, \widehat{u}_i \rangle|^2_{\text{rmt}}$.

3.6.2 Plug-in and RMT Detectors

Using the estimate of our signal subspace, $\widetilde{U}$, formed from our partially observed training data matrix $\widetilde{Y}$, we define

$$\widetilde{w} = \widetilde{U}^T y$$

where y is a testing vector from (3.13). Note that in this setup, we don't assume that our testing observation has any missing entries. Following a similar derivation from

Chapter 2 and the previous section, we have our plug-in and RMT test statistics are

$$\Lambda_{\text{plugin}}(\widetilde{w}) = \widetilde{w}^H \widetilde{w} \sum_{i=1}^{k} \widetilde{w}_i^2 \tag{3.24}$$

$$\Lambda_{\text{plugin}}(\widetilde{w}) = \widetilde{w}^H \widetilde{w} \sum_{i=1}^{k_{\text{eff}}} \widetilde{w}_i^2 \tag{3.25}$$

where we define k_{eff} as the number of signal singular values above the phase transition $\frac{c^{1/4}}{\sqrt{p}}$ shown in Theorem 3.6.1. We may use either test statistic to form a detector of the form

$$\Lambda(\widetilde{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \ln(\eta) \tag{3.26}$$

where η satisfies $P(\Lambda(\widetilde{w}) > \ln(\eta) \, | H_0) = \alpha$.

3.6.3 Theoretical ROC Curve Derivation

A standard way to compare the plug-in and RMT detectors derived in (3.24) and (3.25) respectively is to compute their ROC curves. For a particular statistic $\Lambda(\widetilde{w})$, to compute theoretical ROC curves, we must compute

$$\begin{aligned} P_D &= P(\Lambda(w) > \gamma | w \in H_1) \\ P_F &= P(\Lambda(w) > \gamma | w \in H_0) \end{aligned} \tag{3.27}$$

for $-\infty < \gamma < \infty$. To do this, we explore the conditional CDF under each hypothesis for the statistics (3.24) and (3.25).

This derivation is the same as in Chapter 2 except that we replace $|\langle u_i, \widehat{u}_i \rangle|^2_{\text{rmt}}$ with the expression in Theorem 3.6.1.

3.7 Simulation Results and Discussion

3.7.1 ROC Curves

We consider a setting where $k_{\text{eff}} = 1 < k = 2$. For this setting, as seen in Figure 3.7, for any false alarm rate (P_F), the RMT detector achieves a higher probability of detection (P_D), demonstrating the sub-optimality of the plug-in detector. This is expected because $k_{\text{eff}} < k$ so that the plug-in detector is employing uninformative subspace components. The theoretical ROC curves in (3.4) match the empirically generated ROC curves validating the performance predictions of (3.4) which rely on

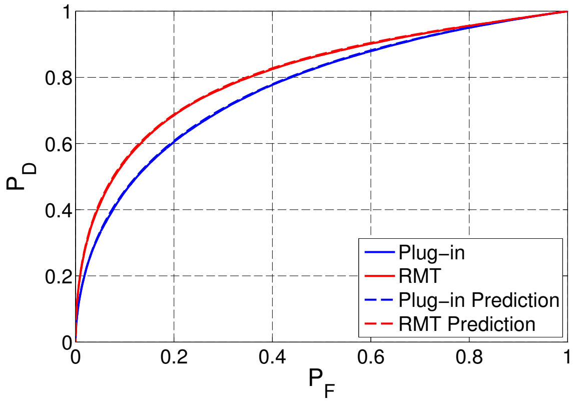


Figure 3.7: Empirical and theoretical ROC curves for the plug-in and RMT matched subspace detectors. Empirical ROC curves were simulated with $n = 500$, $m = 500$, $k = 2$, $\Sigma = \mathbf{diag}(3, 0.1)$, and $p = 0.8$. However, as σ_2 is below the critical threshold, $k_{\text{eff}} = 1$. The empirical ROC curves were computed using 5000 test samples and averaged over 25 trials. x was generated randomly for training samples but fixed for test samples. The theoretical ROC curves were obtained using (3.4). Note the excellent agreement and the performance gain realized by the RMT detector.

Theorem 3.6.1.

3.7.2 Effect of Missing Data

Figure 3.8 examines the performance of each detector as a function of p. Again we observe the sub-optimality of the plug-in detector. The theoretical P_D prediction in (3.4) matches empirically achieved P_D for both detectors. As expected, as p decreases, the achieved probability of detection decreases. We note the presence of a critical $p_{\text{crit.}} := \sqrt{c}/\max_i(\sigma_i^2)$ obtained from Theorem 3.6.1, below which (in the large system limit) we may only achieve $P_D = P_F$; the rounding in Figure 3.8 is attributed to finite system approximation error.

3.8 Conclusion

In this chapter, we considered the problem of designing a signal-plus-noise versus noise detector when the signal is assumed to be a fixed deterministic vector. In such a setting, the GLRT detector is an energy detector, whose statistic is the squared norm of the observation. By examining how the conditional distributions of this test statistic shift when adding additional components, we derived and defined the number of useful components, k_{useful}, that maximize detection ability.

When adding a component to an energy detector, there is a tradeoff between increasing the noise variance and increasing the signal variance by increasing the

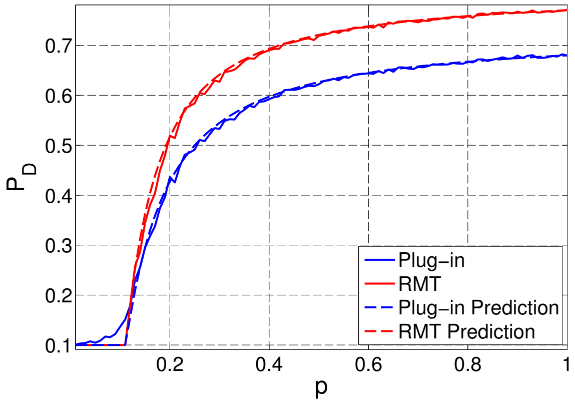


Figure 3.8: Empirically computed probability of detection, P_D, for a fixed probability of false alarm, $P_F = 0.1$, for various p. Here, $n = 1000$, $m = 1000$, $k = 2$, $\Sigma = \mathbf{diag}(3, 0.1)$. P_D was computed using (3.4) and x was generated as described in Figure 3.7. For values of $p \leq 1/9$, $k_{\text{eff}} = 0$ and performance degrades to $P_D = P_F + o(1)$ for both detectors. As p increases, $k_{\text{eff}} = 1$ allowing the detectors to achieve better than random guessing performance. When $k_{\text{eff}} > 0$ the plug-in detector is sub-optimal for all values of p.

non centrality parameter. For a component to be one of the k_{useful} components, the increase in non-centrality parameter must overcome the added noise variance. We explored the necessary increase in non-centrality parameter needed for a component to be useful in Figure 3.4.

We applied the idea of using only k_{useful} components to deterministic matched subspace detection where the unknown signal subspace is estimated from finite, noisy, signal-bearing training data. Both the standard plug-in detector using k subpsace components and RMT detector using k_{eff} subspace components (as defined by (3.16)) are energy detectors. We demonstrated that the new useful subspace detector outperforms both the plug-in and RMT detectors. Importantly, we showed that while a subspace component may be informative ($|\langle u_i, \widehat{u}_i \rangle|^2 > 0$), using that component in a detector may decrease performance.

As detectors using k_{useful} components assume knowledge of the unknown signal vector, they are not realizable. We showed that k_{eff} may be used as an upper bound for k_{useful}, however, deriving other estimates for k_{useful} that can be used in applications other than matched subspace detection is a focus of future work. We also provided a disucssion about the more general weighted energy detector and showed that such a detector can improve detection performance. Determining an efficient algorithm to compute the optimal weighting matrix to use in the weighted energy detector is an important area of future research. Extending the performance analysis of the useful matched subspace detector to the case of unknown Σ or complex valued data is within reach. The work in [107] on eigen-SNR accuracy, estimating k_{eff}, and estimating

$|\langle u_i, \widehat{u}_i \rangle|^2$ is directly applicable.

Finally, we considered a deterministic MSD problem where the unknown low-rank signal subspace is estimated from noisy, limited, signal-bearing training data with missing entries. We used RMT to characterize the resulting performance and showed that using $k_{\text{eff}} \leq k$ subspace components is optimal. The relationship between k_{eff} and p was made explicit in Theorem 3.6.1 and we showed that detection better than random guessing (in the large system limit) is only achievable for $p > p_{\text{crit.}} := \sqrt{c}/\max_i(\sigma_i^2)$.

CHAPTER IV

Using CCA and ICCA to Detect Correlations in Low-Rank Signal-Plus-Noise Datasets

4.1 Introduction

Canonical correlation analysis (CCA) is a joint multidimensional dimensionality reduction algorithm for exactly two datasets [4]. CCA finds a linear transformation for each dataset such that the correlation between the two transformed features is maximized. While CCA itself is not a data fusion algorithm, the correlated features that it returns may be used in data fusion algorithms. Such data fusion algorithms are becoming a necessity with the increased ability to capture high-dimensional multimodal datasets, arising in fields such as computer vision [28, 29, 30, 31, 32, 33, 34, 35], medical signal processing, [36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50], economics [58], climatology [60, 61, 62], and classical signal processing like Wiener filters [53], array processing [7], and cognitive radio [57].

At the heart of the CCA algorithm is a SVD of a matrix product involving the individual covariance matrices of the datasets and the cross-covariance matrix between them. When these matrices are known *a priori*, the non-zero singular values of this matrix represent correlated components between the datasets. These singular values are bounded between zero and one; a larger singular value indicates a stronger correlation. However, in all of the above applications, the true covariance matrices are unknown and must be estimated from data. When using CCA with sample covariance estimates from fewer samples than the combined dimensions of the datasets, the largest singular value of this matrix is deterministically one [6], falsely reporting a perfect correlation between the datasets. In this low-sample, high-dimensionality regime, CCA fails to reliably detect correlations between the datasets and because of this, many have abandoned CCA as a correlation analysis tool.

This chapter shows that detecting correlations in this low-sample high-dimensionality regime is feasible. We first empirically showcase the performance loss of CCA in this regime, provide a statistical significance test for CCA correlations, and derive a consistency bound that elucidates when this test can reliably detect correlations. We then present informative CCA (ICCA), which uses insights from random matrix theory to first trim the number of components used in the singular value decomposition used in CCA [8]. We provide a similar statistical test and consistency bound for ICCA and showcase that it is able to reliably detect correlations in the sample deficient regime. Importantly, we see that while the consistency boundary for CCA depends on the correlation between the datasets, the ICCA consistency boundary is independent of the underlying correlation.

This chapter assumes that each dataset is modeled with a low-rank signal-plus-noise data model, which is ubiquitous in signal processing applications. Surprisingly, the performance of CCA has not been extensively studied for this data model and therefore we use this chapter to complete the discussion by examining the data model in the presence of missing data. To showcase the improved performance of ICCA, we create three real-world audio-video datasets. We note that depending on the application, the linear, low-rank signal-plus-noise data model may be inappropriate. In such a setting, kernel CCA (KCCA) [14, 111], uses the kernel trick to map the data into a higher dimensional space. We leave the performance analysis of such kernel methods for non-linear data models as important future work.

This chapter is organized as follows. We provide the linear low-rank signal-plus-noise data model in Section 4.2. We then derive the solution of CCA in Section 4.3 and show how to estimate the number of correlated components from its solution. In Section 4.4, we derive the empirical version of CCA using sample covariance matrices and discuss the standard Wilk's Lambda Test for correlation detection. This derivation gives rise to the ICCA algorithm. We then provide statistical tests to estimate the number of correlated components between the datasets for both CCA and ICCA. We provide previous known results for empirical CCA in Section 4.5 and then state and prove new results for CCA and ICCA consistency in Section 4.6. In Section 4.7, we extend the consistency analysis to the missing data setting and provide an analogous consistency bound. Finally, we verify our theorems both on simulated data and real-world datasets in Section 4.8.

4.2 Data Model

Let $x_i \in \mathbb{C}^{p\times 1}$ and $y_i \in \mathbb{C}^{q\times 1}$ be modeled as

$$\begin{aligned} x_i &= U_x s_{x,i} + z_{x,i} \\ y_i &= U_y s_{y,i} + z_{y,i}, \end{aligned} \tag{4.1}$$

where $U_x^H U_x = I_{k_x}$, $U_y^H U_y = I_{k_y}$, $z_{x,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_p)$ and $z_{y,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_q)$. Furthermore, assume that

$$\begin{aligned} s_{x,i} &\sim \mathcal{CN}(0, \Theta_x) \\ s_{y,i} &\sim \mathcal{CN}(0, \Theta_y), \end{aligned}$$

where $\Theta_x = \mathbf{diag}\left(\left(\theta_1^{(x)}\right)^2, \dots, \left(\theta_{k_x}^{(x)}\right)^2\right)$ and $\Theta_y = \mathbf{diag}\left(\left(\theta_1^{(y)}\right)^2, \dots, \left(\theta_{k_y}^{(y)}\right)^2\right)$. Assume that $z_{x,i}$ and $z_{y,i}$ are mutually independent and independent from both $s_{x,i}$ and $s_{y,i}$. Finally, assume that

$$\mathbb{E}\left[s_{x,i} s_{y,i}^H\right] =: K_{xy} = \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}$$

where the entries of P_{xy} are $-1 \leq \rho_{kj} \leq 1$ and represent the correlation between $s_{x,i}^{(k)}$ and $s_{y,i}^{(j)}$. For reasons to be made clear later, define

$$\widetilde{K}_{xy} = \left(\Theta_x + I_{k_x}\right)^{-1/2} K_{xy} \left(\Theta_y + I_{k_y}\right)^{-1/2}$$

and define the singular values of $\widetilde{K}_{xy}$ as $\kappa_1, \dots, \kappa_{\min(k_x,k_y)}$. Under this model, we define the following covariance matrices

$$\begin{aligned} \mathbb{E}\left[x_i x_i^H\right] &= U_x \Theta_x U_x^H + I_p =: R_{xx} \\ \mathbb{E}\left[y_i y_i^H\right] &= U_y \Theta_y U_y^H + I_q =: R_{yy} \\ \mathbb{E}\left[x_i y_i^H\right] &= U_x K_{xy} U_y^H =: R_{xy}. \end{aligned} \tag{4.2}$$

Assumption 4.2.1. *Let $Z_x^{(n)} = [z_{x,1}, \dots, z_{x,n}]$ be the $p\times n$ matrix formed by stacking n observations of our noise. Let $Z_x^{(n)}$ have singular values $\sigma_1\left(Z_x^{(n)}\right) \geq \dots \geq \sigma_p\left(Z_x^{(n)}\right)$. Let $\mu_{Z_x^{(n)}}$ be the empirical singular value distribution defined by the probability measure*

$$\mu_{Z_x^{(n)}} = \frac{1}{p}\sum_{i=1}^{p} \delta_{\sigma_i\left(Z_x^{(i)}\right)}.$$

We assume that the probability measure $\mu_{Z_x^{(n)}}$ converges almost surely weakly as $p, n \to$

∞ with $p/n \to c_x$ to a non-random compactly supported probability measure μ_{Z_x} that is supported on $[a_x, b_x]$. We assume that $\sigma_1 \xrightarrow{a.s.} b_x$.

Similarly, we assume that the empirical singular value distribution for the noise matrix of Y converges almost surely to the non-random compactly supported probability measures μ_{Z_y} that is supported on $[a_y, b_y]$ and that $\sigma_1 \xrightarrow{a.s.} b_y$.

4.3 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a dimensionality reduction algorithm that finds linear transformations for x_i and y_i such that in the projected spaces, the transformed variables are maximally correlated. Specifically, CCA solves the following optimization problem

$$\rho_{\text{cca}} = \underset{w_x, w_y}{\text{argmax}} \frac{w_x^H R_{xy} w_y}{\sqrt{w_x^H R_{xx} w_x}\sqrt{w_y^H R_{yy} w_y}}, \tag{4.3}$$

where w_x and w_y are called canonical vectors and ρ_{cca} is called the canonical correlation coefficient. Notice that we can scale w_x and w_y and still achieve the same objective function. Therefore, we may constrain the canonical variates to have unit norm, resulting in

$$\begin{aligned} \underset{w_x, w_y}{\text{argmax}} \quad & w_x^H R_{xy} w_y \\ \text{subject to} \quad & w_x^H R_{xx} w_x = 1 \\ & w_y^H R_{yy} w_y = 1. \end{aligned} \tag{4.4}$$

Substituting the change of variables $\widetilde{w}_x = R_{xx}^{1/2} w_x$ and $\widetilde{w}_y = R_{yy}^{1/2} w_y$ in (4.4) results in the following optimization problem

$$\begin{aligned} \underset{\widetilde{w}_x, \widetilde{w}_y}{\text{argmax}} \quad & \widetilde{w}_x^H R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2} \widetilde{w}_y \\ \text{subject to} \quad & \widetilde{w}_x^H \widetilde{w}_x = 1 \\ & \widetilde{w}_y^H \widetilde{w}_y = 1. \end{aligned} \tag{4.5}$$

Examining the optimization problem in (4.5), we can immediately see that the solution to CCA may be solved via the SVD of the matrix

$$C_{\text{cca}} = R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2}. \tag{4.6}$$

Define $C_{\text{cca}} = FKG^T$ as the SVD of C_{cca} where F is an unitary $p \times p$ matrix with columns $f_1, \ldots, f_p$, G is a unitary $q \times q$ matrix with columns $g_1, \ldots, g_q$, and $K = \mathbf{diag}(k_1, \ldots, k_{\min(p,q)})$ is a $p \times q$ matrix whose diagonal elements are the singular values of C_{cca}. Therefore, the solution to (4.5) is

$$\begin{aligned} \widetilde{w}_x &= f_1 \\ \widetilde{w}_y &= g_1 \\ \rho_{\text{cca}} &= k_1. \end{aligned}$$

We can obtain higher order canonical correlations and vectors by taking sucessive singular value and vector pairs. From this solution, it is clear that the number of non-zero canonical correlation coefficients is exactly equal to the rank of C_{cca}. Noticing that R_{xx} and R_{yy} are non-singular and recalling the definition of R_{xy}, we have that

$$\begin{aligned} \#\text{ canonical correlation coefficients } &= \text{rank}(C_{\text{cca}}) \\ &= \text{rank}(R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2}) \\ &= \text{rank}(R_{xy}) \\ &= \text{rank}(K_{xy}) \\ &=: k. \end{aligned}$$

Therefore, when we know all parameters, k is exactly the number of non-zero singular values of K_{xy}. We note that $k \leq \min(k_x, k_y)$.

4.4 Empirical CCA for Correlation Detection

In many applications, we do not know the covariance matrices R_{xx}, R_{yy}, and R_{xy} *a priori*. In this section, we assume that all parameters in (4.1) are unknown. Therefore, we cannot simply determine the number of canonical correlation coefficient by examining the rank of R_{xy}. Instead, we are given multiple observations of each dataset that we stack columnwise to form the data matrices

$$\begin{aligned} X &= [x_1, \ldots, x_n] \\ Y &= [y_1, \ldots, y_n], \end{aligned}$$

where for $i = 1, \ldots, n$, x_i and y_i are modeled in (4.1). It is important to note that the number of observations of each dataset must be the same and that the observations come in pairs. Defining $Z_x = [z_{x,1}, \ldots, z_{x,n}]$, $Z_y = [z_{y,1}, \ldots, z_{y,n}]$, $V_x =$

$[s_{x,1},\ldots,s_{x,n}]^H$, and $V_y = [s_{y,1},\ldots,s_{y,n}]$, we may write our data matrices as the sum of a low-rank signal matrix and noise matrix

$$\begin{aligned} X &= U_x V_x^H + Z_x \\ Y &= U_y V_y^H + Z_y. \end{aligned} \tag{4.7}$$

Given these data matrices, we form estimates of our unknown covariance matrices via

$$\begin{aligned} \widehat{R}_{xx} &= \frac{1}{n} X X^H \\ \widehat{R}_{yy} &= \frac{1}{n} Y Y^H \\ \widehat{R}_{xy} &= \frac{1}{n} X Y^H. \end{aligned}$$

Define the data SVDs

$$\begin{aligned} X &= \widehat{U}_x \widehat{\Sigma}_x \widehat{V}_y^H \\ Y &= \widehat{U}_y \widehat{\Sigma}_y \widehat{V}_y^H \end{aligned}$$

and trimmed matrices

$$\begin{aligned} \widetilde{U}_x &= \widehat{U}_x\left(:, 1:\min(p,n)\right) \\ \widetilde{V}_x &= \widehat{V}_x\left(:, 1:\min(p,n)\right) \\ \widetilde{U}_y &= \widehat{U}_y\left(:, 1:\min(q,n)\right) \\ \widetilde{V}_y &= \widehat{V}_y\left(:, 1:\min(q,n)\right). \end{aligned}$$

Given these definitions, substituting the sample covariance estimates into C_{cca} yields the estimate (see [8])

$$\widehat{C}_{\text{cca}} = \widetilde{U}_x \widetilde{V}_x^H \widetilde{V}_y \widetilde{U}_y^H.$$

We denote the singular values of this matrix for $j = 1,\ldots,\min(p,q)$ $\widehat{\rho}_{\text{cca}}^{(j)}$, which are the empirical CCA correlation coefficient estimates. Empirical CCA can return up to $\min(p,q)$ canonical correlations, however, we know from the data model in (4.1) that X and Y have k_x and k_y underlying signals, respectively. As k_x and k_y are unknown, let $\widehat{k}_x$ and $\widehat{k}_y$ be estimates of the number of underlying signals in each dataset. It is common to return only $\min(\widehat{k}_x,\widehat{k}_y)$ canonical correlations. Therefore, define the top $\min(\widehat{k}_x,\widehat{k}_y)$ singular values of $\widehat{C}_{\text{cca}}$ as

$$\widehat{\rho}_{\text{cca}}^{(1)},\ldots,\widehat{\rho}_{\text{cca}}^{(\min(\widehat{k}_x,\widehat{k}_y))}. \tag{4.8}$$

For now, we assume that we are given $\widehat{k}_x$ and $\widehat{k}_y$, but we will return to the problem of estimating these parameters from data. To estimate the canonical vectors, we use the corresponding left and right singular vectors of $\widehat{C}_{\text{cca}}$, f_i and g_i to form

$$\begin{aligned} w_x^{(i)} &= \widehat{R}_{xx}^{-1/2} f_i \\ w_y^{(i)} &= \widehat{R}_{yy}^{-1/2} f_i. \end{aligned} \tag{4.9}$$

4.4.1 Classical Wilks Lambda Correlation Test

To test the significance of the empirical CCA canonical correlation estimates, classical methods [112] employ the Wilks likelihood ratio statistic,

$$\prod_{i=1}^{k} \left(1 - \left(\widehat{\rho}_{\text{cca}}^{(i)}\right)^2\right).$$

However, the distribution of this statistic is very unwieldy. For large n it is common to instead use the statistic [113]

$$\Lambda\left(\widehat{\rho}_{\text{cca}}^{(1)}, \ldots, \widehat{\rho}_{\text{cca}}^{(\min(p,q))}\right) = -\left(n - \frac{p+q+3}{2}\right) \log\left(\prod_{i=1}^{k} \left(1 - \widehat{\rho}_{\text{cca}}^{(i)2}\right)\right)$$

because this test statistic approximately follows a chi-square distribution

$$\Lambda\left(\widehat{\rho}_{\text{cca}}^{(1)}, \ldots, \widehat{\rho}_{\text{cca}}^{(\min(p,q))}\right) \sim \chi^2_{pq}.$$

This statistic tests whether the first canonical correlation, $\widehat{\rho}_{\text{cca}}^{(1)}$, is significant (i.e. whether the two datasets are uncorrelated). To determine significance, we compare the statistic against a threshold to achieve a desired false alarm rate via

$$\Lambda\left(\widehat{\rho}_{\text{cca}}^{(1)}, \ldots, \widehat{\rho}_{\text{cca}}^{(\min(p,q))}\right) \underset{\text{not sig}}{\overset{\text{sig}}{\gtrless}} \eta, \tag{4.10}$$

where $\eta = Q_{\chi^2_{pq}}^{-1}(1-\alpha)$, α is a desired probability of false alarm and Q is the inverse cumulative distribution function of the chi squared distribution with $p \times q$ degrees of freedom. To test the significance of successive empirical CCA correlation estimates, we modify the statistic to

$$\Lambda\left(\widehat{\rho}_{\text{cca}}^{(s+1)}, \ldots, \widehat{\rho}_{\text{cca}}^{(\min(p,q))}\right) = -\left(n - \frac{p+q+3}{2}\right) \log\left(\prod_{i=s+1}^{k} \left(1 - \widehat{\rho}_{\text{cca}}^{(i)2}\right)\right),$$

which is approximately

$$\Lambda\left(\widehat{\rho}_{\text{cca}}^{(s+1)},\ldots,\widehat{\rho}_{\text{cca}}^{(\min(p,q))}\right)\sim\chi^2_{(p-s)(q-s)}.$$

Comparing this statistic to an appropriate threshold tests whether the first $s+1$ canonical correlation are significant.

4.4.2 Informative CCA (ICCA)

When the number of samples is less than the combined dimension of the datasets ($n < p+q$), the largest singular value of $\widehat{C}_{\text{cca}}$ is deterministically one [6], regardless of whether an underlying correlation actually exists between the datasets. This observation led Pezeshki and Scharf et al. to correctly conclude that in this regime

> ... the empirical canonical correlations are defective and may not be used as estimates of canonical correlations between random variables.

This is a very unfortunate property of empirical CCA as many of the motivating applications operate in this low-sample, high-dimensionality regime. A key observation by [8] shows that the singular values of $\widehat{C}_{\text{cca}}$ are exactly the same as the singular values of $\widetilde{V}_x^H\widetilde{V}_y$. This is a $\min(p,n)\times\min(q,n)$ matrix that uses all right singular vectors of each dataset corresponding to a non-zero singular value. However, under the low-rank signal-plus-noise model, [8] shows that only a few of the right singular vectors actually contain *informative* signal. Therefore, by trimming $\widetilde{V}_x$ and $\widetilde{V}_y$ to have only $\widehat{k}_x$ and $\widehat{k}_y$ columns, we can avoid the performance loss of CCA in the sample deficient regime. Define the trimmed data SVDs

$$\begin{aligned}\mathring{U}_x&=\widehat{U}_x\left(:,1:\widehat{k}_x\right)\\\mathring{V}_x&=\widehat{V}_x\left(:,1:\widehat{k}_x\right)\\\mathring{U}_y&=\widehat{U}_y\left(:,1:\widehat{k}_y\right)\\\mathring{V}_y&=\widehat{V}_y\left(:,1:\widehat{k}_y\right).\end{aligned}\tag{4.11}$$

Given these definitions, we define the informative CCA (ICCA) matrix

$$\widehat{C}_{\text{icca}}=\mathring{U}_x\mathring{V}_x^H\mathring{V}_y\mathring{U}_y^H.$$

Similar to CCA, define the top $\min(\widehat{k}_x, \widehat{k}_y)$ singular values of $\widehat{C}_{\text{icca}}$ as,

$$\widehat{\rho}_{\text{icca}}^{(1)}, \ldots, \widehat{\rho}_{\text{icca}}^{(\min(\widehat{k}_x, \widehat{k}_y))}. \tag{4.12}$$

To estimate the canonical vectors, we use the corresponding left and right singular vectors of $\widehat{C}_{\text{icca}}$, f_i and g_i to form

$$\begin{aligned} w_x^{(i)} &= \widehat{R}_{xx}^{-1/2} f_i \\ w_y^{(i)} &= \widehat{R}_{yy}^{-1/2} f_i. \end{aligned} \tag{4.13}$$

4.4.3 New Statistical Tests for Correlation Detection

Given the canonical correlation estimates from CCA and ICCA, we can estimate the number of canonical correlations using the test statistics

$$\begin{aligned} \widehat{k}_{\text{cca}} &= \sum_{i=1}^{\min(p,q)} \mathbb{1}_{\left\{\left(\widehat{\rho}_{\text{cca}}^{(i)}\right)^2 > \tau_{\text{cca}}^{\alpha}\right\}} \\ \widehat{k}_{\text{icca}} &= \sum_{i=1}^{\min(\widehat{k}_x, \widehat{k}_y)} \mathbb{1}_{\left\{\left(\widehat{\rho}_{\text{icca}}^{(i)}\right)^2 > \tau_{\text{icca}}^{\alpha}\right\}}, \end{aligned} \tag{4.14}$$

where $\mathbb{1}_{\{.\}}$ is the indicator function and

$$\begin{aligned} \tau_{\text{cca}}^{\alpha} &= F_{\text{cca}}^{-1}(1-\alpha) \\ \tau_{\text{icca}}^{\alpha} &= F_{\text{icca}}^{-1}(1-\alpha). \end{aligned} \tag{4.15}$$

Here F_{cca} and F_{icca} are the distributions of the square of the largest singular value of $\widehat{C}_{\text{cca}}$ and $\widehat{C}_{\text{icca}}$ for the null setting where $\widetilde{V}_x$ and $\widetilde{V}_y$ are the $\min(n,p)$ and $\min(n,q)$ columns of two independent Haar (or isotropically random) distributed $n \times n$ matrices. The exact distribution of the squared singular values of $\widehat{C}_{\text{cca}}$ and $\widehat{C}_{\text{icca}}$ in the null model is given in [114]. The distributions of the square of the largest singular value of $\widehat{C}_{\text{cca}}$ and $\widehat{C}_{\text{icca}}$ in the null model may be approximated to second-order by the Tracy-Widom law [115] as

$$\begin{aligned} \tau_{\text{cca}}^{\alpha} &\approx \sigma_{n,p,q} \mathsf{TW}_{\mathbb{C}}^{-1}(1-\alpha) + \mu_{n,p,q}, \\ \tau_{\text{icca}}^{\alpha} &\approx \sigma_{n,\widehat{k}_x,\widehat{k}_y} \mathsf{TW}_{\mathbb{C}}^{-1}(1-\alpha) + \mu_{n,\widehat{k}_x,\widehat{k}_y}, \end{aligned}$$

where $\sigma_{n,p,q}$ is a scaling parameter and $\mu_{n,p,q}$ is a centering parameter. See Appendix D for values of these parameters as well as a derivation of the Tracy-Widom distribution for CCA and ICCA. The appendix also plots the accuracy of the Tracy-Widom approximation for CCA and ICCA for finite sized systems.

4.5 Empirical CCA Theory

In this section, we provide important, but previous results about empirical CCA. The first proposition specifically quantifies the conditions on when the largest canonical correlation reported by empirical CCA is deterministically one.

Proposition 4.5.1. *Let $n, p, q \to \infty$ such that $p/n \to c_x$ and $q/n \to c_y$. Let $p+q \leq n$. Then the largest singular value of $\widehat{C}_{cca}$ generated from data modeled in (4.1) behaves as*

$$\widehat{\rho}_{cca}^{(1)} = 1.$$

Proof. See [6]. □

This important result shows that the canonical correlation estimates of empirical CCA in the sample deficient regime are unable to detect the presence of correlation between datasets. The proof of this proposition motivates the ICCA algorithm.

The second proposition provides the limiting behavior of the empirical canonical correlations. This very recent result makes contact with a natural phase transition, below which the empirical canonical correlations behave as if the underlying datasets were noise only. Below, we provide a proof to transform our data model in (4.1) to the one used in the original theorem. Because this is such a new result, we provide an similar proof in Appendix C using our own notation. A key insight to this result is that the phase transition boundary is dependent on the underlying correlations between the datasets, which is an another undesirable property of CCA.

Proposition 4.5.2. *Let $n, p, q \to \infty$ such that $p/n \to c_x$ and $q/n \to c_y$. Assume that $p+q < n$. For $i = 1, \ldots, \min(k_x, k_y)$ let $\widehat{\rho}_{cca}^{(i)}$ be the largest singular singular values of $\widehat{C}_{cca}$ generated from data modeled in (4.1). Then these singular values behave as*

$$\widehat{\rho}_{cca}^{(i)} \xrightarrow{a.s.} \begin{cases} \sqrt{\kappa_i^2 \left(1 - c_x + \frac{c_x}{\kappa_i^2}\right)\left(1 - c_y + \frac{c_y}{\kappa_i^2}\right)} & \kappa_i^2 \geq r_c \\ \sqrt{d_r} & \kappa_i^2 < r_c \end{cases} \tag{4.16}$$

where κ_i are the singular values of $\widetilde{K}_{xy}$ and

$$\begin{aligned} r_c &= \frac{c_x c_y + \sqrt{c_y c_y (1-c_x)(1-c_y)}}{(1-c_x)(1-c_y) + \sqrt{c_x c_y (1-c_x)(1-c_y)}} \\ d_r &= c_x + c_y - 2c_x c_y + 2\sqrt{c_x c_y (1-c_x)(1-c_y)}. \end{aligned} \tag{4.17}$$

Proof. Bao et al. [2] proved this result for a slightly simplified model. Here we provide the linear transformations to recover their model. We may write our data matrices X and Y jointly via,

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{xy}^H & R_{yy} \end{bmatrix}^{1/2} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$$

where W_1 is a $p \times n$ matrix with independent $\mathcal{N}(0,1)$ entries and W_2 is an independent $q \times n$ matrix with independent $\mathcal{N}(0,1)$. As $p + q < n$, R_{xx} and R_{yy} are non-singular. Define

$$\begin{bmatrix} \widetilde{X} \\ \widetilde{Y} \end{bmatrix} = \begin{bmatrix} R_{xx}^{-1/2} & 0 \\ 0 & R_{yy}^{-1/2} \end{bmatrix}^{1/2} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} I_p & R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2}. \\ R_{yy}^{-1/2} R_{xy}^H R_{xx}^{-1/2} & I_q \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}.$$

With the definitions of the covariance matrices in (4.2)

$$\begin{aligned} R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2} &= U_x \left(\Theta_x + I_{k_x}\right)^{-1/2} \Theta_x^{1/2} P_{xy} \Theta_y^{1/2} \left(\Theta_y + I_{k_y}\right)^{-1/2} U_y^H \\ &= U_x \widetilde{K}_{xy} U_y^H. \end{aligned}$$

From this expression, it is clear why we defined $\widetilde{K}_{xy}$ as we originally did. Let $U_{\widetilde{K}_{xy}} K V_{\widetilde{K}_{xy}}$ be the SVD of $\widetilde{K}_{xy}$, where K is the $k_x \times k_y$ matrix with κ_j along the diagonal. Define $F = \left[\left(U_x U_{\widetilde{K}_{xy}}\right) \;\; \left(U_x U_{\widetilde{K}_{xy}}\right)^\perp \right]$ and $G = \left[\left(U_y V_{\widetilde{K}_{xy}}\right) \;\; \left(U_y V_{\widetilde{K}_{xy}}\right)^\perp \right]$. Then

$$\begin{aligned} \begin{bmatrix} \widetilde{\widetilde{X}} \\ \widetilde{\widetilde{Y}} \end{bmatrix} &= \begin{bmatrix} F^H & 0 \\ 0 & G^H \end{bmatrix}^{1/2} \begin{bmatrix} \widetilde{X} \\ \widetilde{Y} \end{bmatrix} = \begin{bmatrix} F^H R_{yy}^{-1/2} & 0 \\ 0 & G^H R_{yy}^{-1/2} \end{bmatrix}^{1/2} \begin{bmatrix} X \\ Y \end{bmatrix} \\ &= \begin{bmatrix} I_p & K \\ K^H & I_q \end{bmatrix}^{1/2} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}. \end{aligned}$$

Transforming X and Y to $\widetilde{\widetilde{X}}$ and $\widetilde{\widetilde{Y}}$ preserves the canonical correlation estimates because our transformation matrix is non-singular. After this transformation, we follow the proof from Bao et al. [2] with

$$\sqrt{r_i} = \kappa_i$$

□

4.6 New Results for CCA and ICCA Consistency

The main result of this section is Theorem 4.6.2, which provides conditions on when the statistical tests for CCA and ICCA canonical correlations in (4.14) provide consistent estimates of the true underlying number of canonical correlations, k. In order to prove this theorem, we rely on Corollary 4.6.1, which gives the almost sure convergence of the entries of the ICCA matrix $\mathring{V}_x^H \mathring{V}_y$. We being with a technical lemma.

Lemma 4.6.1. *Let $U = [u_1, \ldots, u_k] \in \mathbb{C}^{p\times k}$ and $V = [v_1, \ldots, v_k] \in \mathbb{C}^{n\times k}$ be independent matrices with orthonormal columns. Let $X \in \mathbb{C}^{p\times n}$ satisfy the hypotheses in Assumption 4.2.1. Then as $n, p \to \infty$ with $p/n \to c$, for $i \neq j$*

$$u_i^H \left(z^2 I_n - XX^H\right)^{-1} u_j \xrightarrow{a.s.} 0.$$

Similarly, for all i, j,

$$u_i \left(z^2 I_n - XX^H\right)^{-1} X v_j \xrightarrow{a.s.} 0.$$

Proof. The proof of Lemma 4.1 in [116] proves both of these statements. □

This lemma is needed in the proof of the following theorem, which we will use to prove Corollary 4.6.1. The result of this theorem may be of interest outside of this thesis for analysis of similar low-rank signal-plus-noise matrix models.

Theorem 4.6.1. *Let $\widetilde{u}_i$ and $\widetilde{v}_i$ be the left and right singular vectors associated with the i-th singular value, $\widetilde{\theta}_i$, of the $p \times n$ matrix*

$$\widetilde{X} = \sum_{i=1}^{k} \underbrace{\theta_i u_i v_i^T}_{P} + X.$$

Assume that X satisfies the hypotheses in Assumption 4.2.1 and suppose that $\theta_i > c^{1/4}$ for $c = p/n$. Let w be an arbitrary unit norm vector that is orthogonal to u_i for some $i \in \{1, \ldots, k\}$. Then as $n, p \to \infty$ such that $p/n \to c$, we have that

$$\langle w, \widetilde{u}_i \rangle \xrightarrow{a.s.} 0.$$

Proof. If $w \in \mathbf{span}(u_1, \ldots, u_k)$, then Theorem 2.10 c) of [116] proves our result. If $w \notin \mathbf{span}(u_1, \ldots, u_k)$, then we may write

$$w = w_u + w_u^{\perp},$$

where $w_u \in \mathbf{span}(u_1, \dots, u_k)$ and $w_u^{\perp}$ is in the orthocomplement of $\mathbf{span}(u_1, \dots, u_k)$. Therefore applying Theorem 2.10 c) of [116]

$$\langle w, \widetilde{u}_i \rangle = \langle w_u, \widetilde{u}_i \rangle + \left\langle w_u^{\perp}, \widetilde{u}_i \right\rangle \xrightarrow{\text{a.s.}} \left\langle w_u^{\perp}, \widetilde{u}_i \right\rangle,$$

so we only must focus on $w_u^{\perp}$.

Based on their definitions, $\widetilde{X}\widetilde{X}^H \widetilde{u}_i = \widetilde{\theta}_i^2 \widetilde{u}_i$ and $\widetilde{X}^H \widetilde{u}_i = \widetilde{\theta}_i \widetilde{v}$. Using the fact that $\widetilde{X} = P + X$, we have

$$\left(PP^H + PX^H + XP^H + XX^H\right) \widetilde{u}_i = \widetilde{\theta}_i^2 \widetilde{u}_i \tag{4.18}$$

and $\left(X^H + P^H\right) \widetilde{u}_i = \widetilde{\theta}_i \widetilde{v}$. Multiplying both sides of this second expression by P, we have

$$PX^H \widetilde{u}_i + PP^H \widetilde{u}_i = \widetilde{\theta}_i P \widetilde{v}_i.$$

Substituting this expression in (4.18) gives

$$\widetilde{\theta}_i P \widetilde{v}_i + XP^H \widetilde{u}_i + XX^H \widetilde{u}_i = \widetilde{\theta}_i^2 \widetilde{u}_i.$$

Rearranging terms gives

$$\widetilde{u}_i = \left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} \left(\widetilde{\theta}_i P \widetilde{v}_i + XP^H \widetilde{u}_i\right).$$

Therefore

$$\begin{aligned}
\langle w_u^{\perp}, \widetilde{u}_i \rangle &= w_u^{\perp H} \left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} \left(\widetilde{\theta}_i P \widetilde{v}_i + XP^H \widetilde{u}_i\right) \\
&= \widetilde{\theta}_i w_u^{\perp H} \left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} \sum_{j=1}^{k} \theta_j \langle v_j, \widetilde{v}_i \rangle u_j \\
&\quad + \widetilde{\theta}_i w_u^{\perp H} \left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} X \sum_{j=1}^{k} \theta_j \langle u_j, \widetilde{u}_i \rangle v_j.
\end{aligned}$$

By Theorem 2.7 c) in [116], we have that for $i \neq j$, $|\langle u_j, \widetilde{u}_i \rangle| \xrightarrow{\text{a.s.}} 0$ and $|\langle v_j, \widetilde{v}_i \rangle| \xrightarrow{\text{a.s.}} 0$. Therefore

$$\begin{aligned}
\langle w_u^{\perp}, \widetilde{u}_i \rangle = &\left(\widetilde{\theta}_i \theta_i \langle v_i, \widetilde{v}_i \rangle\right) w_u^{\perp H} \left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} u_i \\
&+ \left(\widetilde{\theta}_i \theta_i \langle u_i, \widetilde{u}_i \rangle\right) w_u^{\perp H} \left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} X v_i.
\end{aligned} \tag{4.19}$$

By Lemma 4.6.1,

$$w_u^{\perp H}\left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} u_i \xrightarrow{\text{a.s.}} 0$$

and

$$w_u^{\perp H}\left(\widetilde{\theta}_i^2 I_p - XX^H\right)^{-1} Xv_i \xrightarrow{\text{a.s.}} 0.$$

Therefore,

$$\langle w_u^{\perp}, \widetilde{u}_i\rangle \xrightarrow{\text{a.s.}} 0.$$

□

An important corollary to the above theorem allows us to characterize the asymptotic behavior of the entries of the matrix $\mathring{V}_x^H\mathring{V}_y$, which is used in ICCA. We need to characterize these entries in order to prove our consistency result in Theorem 4.6.2.

Corollary 4.6.1. *Let $\mathring{V}_x$ and $\mathring{V}_y$ be the trimmed right singular vectors defined in (4.11) of the data matrices generated from the data model in (4.1). In the asymptotic setting of Theorem 4.6.1 with $p/n \to c_x$ and $q/n \to c_y$*

$$\left|\left[\mathring{V}_x^H\mathring{V}_y\right]_{ij}\right| \xrightarrow{a.s.} \left|k_{ij}^{xy}\right| \alpha_{x,i}\alpha_{y,j},$$

where

$$\begin{aligned}
\alpha_{x,i} &= \begin{cases} \sqrt{1-\frac{c_x+\theta_i^{(x)}}{\theta_i^{(x)}(\theta_i^{(x)}+c_x)}} & \theta_i^{(x)} > c_x^{1/4} \\ 0 & \textit{otherwise} \end{cases} \\
\alpha_{y,j} &= \begin{cases} \sqrt{1-\frac{c_y+\theta_j^{(y)}}{\theta_j^{(y)}(\theta_j^{(y)}+c_x)}} & \theta_j^{(y)} > c_y^{1/4} \\ 0 & \textit{otherwise} \end{cases}
\end{aligned} \tag{4.20}$$

and k_{ij}^{xy} are the entries of K_{xy}.

Proof. The entries of the matrix are the inner products between the columns of $\mathring{V}_x$ and $\mathring{V}_y$

$$\left|\left(\mathring{V}_x^H\mathring{V}_y\right)\right|_{ij} = \left|\mathring{V}_x^H(:,i)\mathring{V}_y(:,j)\right|.$$

Notice that we may write

$$\begin{aligned}
\mathring{V}_x(:,i) &= aV_y(:,j) + bw_y \\
V_y(:,j) &= k_{ij}^{xy}V_x(:,i) + cw_x
\end{aligned} \tag{4.21}$$

for some arbitrary unit-norm vector w_x that is orthogonal to $V_x(:,i)$, some arbitrary unit-norm vector w_y that is orthogonal to $V_y(:,j)$, and constants a, b, and c. With these observations, we have

$$\begin{aligned}\mathring{V}_x^H(:,i)\mathring{V}_y(:,j) \quad &= \left(aV_y(:,j)+bw_y\right)^H \mathring{V}_y(:,j)\\ &= aV_y^H(:,j)\mathring{V}_y(:,j)+bw_y^H\mathring{V}_y(:,j).\end{aligned}$$

By Theorem 4.6.1, $w_y^H\mathring{V}_y(:,j) \xrightarrow{\text{a.s.}} 0$. As derived in [8],

$$\left|V_y^H(:,j)\mathring{V}_y(:,j)\right| \xrightarrow{\text{a.s.}} \alpha_{y,j} =: \begin{cases}\sqrt{1-\frac{c_y+\theta_j^{(y)}}{\theta_j^{(y)}(\theta_j^{(y)}+c_x)}} & \theta_j^{(y)} > c^{1/4}\\ 0 & \text{otherwise}\end{cases}.$$

Therefore, $\left|\mathring{V}_x^H(:,i)\mathring{V}_y(:,j)\right| \xrightarrow{\text{a.s.}} a\alpha_{y,j}$. Using the expression for $\mathring{V}_x(:,i)$ in (4.21), we observe that

$$V_y(:,j)^H\mathring{V}_x(:,i)=a.$$

Using the expression for $V_y(:,j)$ in (4.21), we have

$$\begin{aligned}a \quad &= V_y(:,j)^H\mathring{V}_x(:,i)\\ &= \left(k_{ij}^{xy}V_x(:,i)+cw_x\right)^H \mathring{V}_x(:,i)\\ &= k_{ij}^{xy}V_x^H(:,i)\mathring{V}_x(:,i)+cw_x^H\mathring{V}_x(:,i).\end{aligned}$$

By Theorem 4.6.1, $w_x^H\mathring{V}_x(:,i) \xrightarrow{\text{a.s.}} 0$. As derived in [8],

$$\left|V_y(:,j)\mathring{V}_y(:,j)\right| \xrightarrow{\text{a.s.}} =: \alpha_{x,i} \begin{cases}\sqrt{1-\frac{c_x+\theta_i^{(x)}}{\theta_i^{(x)}(\theta_i^{(x)}+c_x)}} & \theta_i^{(x)} > c^{1/4}\\ 0 & \text{otherwise}\end{cases}.$$

Therefore, $|a| \xrightarrow{\text{a.s.}} \left|k_{ij}^{xy}\right|\alpha_{x,i}$. Therefore,

$$\left|\mathring{V}_x^H(:,i)\mathring{V}_y(:,j)\right| \xrightarrow{\text{a.s.}} \left|k_{ij}^{xy}\right|\alpha_{x,i}\alpha_{y,i}.$$

□

Armed with this corollary, we are now in position to prove the consistency of the CCA and ICCA estimates of the number of correlated signals in (4.14). This result directly allows us to compare the performance of CCA and ICCA across various regimes and showcases the superiority of ICCA.

Theorem 4.6.2. *Let $p, q, n \to \infty$ with $p/n \to c_x$ and $q/n \to c_y$. Given data modeled* *in (4.1), the estimates of the number of correlated components given in (4.14) are consistent under the following conditions*

$$\begin{aligned}\widehat{k}_{cca} &\xrightarrow{a.s.} k \quad \textit{if } \kappa_k^2 > r_c \textit{ and } n > p+q \\ \widehat{k}_{icca} &\xrightarrow{a.s.} k \quad \textit{if } \min_{i=1,\ldots,k_x} \theta_i^{(x)} > c_x^{1/4} \textit{ and } \min_{i=1,\ldots,k_y} \theta_i^{(y)} > c_y^{1/4}\end{aligned}$$

where κ_i are the singular value of $\widetilde{K}_{xy}$ and r_c is given in (4.17).

Proof. The conditions for the consistency of the CCA estimate follows from (4.16). From this equation, we observe that the square of the smallest non-zero singular value of $\widetilde{K}_{xy}$, κ_k, must be larger than the constant r_c for the canonical correlation estimate to be different statistically from noise.

For ICCA, recall that

$$\widehat{k}_{\text{icca}} = \sum_{i=1}^{\min(\widehat{k}_x, \widehat{k}_y)} \mathbb{1}_{\left\{\left(\widehat{\rho}_{\text{icca}}^{(i)}\right)^2 > \tau_{\text{icca}}^{\alpha}\right\}}.$$

First we show that $\widehat{k}_x \to k_x$ and $\widehat{k}_y \to k_y$ under the conditions stated in the theorem. Estimating the number of signals present in such signal-plus-noise models has been extensively studied in [116, 85, 84]. These works show that when the signal-to-noise ratio is larger than a threshold, we can reliably detect the presence of signals in noisy measurements. Specifically, we refer the reader to Algorithm 2 of [89] for a practical implementation using the Tracy Widom approximation of the largest eigenvalues of the sample correlation matrix to detect the number of signals. These results show that the individual estimates of the number of signals are consistent under the following conditions

$$\begin{aligned}\widehat{k}_x &\xrightarrow{\text{a.s.}} k_x \quad \text{if } \min_{i=1,\ldots,k_x} \theta_i^{(x)} > c_x^{1/4} \\ \widehat{k}_y &\xrightarrow{\text{a.s.}} k_y \quad \text{if } \min_{i=1,\ldots,k_y} \theta_i^{(y)} > c_y^{1/4}.\end{aligned}$$

In Appendix D we verify the Tracy-Widom approximation for the detection of signals in individual datasets. When these conditions on Θ_x and Θ_y are met, the estimate of the number of correlated signals becomes

$$\widehat{k}_{\text{icca}} \xrightarrow{\text{a.s.}} \sum_{i=1}^{\min(k_x, k_y)} \mathbb{1}_{\left\{\left(\widehat{\rho}_{\text{icca}}^{(i)}\right)^2 > \tau_{\text{icca}}^{\alpha}\right\}}.$$

To prove the theorem, we want to show that

$$\mathbb{P}\left(\widehat{k}_{\text{icca}} = k\right) \xrightarrow{\text{a.s.}} 1$$

under the above conditions on Θ_x and Θ_y. Momentarily, we assume that $k = \min(k_x, k_y)$. The singular values of $\mathring{V}_x^H \mathring{V}_y$ are ordered and so this probability simply becomes

$$\mathbb{P}\left(\widehat{k}_{\text{icca}} = k\right) = \mathbb{P}\left(\left(\widehat{\rho}_{\text{icca}}^{(\min(k_x,k_y))}\right)^2 > \tau_{\text{icca}}^{\alpha}\right).$$

From Corollary 4.6.1, we also know that

$$\left|\left[\mathring{V}_x^H \mathring{V}_y\right]_{ij}\right| \xrightarrow{\text{a.s.}} \left|k_{ij}^{xy}\right| \alpha_{x,i}\alpha_{y,i}.$$

Using this fact we define

$$A_x = \mathbf{diag}(\alpha_{x,1}, \ldots, \alpha_{x,k_x})$$
$$A_y = \mathbf{diag}(\alpha_{y,1}, \ldots, \alpha_{y,k_y})$$

so that we may write

$$\mathring{V}_x^H \mathring{V}_y = A_x K_{xy} A_y + \Delta,$$

where $\Delta = [\delta_{ij}]$ such that $\delta_{ij} \xrightarrow{\text{a.s.}} 0$. Examining (4.20), we see that under the above conditions on Θ_x and Θ_y, A_x and A_y are both full rank. Define

$$\alpha_{x,\min} = \min_{i=1\ldots,k_x} \alpha_{x,i}$$
$$\alpha_{y,\min} = \min_{i=j\ldots,k_y} \alpha_{y,j}.$$

By properties of singular values

$$\sigma_{\min}(A_x K_{xy} A_y) - \sigma_{\max}(\Delta) \leq \sigma_{\min}(A_x K_{xy} A_y + \Delta) \leq \sigma_{\min}(A_x K_{xy} A_y) + \sigma_{\max}(\Delta).$$

Examining $\sigma_{\max}(\Delta)$, we observe that

$$\sigma_{\max}(\Delta) \leq \|\Delta\|_F = \sqrt{\sum_{i=1}^{k_x}\sum_{j=1}^{k_y} |\delta_{ij}|^2}.$$

Using the fact that $\delta_{ij} \xrightarrow{\text{a.s.}} 0$, we have that $\sigma_{\max}(\Delta) \xrightarrow{\text{a.s.}} 0$. Therefore, almost surely

$$\sigma_{\min}(A_x K_{xy} A_y) \leq \sigma_{\min}(A_x K_{xy} A_y + \Delta) \leq \sigma_{\min}(A_x K_{xy} A_y),$$

which implies that

$$\widehat{\rho}_{\text{icca}}^{(\min(k_x,k_y))} \xrightarrow{\text{a.s.}} \sigma_{\min}(A_x K_{xy} A_y)$$

By properties of singular values

$$\begin{aligned}
\widehat{\rho}_{\text{icca}}^{(\min(k_x,k_y))} &= \sigma_{\min(k_x,k_y)}\left(\overset{\circ}{V}_x^H \overset{\circ}{V}_y\right) \\
&\xrightarrow{\text{a.s.}} \sigma_{\min(k_x,k_y)}\left(A_x K_{xy} A_y\right) \\
&\geq \sigma_{k_x}\left(A_x\right) \sigma_{\min(k_x,k_y)}\left(K_{xy}\right) \sigma_{k_y}\left(A_y\right) \\
&= \alpha_{x,\min} \kappa_{\min(k_x,k_y)} \alpha_{y,\min}.
\end{aligned}$$

Next we turn to our statistical test in the asymptotic setting. Unlike $\widehat{C}_{\text{cca}}$, whose dimension scales with n, the dimension of $\widehat{C}_{\text{icca}}$ remains $k_x \times k_y$ even as n increases. Therefore, in the null setting as $n \to \infty$ the entries $\widehat{C}_{\text{icca}}$ converge almost surely to 0. Therefore, in the asymptotic setting our test becomes

$$\widehat{k}_{\text{icca}} = \sum_{i=1}^{\min(k_x,k_y)} \mathbb{1}_{\left\{\left(\widehat{\rho}_{\text{icca}}^{(i)}\right)^2 > 0\right\}}.$$

Therefore,

$$\begin{aligned}
\mathbb{P}\left(\left(\widehat{k}_{\text{icca}} = k\right)\right) &= \mathbb{P}\left(\left(\widehat{\rho}_{\text{icca}}^{(\min(k_x,k_y))}\right)^2 > 0\right) \\
&\geq \mathbb{P}\left(\left(\alpha_{x,\min} \kappa_{\min(k_x,k_y)} \alpha_{y,\min}\right)^2 > 0\right) \\
&= 1.
\end{aligned}$$

The last equality comes from the fact that under our conditions on Θ_x and Θ_y, the α terms are non-zero and from the fact that we momentarily assumed $k = \min(k_x, k_y)$ so that $\kappa_{\min(k_x,k_y)}$ is non-zero. We note that this holds for all significance levels.

Lastly, we argue that the above analysis holds when $k < \min(k_x, k_y)$. In this setting, the last $\min(k_x, k_y) - k$ singular values of $A_y K_{xy} A_y$ are zero. The above analysis holds for the largest k canonical correlations, showing that they are non-zero in the asymptotic limit. Therefore, the asymptotic statistic will correctly mark these top k singular values as an indicator of the k correlations and correctly identify the smallest $\min(k_x, k_y) - k$ singular values as not containing correlation as they are zero in the asymptotic limit.

□

Finally, we provide a general version of the consistency theorem above for a noise that is not assumed Gaussian as in (4.1).

Corollary 4.6.2. *As in Assumption 4.2.1, let μ_{Z_x} and μ_{Z_y} be the non-random compactly supported probability measures modeling the singular values of the noise matrices X and Y. Let b_x and b_y be the supremums of the supports, respectively. Let $p, q, n \to \infty$ with $p/n \to c_x$ and $q/n \to c_y$. Relax the constraint in (4.1) that the noise is Gaussian but instead drawn from the probability measures above. ICCA returns a consistent estimate of the number of correlated components under the following conditions*

$$\widehat{k}_{icca} \xrightarrow{a.s.} k \text{ if } \min_{i=1,\ldots,k_x} \theta_i^{(x)} > \frac{1}{D_{\mu_X}(b_x^+)} \text{ and } \min_{i=1,\ldots,k_y} \theta_i^{(y)} > \frac{1}{D_{\mu_Y}(b_y^+)}$$

where D_{μ_X} and D_{μ_Y}, the D-transforms of μ_X and μ_Y, are the functions, depending on c_x and c_y, defined by

$$D_{\mu_X}(z) =: \left[\int \frac{z}{z^2 - t^2} d\mu_X(t)\right] \times \left[c_x \int \frac{z}{z^2 - t^2} d\mu_X(t) + \frac{1 - c_x}{z}\right] \text{ for } z > b_x$$
$$D_{\mu_Y}(z) =: \left[\int \frac{z}{z^2 - t^2} d\mu_Y(t)\right] \times \left[c_y \int \frac{z}{z^2 - t^2} d\mu_Y(t) + \frac{1 - c_y}{z}\right] \text{ for } z > b_y.$$

Define the notation

$$D_\mu(b^+) =: \lim_{z \downarrow b} D_\mu(z).$$

Proof. This result follows from the proof of Theorem 4.6.2 using the analysis in [116] for the D-transform. □

This is the more general result to Theorem 4.6.2 as it is applicable to non-Gaussian noise. See Chapter 5 or [117] for a discussion on computing D-transforms in practice.

4.7 Extension to Missing Data

We now consider the setting where our data matrices X and Y have missing entries. In such as setting, our matrices are modeled similar to (4.7) but with additional masking matrices

$$\begin{aligned} X &= \left(U_x V_x^H + Z_x\right) \odot M_x \\ Y &= \left(U_y V_y^H + Z_y\right) \odot M_y \end{aligned} \tag{4.22}$$

where

$$M_{ij}^x = \begin{cases} 1 & \text{with probability } \gamma_x \\ 0 & \text{with probability } 1-\gamma_x \end{cases} \qquad M_{ij}^y = \begin{cases} 1 & \text{with probability } \gamma_y \\ 0 & \text{with probability } 1-\gamma_y \end{cases}$$

and $\odot$ denotes the Hadamard or element-wise product. Throughout this section we make the following assumption on the entries of U_x, U_y, V_x, V_y. This assumption ensures that the columns of these matrices are not "spiked". For our simulations, we sample columns from the unit hypersphere so that these conditions are met.

Assumption 4.7.1. *In the missing data setting, assume that the columns of U_x, U_y, V_x, and V_y satisfy a 'low-coherence' condition in the following sense: we suppose that there exist non-negative constants $\eta_{u,x}$, $C_{u,x}$, $\eta_{u,y}$, $C_{u,y}$, $\eta_{v,x}$, $C_{v,x}$,$\eta_{v,y}$, $C_{v,y}$ independent of n, such that for $i = 1, \ldots, k_x$ and $\jmath = 1, \ldots, k_y$,*

$$\max_i \|u_i^{(x)}\|_\infty \leq \eta_{u,x} \frac{\log^{C_{u,x}} p}{\sqrt{p}}, \quad \max_i \|u_j^{(y)}\|_\infty \leq \eta_{u,y} \frac{\log^{C_{u,y}} q}{\sqrt{q}}$$
$$\max_i \|v_i^{(x)}\|_\infty \leq \eta_{v,x} \frac{\log^{C_{v,x}} n}{\sqrt{n}}, \quad \max_i \|v_j^{(y)}\|_\infty \leq \eta_{v,y} \frac{\log^{C_{v,y}} n}{\sqrt{n}}.$$

In the same manner of Section 4.6, we wish to know when the estimates in (4.14) are consistent in the presence of missing data. The theorem below characterizes this behavior. We proceed similarly to the proof of [117] and then invoke the proof of Theorem 4.6.2. The two theorems are very similar except that in the case of missing data, we simply replace Θ_x with $\gamma_x\Theta_x$ and Θ_y with $\gamma_y\Theta_y$. Therefore, missing data has the effect of decreasing the SNR of our problem.

Theorem 4.7.1. *Let $p, q, n \to \infty$ with $p/n \to c_x > 0$ and $q/n \to c_y > 0$ and assume the coherence conditions given in Assumption 4.7.1. Given data modeled in (4.22), the estimates of the number of correlated components given in (4.14) are consistent under the following conditions*

$$\widehat{k}_{cca} \xrightarrow{a.s.} k \quad \text{if } \min_{i=1,\ldots,k} \mathring{\kappa}_i^2 > r_c \text{ and } n > p+q$$
$$\widehat{k}_{icca} \xrightarrow{a.s.} k \quad \text{if } \min_{i=1,\ldots,\widehat{k}_x} \theta_i^{(x)} > \frac{c_x^{1/4}}{\sqrt{\gamma_x}} \text{ and } \min_{i=1,\ldots,\widehat{k}_y} \theta_i^{(y)} > \frac{c_y^{1/4}}{\sqrt{\gamma_y}}$$

where $\mathring{\kappa}_i$ are the singular values of

$$\left(\gamma_x\Theta_x + I_{k_x}\right)^{-1/2} \left(\gamma_x\Theta_x\right)^{1/2} P_{xy} \left(\gamma_y\Theta_y\right)^{1/2} \left(\gamma_y\Theta_y + I_{k_y}\right)^{-1/2}$$

and r_c is given in (4.17).

Proof. Defining $P_x = U_x V_x^H$ We may write (4.22) as

$$\begin{aligned} X &= \underbrace{P_x \odot M_x}_{\widehat{P}_x} + \underbrace{Z_x \odot M_x}_{\widehat{Z}_x} \\ &= \mathbb{E}\left[\widehat{P}_x\right] + \widehat{Z}_x + \Delta_{\widehat{P}_x} \\ &= \underbrace{\gamma_x P_x + \widehat{Z}_x}_{\widetilde{X}} + \Delta_{\widehat{P}_x}. \end{aligned}$$

Similarly, we may write $Y = \widetilde{Y} + \Delta_{\widehat{P}_y}$ where $\widetilde{Y} = \gamma_y P_y + \widehat{Z}_y$.

First we show that the maximum singular value of $\Delta_{\widehat{P}_x}$ and $\Delta_{\widehat{P}_y}$ converge almost surely to 0. Under the low-coherence assumption, we have that

$$\max_{ij} |P_x|_{ij} \leq \max_i \theta_i^{(x)} \max_k \|u_k^{(x)}\|_\infty \max_\ell \|v_\ell^{(x)}\|_\infty = \max_i \theta_i^{(x)} \mathcal{O}\left(\frac{\log n, p \text{ factors}}{\sqrt{np}}\right). \tag{4.23}$$

By assumption that $c_x > 0$, we have that $n = \mathcal{O}(p)$. This fact, coupled with the fact that $\theta_i^{(x)}$ is not dependent on n gives

$$\max_{ij} |P_x|_{ij} \leq \mathcal{O}\left(\frac{\log n \text{ factors}}{n}\right). \tag{4.24}$$

To characterize the largest singular value of $\Delta_{\widehat{P}_x}$, we want to use Latala's theorem [118], which states that for a matrix A with independent mean zero random entries with bounded fourth moment

$$\mathbb{E}\left[\sigma_1(A)\right] \leq C\left[\max_i \left(\sum_j \mathbb{E}\left[A_{ij}^2\right]\right)^{1/2} + \max_j \left(\sum_i \mathbb{E}\left[A_{ij}^2\right]\right)^{1/2} + \left(\sum_{i,j} \mathbb{E}\left[A_{ij}^4\right]\right)^{1/4}\right]$$

for some universal constant C that does not depend on n or p. Through basic calculation, one can show that

$$\begin{aligned} \mathbb{E}\left[\left(\Delta_{\widehat{P}_x}\right)_{ij}^2\right] &= \gamma_x (1-\gamma_x)(P_x)_{ij}^2 \\ \mathbb{E}\left[\left(\Delta_{\widehat{P}_x}\right)_{ij}^4\right] &= \left(-3\gamma_x^4 + 6\gamma_x^3 - 4\gamma_x^2 + \gamma\right)(P_x)_{ij}^4. \end{aligned}$$

These expressions satisfy the conditions on Latala's theorem. Therefore, by substituting these expressions into Latala's theorem with the bound in (4.24), we have

$$\mathbb{E}\left[\sigma_1\left(\Delta_{\widehat{P}_x}\right)\right] \leq \mathcal{O}\left(\frac{\log n \text{ factors}}{\sqrt{n}}\right).$$

By concentration and convexity of the largest singular value (see [117]), we have that in our asymptotic regime

$$\sigma_1(\Delta_{\widehat{P}_x}) \xrightarrow{\text{a.s.}} 0.$$

Using a similar argument

$$\sigma_1(\Delta_{\widehat{P}_y}) \xrightarrow{\text{a.s.}} 0.$$

Therefore we have that

$$\begin{aligned} X &\to \gamma_x P_x + \widehat{Z}_x \\ Y &\to \gamma_y P_y + \widehat{Z}_y. \end{aligned}$$

Examining $\widehat{Z}_x$, we have

$$\begin{aligned} \mathbb{E}\left[\widehat{Z}_{ij}^{(x)}\right] = & \ \mathbb{E}\left[\widehat{Z}_{ij}^{(x)}|M_{ij}^{(x)}=0\right]\mathbb{P}\left(M_{ij}^{(x)}=0\right) \\ & + \mathbb{E}\left[\widehat{Z}_{ij}^{(x)}|M_{ij}^{(x)}=1\right]\mathbb{P}\left(M_{ij}^{(x)}=1\right) = 0 \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}\left[\left(\widehat{Z}_{ij}^{(x)}\right)^2\right] &= \mathbb{E}\left[\left(\widehat{Z}_{ij}^{(x)}\right)^2|M_{ij}^{(x)}=0\right]\mathbb{P}\left(M_{ij}^{(x)}=0\right) + \\ & \qquad \mathbb{E}\left[\left(\widehat{Z}_{ij}^{(x)}\right)^2|M_{ij}^{(x)}=1\right]\mathbb{P}\left(M_{ij}^{(x)}=1\right) \\ &= 0 + \gamma_x. \end{aligned}$$

Therefore, $\widehat{Z}_{ij}^{(x)}$ are i.i.d. zero mean with variance γ_x and $\widehat{Z}_x \to \sqrt{\gamma_x}Z_x$. Using this observation,

$$\begin{aligned} X &\xrightarrow{\text{a.s.}} \gamma_x P_x + \widehat{Z}_x \\ &\to \gamma_x P_x + \sqrt{\gamma_x}Z_x \\ &= \sqrt{\gamma_x}\left(\sqrt{\gamma_x}P_x + Z_x\right). \end{aligned}$$

Similarly, $Y \to \sqrt{\gamma_y}\left(\sqrt{\gamma_y}P_y + Z_y\right)$.

From this we can conclude that eigenvector expressions of the form $\langle u, \widehat{u}\rangle$ behave as if we replace Θ_x with $\gamma_x\Theta_x$ and Θ_y with $\gamma_y\Theta_y$. Consider

$$\left|u_i^H\left(zI - (Z_x + \Delta_{\widehat{P}_x})(Z_x + \Delta_{\widehat{P}_x})^H\right)^{-1}u_j - u_i^H\left(zI - Z_xZ_x^H\right)^{-1}u_j\right|, \tag{4.25}$$

which as a a consequence of the variational characterization of the largest singular value is upper bounded by

$$\sigma_1\left(\left(zI-(Z_x+\Delta_{\widehat{P}_x})(Z_x+\Delta_{\widehat{P}_x})^H\right)^{-1}-\left(zI-Z_xZ_x^H\right)^{-1}\right).$$

Following a similar argument in [117], we have that (4.7) is upper bounded by

$$\frac{3\sigma_z(Z_x)}{\Im w}\sigma_1(\Delta_{\widehat{P}_x}),$$

where $\Im w>0$. Using the facts that $\sigma_1(Z_x)\xrightarrow{\text{a.s.}}\sqrt{\gamma_x}b_x$ by the above relationship and $\sigma_1(\Delta_{\widehat{P}_x})\xrightarrow{\text{a.s.}}0$ combined with Assumption 4.2.1, we have that (4.25) converges to 0. Using a similar argument, one can prove the same result for quadratic forms with Z_y and $\Delta_{\widehat{P}_y}$.

Therefore, an analogous version of Theorem 4.6.1 holds for the missing data section, as the quadratic forms used by Lemma 4.6.1 still hold. Therefore, we prove the theorem following the same rank argument as Theorem 4.6.2, except that we replace Θ_x with $\gamma_x\Theta_x$ and replace Θ_y with $\gamma_y\Theta_y$. □

4.8 Empirical Results

4.8.1 Simulated Data

We first showcase the accuracy of the consistency boundary for both CCA and ICCA described in Theorem 4.6.2. We consider a rank-1 setting ($k_x=k_y=1$) and generate data from (4.1) for fixed $p=q=150$ over various number of samples n, signal-to-noise ratio (SNR) $\theta=\theta_1^{(x)}=\theta_1^{(y)}$, and various $\rho=P_{xy}$. In this setting, there is only one correlated signal so $k=1$ and the consistency boundary becomes a phase transition. We then the data into data matrices X and Y, and compute $\widehat{\rho}_{\text{cca}}^{(1)}$ and $\widehat{\rho}_{\text{icca}}^{(1)}$ from the SVD of of $\widehat{C}_{\text{cca}}$ and $\widehat{C}_{\text{icca}}$, respectively. Using these correlation estimates, we compute the estimated number of correlated components via (4.14) for a significance level of $\alpha=0.01$. For a fixed set of parameters (n, θ, ρ) we repeat the above process for 10000 trials and determine the percentage of trials where we detect $\widehat{k}_{\text{cca}}=1$ and $\widehat{k}_{\text{icca}}=1$. In all simulations, we use Algorithm 2 of [89] to estimate $\widehat{k}_x$ and $\widehat{k}_y$ using a significance level of $\alpha=0.01$ (See Appendix D for discussion). Figure 4.1 plots the $\log_{10}$ of this percentage for empirical CCA and ICCA for two values of ρ. On each plot, we overlay the consistency boundary given by Theorem 4.6.2 using a solid white line for empirical CCA and a dashed white line for ICCA..

From these figures, we see that for smaller ρ, it is more difficult for empirical CCA to detect the presence of the correlated signal. However, ICCA is very robust to the underlying correlation; the ICCA consistency boundary in (4.6.2) does not depend on the value of ρ. We also verify Proposition 4.5.1 showing that when $n < 300$, it is impossible to detect the presence of correlated signals using empirical CCA because $\widehat{\rho}_{\text{cca}} = 1$ deterministically. With ICCA, we avoid this undesirable property and can still detect the presence of a correlated signal for very small n and θ.

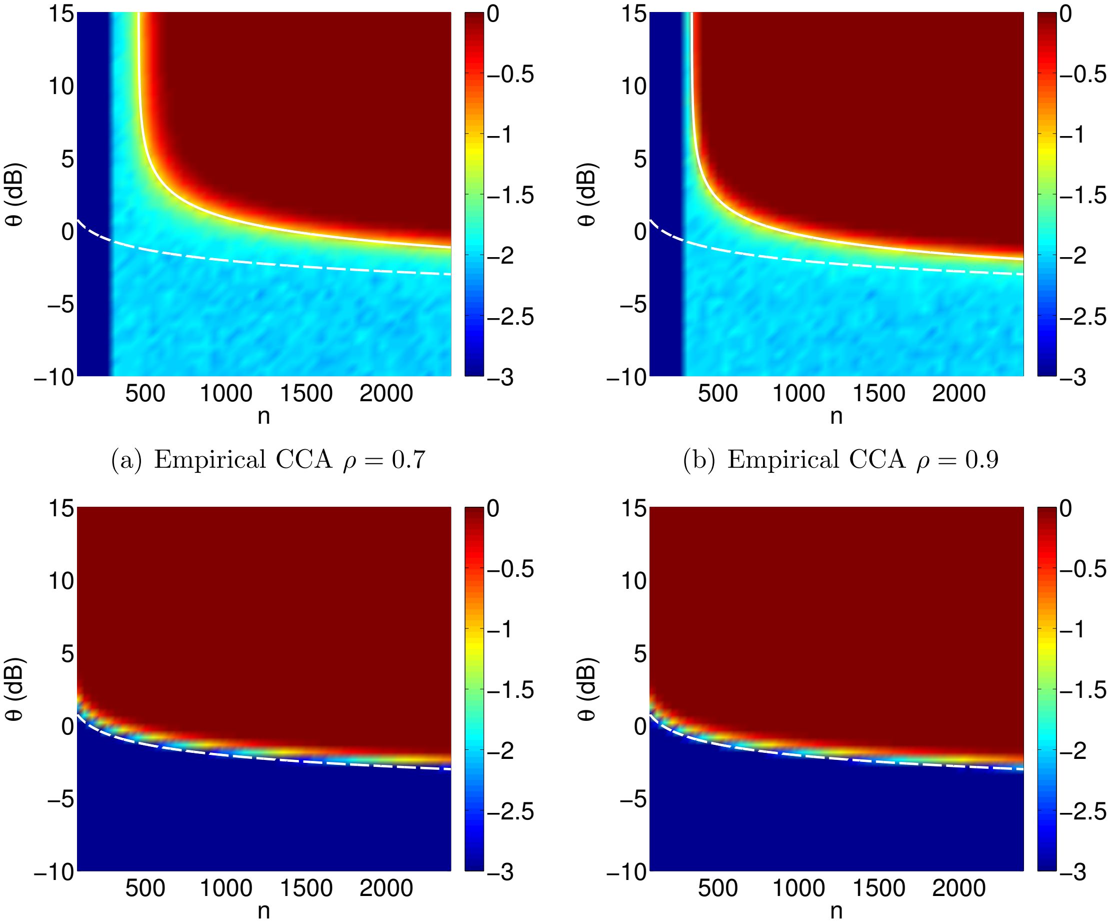
Next, we explore the minimum $1/c$ for $c = c_x = c_y$ needed to reliably detect $k = 1$ correlated signal in the experiment setting described for Figure 4.1. As $c = p/n = q/n$, the minimum $1/c$ is equivalent to the minimum number of samples needed for fixed dimensions. Using the theoretical phase transitions in Theorem 4.6.2, we have that this critical value of c is $c_{\text{crit}} = \theta^4$ for ICCA and $c_{\text{crit}} = \min\left(\frac{r_c^{\text{crit}}}{1+r_c^{\text{crit}}}, 0.5\right)$ for empirical CCA, where

$$r_c^{\text{crit}} = \left(\frac{-\rho + \sqrt{\rho^2 + 4\theta^2\rho^2(1+\theta^2\rho^2)}}{2(1+\theta^2\rho^2)}\right)^2 .$$

Figure 4.2 plots level sets of c_{crit} for empirical CCA and ICCA for various values of $\theta = \theta_1^{(x)} = \theta_1^{(y)}$ and $\rho = P_{xy}$. Recall that if $c > 0.5$, empirical CCA fails entirely, so for comparison we only show contour lines for $1/c = 10$ and $1/c = 3$.

From this figure, we once again observe that the performance of ICCA is independent of the value of $\rho = P_{xy}$, while the performance of empirical CCA is highly dependent on the correlation. This figure allows us to showcase that ICCA is theoretically better than empirical CCA in all parameter regimes as ICCA can achieve the same performance of empirical CCA given fewer samples at a lower SNR.

Finally, we show the detection ability of ICCA as a function of θ_x and θ_y for a rank-1 setting with $c_x = c_y = 1$ in Figure 4.3. This figure succinctly summarizes Theorem 4.6.2. When either θ_x or θ_y is less than the critical value of 1, we cannot reliably detect the presence of correlation between the two datasets. This corresponds to the blue and green regions in the figure. The blue region corresponds to when both SNRs are below the phase transition and neither signal is detectable. The green region corresponds to when only one SNR is above the phase transition; however, as the other SNR is below the phase transition, we still cannot detect the presence of correlation. Only when both SNRs are above the phase transition (yellow region) can we reliably detect the presence of correlation. Most importantly, the regions in this figure are independent of the correlation between the two datasets.



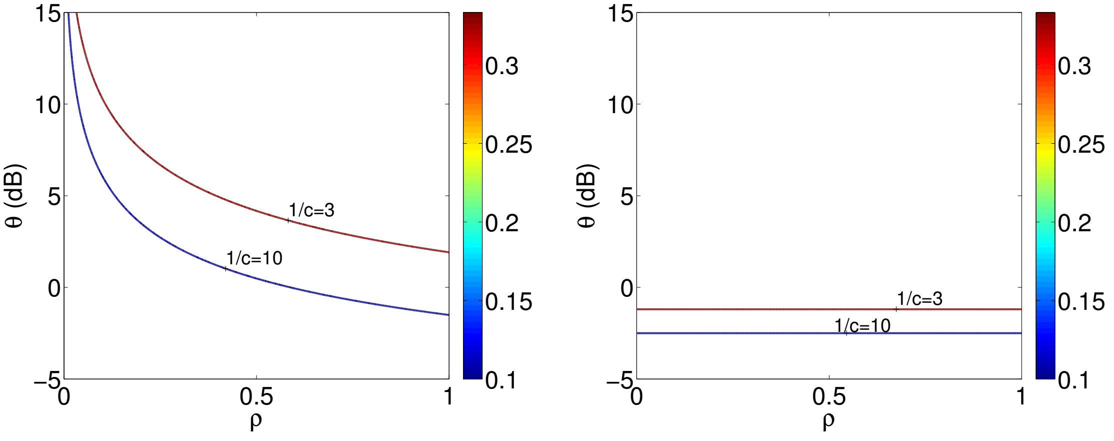
(a) Empirical CCA $\rho = 0.7$

(b) Empirical CCA $\rho = 0.9$

(c) ICCA $\rho = 0.7$

(d) ICCA $\rho = 0.9$

Figure 4.1: We generate data from (4.1) for $p = q = 150$, $k_x = k_y = 1$, $k = 1$, and various $\rho = P_{xy}$ and sweep over $\theta = \theta_1^{(x)} = \theta_1^{(y)}$ and n. We compute $\widehat{k}_x$ and $\widehat{k}_y$ as outlined in Appendix D for a significance value of $\alpha = 0.01$. Using these estimates, we compute $\widehat{\rho}_{\text{cca}}^{(1)}$ as the largest singular value of $\widehat{C}_{\text{cca}}$ as in (4.8) and $\widehat{\rho}_{\text{icca}}^{(1)}$ as the largest singular value of $\widehat{C}_{\text{icca}}$ as in (4.12). We then estimate the number of correlated signals $\widehat{k}_{\text{cca}}$ and $\widehat{k}_{\text{icca}}$ via (4.14) for a significance level of $\alpha = 0.01$. We repeat this for 10000 trials and compute the percentage of trials where $\widehat{k}_{\text{cca}} = 1$ and $\widehat{k}_{\text{icca}} = 1$. We plot $\log_{10}$ of these percentages for multiples values of θ and n. We plot the theoretical consistency boundary of CCA (given in Theorem 4.6.2 that relies on [2]) in a solid white line and the theoretical consistency boundary of ICCA (given in Theorem 4.6.2) in a dashed white line.



(a) Empirical CCA

(b) ICCA

Figure 4.2: Contour lines for minimum $1/c$ necessary for reliable detection of $k = 1$ correlated component. The quantity $1/c = n/p$ is equivalent to the number of samples per dimension of data. Figure 4.2(a) plots the contours for empirical CCA and Figure 4.2(b) plots the ICCA contours using the limits give in Theorem 4.6.2 for $c = c_x = c_y$. We plot the contours for $1/c = 10$ to $1/c = 3$. These plots clearly demonstrate the ICCA limits are independent of $\rho = P_{xy}$ while CCA is highly dependent on $\rho = P_{xy}$. For a fixed number of samples (fixed c), ICCA is reliably detect the presence of a correlated signal at lower SNR values than empirical CCA.

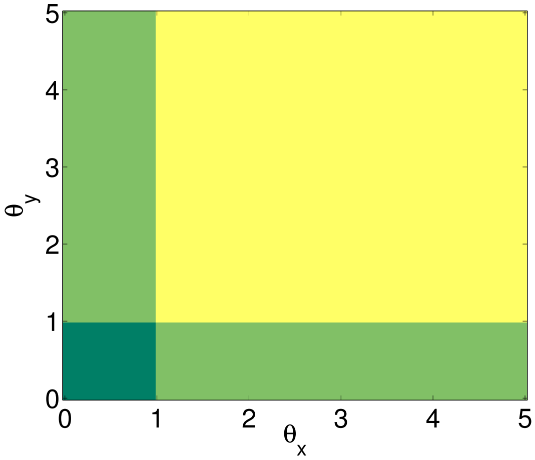


Figure 4.3: Theoretical detection regions for ICCA for a rank-1 setting where $c_x = c_y = 1$. In this setting, $\theta > 1$ implies that the corresponding subspace is informative. Therefore, in light of Theorem 4.6.2, we see that when both $\theta_x < 1$ and $\theta_y < 1$, neither subspace component is informative and we cannot detect the presence of a correlated signal. This corresponds to the blue region. When only one of θ_x or θ_y is above the phase transition (green region), we still cannot detect the presence of a correlated signal even though we have one informative signal. However, when both θ_x and θ_y are above the phase transition (yellow region), we can detect the presence of a correlated signal between the datasets. This detection ability is independent of the value of correlation between the datasets.

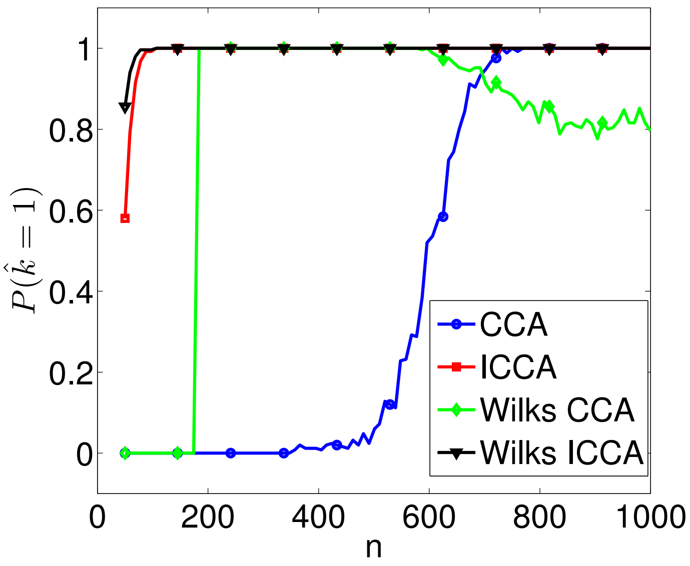


Figure 4.4: We generate data from (4.1) for $p = q = 150$, $k_x = k_y = 1$, $k = 1$, $\theta_1^{(x)} = \theta_1^{(y)} = 2$, $P_{xy} = 1$, and $\widehat{k}_x = \widehat{k}_y = 1$. We compute $\widehat{\rho}_{\text{cca}}^{(1)}$ as the largest singular value of $\widehat{C}_{\text{cca}}$ as in (4.8) and $\widehat{\rho}_{\text{icca}}^{(1)}$ as the largest singular value of $\widehat{C}_{\text{icca}}$ as in (4.12). We then estimate the number of correlated signals $\widehat{k}_{\text{cca}}$ and $\widehat{k}_{\text{icca}}$ using the Wilks test in (4.10) and our new test in (4.14) for a significance level of $\alpha = 0.01$. We repeat this for 250 trials and compute the percentage of trials where $\widehat{k}_{\text{cca}} = 1$ and $\widehat{k}_{\text{icca}} = 1$ for each significance test. We repeat this for multiple values of n and plot the results. We observe that the classical Wilk's Lambda test is suboptimal and results in a large number of false alarms.

4.8.2 Comparison to Wilks Lambda Test

Next we compare the classical Wilk's Lambda Test presented in Section 4.4.1 to the statistical tests developed in this section in (4.14). We create two synthetic rank-1 signal-plus-noise data matrices of dimension $p = 150$ and $q = 200$. We set the SNR of the signal in each dataset to $\theta_1^{(x)} = \theta_1^{(y)} = 2$ and the correlation between the signals to $\rho = P_{xy} = 0.9$. For multiple value of n, we compute the correlations returned by both ICCA and CCA. Using these correlations, we use our statistical tests in (4.14) to determine whether the largest correlations, $\widehat{\rho}_{\text{cca}}^{(1)}$ and $\widehat{\rho}_{\text{icca}}^{(1)}$, are significant. Similarly we use the Wilk's test in (4.10) to determine for both CCA and ICCA whether the largest correlation is significant. We note that for the Wilk's test, we need all $\min(p, q)$ correlations returned by CCA and all $\min(k_x, k_y)$ correlations returned by ICCA. We repeat this process for 250 trials for each values of n. Figure 4.4 plots the average percentage of trials where the correlation was significant for each statistical test for each algorithm.

From this figure, we observe that the classical Wilk's Lambda test is suboptimal for testing the presence of a correlation in the low-rank signal-plus-noise model for both empirical CCA and ICCA. We know from our analysis in Theorem 4.6.2 and the empirical exploration on synthetic datasets from the previous section that the empirical CCA and ICCA test statistics that we developed consistently estimate the presence of correlated signals in low-rank signal-plus-noise datasets. Examining the performance of the Wilk's test for determining the significance of the top ICCA correlation, we observe that it predicts the presence of correlation before our consistent test statistic. Therefore, using the Wilk's test statistic for ICCA will result in a high false alarm rate. Similarly for empirical CCA, we see that the Wilk's test is very bad in the sample-starved regime. Here our test statistic correctly predicts that the top empirical CCA correlation is not significant because it is deterministically one. However, the Wilk's test returns that the correlation is significant. Even worse, once we have enough samples so that the top empirical CCA correlation is indeed significant, the Wilk's test does not always predict a that the correlation is significant. Therefore, for empirical CCA, the Wilk's test will have a large false-alarm rate in the low-sample regime and a lower detection rate in the moderate-sample regime. The classical Wilk's test is suboptimal for determining the significance of correlations of both empirical CCA and ICCA and we encourage practitioners to instead use our consistent test statistics.

4.8.3 Simulated Missing Data

Next, we demonstrate the accuracy of the consistency bound for both empirical CCA and ICCA in the setting of missing data described in Theorem 4.7.1. Again, we consider a rank-1 setting ($k_x = k_y = 1$) but generate data from (4.22) for fixed $p = q = 150$ over various number of samples n, signal-to-noise ratio (SNR) $\theta = \theta_1^{(x)} = \theta_1^{(y)}$, various $\rho = P_{xy}$ (so that $k = 1$), and also various percentages of missing data $\gamma = \gamma_x = \gamma_y$. In all simulations, we use Algorithm 2 of [89] to estimate $\widehat{k}_x$ and $\widehat{k}_y$ using a significance level of $\alpha = 0.01$. We stack the data into matrices X and Y, and compute $\widehat{\rho}_{\text{cca}}^{(1)}$ and $\widehat{\rho}_{\text{icca}}^{(1)}$ from the SVD of of $\widehat{C}_{\text{cca}}$ and $\widehat{C}_{\text{icca}}$, respectively. Using these correlation estimates, we compute the estimated number of correlated components via (4.14) for a significance level of $\alpha = 0.01$. For a fixed set of parameters (n, θ, ρ, γ) we repeat the above process for 10000 trials and determine the percentage of trials where we detect $\widehat{k}_{\text{cca}} = 1$ and $\widehat{k}_{\text{icca}} = 1$. Figure 4.5 plots the $\log_{10}$ of this percentage for empirical CCA and ICCA, respectively. On each plot, we overlay the consistency boundary given by Theorem 4.7.1.

From these figures, we observe that Theorem 4.7.1 accurately predicts the phase transition for both empirical CCA and ICCA in the presence of missing data for a wide array of parameters. When $n < p + q$ empirical CCA is unable to detect the correlated signal while ICCA can reliably detect the correlated signal, even in the presence of missing data. In this missing data setting, we once again observe that the value of ρ affects the phase transition for empirical CCA but not for ICCA; it is harder for empirical CCA to detect signals with small correlations.

4.8.4 Controlled Flashing Lights Experiment

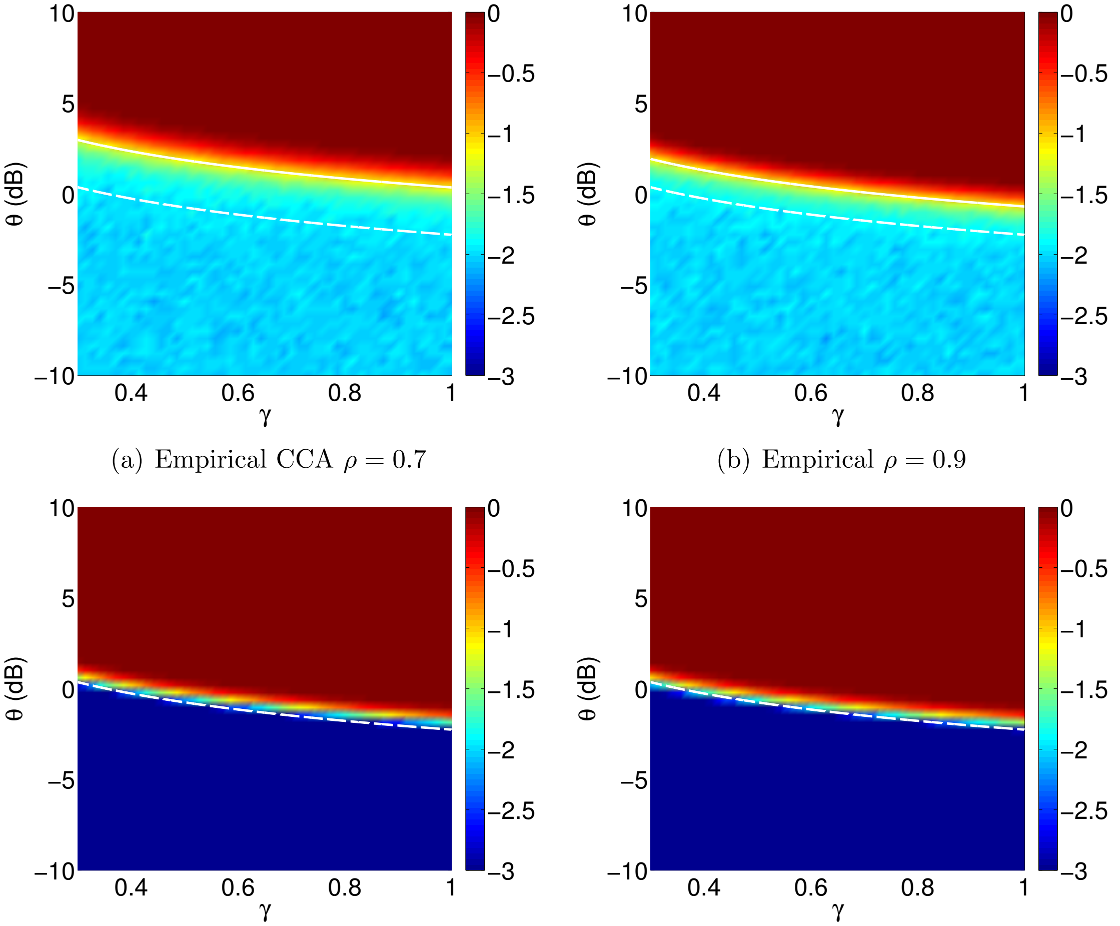
To verify the effectiveness of ICCA for real world applications, we conducted a controlled experiment consisting of 5 stationary flashing lights and two stationary iPhone cameras. Figure 4.6 shows the left and right camera views at one time point of our experiment and manually identifies each source. The 5 sources are a blue flashing police light (BPL) outlined in the green rectangle, one phone with a flashing strobe light (PH1) outlined in the dark blue rectangle, another phone with a flashing strobe light (PH2) outlined in a red rectangle, a tablet with a flashing screen (T1) outlined in the magenta rectangle, and a red flashing police light (RPL) outlined in the cyan rectangle. From left to right, the left camera can see BPL, PH1, and PH2. From left to right, the right camera can see PH2, T1, and RPL. Therefore, both cameras share the common signal of PH2.

To synchronize the cameras we used the RecoLive MultiCam iPhone app [1]. After turning on all light sources, we recorded 30 seconds of video at 30 frames per second. The resolutions of the iPhone's cameras were both 1920 × 1080 pixels.

To post-process the video data, we first converted the video streams to grayscale and then downsampled each spatial dimension by a factor of 8, resulting in a resolution of 240 × 135. We then vectorized each image and stacked the 900 frames into data matrices, both of dimension 32400 × 900. Finally, we subtract the mean from each dataset so that we may run PCA, empirical CCA, and ICCA on the zero-mean datasets, X_{left} and Y_{right}.

First, we run PCA on X_{left} and Y_{right} to identify the number of signals in each dataset. We know from our setup that each camera has 3 independent sources. Figure 4.7 plots the singular values of X_{left} and Y_{right}. Figures 4.8 and 4.9 plot the singular vector heatmaps corresponding to the top 3 singular values of X_{left} and Y_{right}, respectively. Each figure also overlays a thresholded version of the singular vectors onto the raw video. The threshold that we use is $\sqrt{\log(n)/n}$. From these figures, PCA does a good job at identifying the pixels containing a signal (flashing light).

[1] http://recolive.com/en/



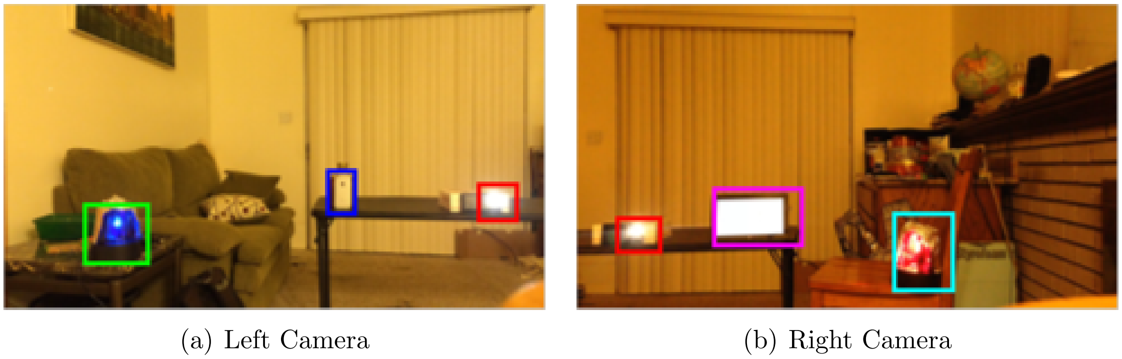
(a) Empirical CCA $\rho = 0.7$

(b) Empirical $\rho = 0.9$

(c) ICCA $\rho = 0.7$

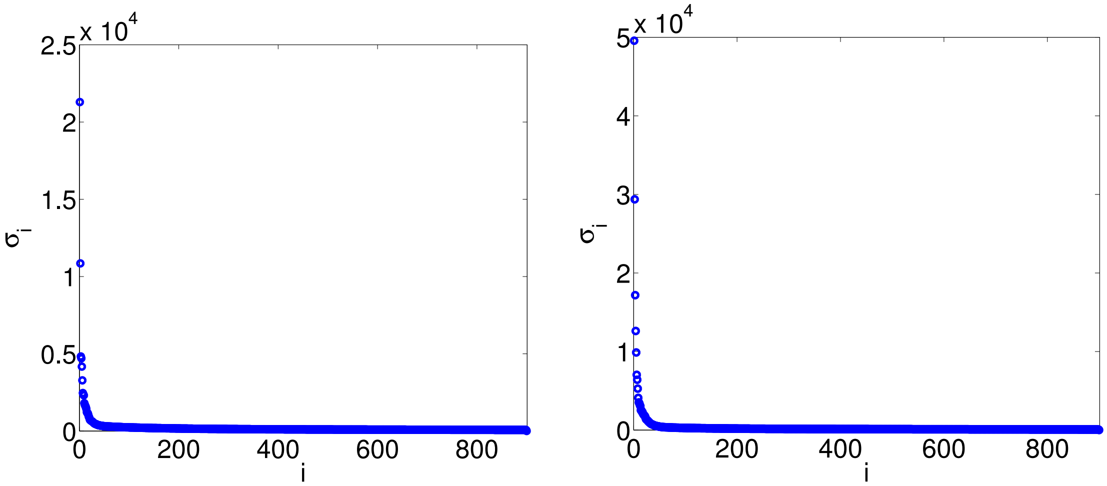
(d) ICCA $\rho = 0.9$

Figure 4.5: We generate data from (4.22) for $p = q = 150$, $k_x = k_y = 1$, $k = 1$, $n = 1200$, and various $\rho = P_{xy}$ and sweep over $\theta = \theta_1^{(x)} = \theta_1^{(y)}$ and $\gamma = \gamma_x = \gamma_y$. We compute $\widehat{k}_x$ and $\widehat{k}_y$ as outline in Appendix D for a significance value of $\alpha = 0.01$. Using these estimates, we compute $\widehat{\rho}_{\text{cca}}^{(1)}$ as the largest singular value of $\widehat{C}_{\text{cca}}$ as in (4.8) and $\widehat{\rho}_{\text{icca}}^{(1)}$ as the largest singular value of $\widehat{C}_{\text{icca}}$ as in (4.12). We then estimate the number of correlated signals $\widehat{k}_{\text{cca}}$ and $\widehat{k}_{\text{icca}}$ via (4.14) for a significance level of $\alpha = 0.01$. We repeat this for 10000 trials and compute the percentage of trials where $\widehat{k}_{\text{cca}} = 1$ and $\widehat{k}_{\text{icca}} = 1$. We plot $\log_{10}$ of these percentages for multiples values of θ and n. We plot the theoretical consistency boundary of empirical CCA (given in Theorem 4.7.1) in a solid white line and the theoretical consistency boundary of ICCA (given in Theorem 4.7.1) in a dashed white line.



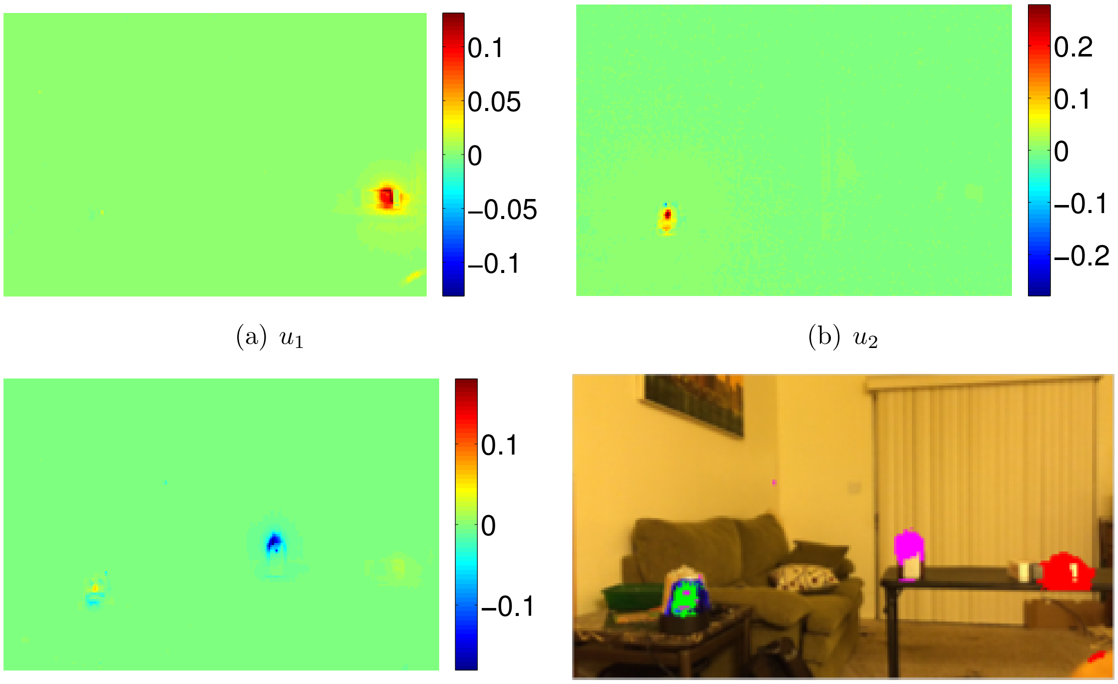
(a) Left Camera (b) Right Camera

Figure 4.6: Left and right camera views of our experiment with boxes manually identifying each source. Both cameras share a common flashing phone, outlined in a red rectangle. Each camera has two independent sources besides the shared flashing phone.



(a) Left camera (b) Right camera

Figure 4.7: Singular value spectra of X_{left} and Y_{right} for the flashing light experiment.



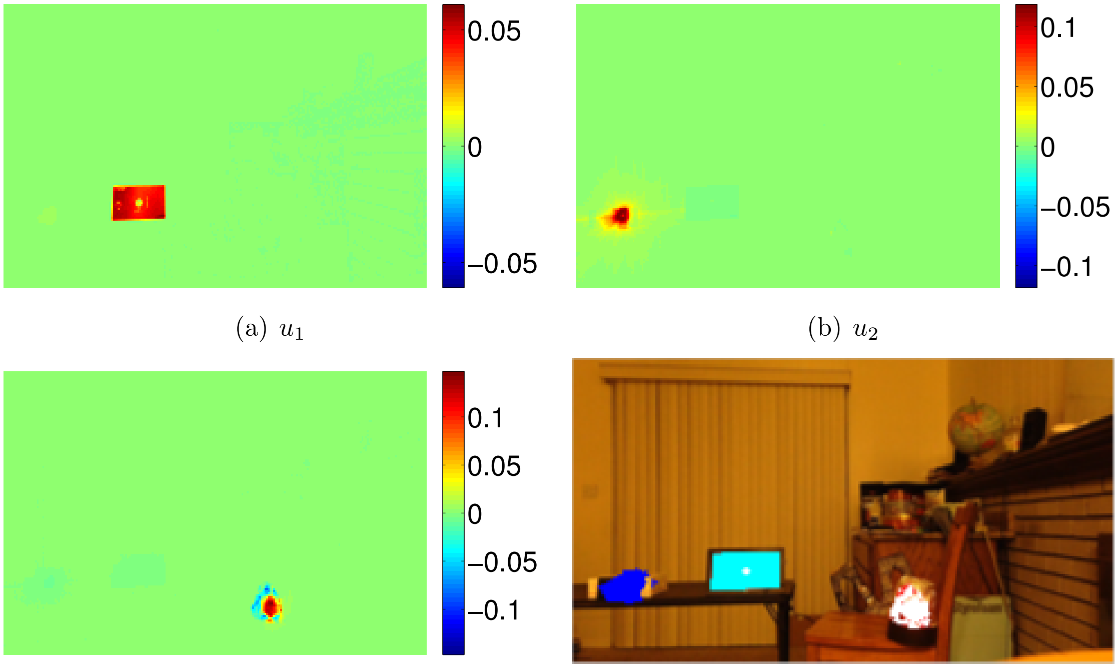
(a) u_1

(b) u_2

(c) u_3

(d) Overlay

Figure 4.8: (a)-(c) Left singular vectors of X_{left} corresponding to the top 3 singular values in Figure 4.7(a). (d) Thresholded singular vectors from (a)-(c) overlayed onto original scene. We use a threshold of $\log(p)/\sqrt{p}$ where $p = 32400$ pixels. These correspond to the 3 light sources visible in the left camera.The green pixels identify BPL; the magenta pixels identify PH1; the red pixels identify PH2.



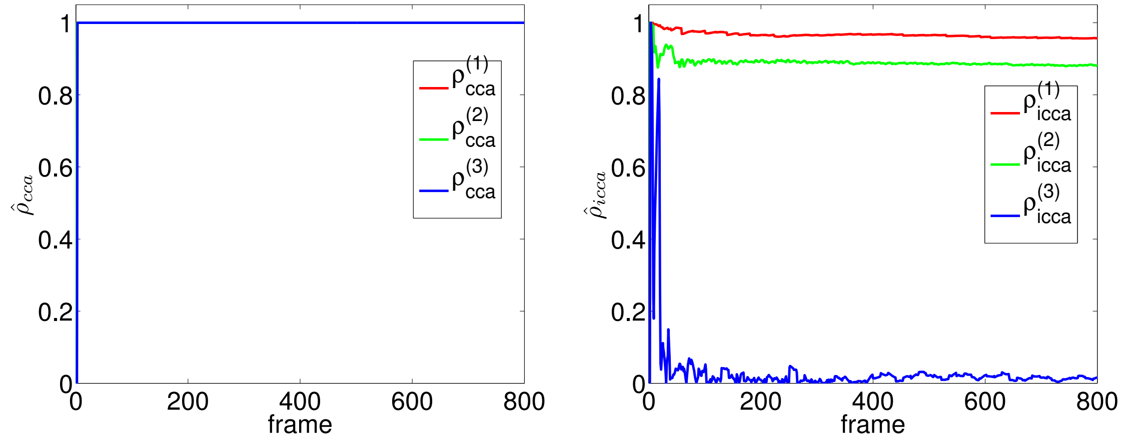
(a) u_1

(b) u_2

(c) u_3

(d) Overlay

Figure 4.9: (a)-(c) Left singular vectors of Y_{right} corresponding to the top 3 singular values in Figure 4.7(b). (d) Thresholded singular vectors from (a)-(c) overlayed onto original scene. We use a threshold of $\log(p)/\sqrt{p}$ where $p = 32400$ pixels. These correspond to the 3 light sources visible in the right camera. The dark blue pixels identify PH2; the cyan pixels identify T1; the white pixels identify RPL.



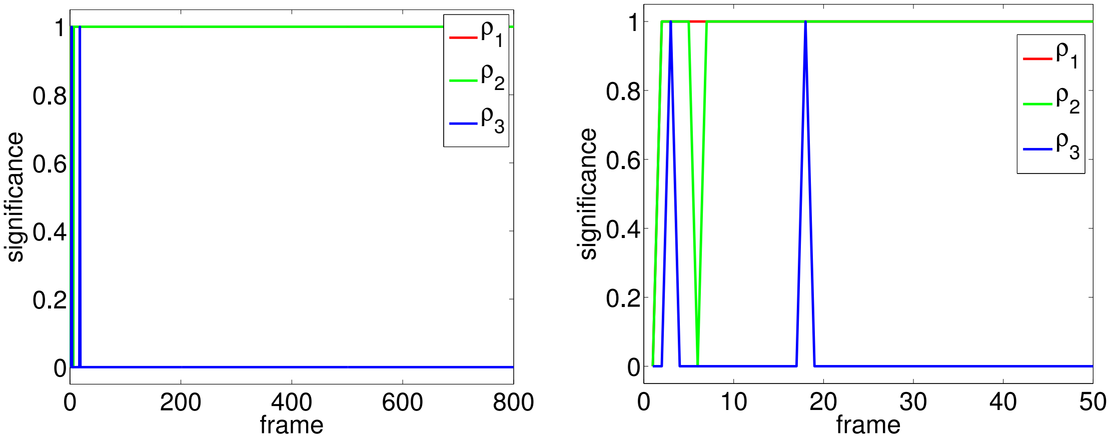
(a) CCA

(b) ICCA

Figure 4.10: (a) Top three singular values returned by empirical CCA as defined in (4.8). As we are in the sample deficient regime, these singular values are deterministically 1. (b) Top three singular values returned by ICCA as defined in (4.12). ICCA correctly identifies two sources of correlation.

However, PCA does not provided any information about whether the identified signals are correlated across cameras. To identify correlated pixels between the cameras, we run empirical CCA and ICCA after each new video frame. For frame ℓ, we construct the $32400 \times \ell$ submatrices X^{ℓ}_{left} and Y^{ℓ}_{right} by taking the matrix of the first ℓ original vectorized frames and zero meaning it. We then use these matrices as the input to empirical CCA and ICCA. Using our knowledge of 3 sources present in each camera, we set $\widehat{k}_x = \widehat{k}_y = 3$. Figure 4.10 plots the top 3 correlation coefficients returned by empirical CCA and ICCA over the first 800 frames. Intuitively, empirical CCA returns perfect correlation as we have only a few frames but a large dimension (pixels).

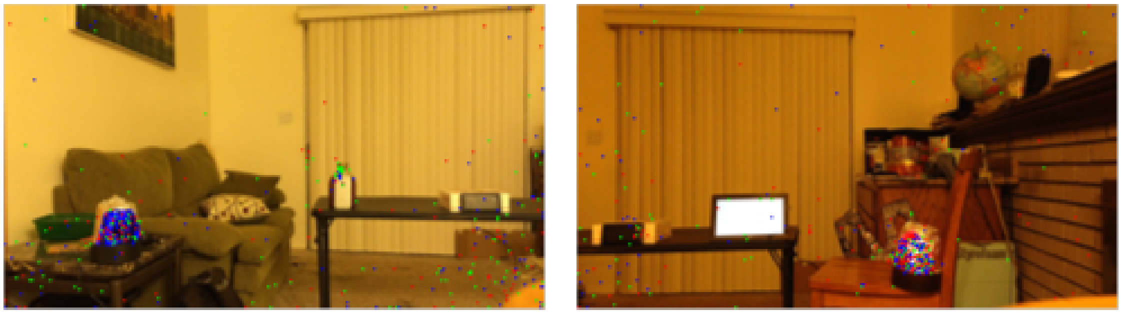
Using these correlation coefficients, we determine which ones are significant for a significance level of $\alpha = 0.01$ using (4.14). Unsurprisingly, all correlations returned by empirical CCA are insignificant and we do not plot the results. However, we plot whether the ICCA correlations are significant in Figure 4.11. After about 20 frames, ICCA identifies 2 significant correlations. Similar to Figures 4.8(d) and 4.9(d), we overlay the thresholded unit-norm canonical vectors onto the original images in Figures 4.12 and 4.13 for empirical CCA and ICCA, respectively. The empirical CCA canonical vectors appear to be very random and noisy. The canonical vector corresponding to the largest ICCA correlation selects the pixels of the shared flashing camera.



(a) ICCA

(b) ICCA - first 50 frames

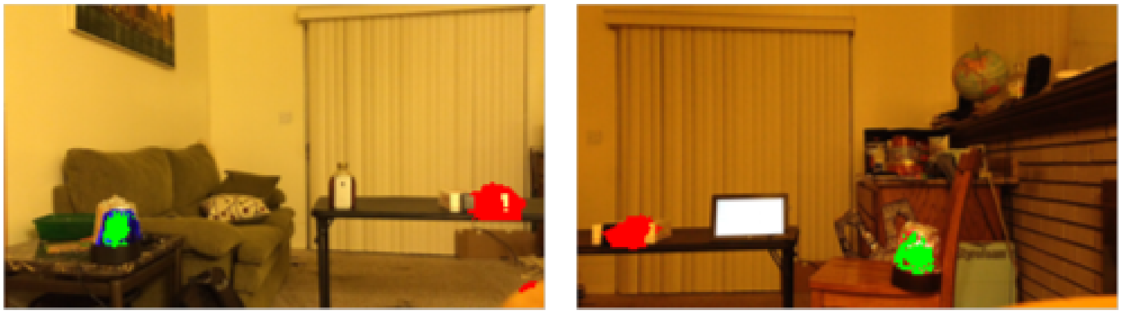
Figure 4.11: Significance of the top singular value returned by ICCA in Figure 4.10(b) using (4.14) with $\alpha = 0.01$. A value of zero represents that the singular value is not significant. A value of one represents that the singular value is significant. (a) Significance over all 800 frame. (b) Zoomed in to the first 50 frames in (a).



(a) Left Camera

(b) Right Camera

Figure 4.12: Top 3 threholded empirical CCA canonical vectors overlayed on the original scene after 800 frames as computed in (4.9). The red pixels correspond to the vector with the highest correlation, the green pixels correspond to the vector with the second highest correlation, and the blue pixels correspond to the vector with the third highest correlation. We use a threshold of $\log(p)/\sqrt{p}$ where $p = 32400$ pixels.



(a) Left Camera

(b) Right Camera

Figure 4.13: Top 2 threholded ICCA canonical vectors overlayed on video after 800 frames as computed in (4.13). The red pixels correspond to the vector with the highest correlation and the green pixels correspond to the vector with the second highest correlation. We use a threshold of $\log(p)/\sqrt{p}$ where $p = 32400$ pixels.

Given that our experiment setup has only one shared flashing light, it is initially surprising that ICCA returns a second significant correlation. Examining the canonical vector overlay in Figure 4.13, we observe that this correlation corresponds to RPL and BPL. Figure 4.14 examines the right singular vectors returned by PCA corresponding to RPL and BPL. We observe that these light sources have approximately the same period and even though they were started at random times, they are in approximate antiphase, making them correlated. This is especially interesting because neither camera can see both sources, but ICCA is still able to reveal a latent correlation inherent in the period and phase of these lights.

4.8.5 Controlled Flashing Lights with Missing Data

Using the same dataset in the previous section, we add missing data to each frame independently. We set $\gamma = \gamma_x = \gamma_y = 0.75$ so that about 25% of the pixels are set to 0. We generate the missing pixels independently for each camera and for each frame. We then process the data exactly as above without missing data. We note that in this setup, our light sources do not obey the low-coherence condition, but we still run ICCA to demonstrate it's robustness. Particularly, source PH1 is very small and it's signal is very spiked and violates the low-coherence assumption the most. It is unsurprising that it is not detected by PCA, as we will see.

Figure 4.15 plots an example of a frame for each camera with missing data. It is much more difficult to make out the scene even while retaining 75% of the data. Figure 4.16 overlays the thresholded PCA vectors onto each camera after 800 frames. For the right camera, these vectors still identify all three visible sources. However for the left

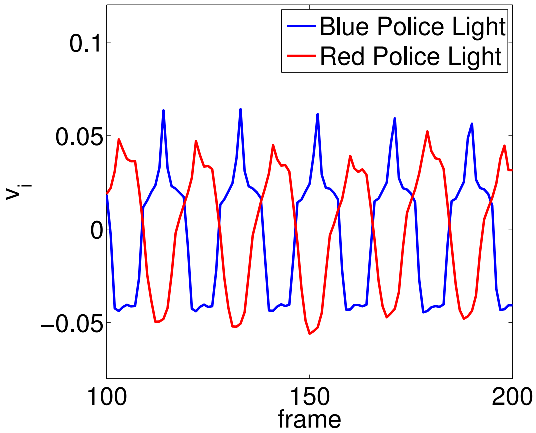
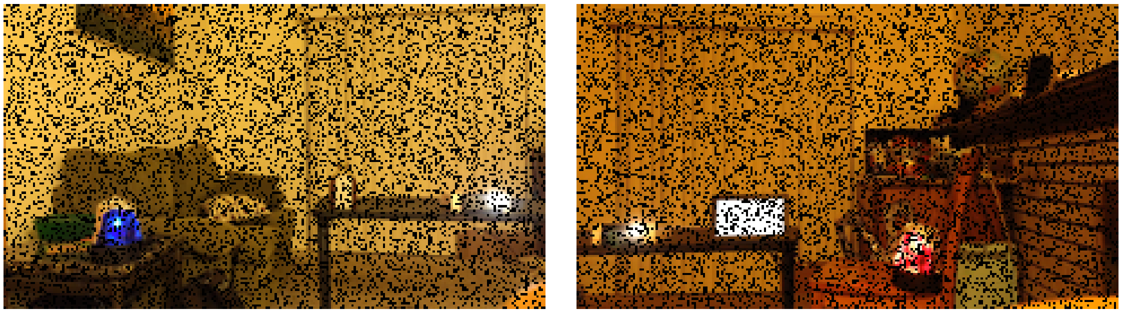


Figure 4.14: A portion of the right singular vectors of X_{left} (blue) and Y_{right} (red) corresponding the flashing police lights in each camera view. Both sources have very similar periods and are approximately in antiphase and therefore are correlated.



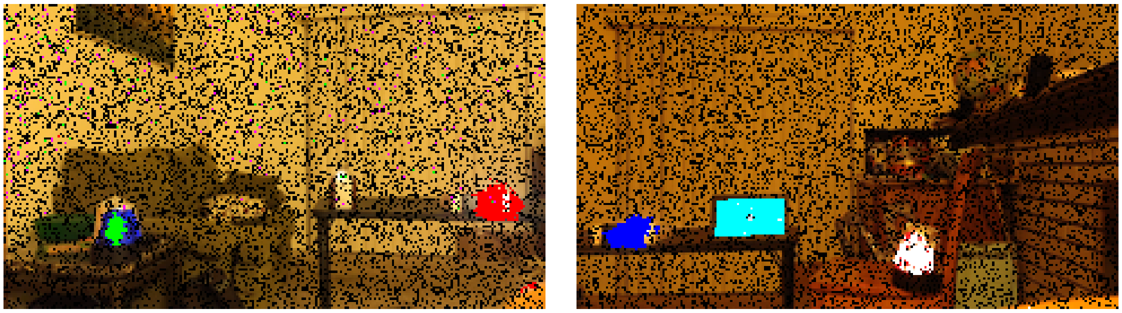
(a) Left Camera

(b) Right Camera

Figure 4.15: Left and right camera views of our five sources in the presence of missing data.

camera, we only identify BPL and PH2. This makes sense because PH1 drastically violates the low-coherence condition. However, as this signal is not correlated with the others, we can still attempt to use ICCA to find correlated signals.

Figure 4.17 overlays the thresholded canonical vectors corresponding to the top 2 empirical CCA canonical correlations onto the original scene after 800 frames. Unsurprisingly, empirical CCA is still unable to detect the two correlated signals because in this regime the top correlations are deterministically one and the corresponding canonical vectors are uninformative. However, ICCA is able to detect our correlated signals even in the presence of missing data. Figure 4.18 overlays the thresholded canonical vectors corresponding to the top 2 ICCA canonical correlations onto the original scene after 800 frames. The colored pixels clearly identify our two sources of



(a) Left Camera

(b) Right Camera

Figure 4.16: Top 3 threholded PCA vectors overlayed on each video after 800 frames. This is analogous to Figures 4.8 and 4.9 but with missing data. Again we use the threshold $\log(p)/\sqrt{p}$ for $p = 32400$. A different color is used for the every vector. We note that the middle source of the left camera violates the low-coherence assumption in Assumption 4.7.1 and so PCA does not detect it.

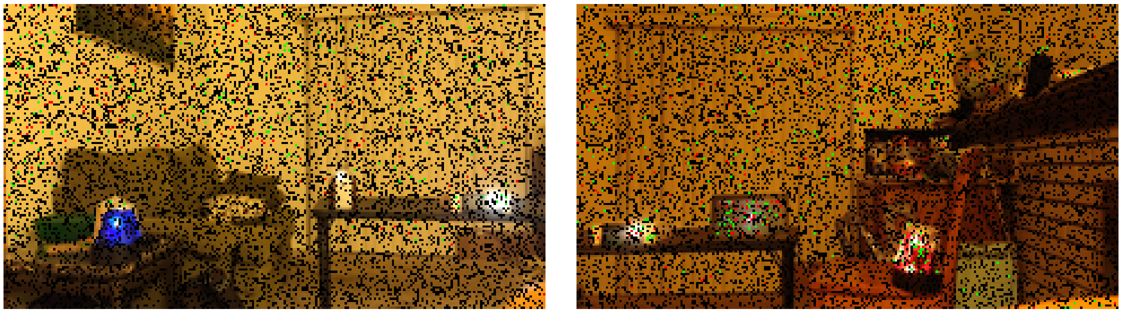
correlation.

Figure 4.19 plot the top 3 canonical correlations for empirical CCA and ICCA. Unsurprisingly, the correlations reported by empirical CCA are 1 and are not significant. We plot the corresponding significance of the ICCA correlations in Figure 4.20. From these two figures, we see that ICCA is very quickly able to determine that the shared source PH1 is correlated. At first, this is the only significant correlation. However, after about 200 frames (about 7 seconds), the correlation corresponding to the police lights becomes significant.

4.8.6 Controlled Audio Visual Experiment

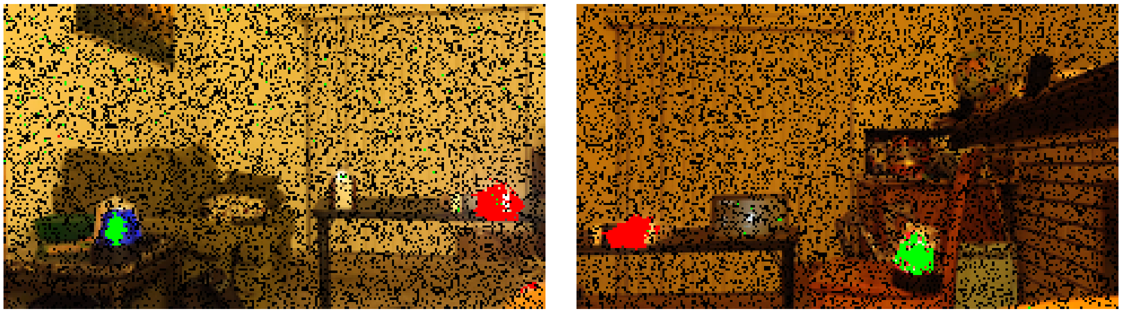
Similar to the flashing light experiment, we verify the effectiveness of ICCA with an audio visual controlled experiment. In this experiment, we play an audio sequence containing three different pure-tones, each amplitude modulated (AM) at a different frequency. In addition, we add uncorrelated coffee shop noise [2]. In the video sequence there are two flashing block-M's, one of which is flashing at the same AM frequency as one of the pure tone audio signals. The audio sequence is sampled at 44.1kHz and the images are each 553 × 1000 pixels, for a total of 20 seconds. Figure 4.21 shows the images and identifies the two sources. Figure 4.22 plots the full spectrogram of our audio signal and zooms in on a smaller portion of the spectrum to see the three AM signals, which are described in Table 4.1. Our audio waveform is

[2] https://www.youtube.com/watch?v=TpdFVSi7PZ8


(a) Left Camera

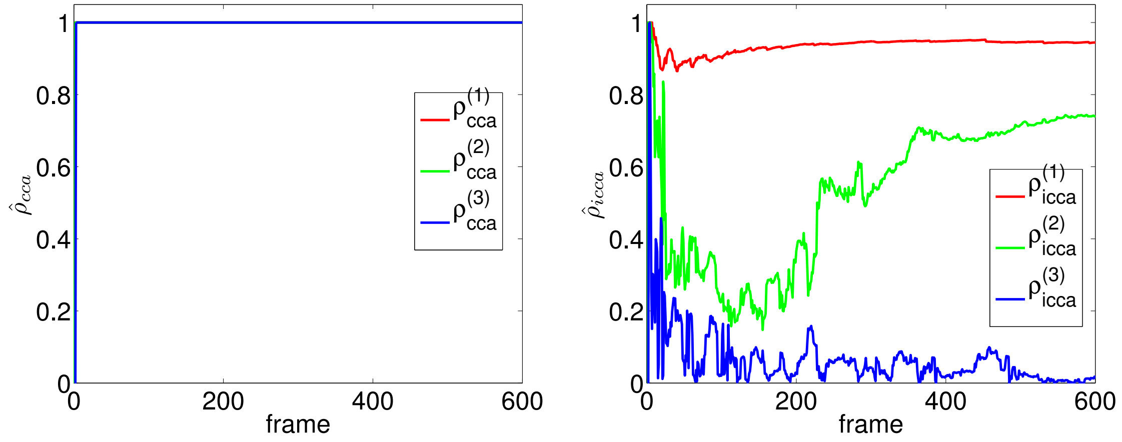
(b) Right Camera

Figure 4.17: Top 2 threholded empirical CCA canonical vectors overlayed on missing data video as computed in (4.9). Again we use the threshold $\log(p)/\sqrt{p}$ for $p = 32400$. The red pixels correspond to the vector with the highest correlation and the green pixels correspond to the vector with the second highest correlation in Figure 4.19(a).


(a) Left Camera

(b) Right Camera

Figure 4.18: Top 2 threholded ICCA canonical vectors overlayed on missing data video as computed in (4.13). Again we use the threshold $\log(p)/\sqrt{p}$ for $p = 32400$. The red pixels correspond to the vector with the highest correlation and the green pixels correspond to the vector with the second highest correlation in Figure 4.19(a).



(a) CCA

(b) ICCA

Figure 4.19: (a) Top three singular values returned by empirical CCA as defined in (4.8). As we are in the sample deficient regime, these singular values are deterministically 1. (b) Top three singular values returned by ICCA as defined in (4.12). ICCA correctly identifies two sources of correlation. As our data matrices now have missing data, it takes more frames for ICCA to identify the two sources of correlations.

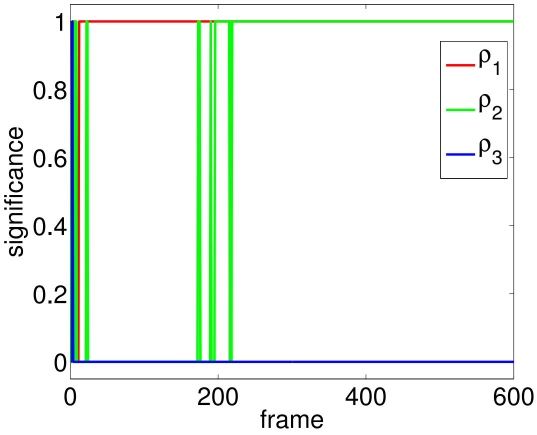


Figure 4.20: Significance of the top singular value returned by ICCA in Figure 4.19(b) using (4.14) with $\alpha = 0.01$. A value of zero represents that the singular value is not significant. A value of one represents that the singular value is significant.

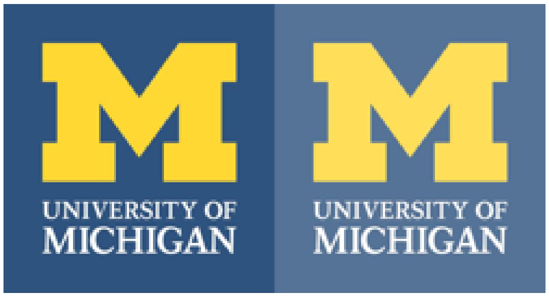
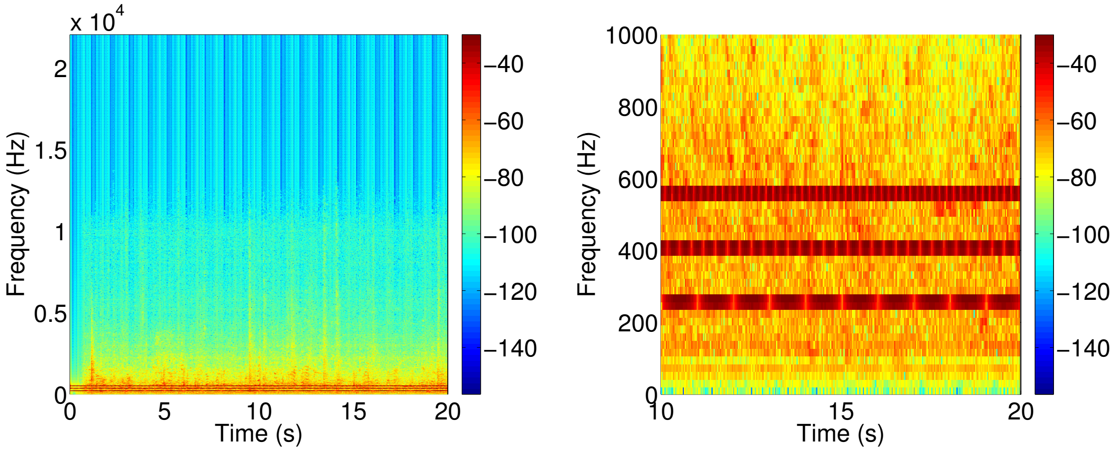


Figure 4.21: Still shot of the video. The block M on the left is amplitude modulated at 1 Hz, the same rate as $s_1(t)$ in (4.27), while the block M on the right is amplitude modulate at 2.15 Hz, which is different than all other audio and visual sources.



(a) Full Spectrogram

(b) Zoomed Spectrogram

Figure 4.22: (a) Full spectrogram of the audio signal in (4.26). (b) Zoomed in spectrogram of (a) to see the 3 audio sources at 250 Hz, 400 Hz, and 550 Hz. Each sources is also amplitude modulated at a different frequency as described in (4.27).

$$a(t) = \frac{1}{4}s_1(t) + \frac{1}{4}s_2(t) + \frac{1}{4}s_3(t) + \frac{1}{4}n(t) \tag{4.26}$$

where

$$\begin{aligned} s_1(t) &= |\sin(2\pi t/2)| \sin(2\pi(250t)) \\ s_2(t) &= |\cos(2\pi(3/2t))| \sin(2\pi(400t)) \\ s_3(t) &= |\cos(2\pi(5/2t))| \sin(2\pi(550t)) \\ n(t) &= \text{coffee shop noise.} \end{aligned} \tag{4.27}$$

To post-process the video data, we first converted the video streams to grayscale and then downsampled each spatial dimension by a factor of 4, resulting in a resolution of 133 × 250. We then ignore the first and last second of data and vectorized the image

Type	Source	AM Frequency
Visual	Left Block M	1 Hz
	Right Block M	2.15 Hz
Audio	250 Hz pure tone	1 Hz
	400 Hz pure tone	3 Hz
	550 Hz pure tone	7 Hz
	coffee shop noise	

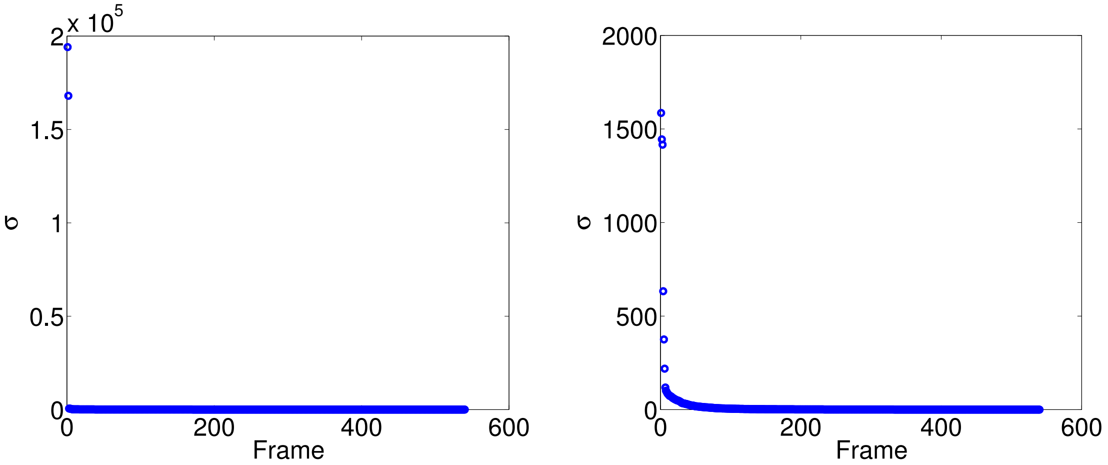
Table 4.1: Summary of the audio and visual sources and their amplitude modulated signals. The audio sources are described in (4.27) and the video sources are shown in Figure 4.21. The 250 Hz pure tone is amplitude modulated at the same frequency as the left block M and is thus correlated with it.

and stacked the resulting 540 frames into a data matrix of dimension 33250 × 540. Finally, we subtract the mean from the video dataset so that we may run PCA, empirical CCA, and ICCA on the zero-mean dataset, X_{video}.

To post-process the audio data, we separate the audio stream into equal window sizes of 1470 time points centered on every 1/30 second of data. On each 1470 time point segment, we run a 2048 point FFT and take the magnitude of the first 1025 points as a feature vector. We process the 18 seconds of data, stack the feature vectors into a matrix, and then subtract the mean, resulting in a 1025 × 540 matrix Y_{audio}.

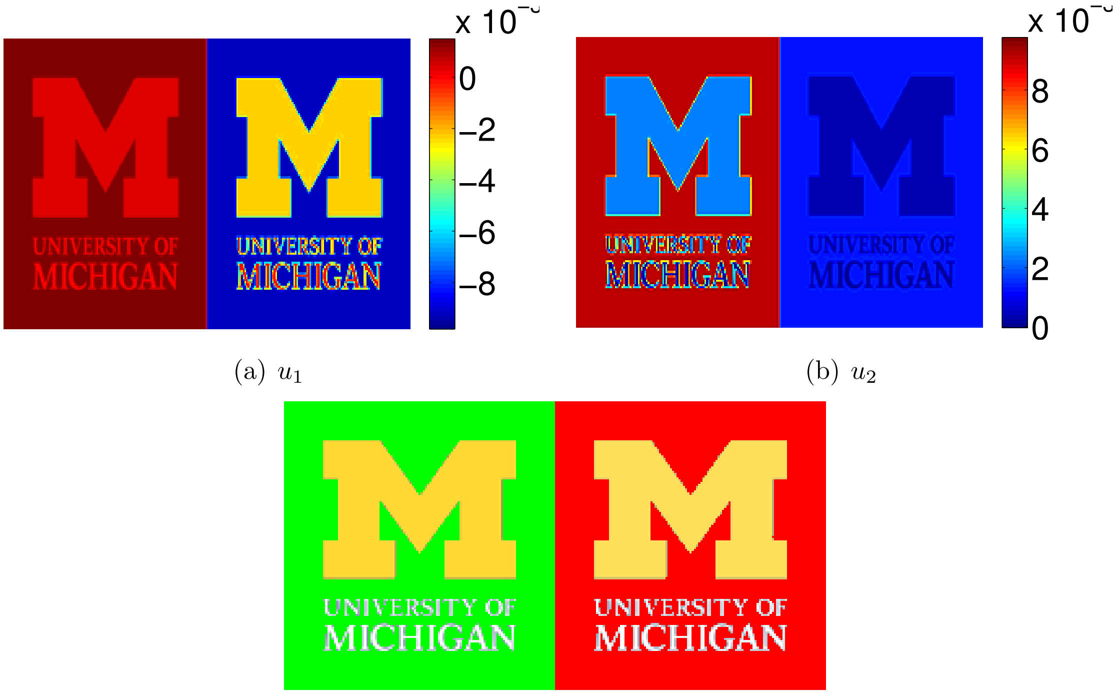
First, we run PCA on X_{video} and Y_{audio}. Figure 4.23 plots the singular value of each dataset. From our setup, we know that the video dataset has 2 signals and the audio dataset has 3 sources. Figure 4.24 plots the singular vector heatmaps corresponding to the top 2 singular values of X_{video}. These heatmaps clearly identify the two block M's. We then threshold the absolute value of the singular vectors with the threshold $1/\sqrt{p}$ and overlay it on top of the original scene. Similarly, Figure 4.25 plots the singular vectors of Y_{audio} corresponding to the top 3 singular values. By thresholding these singular vectors, we create audio filters, as Figure 4.25(c) shows.

However, PCA does not provide any information about whether the identified audio visual signals are correlated. To identify such correlations, we run empirical CCA and ICCA after each new video frame. For frame ℓ, we construct the $33250 \times \ell$ submatrix X_{video}^{ℓ} and $1025 \times \ell$ submatrix Y_{audio}^{ℓ} by taking the matrix of the first ℓ original vectorized frames and zero meaning it. We then use these matrices as the input to empirical CCA and ICCA. Using knowledge of our experimental setup, we set $\widehat{k}_x = 2$ and $\widehat{k}_y = 3$. Figure 4.26 plots the top 2 correlation coefficients returned by empirical CCA and ICCA over the first 540 frames. Intuitively, empirical CCA returns perfect correlation as we have only a few frames but a large dimension (pixels).



(a) Video (b) Audio

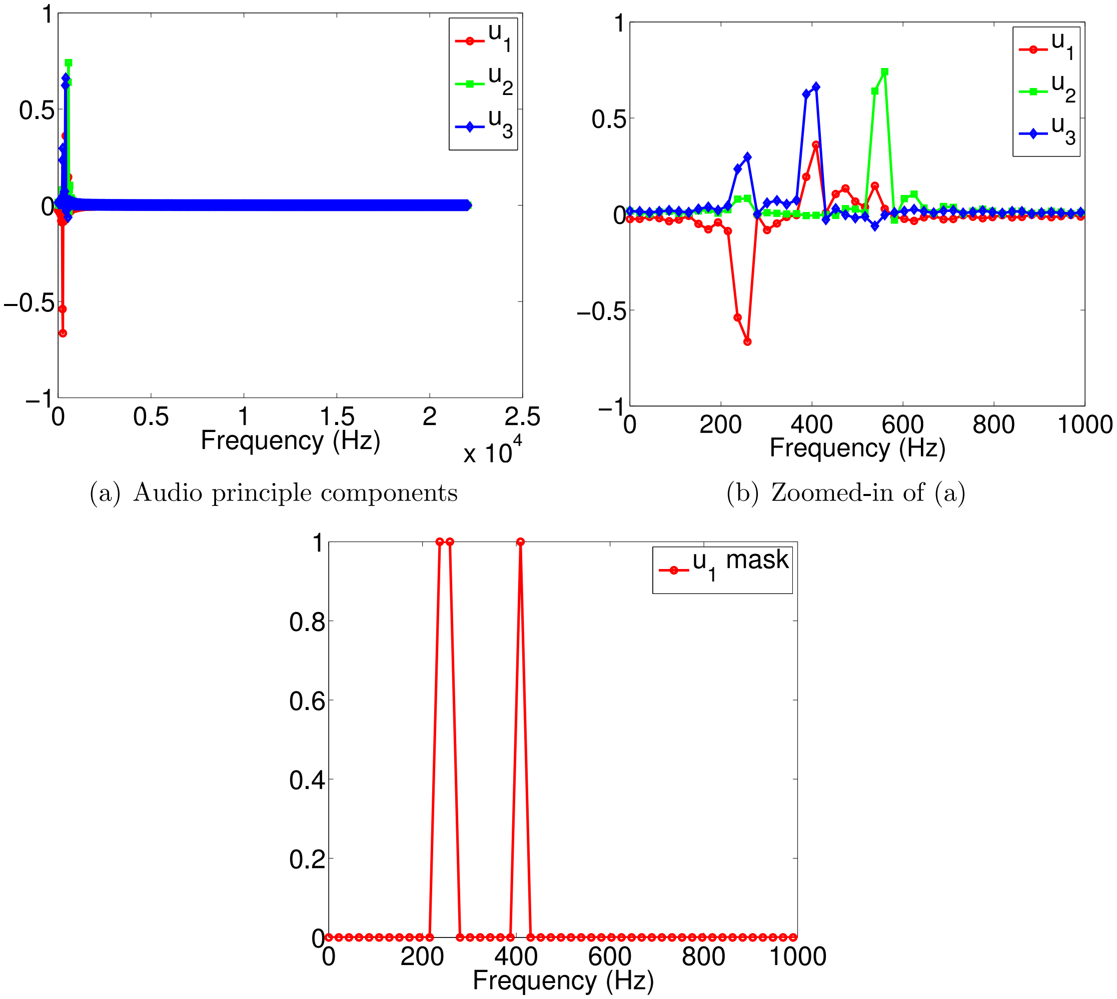
Figure 4.23: Singular value spectra of X_{video} and Y_{audio}.



(a) u_1 (b) u_2

(c) u_3

Figure 4.24: (a)-(b)) Left singular vectors of X_{video} corresponding to the top 2 singular values. (c) Thresholded singular vectors from (a)-(b) overlayed onto original scene with pixels from (a) in red and pixels from (b) in green. We use the threshold $\log(p)/\sqrt{p}$ for $p = 33250$. These vectors correspond to the 2 block M's.

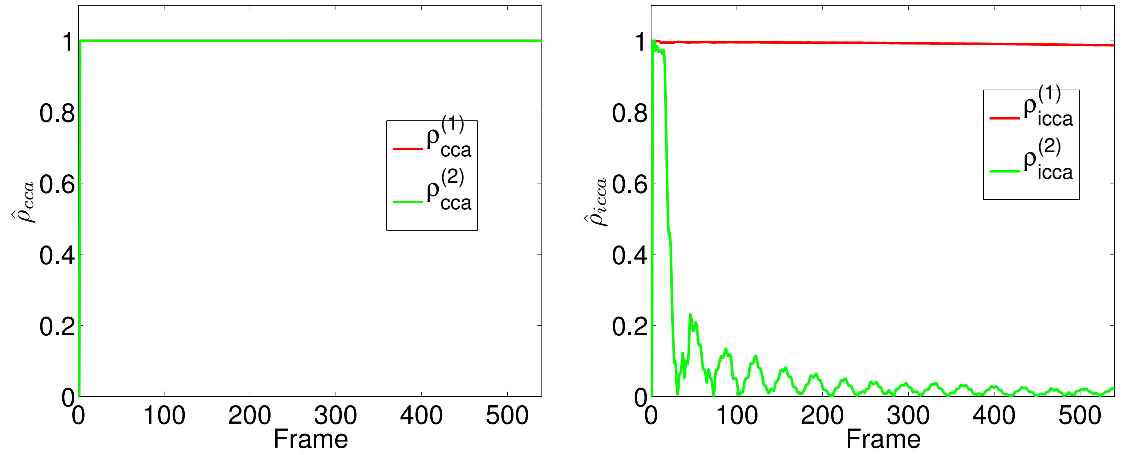


(a) Audio principle components

(b) Zoomed-in of (a)

(c) Example Filter

Figure 4.25: (a) Left singular vectors of Y_{audio} corresponding to the top 3 singular values. (b) Zoomed in version of (a) to see the three audio sources. (c) Masked principle component formed by thresholding the singular vectors with $\sqrt{\log(q)/q}$ for $q = 1025$.



(a) Empirical CCA

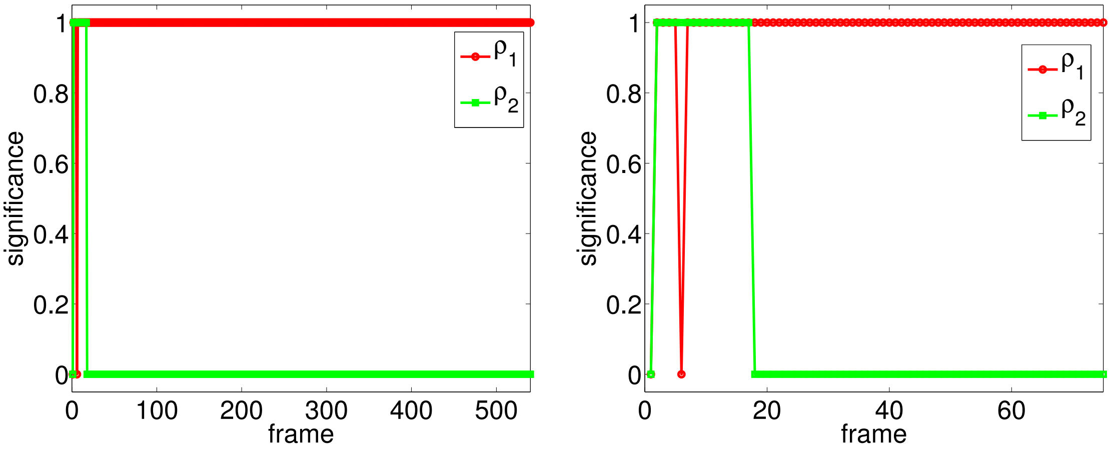
(b) ICCA

Figure 4.26: (a) Top two singular values returned by empirical CCA as defined in (4.8). As we are in the sample deficient regime, these singular values are deterministically 1. (b) Top two singular values returned by ICCA as defined in (4.12). ICCA correctly identifies the one source of correlation present in the audio-video dataset.

While the canonical correlation coefficients returned by CCA are insignificant because we operate in the sample deficient regime, we may use (4.14) to determine whether the canonical correlation coefficients returned by ICCA are significant. Using a significance level of $\alpha = 0.01$, Figure 4.27 plots whether the ICCA canonical correlations are significant. Unsurprisingly, after 20 frames (2/3 seconds) we identify the only correlated source and show that the second correlation is insignificant. The only significant correlation corresponds to the one correlated audio-visual signal.

Figure 4.28 uses the first empirical CCA canonical vectors to filter both the audio and video stream to highlight the correlated component. Using the thresholded first audio canonical vector as a bandpass filter, we filter the original audio stream using the overlap-save method and plot the resulting spectrogram in Figures 4.28(a) and 4.28(b). From the spectrogram, we see that the canonical vectors filters random frequencies and as time goes on, filter almost everything. Similarly, we threshold the video canonical vector and plot the pixels that are above the threshold in Figure 4.28(c). The pixels congregate around the text in both block M's, which is not correlated. These figures demonstrate that empirical CCA fails to identify the pixels that are correlated to any audio source and instead returns random filters for both the audio and video sources.

However, ICCA is able to find the underlying correlated audio-video source, as shown in Figure 4.29. Just as in the empirical CCA analysis, we use the thresholded



(a) ICCA

(b) ICCA - first 75 frames

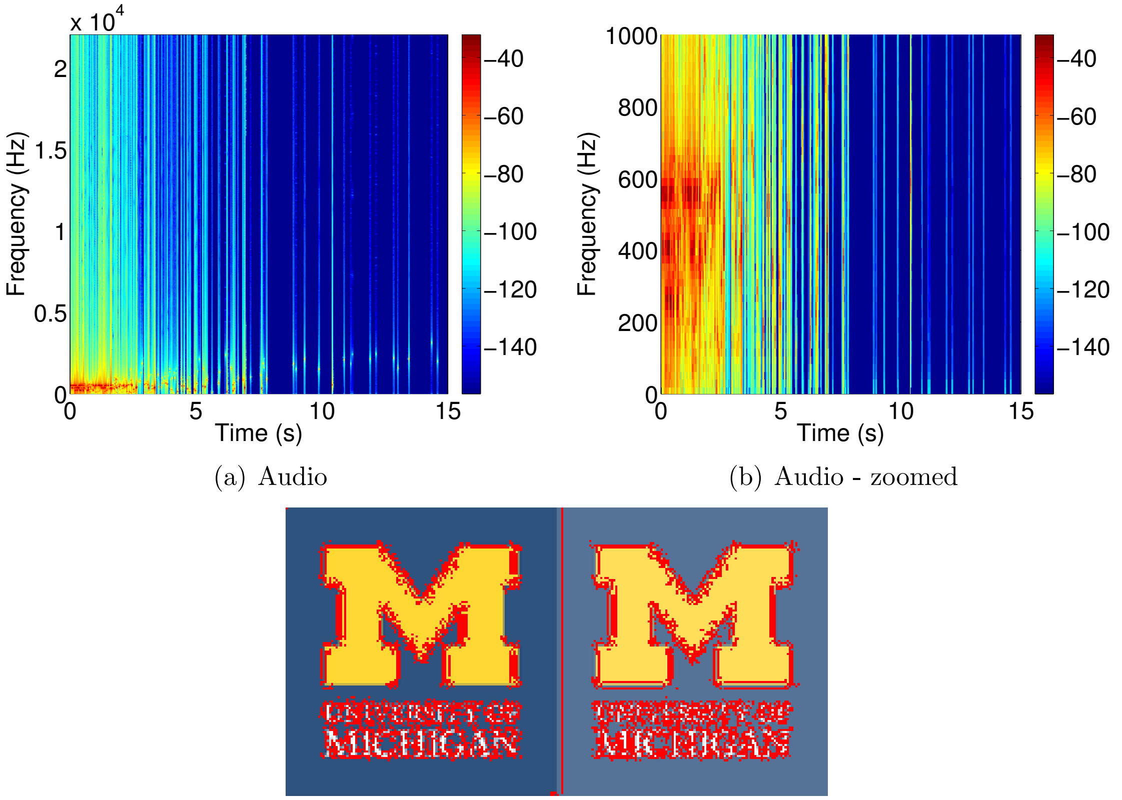
Figure 4.27: Significance of the top singular value returned by ICCA in Figure 4.26 using (4.14) with $\alpha = 0.01$. A value of zero represents that the singular value is not significant. A value of one represents that the singular value is significant. (a) Significance for all 540 frames. (b) Zoomed in of (a) to examine the first 75 frames.

first ICCA audio canonical vector to bandpass filter the original audio stream using the overlap-save method and plot the resulting spectrogram in Figures 4.29(a) and 4.29(b). Unlike empirical CCA, the ICCA audio filter correctly filters the audio to contain the correlated 250 Hz audio tone. Figure 4.29(c) plots the pixels in our scene that are correlated with the audio source and ICCA is able to correctly identify the left block M.

4.8.7 Controlled Audio Audio Experiment

Our final experiment explores the inherent correlation between two audio streams. In this experiment, we generate two 30 second audio sequences. Each sequence contains two pure-tones, which are amplitude modulated (AM) at different frequencies. In addition we add uncorrelated coffee shop noise, which is independent between each audio sequence. One pure-tone in each sequence is AM at a shared rate, inducing correlation between the audio sequences. The remaining pure-tones are AM at different rates, making them independent of the shared AM tones. Our waveforms are

$$\begin{aligned} a_1(t) &= \frac{1}{3}s_1(t) + \frac{1}{3}s_2(t) + \frac{1}{3}n_1(t) \\ a_2(t) &= \frac{1}{3}s_3(t) + \frac{1}{3}s_4(t) + \frac{1}{3}n_2(t) \end{aligned} \tag{4.28}$$

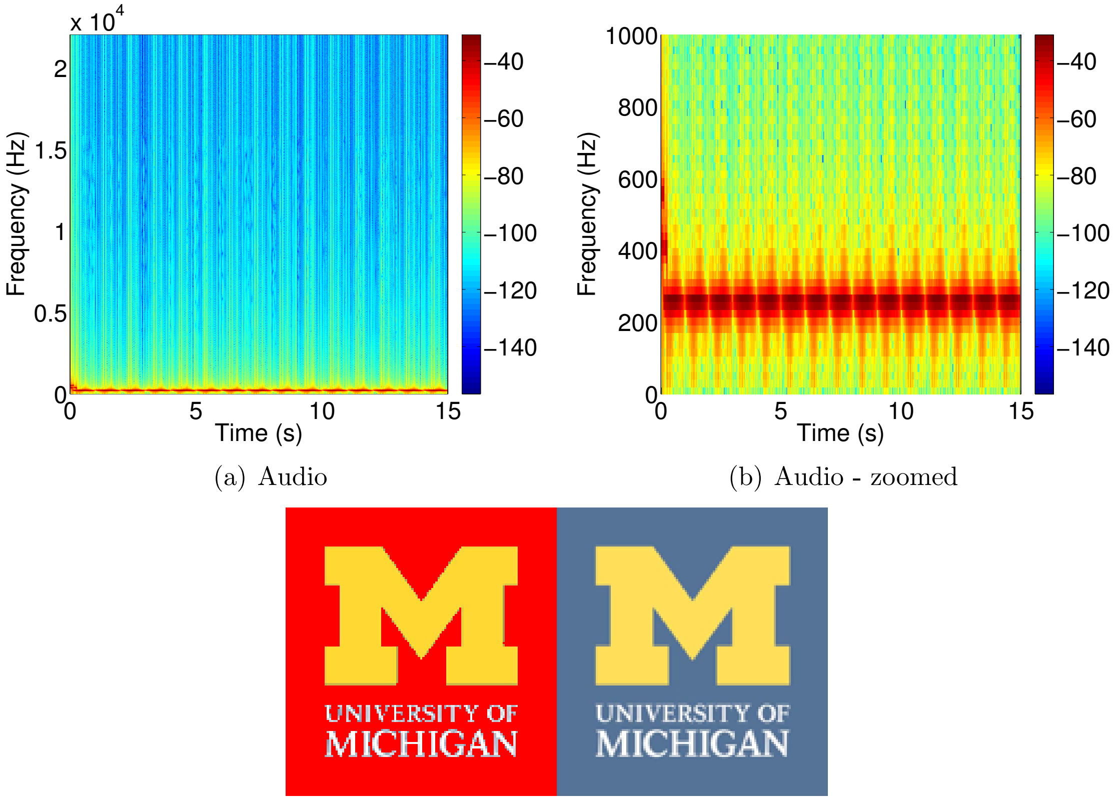


(a) Audio

(b) Audio - zoomed

(c) Video

Figure 4.28: Thresholded canonical vectors (as computed in (4.9)) corresponding to the top singular value returned by empirical CCA in Figure 4.26(a). We use a threshold of $\log(p)/\sqrt{p}$ for $p = 33250$ and $p = 1025$ for video and audio vectors, respectively. (a) Spectrogram of the original audio stream filtered using the thresholded empirical CCA top canonical vector and the overlap-save filter method. (b) Zoomed in spectrogram of (a). (c) Red colored pixels represent the pixels that empirical CCA marks as correlated to the audio stream in (a).



(a) Audio

(b) Audio - zoomed

(c) Video

Figure 4.29: Thresholded canonical vectors (as computed in (4.13)) corresponding to the top singular value returned by ICCA in Figure 4.26(b). We use a threshold of $\log(p)/\sqrt{p}$ for $p = 33250$ and $p = 1025$ for video and audio vectors, respectively. (a) Spectrogram of the original audio stream filtered using the thresholded ICCA top canonical vector and the overlap-save filter method. (b) Zoomed in spectrogram of (a). (c) Red colored pixels represent the pixels that ICCA marks as correlated to the audio stream in (a).

where

$$\begin{aligned}
s_1(t) &= \frac{(1+\sin(2\pi t))}{2}\sin\left(2\pi\left(250t\right)\right)\\
s_2(t) &= \frac{(1+\cos(2\pi(3t))}{2}\sin\left(2\pi\left(400t\right)\right)\\
s_3(t) &= \frac{(1+\sin(2\pi t))}{2}\sin\left(2\pi\left(300t\right)\right)\\
s_4(t) &= \frac{(1+\cos(2\pi(5t))}{2}\sin\left(2\pi\left(550t\right)\right)\\
n_1(t) &= \text{independent coffee shop noise}\\
n_2(t) &= \text{independent coffee shop noise.}
\end{aligned} \tag{4.29}$$

All time sequences are generated with a sample rate of 44.1 kHz. Figure 4.2 plots the spectrogram of each sequence and zooms in on a smaller portion of the spectrum to see the AM sequences. Table 4.30 summarizes each of our signals in each audio sequences.

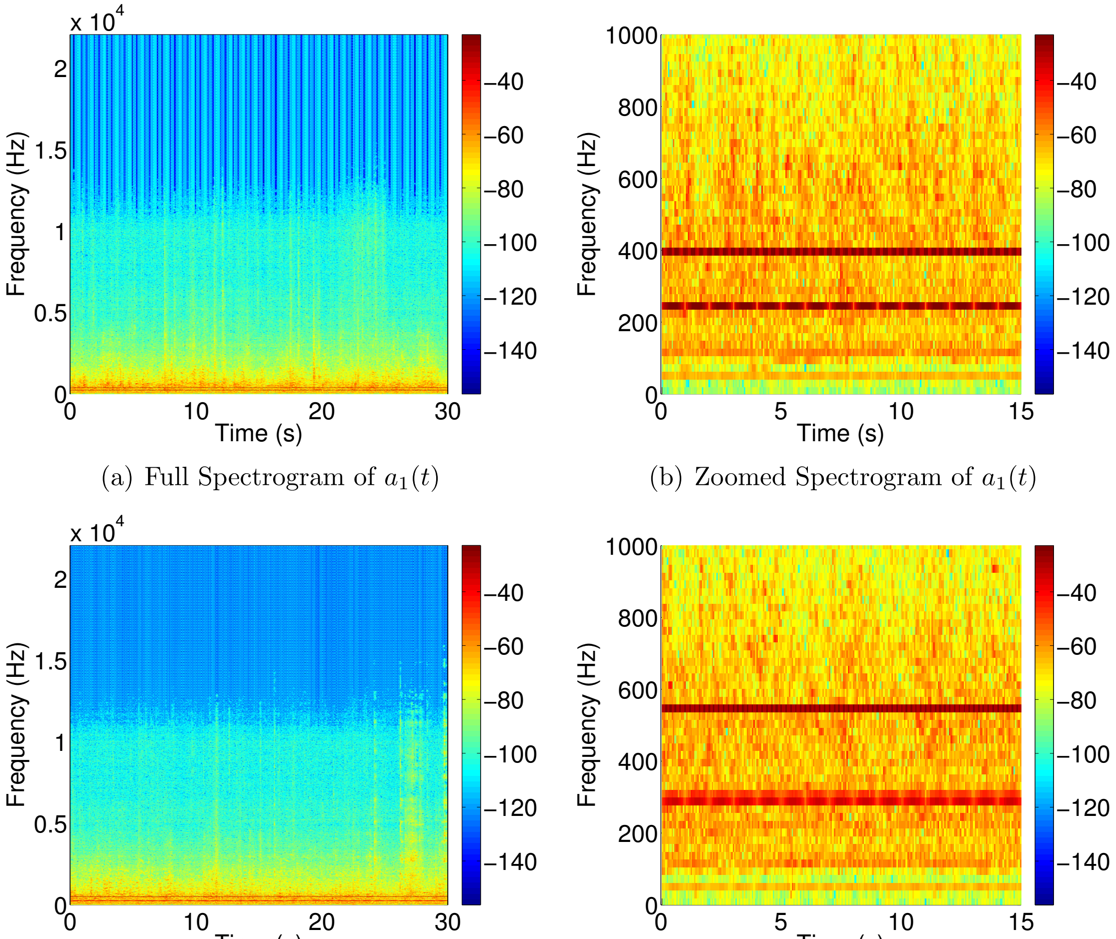
View	Source	AM Frequency
$a_1(t)$	250 Hz pure tone 400 Hz pure tone coffee shop noise 1	1 Hz 3 Hz
$a_2(t)$	300 Hz pure tone 550 Hz pure tone coffee shop noise 2	1 Hz 5 Hz

Table 4.2: Summary of the audio sources in (4.28) and their components in (4.29). The 250 Hz pure tone in $a_1(t)$ is amplitude modulated at the same frequency as the 300 Hz pure tone in $a_2(t)$ and is thus correlated with it.

To post-process the data, we separate the audio streams into equal window sizes of 2940 time points, corresponding to a time interval of 1/15 second. On each window, we run a 4096 point FFT and take the magnitude of the first 2049 points as a feature vector. We then stack the feature vectors for all windows into a matrix and subtract the mean, resulting in 2049 × 450 matrices X_{a_1} and Y_{a_2}.

First, we run PCA on X_{a_1} and Y_{a_2}. Figure 4.31 plots the singular values of each dataset. From our setup, we know that each audio sequence has 2 pure-tone signals plus coffee shop noise. Figure 4.32 plots the corresponding singular vectors for the top 2 singular vectors of each audio dataset.

While the PCA vectors in Figure 4.32 can identify sources within a dataset, by themselves, they do not provide any information about whether the sources are correlated between datasets. To identify such correlations, we run empirical CCA and



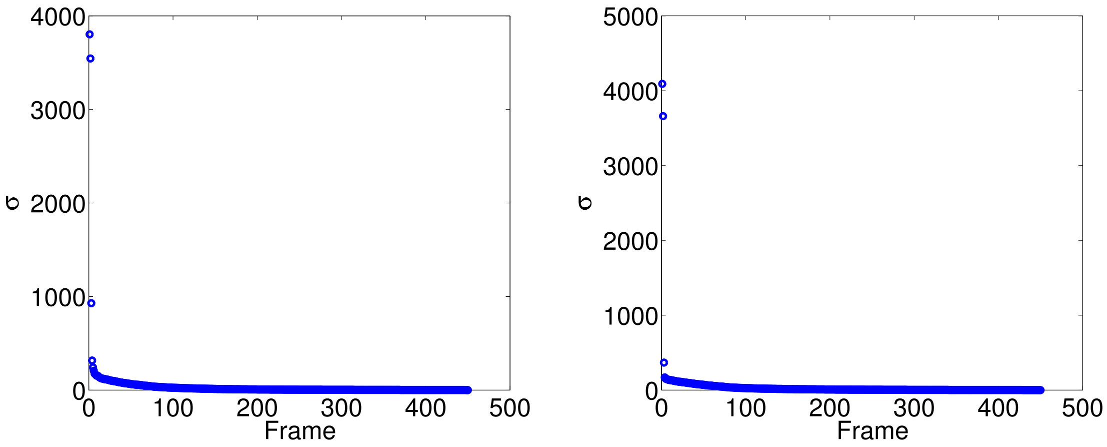
(a) Full Spectrogram of $a_1(t)$

(b) Zoomed Spectrogram of $a_1(t)$

(c) Full Spectrogram of $a_2(t)$

(d) Zoomed Spectrogram of $a_2(t)$

Figure 4.30: (a) Full spectrogram of $a_1(t)$ defined in (4.28). (b) Zoomed in spectrogram of $a_1(t)$ to see the 2 sources at 250 Hz and 400 Hz. (c) Full spectrogram of $a_2(t)$ defined in (4.28) (d) Zoomed in spectrogram of $a_2(t)$ to see the 2 sources at 300 Hz and 550 Hz. The 250 Hz signal in $a_1(t)$ is amplitude modulated at the same frequency as the 300 Hz signal in $a_2(t)$.



(a) Video

(b) Audio

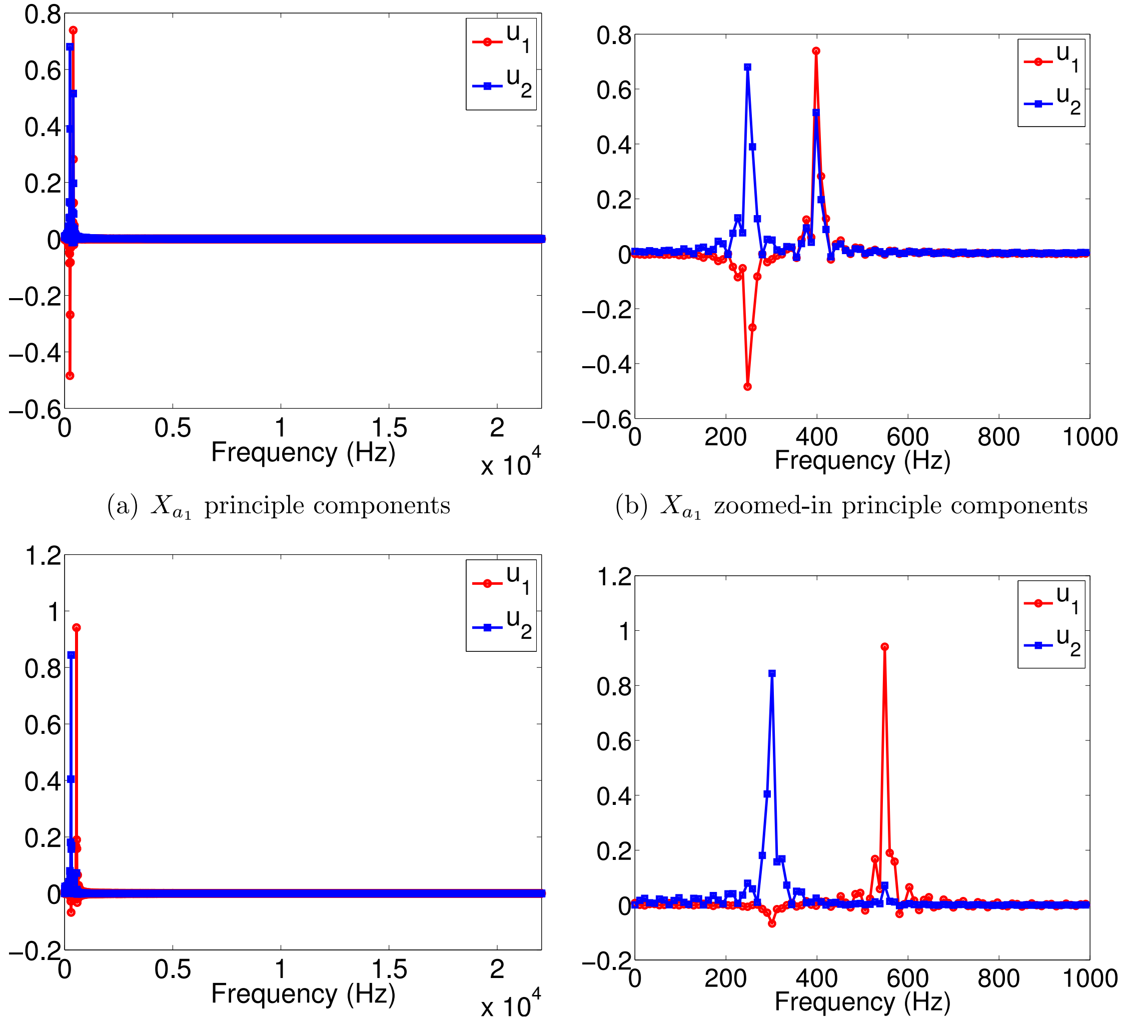
Figure 4.31: Singular value spectra of X_{a_1} and Y_{a_2}.

ICCA after each new video frame. For frame ℓ, we construct the $2049 \times \ell$ submatrix $X_{a_1}^{\ell}$ and $2049 \times \ell$ submatrix $Y_{a_2}^{\ell}$ by taking the matrix of the first ℓ original vectorized frames and zero meaning it. We then use these matrices as the input to empirical CCA and ICCA. Using knowledge of our experiment setup, we set $\widehat{k}_x = \widehat{k}_y = 2$. Figure 4.33 plots the top 5 correlation coefficients returned by empirical CCA and ICCA over the first 450 frames.

Intuitively, empirical CCA returns perfect correlation as we have only a few frames but a large dimension (frequencies). These correlations are insignificant because we operate in the sample deficient regime. However, we may use (4.14) to determine whether the ICCA canonical correlations are significant. Using a significance level of $\alpha = 0.01$, Figure 4.34 plots the binary decision (0 is insignificant, 1 is significant).

Finally, we threshold the audio canonical vectors for empirical CCA and ICCA to create bandpass filters. Similar to the above experiments, we use the threshold $\sqrt{\log(p)/p}$. Using these filters, we filter the original audio streams using the overlap-save method and plot the resulting empirical CCA filtered spectrograms in Figure 4.35 and ICCA filtered spectrograms in Figure 4.36.

From these spectrograms we observe that once again, empirical CCA fails to detect the correlated 1 Hz AM signals in each of the datasets. Examining 4.35, we see that the correlated filter that empirical CCA returns keeps much of the frequency content below 1000 Hz. However, ICCA is able to very quickly detect the presence the correlated signals, while determining that the second canonical vector is insignificant. Thus ICCA detects exactly the number of underlying correlated components.



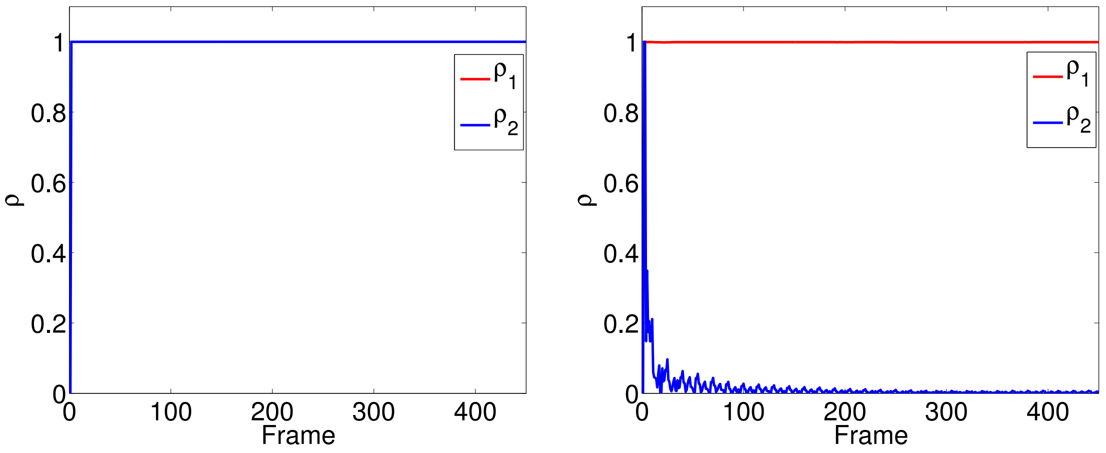
(a) X_{a_1} principle components

(b) X_{a_1} zoomed-in principle components

(c) Y_{a_2} principle components

(d) Y_{a_2} zoomed-in principle components

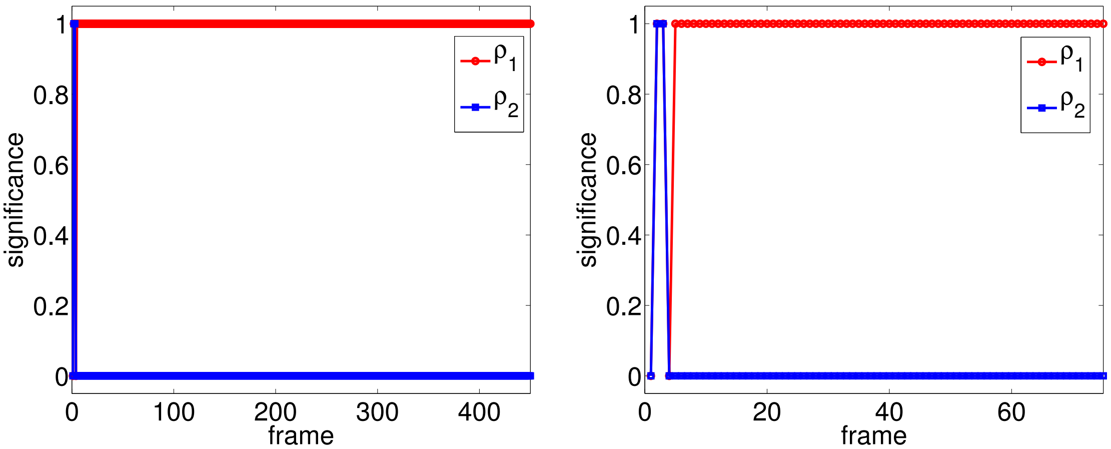
Figure 4.32: (a) Left singular vectors of X_{a_1} corresponding to the top 2 singular values in Figure 4.31(a). (b) Zoomed in version of (a). (c) Left singular vectors of Y_{a_2} corresponding to the top 2 singular values in Figure 4.31(b). (d) Zoomed in version of (c).



(a) Empirical CCA

(b) ICCA

Figure 4.33: (a) Top two singular values returned by empirical CCA as defined in (4.8) for the audio-audio experiment. As we are in the sample deficient regime, these singular values are deterministically 1. (b) Top two singular values returned by ICCA as defined in (4.12). ICCA correctly identifies the one source of correlation present in the audio-audio dataset.

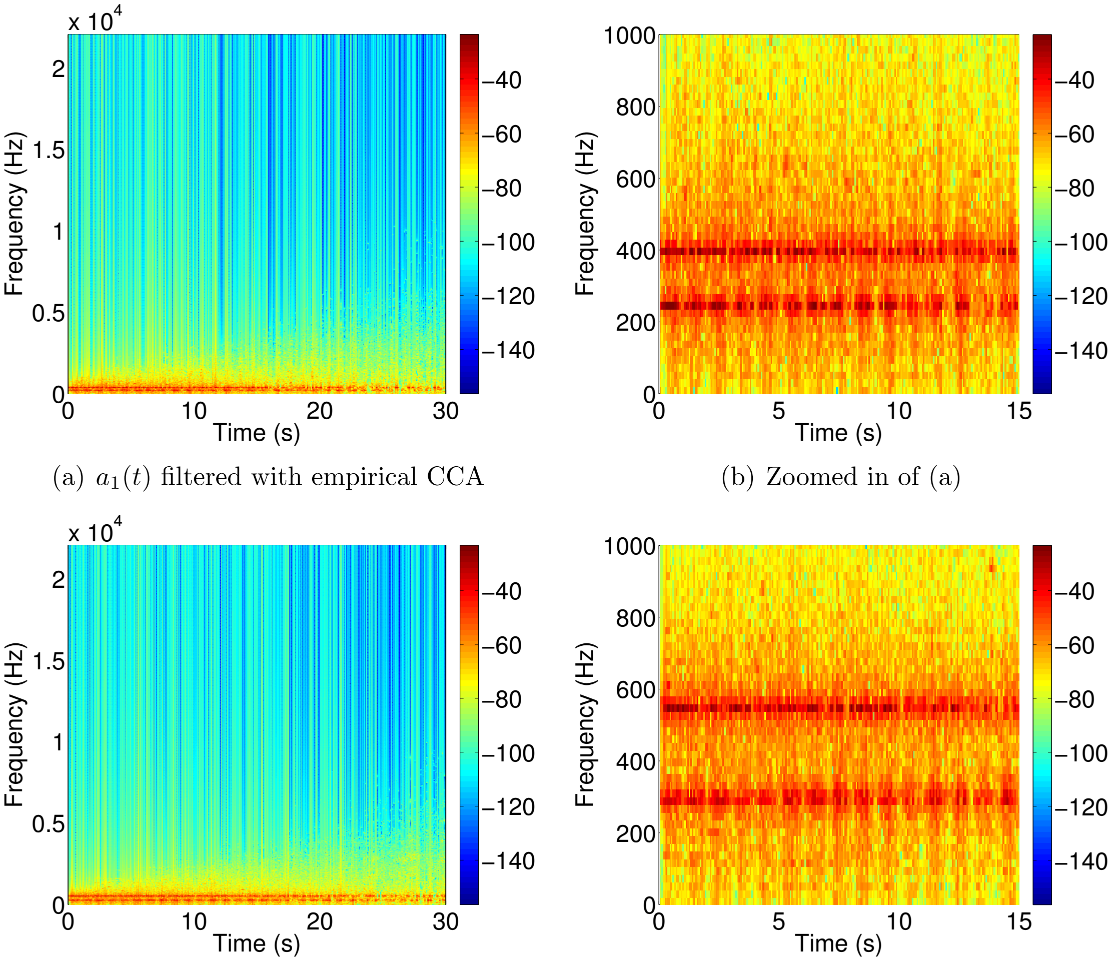


(a) ICCA

(b) ICCA - first 75 frames

Figure 4.34: Significance of the top singular value returned by ICCA in Figure 4.33 using (4.14) with $\alpha = 0.01$. A value of zero represents that the singular value is not significant. A value of one represents that the singular value is significant. (a) Significance for all 450 frames. (b) Zoomed in of (a) to examine the first 75 frames.

Examining Figure 4.36, we see that ICCA correctly filters all but the 250 Hz pure-tone in a_1 and the 300 Hz pure-tone in a_2, which are both amplitude modulated at 1 Hz.



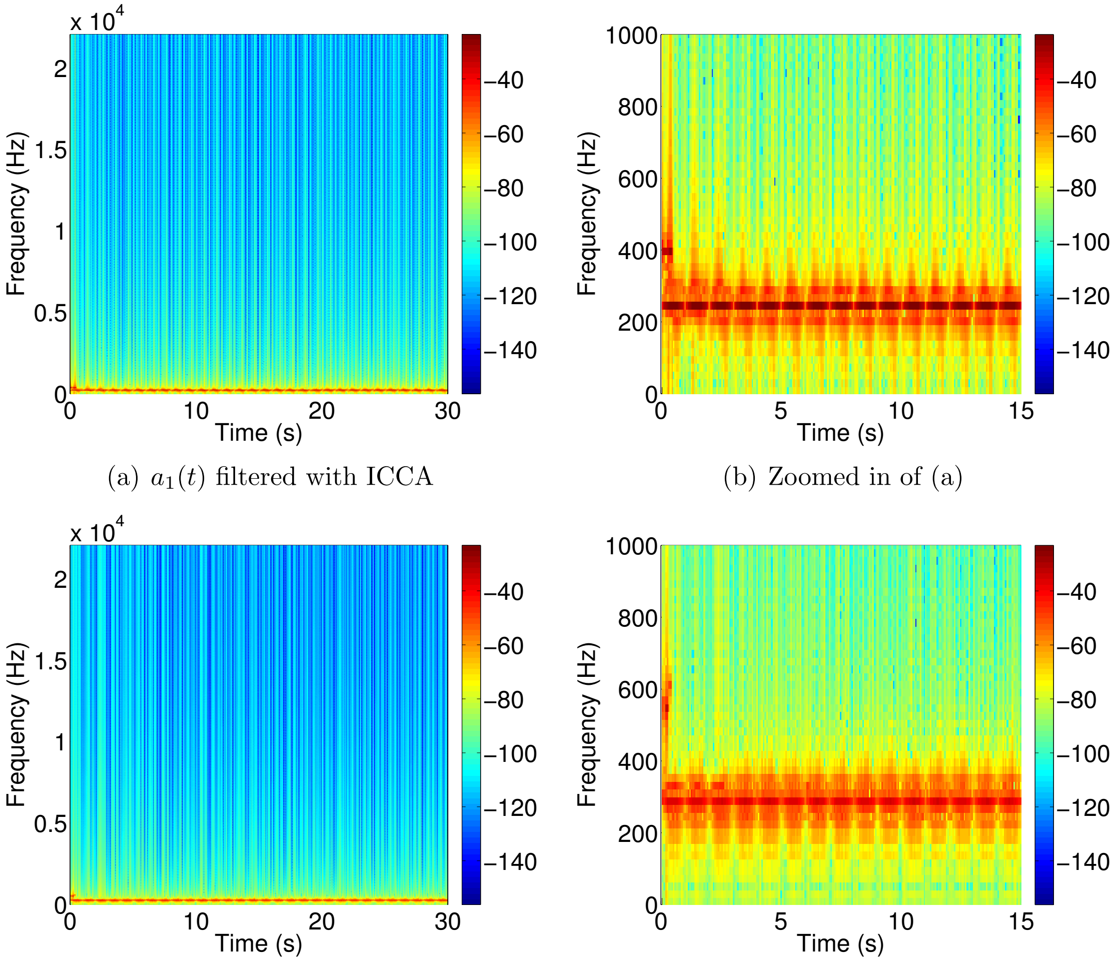
(a) $a_1(t)$ filtered with empirical CCA

(b) Zoomed in of (a)

(c) $a_2(t)$ filtered with empirical CCA

(d) Zoomed in of (c)

Figure 4.35: (a) Full spectrogram of $a_1(t)$ (as defined in (4.28)) filtered using the thresholded top canonical vector of empirical CCA computed via (4.9). We use a threshold of $\log(p)/\sqrt{p}$ for $p = 2049$ and use the overlap-save filter method. (b) Zoomed in spectrogram of (a). (c) Full spectrogram of $a_2(t)$ (as defined in (4.28)) filtered using the thresholded top canonical vector of empirical CCA computed using (4.9). We use a threshold of $\log(p)/\sqrt{p}$ for $p = 2049$ and use the overlap-save filter method. (d) Zoomed in spectrogram of (c). Empirical CCA fails to detect the correlated 250 Hz signal in $a_1(t)$ and the 300 Hz signal in $a_2(t)$. Instead, empirical CCA has random bandpass filters across the spectrum.



(a) $a_1(t)$ filtered with ICCA

(b) Zoomed in of (a)

(c) $a_2(t)$ filtered with ICCA

(d) Zoomed in of (c)

Figure 4.36: (a) Full spectrogram of $a_1(t)$ (as defined in (4.28)) filtered using the thresholded top canonical vector of ICCA computed using (4.13). We use a threshold of $\log(p)/\sqrt{p}$ for $p = 2049$ and use the overlap-save filter method. (b) Zoomed in spectrogram of (a). (c) Full spectrogram of $a_2(t)$ (as defined in (4.28)) filtered using the thresholded top canonical vector of ICCA computed using (4.13). We use a threshold of $\log(p)/\sqrt{p}$ for $p = 2049$ and use the overlap-save filter method. (d) Zoomed in spectrogram of (c). ICCA correctly detects the correlated 250 Hz signal in $a_1(t)$ and the 300 Hz signal in $a_2(t)$ without including any spurious frequencies.

CHAPTER V

On Estimating Population Canonical Vectors

5.1 Introduction

In Chapter IV, we presented statistical tests for empirical CCA and informative CCA. This analysis showed that in the sample deficient regime, the canonical correlations returned by ICCA can reliably detect the presence of correlated signals between two datasets. In this chapter, we complete the analysis of empirical CCA and ICCA by examining the accuracy of the canonical vectors associated with the canonical correlations.

Using the same data model as Chapter IV, we begin by deriving the CCA population canonical vectors if all parameters are known. From this analysis we see that the canonical vectors are a linear combination of signal vectors that form the linear subspace of each dataset. This linear combination involves the eigen-structure of the cross-correlation matrix and the inverse signal-to-noise ratios (SNRs) of the individual correlation matrices. We show that the canonical vectors returned by empirical CCA are very inaccurate in the low-sample, low-SNR regime while the canonical vectors returned by ICCA are able to properly estimate the true population canonical vectors in this regime.

This analysis of the canonical vector accuracy leads to some nice observations. First, we notice that the canonical vector estimation is very sensitive to the estimation of the underlying SNRs since we need to use their inverses. With this observation in mind, we form an asymptotically optimal estimator, which we call ICCA+ , that provides an optimal linear combination of the estimated components of the signal subspaces, where these weights incorporate the accuracy each component of the estimated signal subspace. These weights make contact with the accuracy of subspace components that we used in Chapters II and III to improve matched subspace detection. We finally consider an orthogonal estimate to the canonical vectors and discuss when this estimate is equivalent to ICCA and ICCA+ .

5.1.1 Data Model

We use the same data model in Chapter IV, but repeat it here to facilitate exposition. Let $x_i \in \mathbb{C}^{p\times 1}$ and $y_i \in \mathbb{C}^{q\times 1}$ be modeled as

$$\begin{aligned} x_i &= U_x s_{x,i} + z_{x,i} \\ y_i &= U_y s_{y,i} + z_{y,i}, \end{aligned} \tag{5.1}$$

where $U_x^H U_x = I_{k_x}$, $U_y^H U_y = I_{k_y}$, $z_{x,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_p)$ and $z_{y,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_q)$. Furthermore, assume that

$$\begin{aligned} s_{x,i} &\sim \mathcal{CN}(0, \Theta_x) \\ s_{y,i} &\sim \mathcal{CN}(0, \Theta_y), \end{aligned}$$

where $\Theta_x = \mathbf{diag}\left(\left(\theta_1^{(x)}\right)^2, \ldots, \left(\theta_{k_x}^{(x)}\right)^2\right)$ and $\Theta_y = \mathbf{diag}\left(\left(\theta_1^{(y)}\right)^2, \ldots, \left(\theta_{k_y}^{(y)}\right)^2\right)$. Assume that $z_{x,i}$ and $z_{y,i}$ are mutually independent and independent from both $s_{x,i}$ and $s_{y,i}$. Finally, assume that

$$\mathbb{E}\left[s_{x,i} s_{y,i}^H\right] =: K_{xy} = \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}$$

where the entries of P_{xy} are $-1 \leq \rho_{kj} \leq 1$ and represent the correlation between $s_{x,i}^{(k)}$ and $s_{y,i}^{(j)}$. For reasons to be made clear later, define

$$\widetilde{K}_{xy} = \left(\Theta_x + I_{k_x}\right)^{-1/2} K_{xy} \left(\Theta_y + I_{k_y}\right)^{-1/2}$$

and define the singular values of $\widetilde{K}_{xy}$ as $\kappa_1, \ldots, \kappa_{\min(k_x,k_y)}$. Under this model, we define the following covariance matrices

$$\begin{aligned} \mathbb{E}\left[x_i x_i^H\right] &= U_x \Theta_x U_x^H + I_p =: R_{xx} \\ \mathbb{E}\left[y_i y_i^H\right] &= U_y \Theta_y U_y^H + I_q =: R_{yy} \\ \mathbb{E}\left[x_i y_i^H\right] &= U_x K_{xy} U_y^H =: R_{xy}. \end{aligned} \tag{5.2}$$

Finally, define the random matrices Z_n^x and Z_n^y formed by stacking n realizations of $z_{x,i}$ and $z_{y,i}$ columnwise via

$$\begin{aligned} Z_n^x &= \left[z_{x,1}, \ldots, z_{x,n}\right] \\ Z_n^y &= \left[z_{y,1}, \ldots, z_{y,n}\right]. \end{aligned}$$

Denote the singular values of these matrices as

$$\begin{aligned}\sigma_1(Z_n^x) &\geq \cdots \geq \sigma_p(Z_n^x)\\ \sigma_1(Z_n^y) &\geq \cdots \geq \sigma_q(Z_n^y)\end{aligned}$$

where without loss of generality we let $p < n$ and $q < n$ to simplify the definition of the empirical singular value distribution. Let $\mu_{Z_n^x}$ and $\mu_{Z_n^y}$ be the empirical singular value distribution defined as

$$\begin{aligned}\mu_{Z_n^x} &= \frac{1}{p}\sum_{i=1}^{p}\delta_{\sigma_i(Z_n^x)}\\ \mu_{Z_n^y} &= \frac{1}{q}\sum_{i=1}^{q}\delta_{\sigma_i(Z_n^y)}\end{aligned}$$

.

Assume that the probability measures $\mu_{Z_n^x}$ and $\mu_{Z_n^y}$ converge almost surely as $p, q, n \to \infty$ to non-random compactly supported probability measures μ_{Z_x} and μ_{Z_y} respectively. Finally, we assume that $\sigma_1(Z_n^x) \xrightarrow{\text{a.s.}} b_x$ and $\sigma_1(Z_n^y) \xrightarrow{\text{a.s.}} b_y$.

5.2 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a dimensionality reduction algorithm that finds linear projections for x_i and y_i such that in the projected spaces, the variables are maximally correlated. Specifically, CCA solves the following optimization problem

$$\rho_{\text{cca}} = \underset{w_x, w_y}{\operatorname{argmax}} \frac{w_x^H R_{xy} w_y}{\sqrt{w_x^H R_{xx} w_x}\sqrt{w_y^H R_{yy} w_y}}, \tag{5.3}$$

where w_x and w_y are called canonical vectors and ρ_{cca} is called the canonical correlation coefficient. Notice that we can scale w_x and w_y and still achieve the same objective function. Therefore, we may constrain the canonical variates to have unit norm, resulting in

$$\begin{aligned}\underset{w_x, w_y}{\operatorname{argmax}} \quad & w_x^H R_{xy} w_y\\ \text{subject to} \quad & w_x^H R_{xx} w_x = 1\\ & w_y^H R_{yy} w_y = 1.\end{aligned} \tag{5.4}$$

Substituting the change of variables $\widetilde{w}_x = R_{xx}^{1/2} w_x$ and $\widetilde{w}_y = R_{yy}^{1/2} w_y$ in (5.4) results in the following optimization problem

$$\begin{aligned} \underset{\widetilde{w}_x, \widetilde{w}_y}{\text{argmax}} \quad & \widetilde{w}_x^H R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2} \widetilde{w}_y \\ \text{subject to} \quad & \widetilde{w}_x^H \widetilde{w}_x = 1 \\ & \widetilde{w}_y^H \widetilde{w}_y = 1. \end{aligned} \tag{5.5}$$

Examining the optimization problem in (5.5), we can immediately see that the solution to CCA may be solved via the SVD of the matrix

$$C_{\text{cca}} = R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2}. \tag{5.6}$$

Define $C_{\text{cca}} = FKG^T$ as the SVD of C_{cca} where F is an unitary $p \times p$ matrix with columns $f_1, \ldots, f_p$, G is a unitary $q \times q$ matrix with columns $g_1, \ldots, g_q$, and $K = \textbf{diag}(k_1, \ldots, k_{\min(p,q)})$ is a $p \times q$ matrix whose diagonal elements are the singular values of C_{cca}. Therefore, the solution to (5.5) is

$$\begin{aligned} \widetilde{w}_x &= f_1 \\ \widetilde{w}_y &= g_1 \\ \rho &= k_1. \end{aligned}$$

We can obtain higher order canonical correlations and vectors by taking successive singular value and vector pairs. Thus our canonical correlations are simply the singular values of C_{cca} and the canonical vectors are transformations of the singular vectors of C_{cca}

$$w_x = R_{xx}^{-1/2} \widetilde{w}_x \quad w_y = R_{yy}^{-1/2} \widetilde{w}_y. \tag{5.7}$$

5.3 Empirical CCA

In many applications, we do not know the covariance matrices R_{xx}, R_{yy}, and R_{xy} *a priori*. Therefore, we cannot know the true canonical vectors in (5.7) and must estimate them from training data. Typically, we are given multiple snapshots that we stack columnwise to form the data matrices

$$\begin{aligned} X &= [x_1, \ldots, x_n] \\ Y &= [y_1, \ldots, y_n], \end{aligned}$$

where for $i = 1, \ldots, n$, x_i and y_i are modeled in (5.1). Defining $Z_x = [z_{x,1}, \ldots, z_{x,n}]$, $Z_y = [z_{y,1}, \ldots, z_{y,n}]$, $V_x = [s_{x,1}, \ldots, s_{x,n}]^H$, and $V_y = [s_{y,1}, \ldots, s_{y,n}]$, we may write

$$\begin{aligned} X &= U_x V_x^H + Z_x \\ Y &= U_y V_y^H + Z_y. \end{aligned}$$

Given these data matrices, we form estimates of our unknown covariance matrices via

$$\begin{aligned} \widehat{R}_{xx} &= \frac{1}{n} X X^H \\ \widehat{R}_{yy} &= \frac{1}{n} Y Y^H \\ \widehat{R}_{xy} &= \frac{1}{n} X Y^H. \end{aligned}$$

Define the data SVDs

$$\begin{aligned} X &= \widehat{U}_x \widehat{\Sigma}_x \widehat{V}_y^H \\ Y &= \widehat{U}_y \widehat{\Sigma}_y \widehat{V}_y^H \end{aligned}$$

and trimmed matrices

$$\begin{aligned} \widetilde{U}_x &= \widehat{U}_x \left(:, 1 : \min(p, n)\right) \\ \widetilde{\Sigma}_x &= \widetilde{\Sigma}_x \left(1 : \min(p, n), 1 : \min(p, n)\right) \\ \widetilde{V}_x &= \widehat{V}_x \left(:, 1 : \min(p, n)\right) \\ \widetilde{U}_y &= \widehat{U}_y \left(:, 1 : \min(q, n)\right) \\ \widetilde{\Sigma}_y &= \widetilde{\Sigma}_y \left(1 : \min(q, n), 1 : \min(q, n)\right) \\ \widetilde{V}_y &= \widehat{V}_y \left(:, 1 : \min(q, n)\right). \end{aligned}$$

Given these definitions, substituting the sample covariance estimates into C_{cca} yields (see [8])

$$\widehat{C}_{\text{cca}} = \widetilde{U}_x \widetilde{V}_x^H \widetilde{V}_y \widetilde{U}_y^H.$$

The singular values of this matrix are exactly the canonical correlation estimates, $\widehat{\rho}_{\text{cca}}$, returned by CCA. Empirical CCA can return up to $\min(p, q)$ canonical correlations. However, X and Y have k_x and k_y underlying signals, respectively, based on the model in (5.1). As k_x and k_y are unknown, let $\widehat{k}_x$ and $\widehat{k}_y$ be estimates of the number of underlying signals in each dataset. It is common to return only $\min(\widehat{k}_x, \widehat{k}_y)$ canonical correlations.

However, we showed in Chapter IV that empirical CCA fails in the sample deficient regime. When $n < p + q$, the top estimated canonical correlation is deterministically one. Instead we showed that ICCA [8], an algorithm that first trims the singular vectors of the individual datasets to only include *informative* singular vectors, can reliably detect correlations in the sample starved regime. Define the trimmed data SVDs

$$\begin{aligned}
\mathring{U}_x &= \widehat{U}_x\left(:, 1:\widehat{k}_x\right)\\
\mathring{\Sigma}_x &= \widehat{\Sigma}_x\left(1:\widehat{k}_x, 1:\widehat{k}_x\right)\\
\mathring{V}_x &= \widehat{V}_x\left(:, 1:\widehat{k}_x\right)\\
\mathring{U}_y &= \widehat{U}_y\left(:, 1:\widehat{k}_y\right)\\
\mathring{\Sigma}_y &= \widehat{\Sigma}_y\left(1:\widehat{k}_y, 1:\widehat{k}_y\right)\\
\mathring{V}_y &= \widehat{V}_y\left(:, 1:\widehat{k}_y\right).
\end{aligned}$$

Given these definitions, we define the ICCA matrix

$$\widehat{C}_{\text{icca}} = \mathring{U}_x\mathring{V}_x^H\mathring{V}_y\mathring{U}_y^H.$$

Similar to CCA, the SVD of this matrix gives the ICCA canonical correlations and vectors. Define $\widehat{C}_{\text{icca}} = FKG^T$ as the SVD of $\widehat{C}_{\text{icca}}$ where F is an unitary $p \times p$ matrix with columns $f_1, \dots, f_p$, G is a unitary $q \times q$ matrix with columns $g_1, \dots, g_q$, and $K = \mathbf{diag}(k_1, \dots, k_{\min(p,q)})$ is a $p \times q$ matrix whose diagonal elements are the singular values of $\widehat{C}_{\text{icca}}$. We note that by construction, there will be at most $\min(\widehat{k}_x, \widehat{k}_y)$ non-zero singular values of $\widehat{C}_{\text{icca}}$. Then ICCA canonical correlation and vector pairs are

$$\begin{aligned}
\widehat{\rho}_{\text{icca}} &= k_1\\
\widehat{w}_x^{\text{icca}} &= \widehat{R}_{xx}^{-1/2}f_1\\
\widehat{w}_y^{\text{icca}} &= \widehat{R}_{yy}^{-1/2}g_1.
\end{aligned}$$

Again, successive canonical correlation and vectors are found via successive singular value and vectors pairs from $\widehat{C}_{\text{icca}}$.

5.4 Estimating Population Canonical Vectors

In this section, we derive the population canonical vectors of CCA assuming known parameters. We observe that these population canonical vectors are a linear combination of the signal vectors U_x and U_y of the individual datasets. This linear combination is dependent on the individual SNRs Θ_x and Θ_y and the eigen-structure of the previously alluded to matrix $\widetilde{K}_{xy}$. We then show that ICCA is equivalent to substituting plug-in estimates for unknown quantities in these estimates. Finally, we provide the definition for a new asymptotically optimal estimate, which we cal ICCA+, of these vectors that uses the accuracy of the estimated subspaces U_x and U_y.

5.4.1 Population canonical vectors

We first determine the population canonical vectors of our data model in (5.1). To do so, we need the singular vectors of C_{cca}.

$$\begin{aligned}
C_{\text{cca}} &= R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2} \\
&= \left(U_x \Theta_x U_x^H + I_{k_x}\right)^{-1/2} U_x K_{xy} U_y^H \left(U_x \Theta_y U_y^H + I_{k_y}\right)^{-1/2} \\
&= U_x \left(\Theta_x + I_{k_x}\right)^{-1/2} K_{xy} \left(\Theta_y + I_{k_y}\right)^{-1/2} U_y^H \\
&= U_x \widetilde{K}_{xy} U_y^H .
\end{aligned}$$

Define $U_{\widetilde{K}} K_{\widetilde{K}} V_{\widetilde{K}}^H$ as the SVD of $\widetilde{K}_{xy}$. First note that from this observation the rank of C_{cca} is $k =: \min(k_x, k_y)$. Recall that $w_x = R_{xx}^{-1/2} \widetilde{w}_x$ and $w_y = R_{yy}^{-1/2} \widetilde{w}_y$. Therefore if we define the matrices of the canonical vectors $W_x = [w_x^{(1)}, \dots, w_x^{(k)}]$ and $W_y = [w_y^{(1)}, \dots, w_y^{(k)}]$, we have that

$$\begin{aligned}
W_x &= U_x \left(\Theta_x + I_{k_x}\right)^{-1/2} U_{\widetilde{K}} \\
W_y &= U_y \left(\Theta_y + I_{k_y}\right)^{-1/2} V_{\widetilde{K}} .
\end{aligned} \tag{5.8}$$

Therefore, we see that the individual canonical vectors $w_x^{(i)}$ and $w_y^{(i)}$ are linear combinations of U_x and U_y dependent on Θ_x, Θ_y, $U_{\widetilde{K}}$, and $V_{\widetilde{K}}$.

5.4.2 Empirical CCA canonical vector estimates

We may use empirical CCA to estimate the population CCA canonical vectors. This requires taking the SVD of $\widehat{C}_{\text{cca}}$. Notice that inner matrix product of this matrix is $\widetilde{V}_x^H \widetilde{V}_y$. Define the SVD of this $\min(p, n) \times \min(q, n)$ matrix as $\widetilde{U}_{\widetilde{K}} \widetilde{K}_{\widetilde{K}} \widetilde{V}_{\widetilde{K}}^H$. Then the empirical CCA canonical vector estimates are

$$\begin{aligned}
\widehat{w}_{x,i}^{\text{cca}} &= \widehat{R}_{xx}^{-1/2} \widetilde{w}_x \\
&= \left(\widetilde{U}_x \widetilde{\Sigma}_x^{-1} \widetilde{U}_x^H\right) \left(\widetilde{U}_x \widetilde{U}_{\widetilde{K}}(:, i)\right) \\
&= \widetilde{U}_x \widetilde{\Sigma}_x^{-1} \widetilde{U}_{\widetilde{K}}(:, i)
\end{aligned}$$

Stacking these empirical CCA canonical vectors estimates in a matrix yields

$$\begin{aligned}\widehat{W}_x^{\text{cca}} &= \widetilde{U}_x \left(\widetilde{\Sigma}_x\right)^{-1} \widetilde{U}_{\widetilde{K}} \\ \widehat{W}_y^{\text{cca}} &= \widetilde{U}_y \left(\widetilde{\Sigma}_y\right)^{-1} \widetilde{V}_{\widetilde{K}}.\end{aligned} \tag{5.9}$$

We can immediately expect that empirical CCA will do a very poor job at estimating the canonical vectors because it uses the entire left singular vectors $\widetilde{U}_x$ and $\widetilde{U}_y$ of each data matrix. CCA assumes that the SVD of $\widetilde{V}_x^H\widetilde{V}_y$ is very accurate. However, when we have high dimensions and low samples, this matrix is incredibly inaccurate, as we will see. We note here that the singular values of the individual data matrices may be used to estimate the SNRs via $\widehat{\Theta}_x = \widetilde{\Sigma}_x^2 - I$ and $\widehat{\Theta}_y = \widetilde{\Sigma}_y^2 - I$. In empirical CCA, the above canonical vector estimates use the full data SVDs, which assumes that the rank of underlying signals are $\min(p, n)$ and $\min(q, n)$. This is obviously quite incorrect.

5.4.3 ICCA canonical vectors

In (5.8), we do not know U_x, U_y, Θ_x, Θ_y, $U_{\widetilde{K}}$, or $V_{\widetilde{K}}$. Therefore, in the spirit of many algorithms, we may plug-in estimates of all of these parameters. From the above sections we obtain the estimates $\widehat{U}_x$,$\widehat{U}_y$, $\widehat{\Theta}_x$, and $\widehat{\Theta}_y$ from the individual data SVDs of X and Y. We obtain estimates $\widehat{U}_{\widetilde{K}}$ and $\widehat{V}_{\widetilde{K}}$ from the left and right singular vectors of $\mathring{V}_x^H\mathring{V}_y$. Then our plug-in estimate of the canonical vectors is

$$\begin{aligned}\widehat{W}_x^{\text{icca}} &= \mathring{U}_x \left(\widehat{\Theta}_x + I_{\widehat{k}_x}\right)^{-1/2} \widehat{U}_{\widetilde{K}} \\ \widehat{W}_y^{\text{icca}} &= \mathring{U}_y \left(\widehat{\Theta}_y + I_{\widehat{k}_y}\right)^{-1/2} \widehat{V}_{\widetilde{K}}.\end{aligned} \tag{5.10}$$

These plug-in estimates are exactly the ICCA canonical vector estimates. Recall that the key matrix in ICCA, $\widehat{C}_{\text{icca}}$, has an inner matrix product of exactly $\mathring{V}_x^H\mathring{V}_y$. The process of trimming the individual singular vectors causes $\widehat{C}_{\text{icca}}$ to be rank $\min(\widehat{k}_x, \widehat{k}_y)$, which in turn causes the ICCA canonical vector estimates to correctly take only a linear combination of the top $\widehat{k}_x$ and $\widehat{k}_y$ signal vectors.

5.4.4 ICCA+

We expect the estimates in (5.10) to greatly outperform the estimates in (5.9) for reasons mentioned above. However, we still expect the estimates in (5.10) to be sub-optimal because they substitute parameter estimates without considering their accuracy. To consider an improved estimate, we first assume that $\widehat{U}_{\widetilde{K}}$ and $\widehat{V}_{\widetilde{K}}$ are consistent estimators of the true $U_{\widetilde{K}}$ and $V_{\widetilde{K}}$, respectively.

The population, empirical CCA, and ICCA canonical vector estimates all take a linear combination of the known or unknown signal subspace. With this observation, we consider the following canonical vector estimates

$$\begin{aligned} \widetilde{W}_x^{\text{icca+}} &= \mathring{U}_x \Lambda_x^{\text{opt}} \widehat{U}_{\widetilde{K}} \\ \widetilde{W}_y^{\text{icca+}} &= \mathring{U}_y \Lambda_y^{\text{opt}} \widehat{V}_{\widetilde{K}}, \end{aligned} \tag{5.11}$$

where $\Lambda_x^{\text{opt}} = \mathbf{diag}(\lambda_x^{\text{opt}})$ and $\Lambda_y^{\text{opt}} = \mathbf{diag}(\lambda_y^{\text{opt}})$ such that $\lambda_x^{\text{opt}} = \left[\lambda_x^{(1)}, \ldots, \lambda_x^{(k_x)}\right]$ and $\lambda_y^{\text{opt}} = \left[\lambda_y^{(1)}, \ldots, \lambda_y^{(k_y)}\right]$ and are the solutions to the following optimization problems

$$\begin{aligned} \lambda_x^{\text{opt}} &= \underset{\lambda_x}{\text{argmin}} \left\| W_x - \widehat{U}_x \, \mathbf{diag}(\lambda_x) \widehat{U}_{\widetilde{K}} \right\|_F \\ \lambda_y^{\text{opt}} &= \underset{\lambda_y}{\text{argmin}} \left\| W_y - \widehat{U}_y \, \mathbf{diag}(\lambda_x) \widehat{V}_{\widetilde{K}} \right\|_F . \end{aligned} \tag{5.12}$$

This matrix approximation is similar to [117], which examines the optimal approximation to a signal matrix from noisy observations. Nadakuditi shows that the classical Eckart-Young-Mirsky (EYM) low-rank matrix approximation is suboptimal when trying to estimate a low-rank signal matrix from a low-rank signal-plus-noise matrix. The EYM approximation is the optimal low-rank approximation of the low-rank signal-plus-noise matrix but *not* the low-rank signal matrix. Similarly here, the ICCA estimates find the best representation of noisy canonical vectors and not the true underlying canonical vectors. Instead we want the optimal estimates of the population canonical vectors.

5.5 Main Results

In this section we state our main results in the form of theorems and corollaries. We prove all these results in Sections 5.9 and 5.10. We begin by providing the asymptotic limit of the optimal weights to use in the ICCA+ canonical vector estimates. The general form of these weights is independent of the data model and so we provided closed form expressions of these weights when using data modeled in (5.1). In the general case, we provide an algorithm to compute the optimal weights using the spectrum of our individual data matrices. We then define a notion of vector accuracy and provide results for the accuracy of the different estimates proposed herein. Finally, we provide the closed form expressions for the optimal weights when our data modeled in (5.1) also contains missing data.

Theorem 5.5.1. *The solutions to (5.12) are given by*

$$\begin{aligned} \lambda_x^{opt} &= \mathbf{diag}\left(\overset{\circ}{U}_x^H U_x \left(\Theta_x + I_{k_x}\right)^{-1/2}\right) \\ \lambda_y^{opt} &= \mathbf{diag}\left(\overset{\circ}{U}_y^H U_y \left(\Theta_y + I_{k_y}\right)^{-1/2}\right). \end{aligned} \tag{5.13}$$

The proof of this is very straightforward. The key observation is that the optimal weights are dependent on the matrix products $\overset{\circ}{U}_x^H U_x$ and $\overset{\circ}{U}_y^H U_y$. We made contact with these weights in Chapters II and III. The diagonal elements of these matrices are the accuracies of the estimated components of our signal subspaces. It makes sense then that the optimal weights tell us to place less weight on inaccurately estimated signal subspaces. Next we provide the asymptotic limit of these weights, which relies on the asymptotic limit of entries of the matrices $\overset{\circ}{U}_x^H U_x$ and $\overset{\circ}{U}_y^H U_y$. This theorem is for a general noise distribution and does not assume that the noise is Gaussian.

Theorem 5.5.2. *For the data model in (5.1) without the Gaussian noise assumption, the solution in (5.13) exhibits the following behavior in the asymptotic regime where $p, q, n \to \infty$ with $p/n \to c_x$ and $q/n \to c_y$.*

a) For $i = 1, \ldots, k_x$,

$$\lambda_{x,opt}^{(i)} \xrightarrow{a.s.} \begin{cases} D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)\sqrt{\frac{-2\varphi_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)}{D'_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)\left(1+D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)\right)}} & \text{if } \left(\theta_i^{(x)}\right)^2 > 1/D_{\mu_{Z_x}}(b_x^+) \\ 0 & \text{otherwise} \end{cases}$$

and for $i = 1, \ldots, k_y$,

$$\lambda_{y,opt}^{(i)} \xrightarrow{a.s.} \begin{cases} D_{\mu_{Z_y}}\left(\sigma_y^{(i)}\right)\sqrt{\frac{-2\varphi_{\mu_{Z_y}}\left(\sigma_y^{(i)}\right)}{D'_{\mu_{Z_y}}\left(\sigma_y^{(i)}\right)\left(1+D_{\mu_{Z_y}}\left(\sigma_y^{(i)}\right)\right)}} & \text{if } \left(\theta_i^{(y)}\right)^2 > 1/D_{\mu_{Z_y}}(b_y^+) \\ 0 & \text{otherwise} \end{cases}$$

where $\sigma_x^{(i)} = D_{\mu_{Z_x}}^{-1}\left(1/\left(\theta_i^{(x)}\right)^2\right)$, $\sigma_y^{(i)} = D_{\mu_{Z_y}}^{-1}\left(1/\left(\theta_i^{(y)}\right)^2\right)$ and

$$\begin{aligned} D_{\mu_{Z_x}}(z) &=: \left[\int \frac{z}{z^2 - t^2} d_{\mu_{Z_x}}(t)\right] \times \left[c_x \int \frac{z}{z^2 - t^2} d_{\mu_{Z_x}}(t) + \frac{1 - c_x}{z}\right] \quad \text{for } z \notin \text{supp } \mu_{Z_x} \\ D_{\mu_{Z_y}}(z) &=: \left[\int \frac{z}{z^2 - t^2} d_{\mu_{Z_y}}(t)\right] \times \left[c_y \int \frac{z}{z^2 - t^2} d_{\mu_{Z_y}}(t) + \frac{1 - c_y}{z}\right] \quad \text{for } z \notin \text{supp } \mu_{Z_y} \end{aligned}$$

b) The weights used by the ICCA canonical vector estimates exhibit the following behavior

$$\lambda_{x,icca}^{(i)} = \frac{1}{\sqrt{\left(\widehat{\theta}_i^{(x)}\right)^2 + 1}} \xrightarrow{a.s.} \begin{cases} \frac{1}{D_{\mu_{Z_x}}^{-1}\left(1/\left(\theta_i^{(x)}\right)^2\right)} & \textit{if } \left(\theta_i^{(x)}\right)^2 > 1/D_{\mu_{Z_x}}(b_X^+) \\ \frac{1}{\sqrt{b_X^2+1}} & \textit{otherwise} \end{cases}$$

and

$$\lambda_{y,icca}^{(i)} = \frac{1}{\sqrt{\left(\widehat{\theta}_i^{(y)}\right)^2 + 1}} \xrightarrow{a.s.} \begin{cases} \frac{1}{D_{\mu_{Z_y}}^{-1}\left(1/\left(\theta_i^{(y)}\right)^2\right)} & \textit{if } \left(\theta_i^{(y)}\right)^2 > 1/D_{\mu_{Z_y}}(b_Y^+) \\ \frac{1}{\sqrt{b_Y^2+1}} & \textit{otherwise} \end{cases}$$

This theorem highlights some key similarities between the ICCA and optimal weights used in ICCA+ . First, both sets of weights exhibit a phase transition. When the corresponding SNR for a subspace component is below the critical, the weights are constant. When the SNR is below this phase transition, the corresponding subspace component is *uninformative*. Below this phase transition, the ICCA weights are a non-zero constant, however, the optimal weights are zero. We expect the optimal weights to perform better in this uninformative regime since they place no weight on estimated subspaces that are simply noise.

Theorem 5.5.1 motivates Algorithm 5.1 to compute the ICCA+ canonical vectors estimates given two data matrices. These data matrices are assumed to be noisy observation of low-rank signals, but we place no model on the noise. To estimate the D transform and its derivative we follow [117]. For a matrix$p \times n$ matrix X, define

$$\widehat{D}\left(z, X\right) =: \frac{1}{p}\mathbf{tr}\left(z\left(z^2 I_p - XX^H\right)^{-1}\right) \cdot \frac{1}{n}\mathbf{tr}\left(z\left(z^2 I_n - X^H X\right)^{-1}\right) \tag{5.14}$$

and

$$\begin{aligned} \widehat{D}'(z;X) =: &\frac{1}{p}\mathbf{tr}\left(z\left(z^2 I_p - XX^H\right)^{-1}\right) \cdot \frac{1}{m}\mathbf{tr}\left(-2z^2\left(z^2 I_m - X^H\right)^{-2} + \left(z^2 I_n - X^H X\right)^{-1}\right) + \\ &\frac{1}{m}\mathbf{tr}\left(z\left(z^2 I_m - X^H X\right)^{-1}\right) \cdot \frac{1}{n}\mathbf{tr}\left(-2z^2\left(z^2 I_p - XX^H\right)^{-2} + \left(z^2 I_p - XX^H\right)^{-1}\right). \end{aligned} \tag{5.15}$$

Next, we characterize the limiting behavior of the weights when using Gaussian noise.

```
Input: Zero-meaned Dataset 1: $X = p \times n$ matrix
Input: Zero-meaned Dataset 2: $Y = q \times n$ matrix
Input: Rank estimates $\widehat{k}_x$,$\widehat{k}_y$
Compute individual data SVDs $X = \widehat{U}_x \widehat{\Sigma}_x \widehat{V}_x^H$, $Y = \widehat{U}_y \widehat{\Sigma}_y \widehat{V}_y^H$
Compute $\widehat{\Sigma}_x^{\widehat{k}_x} = \mathbf{diag}(\widehat{\sigma}_{\widehat{k}_x+1}, \cdots, \widehat{\sigma})$
Compute $\widehat{\Sigma}_x^{\widehat{k}_x} = \mathbf{diag}(\widehat{\sigma}_{\widehat{k}_x+1}, \cdots, \widehat{\sigma})$
for $i = 1, \ldots, \widehat{k}_x$ do
    Compute $\widehat{D}(\widehat{\sigma}_i^{(x)}, \widehat{\Sigma}_x^{\widehat{k}_x})$ using (5.14) and $\widehat{D}'(\widehat{\sigma}_i^{(x)}, \widehat{\Sigma}_x^{\widehat{k}_x})$ using (5.15)
    Compute $\lambda_{x,\text{opt}}^{(i)}$ using Theorem 5.5.2
for $i = 1, \ldots, \widehat{k}_y$ do
    Compute $\widehat{D}(\widehat{\sigma}_i^{(y)}, \widehat{\Sigma}_x^{\widehat{k}_y})$ using (5.14) and $\widehat{D}'(\widehat{\sigma}_i^{(y)}, \widehat{\Sigma}_x^{\widehat{k}_y})$ using (5.15)
    Compute $\lambda_{y,\text{opt}}^{(i)}$ using Theorem 5.5.2
Compute $\widehat{U}_{\widetilde{K}}$ and $\widehat{V}_{\widetilde{K}}$ from the SVD of $\mathring{V}_x^H \mathring{V}_y$
Compute $\widehat{w}_{x,\text{icca+}}^{(i)} = \mathring{U}_x \, \mathbf{diag}(\lambda_{x,\text{opt}}^{(1)}, \ldots, \lambda_{x,\text{opt}}^{(\widehat{k}_x)}) \widehat{U}_{\widetilde{K}}$
Compute $\widehat{w}_{y,\text{icca+}}^{(i)} = \mathring{U}_y \, \mathbf{diag}(\lambda_{y,\text{opt}}^{(1)}, \ldots, \lambda_{y,\text{opt}}^{(\widehat{k}_y)}) \widehat{V}_{\widetilde{K}}$
Output: $\widehat{W}_x^{\text{icca+}} = \left[\widehat{w}_{x,\text{icca+}}^{(1)}, \ldots, \widehat{w}_{x,\text{icca+}}^{(\widehat{k}_x)}\right]$
Output: $\widehat{W}_y^{\text{icca+}} = \left[\widehat{w}_{y,\text{icca+}}^{(1)}, \ldots, \widehat{w}_{y,\text{icca+}}^{(\widehat{k}_y)}\right]$
```

Figure 5.1: Algorithm to compute the ICCA+ canonical vectors.

Corollary 5.5.1. *In the data model of (5.1), we have that*

$$\lambda_{x,opt}^{(i)} \xrightarrow{a.s.} \begin{cases} \sqrt{\frac{\left(\theta_i^{(x)}\right)^4 - c_x}{\left(\theta_i^{(x)}\right)^2\left(\left(\theta_i^{(x)}\right)^2 + c_x\right)\left(\left(\theta_i^{(x)}\right)^2 + 1\right)}} & \text{if } \left(\theta_i^{(x)}\right)^2 > c_x^{1/2} \\ 0 & \text{otherwise} \end{cases}$$

and for $i = 1, \ldots, k_y$,

$$\lambda_{y,opt}^{(i)} \xrightarrow{a.s.} \begin{cases} \sqrt{\frac{\left(\theta_i^{(y)}\right)^4 - c_y}{\left(\theta_i^{(y)}\right)^2\left(\left(\theta_i^{(y)}\right)^2 + c_y\right)\left(\left(\theta_i^{(y)}\right)^2 + 1\right)}} & \text{if } \left(\theta_i^{(y)}\right)^2 > c_y^{1/2} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\lambda_{x,icca}^{(i)} \xrightarrow{a.s.} \begin{cases} \frac{\theta_i^{(x)}}{\sqrt{\left(1+\left(\theta_i^{(x)}\right)^2\right)\left(c_x + \left(\theta_i^{(x)}\right)^2\right)}} & \text{if } \left(\theta_i^{(x)}\right)^2 > c_x^{1/2} \\ \frac{1}{1+\sqrt{c_x}} & \text{otherwise} \end{cases}$$

and

$$\lambda_{y,icca}^{(i)} \xrightarrow{a.s.} \begin{cases} \frac{\theta_i^{(y)}}{\sqrt{\left(1+\left(\theta_i^{(y)}\right)^2\right)\left(c_y + \left(\theta_i^{(y)}\right)^2\right)}} & \text{if } \left(\theta_i^{(y)}\right)^2 > c_y^{1/2} \\ \frac{1}{1+\sqrt{c_y}} & \text{otherwise} \end{cases}$$

Under the Gaussian noise assumption, we see that the phase transition is the same as the consistency results in Chapter IV. We also notice that the limiting behavior of these weights may be calculated simply from the system parameters Θ_x, Θ_y, n, p, and q.

We next turn toward the accuracy of the estimates using these weights. We define the accuracy of the canonical vector estimates of $w_x^{(i)}$ using weights, $\lambda = [\lambda_1, \ldots, \lambda_{k_x}]$ as

$$\begin{aligned} \text{ACC}^{(i)}(\lambda) &= \left| \frac{w_x^{(i)H} \widehat{w}_x^{(i)}(\lambda)}{\|w_x^{(i)}\| \|\widehat{w}_x^{(i)}(\lambda)\|} \right|^2 \\ &= \frac{\left(U_{\widetilde{K}}^{(i)H} \left(\Theta_x + I_{k_x}\right)^{-1/2} U_x^H \mathring{U}_x \Lambda \widehat{U}_{\widetilde{K}}^{(i)}\right)^2}{\left(U_{\widetilde{K}}^{(i)H} \left(\Theta_x + I_{k_x}\right)^{-1} U_{\widetilde{K}}^{(i)}\right)\left(\widehat{U}_{\widetilde{K}}^{(i)H} \Lambda^2 \widehat{U}_{\widetilde{K}}^{(i)}\right)} \end{aligned} \tag{5.16}$$

We may similarly define the accuracy for the canonical vector estimates of $w_y^{(i)}$. We do not report the analogous theorem to save space. Simply replace all x subscripts with y.

Theorem 5.5.3. *Assume that for $i = 1, \ldots, k_x$ $\theta_i^{(x)} > 1/D_{\mu_{Z_x}}(b_X^+)$ and that for $i = 1, \ldots, k_y$, $\theta_i^{(y)} > 1/D_{\mu_{Z_y}}(b_Y^+)$. Then in the asymptotic regime considered in Theorem 5.5.2, the accuracy defined in (5.16) exhibits the following behavior:*

$$ACC^{(i)}(\lambda) \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j\lambda_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \lambda_j^2\right)},$$

where

$$\alpha_j = \sqrt{\frac{-2\phi_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)D_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}{D'_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}}.$$

Consequently,
a)

$$ACC^{(i)}(\lambda_{x,opt}) \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j^2\sqrt{D_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}\sqrt{D_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)+1}}\right)^2}{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j^2 D_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}{1+D_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}\right)},$$

b)

$$ACC^{(i)}(\lambda_{x,icca}) \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}\sqrt{\left(\sigma_x^{(j)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\sigma_x^{(j)}\right)^2+1}\right)}.$$

Similar expressions exist for $ACC^{(i)}(\lambda_{y,opt})$ and $ACC^{(i)}(\lambda_{y,icca})$ and are found by replacing the quantities dependent on X with those dependent on Y.

This theorem holds for low-rank signals with non-Gaussian noise. Similar to Corollary 5.5.1, we may explicitly solve the D-transforms for the Gaussian settings to recover closed form expressions of the accuracy in terms of Θ_x, Θ_y, p, q, and n. A similarly corollary exists for the accuracy of the canonical vector estimates of $w_y^{(i)}$ by replacing the quantities dependent on x with those dependent on y.

Corollary 5.5.2. *In the same setting as Theorem 5.5.3, under the data model of (5.1), we have that,*

$$ACC^{(i)}(\lambda) \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j \lambda_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \lambda_j^2\right)},$$

where

$$\alpha_j = \frac{\left(\theta_i^{(x)}\right)^4 - c_x}{\left(\theta_1^{(x)}\right)^4 + \left(\theta_i^{(x)}\right)^2 c_x}.$$

Consequently,

a)

$$ACC^{(i)}(\lambda_{x,opt}) \xrightarrow{a.s.} \frac{\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j^2}{\left(\theta_j^{(x)}\right)^2+1}}{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)},$$

b)

$$ACC^{(i)}(\lambda_{x,icca}) \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j}{\left(\theta_j^{(x)}\right)^2+1} \frac{\theta_j^{(x)}}{\sqrt{\left(\theta_j^{(x)}\right)^2+c}}\right)^2}{\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1} \frac{\left(\theta_j^{(x)}\right)^2}{\left(\theta_j^{(x)}\right)^2+c}\right)}$$

Conjecture 5.5.1. *Based on the observations by Bao et. al [2], we believe that the canonical vectors used by empirical CCA will be uninformative when $\kappa_i^2 < r_c$. In this regime, we believe that ACC(λ) = 0. See Appendix C for a proof of the empirical CCA canonical vector accuracy. There is still one term that we cannot approximate in closed form but using a numerically simulation for this term in the accuracy approximation yields good results.*

5.5.1 Extension to missing data

We now consider the setting where our data matrices X and Y have missing entries. In such as setting, our matrices are modeled as

$$
\begin{aligned}
X &= \big(U_x V_x^H + Z_x\big) \odot M_x \\
Y &= \big(U_y V_y^H + Z_y\big) \odot M_y
\end{aligned}
\tag{5.17}
$$

where

$$
M_{ij}^x = \begin{cases} 1 & \text{with probability } \gamma_x \\ 0 & \text{with probability } 1-\gamma_x \end{cases} \qquad M_{ij}^y = \begin{cases} 1 & \text{with probability } \gamma_y \\ 0 & \text{with probability } 1-\gamma_y \end{cases}
$$

and $\odot$ denotes the Hadamard or element-wise product. Similar to Chapter IV, we make the following low-coherence assumption about our data.

Assumption 5.5.1. *In the missing data setting, assume that the columns of U_x, U_y, V_x, and V_y satisfy a 'low-coherence' condition in the following sense: we suppose that there exist non-negative constants $\eta_{u,x}$, $C_{u,x}$, $\eta_{u,y}$, $C_{u,y}$, $\eta_{v,x}$, $C_{v,x}$,$\eta_{v,y}$, $C_{v,y}$ independent of n, such that for $i = 1, \dots, k_x$ and $\jmath = 1, \dots, k_y$,*

$$
\begin{aligned}
\max_i \|u_i^{(x)}\|_\infty &\leq \eta_{u,x} \frac{\log^{C_{u,x}} p}{\sqrt{p}}, \quad \max_i \|u_j^{(y)}\|_\infty \leq \eta_{u,y} \frac{\log^{C_{u,y}} q}{\sqrt{q}} \\
\max_i \|v_i^{(x)}\|_\infty &\leq \eta_{v,x} \frac{\log^{C_{v,x}} n}{\sqrt{n}}, \quad \max_i \|v_j^{(x)}\|_\infty \leq \eta_{v,y} \frac{\log^{C_{v,y}} n}{\sqrt{n}}.
\end{aligned}
$$

In the missing data setting, we consider the analogous optimization problem to (5.12. The main difference is assuming that the entries of our population canonical vectors are observed with the same probability as our data.

$$
\begin{aligned}
\lambda_x^{\text{opt}} &= \underset{\lambda_x}{\operatorname{argmin}} \left\| \gamma_x W_x - \widehat{U}_x \,\mathbf{diag}(\lambda_x) \widehat{U}_{\widetilde{K}} \right\|_F \\
\lambda_y^{\text{opt}} &= \underset{\lambda_y}{\operatorname{argmin}} \left\| \gamma_y W_y - \widehat{U}_y \,\mathbf{diag}(\lambda_x) \widehat{V}_{\widetilde{K}} \right\|_F.
\end{aligned}
\tag{5.18}
$$

Using these optimization problems, we have an analogous Theorem to Corollary 5.5.1 for missing data. Again, these weights may be computed in closed form with knowledge of Θ_x, Θ_y, p, q, and n. A key observation of this theorem is that missing data only decreases the relative SNR and therefore we may still use Algorithm 5.1 to compute these weights if the noise is non-Gaussian.

Theorem 5.5.4. *Let $p, q, n \to \infty$ with $p/n \to c_x$ and $q/n \to c_y$ and assume the coherence conditions given in Assumption 5.5.1. Given data modeled in (5.17), then the solution to (5.18) exhibits the following behavior for $\gamma_x, \gamma_y \in (0, 1]$*

$$\lambda_{x,opt}^{(i)} \xrightarrow{a.s.} \begin{cases} \sqrt{\frac{\gamma_x^2\left(\theta_i^{(x)}\right)^4 - c_x}{\gamma_x\left(\theta_i^{(x)}\right)^2\left(\gamma_x\left(\theta_i^{(x)}\right)^2 + c_x\right)\left(\gamma_x\left(\theta_i^{(x)}\right)^2 + 1\right)}} & \text{if } \left(\theta_i^{(x)}\right)^2 > \frac{c_x^{1/2}}{\gamma_x} \\ 0 & \text{otherwise} \end{cases}$$

and for $i = 1, \ldots, k_y$,

$$\lambda_{y,opt}^{(i)} \xrightarrow{a.s.} \begin{cases} \sqrt{\frac{\gamma_y^2\left(\theta_i^{(y)}\right)^4 - c_y}{\gamma_y\left(\theta_i^{(y)}\right)^2\left(\left(\theta_i^{(y)}\right)^2 + c_y\right)\left(\gamma_y\left(\theta_i^{(y)}\right)^2 + 1\right)}} & \text{if } \left(\theta_i^{(y)}\right)^2 > \frac{c_y^{1/2}}{\gamma_y} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\lambda_{x,icca}^{(i)} \xrightarrow{a.s.} \begin{cases} \frac{\sqrt{\gamma_x}\theta_i^{(x)}}{\sqrt{\left(1 + \gamma_x\left(\theta_i^{(x)}\right)^2\right)\left(c_x + \gamma_x\left(\theta_i^{(x)}\right)^2\right)}} & \text{if } \left(\theta_i^{(x)}\right)^2 > \frac{c_x^{1/2}}{\gamma_x} \\ \frac{1}{\sqrt{\gamma_x}(1 + \sqrt{c_x})} & \text{otherwise} \end{cases}$$

and

$$\lambda_{y,icca}^{(i)} \xrightarrow{a.s.} \begin{cases} \frac{\sqrt{\gamma_y}\theta_i^{(y)}}{\sqrt{\left(1 + \gamma_y\left(\theta_i^{(y)}\right)^2\right)\left(c_y + \gamma_y\left(\theta_i^{(y)}\right)^2\right)}} & \text{if } \left(\theta_i^{(y)}\right)^2 > \frac{c_y^{1/2}}{\gamma_y} \\ \frac{1}{\sqrt{\gamma_y}\left(1 + \sqrt{c_y}\right)} & \text{otherwise} \end{cases}$$

5.6 Orthogonal Canonical Vector Estimates

Algorithm 5.1 requires the entire spectrum of both X and Y, which requires computing a $p \times n$ and $q \times n$ SVD. As $p, q, n \to \infty$, these SVDs become more expensive. Motivated by this drawback of computing the optimal weights, we are curious to explore the performance of an orthogonal approximation to (5.8). Define the orthogonal canonical vector estimates as

$$\begin{aligned} \widehat{W}_x^{\text{orth}} &= \mathring{U}_x \widehat{U}_{\widetilde{K}} \\ \widehat{W}_y^{\text{orth}} &= \mathring{U}_y \widehat{V}_{\widetilde{K}}. \end{aligned} \tag{5.19}$$

In light of our optimal weighting matrices Λ_x^{opt} and Λ_y^{opt}, the orthogonal approximation set $\Lambda_x^{\text{opt}} = I_{\widehat{k}_x}$ and $\Lambda_y^{\text{opt}} = I_{\widehat{k}_y}$. With this observation, we determine the asymptotic accuracy of the orthogonal estimates with the following Theorem.

Theorem 5.6.1. *In the same setting as Theorem 5.5.3, we have the limiting accuracy of the orthogonal approximation is*

$$ACC^{(i)}(\lambda_{x,orth}) \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2\right)}.$$

A similar expression exists for the accuracy of the canonical vectors of Y by substituting the appropriate parameters.

A natural question arises: When are the orthogonal canonical correlation vectors equivalent to the ICCA and ICCA+ estimates? By examining the accuracy expression in Theorems 5.5.3 and 5.6.1, we have the following conditions for equivalency between the estimates:

1. $U_{\widetilde{K}} = I_{k_x}$
2. $\Theta_x = \alpha I_{k_x}$

Similar conditions hold for the canonical vectors of Y. The first condition is the most interesting. The matrix $U_{\widetilde{K}}$ controls how the signals between datasets interact. Therefore, when $U_{\widetilde{K}} = I$, each canonical vector is a scaled version of a column of U_x, representing one signal component from the dataset. While this scaling will be different for each estimate, it does not affect the accuracy of the estimates, which are all the same. When the SNRs are all the same, a similar behavior occurs and the weights on each component of $U_{\widetilde{K}}$ are all the same regardless of estimate.

5.7 Empirical Results - Synthetic Data

In this section we explore the empirical accuracy of the four estimates of the population canonical vectors, empirical CCA, ICCA, ICCA+, and orthogonal. In our experiments, we show the extreme sub-optimality of the empirical CCA estimates; all other estimates outperform empirical CCA in the sample deficient regime. More interestingly, we compare the performance of the other three estimates for a few parameter choices to highlight key differences.

5.7.1 Performance on non identity $U_{\widetilde{K}}$

As discussed in Section 5.6, when $U_{\widetilde{K}}$ is identity, the ICCA, ICCA+, and orthogonal estimates all return a scaled version of the same estimate. Therefore in this section, we consider the case where $U_{\widetilde{K}}$ is not identity. We consider a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \mathbf{diag}(16, 1)$, $P_{xy} = \mathbf{diag}(0.9, 0.9)$, $V_K = I_2$, and

$$U_K = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}.$$

In this setup,

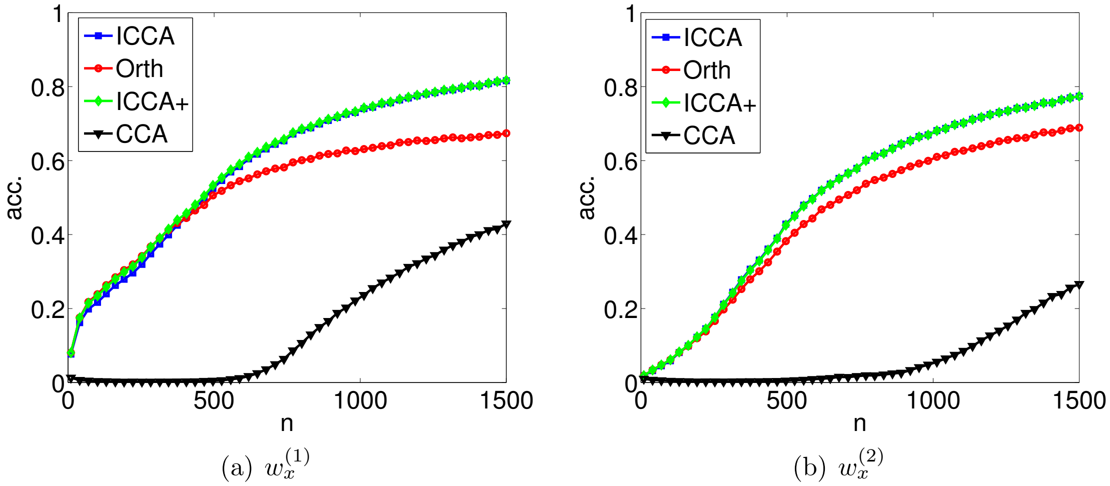
$$U_{\widetilde{K}} = \begin{bmatrix} -0.8559 & -0.5172 \\ -0.5172 & 0.8559 \end{bmatrix}.$$

In this simulation, we sweep over n and compute the accuracy given in (5.16) of our four estimates for each of the two canonical vectors. For each value of n, we average over 750 different generated data matrices from (5.1). Figure 5.2 plots the results.

For this parameter setup, we see that the ICCA+ estimate performs well throughout all values of n. Because $U_{\widetilde{K}}$ is non-identity, we get the strange behavior that, when the second subspace component is uninformative, the orthogonal estimate outperforms the ICCA estimate. This occurs for low values of n, around 100-200. Once n is large enough that the second component is informative, the orthogonal approximation becomes suboptimal but still outperforms the empirical CCA estimate. The beauty of the ICCA+ estimate is that it knows when the subspace estimates are inaccurate. In this low n regime, it does not give much weight to the inaccurate second subspace and so outperforms the ICCA estimate, which places a non-zero weight on the second very noisy subspace estimate. However for large values of n where both subspace estimates are accurate, the ICCA estimate outperforms the orthogonal approximation and achieves the same performance as the ICCA+ estimate.

5.7.2 Convergence

The theorems presented in this chapter state their results for the asymptotic regime of $p, q, n \to \infty$ with $p/n \to c_x$ and $q/n \to c_y$. Here, for three fixed values of $c_x = 0.5, 1, 2$, we generate data from (5.1) with the parameters from Figure 5.2 for 3 values of $p = 100, 500, 1000$ to ensure that the estimates do indeed converge. Figure 5.3 plots the accuracy as defined in (5.16) for the first canonical vector for all four estimators. We also plot one standard deviation errorbars from the simulation. Figures 5.4 and 5.5 plot the accuracy convergence for the individual estimates for



(a) $w_x^{(1)}$

(b) $w_x^{(2)}$

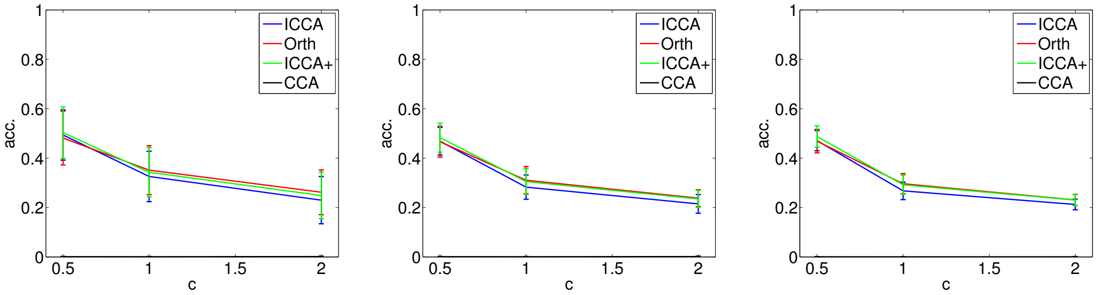
Figure 5.2: Accuracy plots as a function of n for a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \mathbf{diag}(16, 1)$, $P_{xy} = \mathbf{diag}(0.9, 0.9)$, $V_K = I_2$, and non-identity U_K. Accuracy is defined in (5.16). The left figure plots the accuracy of the first canonical vector and the right figure plots the accuracy of the second canonical vector.

both the first and second canonical vectors.

These three figures paint a nice picture of the different estimators. We first see that the CCA estimator fails for all three values of c_x. This gives credence to Conjecture 5.5.1 that when $n < p + q$ the canonical vectors returned by empirical CCA are random. For the other three estimators, we see accuracy increase as c_x decreases. This makes sense as we expect our estimates to perform better given more samples relative to the dimension size. Next we note that as p increases, the errorbars on all estimates decrease, empirically verifying the belief that the accuracy does indeed have an asymptotic limit. Finally, we note that these figures reinforce the fact that the optimal weights are optimal in the asymptotic regime. For small p, the orthogonal estimate slightly outperforms the ICCA+ estimate. However, when $p = 1000$, we see that this gap closes and that the ICCA+ estimate starts to outperform all other estimates.

5.7.3 Robustness to $\widehat{k}_x$

Finally we explore the performance of our estimators when we change the estimate of the number of subspace components, $\widehat{k}_x$. The theorems presented in this chapter assume that $\widehat{k}_x = k_x$. However, we explore the performance of these estimates when this assumption is not valid. Figure 5.6 and 5.7 plot the performance of the estimates while sweeping over $\widehat{k}_x = \widehat{k}_y$ for $c_x = 0.2$ and $c_x = 1$, respectively. We use a different



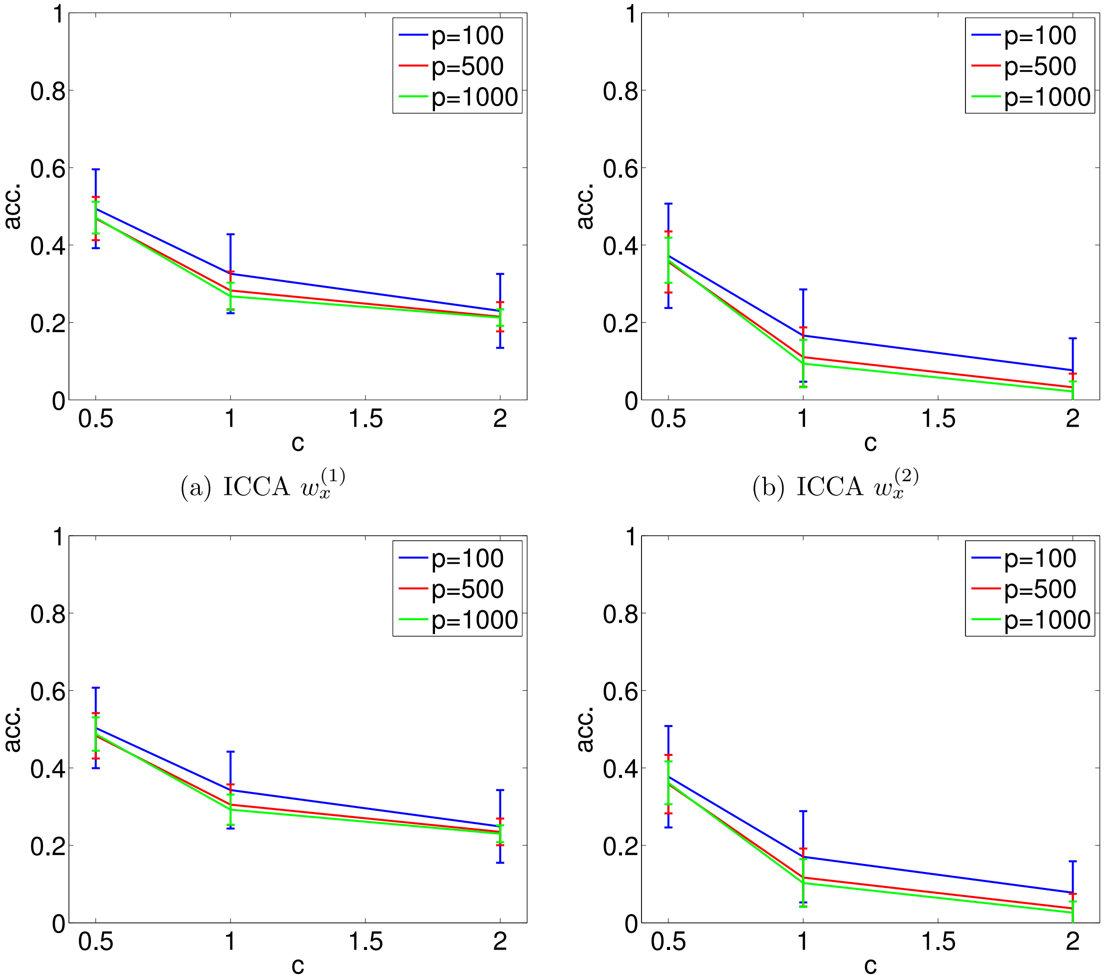
(a) $p = 100$ (b) $p = 500$ (c) $p = 1000$

Figure 5.3: Convergence plots of the first canonical vector of each estimate for three values of p. Results are plotted for three fixed values of $c_x = 0.5, 1, 2$. The simulation setting is the same as Figure 5.2 for a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \mathbf{diag}(16, 1)$, $P_{xy} = \mathbf{diag}(0.9, 0.9)$, $V_K = I_2$, and non-identity U_K. Errorbars are 1 standard deviation.

simulation setting than Figure 5.2. Here we set setting $k_x = k_y = 3$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \mathbf{diag}(3, 2, 1)$, $P_{xy} = \mathbf{diag}(0.9, 0.5, 0.3)$,$V_k = I_3$ and

$$U_k = \left[\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \right].$$

From these Figures we see that all estimates suffer greatly when $\widehat{k}_x < k_x = 3$. This makes sense as we don't use all of the possible signals that are present. The more interesting behavior occurs when we overestimate k_x, which is a common practice in many applications. We see that the first canonical vector in both cases remains robust to overestimating $\widehat{k}_x$. This first canonical vector corresponds to the largest singular value of $\widetilde{K}_{xy}$ and so the corresponding singular vector estimate is accurate. However, the higher order canonical vectors are less accurate as we overestimate $\widehat{k}_x$. This accuracy suffers more for $w_x^{(3)}$ with increasing $\widehat{k}_x$. For the case of $c = 0.2$ in Figure 5.6, we see that for these parameters, ICCA+ is the most robust even as we greatly overestimate $\widehat{k}_x$ and that the orthogonal estimate is more accurate than the ICCA estimate when we overestimate $\widehat{k}_x$. This makes sense as Θ_x and Θ_y are very close to identity. For all cases though, the CCA estimate is very bad and all of the ICCA, orthogonal, and ICCA+ estimates greatly outperform it. In Figure 5.7 when $c = 1$, we see that the accuracy is much lower for all estimates; this makes sense as a larger c corresponds to fewer samples. Still though, the ICCA+ estimate is the most robust estimate and the CCA estimate is completely random.



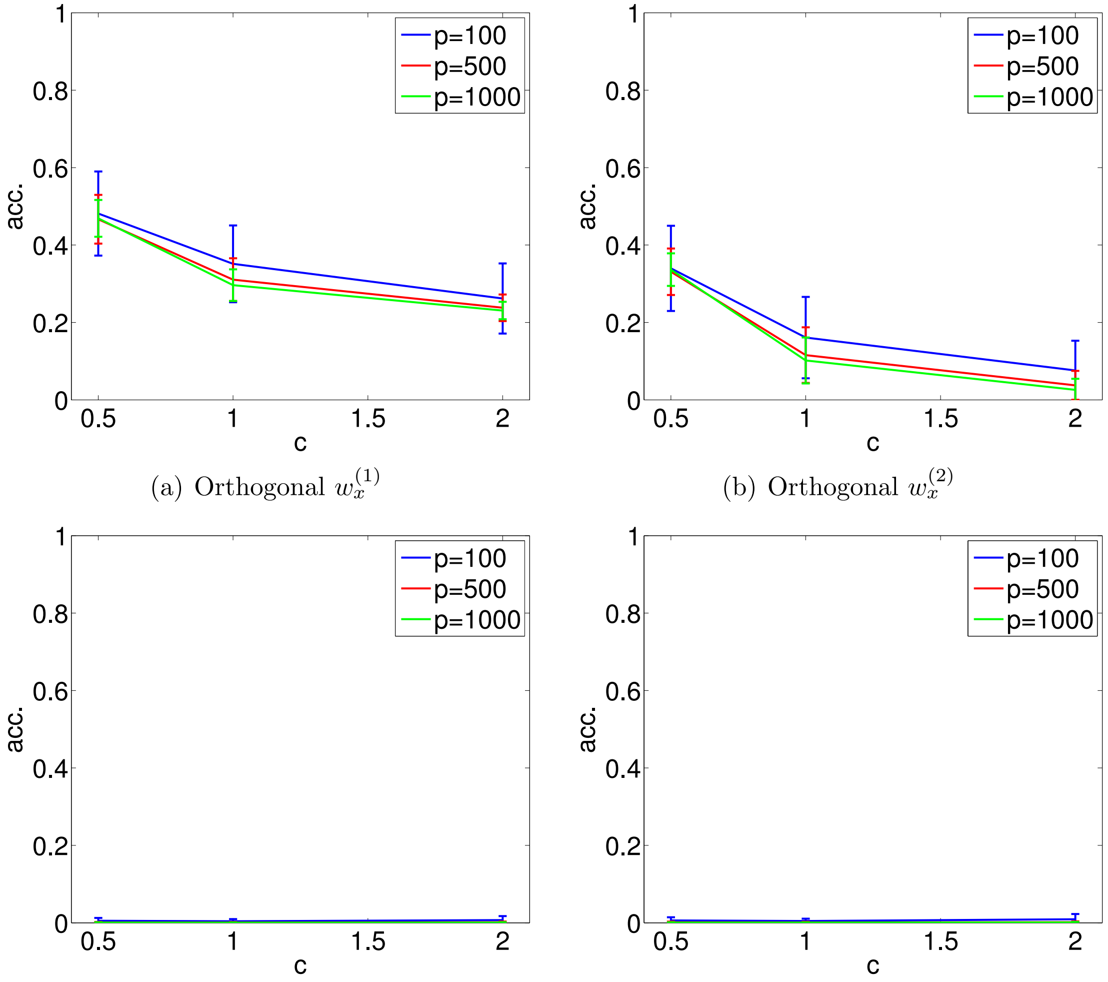
(a) ICCA $w_x^{(1)}$

(b) ICCA $w_x^{(2)}$

(c) ICCA+ $w_x^{(1)}$

(d) ICCA+ $w_x^{(2)}$

Figure 5.4: Accuracy convergence plots for the top two canonical vectors of the ICCA and ICCA+ estimates. Results are plotted for three fixed values of $c_x = 0.5, 1, 2$ for three different values of p. The simulation setting is the same as Figure 5.3 for a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \mathbf{diag}(16, 1)$, $P_{xy} = \mathbf{diag}(0.9, 0.9)$, $V_K = I_2$, and non-identity U_K. Errorbars are 1 standard deviation.



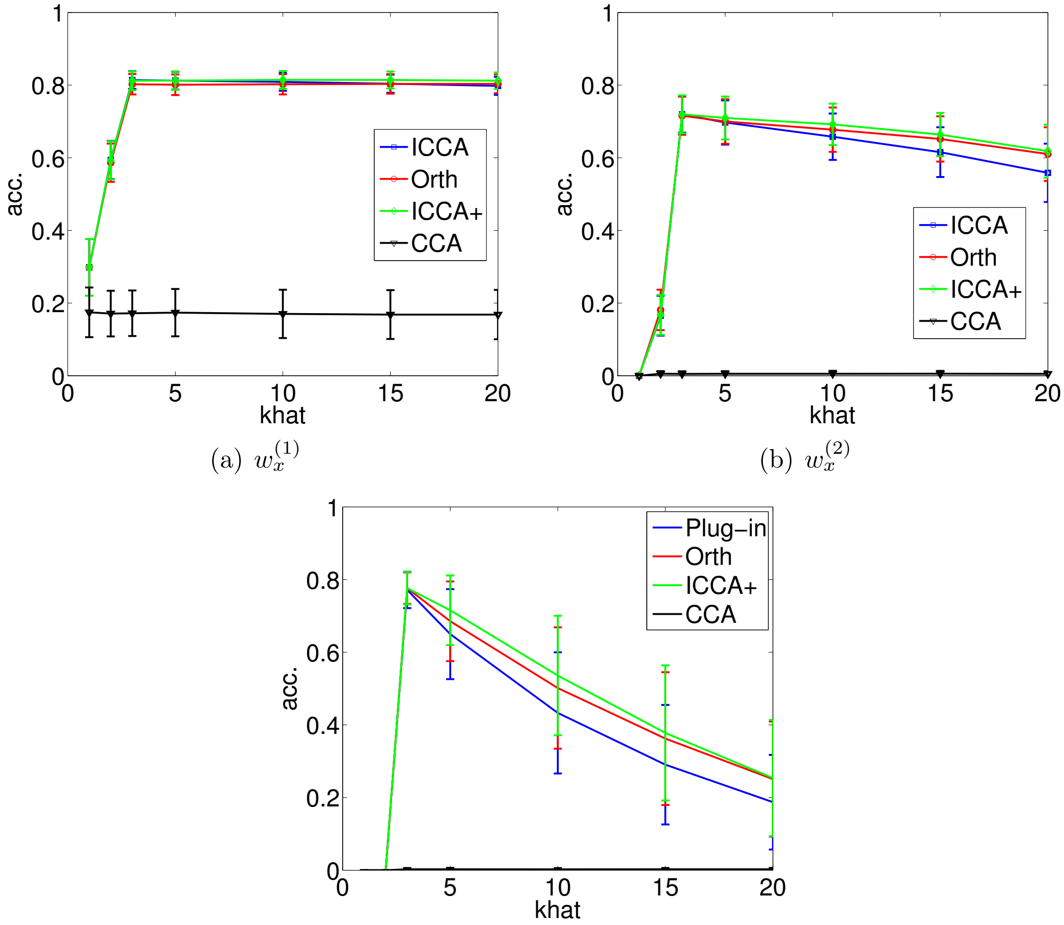
(a) Orthogonal $w_x^{(1)}$

(b) Orthogonal $w_x^{(2)}$

(c) Empirical CCA $w_x^{(1)}$

(d) Empirical CCA $w_x^{(2)}$

Figure 5.5: Accuracy convergence plots for the top two canonical vectors of the orthogonal and empirical CCA estimates. Results are plotted for three fixed values of $c_x = 0.5, 1, 2$ for three different values of p. The simulation setting is the same as Figure 5.3 for a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \mathbf{diag}(16, 1)$, $P_{xy} = \mathbf{diag}(0.9, 0.9)$, $V_K = I_2$, and non-identity U_K. Errorbars are 1 standard deviation.

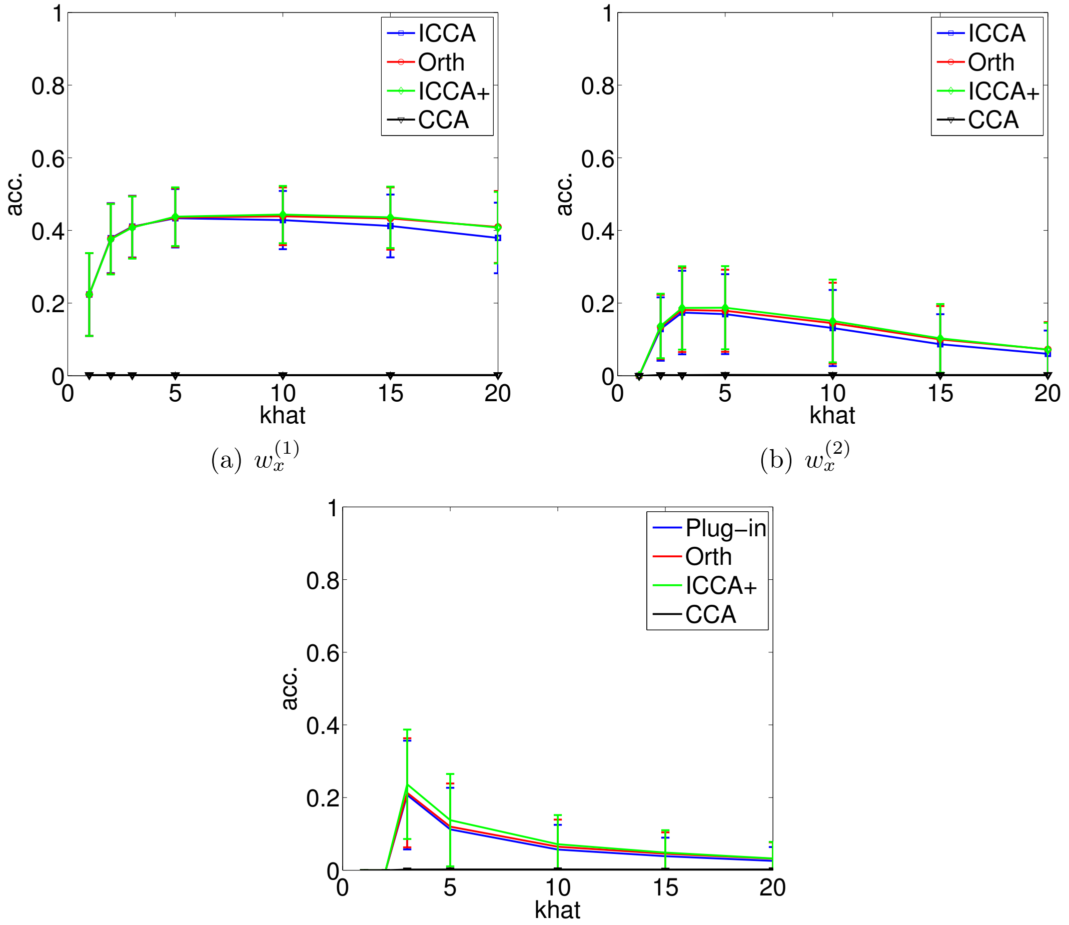


(a) $w_x^{(1)}$

(b) $w_x^{(2)}$

(c) $w_x^{(3)}$

Figure 5.6: Accuracy plots of the first two canonical vector estimates a function of $\widehat{k}_x$ for $c = 0.2$. The simulation setting is the same as Figure 5.3 for a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \textbf{diag}(16, 1)$, $P_{xy} = \textbf{diag}(0.9, 0.9)$, $V_K = I_2$, and non-identity U_K. Errorbars are 1 standard deviation.



(a) $w_x^{(1)}$

(b) $w_x^{(2)}$

(c) $w_x^{(3)}$

Figure 5.7: Accuracy plots of the first two canonical vector estimates a function of $\widehat{k}_x$ for $c = 1$. The simulation setting is the same as Figure 5.3 for a rank-2 setting where $k_x = k_y = 2$, $p = 200$, $q = 250$, $\Theta_x = \Theta_y = \textbf{diag}(16, 1)$, $P_{xy} = \textbf{diag}(0.9, 0.9)$, $V_K = I_2$, and non-identity U_K. Errorbars are 1 standard deviation.

5.8 Empirical Results - Real-World Data

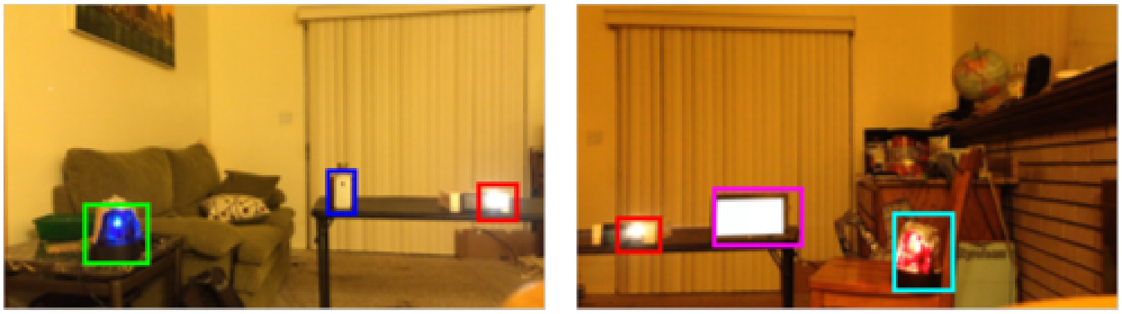
To compare the performance of these canonical vector estimates on real world applications, we reuse two of the controlled experiments we created in Chapter IV. These examples showcase quite well the very nuanced behavior of the ICCA, orthogonal, and ICCA+ estimates. We recall that the ICCA+ estimate is optimal in an asymptotic sense and so that, as in some of these examples, finite p, q, n cause the ICCA+ estimates to perform slightly worse than the ICCA or orthogonal estimates. The ICCA canonical vector estimate requires the inversion of $\widehat{R}_{xx}$ and $\widehat{R}_{yy}$ (or equivalently computing the SVD of X and Y), which involves inverting the estimated singular values of R_{xx} and R_{yy}. In some cases, inaccurate singular value estimates actually improve the weightings applied to the singular vectors, which thus improves the canonical vector accuracy. While the orthogonal estimate works quite well when $U_{\widetilde{K}}$ is identity, it suffers a performance loss when this assumption is not true. Therefore, even in the finite p, q, n applications, the ICCA+ canonical vector estimate is the most robust estimator.

5.8.1 Video-Video Experiment

First, we use the video-video experiment consisting of 5 stationary flashing lights and two stationary iPhone cameras. Figure 5.8 shows the views from the left and right cameras and manually identifies each source. The 5 sources are a blue flashing police light (BPL) outlined in the green rectangle, one phone with a flashing strobe light (PH1) outlined in the dark blue rectangle, another phone with a flashing strobe light (PH2) outlined in a red rectangle, a tablet with a flashing screen (T1) outlined in the magenta rectangle, and a red flashing police light (RPL) outlined in the cyan rectangle. From left to right, the left camera can see BPL, PH1, and PH2. From left to right, the right camera can see PH2, T1, and RPL. Therefore, both cameras share the common signal of PH2. As we saw in Chapter IV, the police lights RPL and BPL are in antiphase and thus also correlated. Therefore, for this experiment each view has 3 signals, two of which are correlated.

To synchronize the cameras we used the RecoLive MultiCam iPhone app [1]. After turning on all light sources, we recorded 30 seconds of video at 30 frames per second. The resolutions of the iPhone's cameras were both 1920 × 1080 pixels. To post-process the video data, we first converted the video streams to grayscale and then downsampled each spatial dimension by a factor of 8, resulting in a resolution of

[1] http://recolive.com/en/



(a) Left Camera

(b) Right Camera

Figure 5.8: Manual source identification of each camera. Both cameras share a common flashing phone, outlined in a red rectangle. Each camera has two independent sources besides the shared flashing phone.

240 × 135. We then vectorized each image and stacked the 900 frames into data matrices , both of dimension 32400 × 900. Finally, we subtract the mean from each dataset so that we may run PCA, CCA, and ICCA on the zero-mean datasets, X_{left} and Y_{right}.

To run these algorithms, we use knowledge of the simulation setup and set $\widehat{k}_x = \widehat{k}_y = 3$. Figures 5.9 - 5.11 plot the first canonical vector estimates for the left camera after frame 5, 30, and 600, respectively. Each figure plots the absolute value of the ICCA, orthogonal, ICCA+, and empirical CCA canonical vector estimates. We plot the absolute value or each vector to discover correlated pixels; a left canonical vector gives more weight to left camera pixels it believe are correlated with the pixels in the right camera. Each figure also plots the difference between the ICCA and ICCA+ canonical vectors and the difference between the orthogonal and ICCA+ canonical vectors. In these difference figures, pixels with negative values represent pixels that the ICCA+ estimate believes are more correlated while positive values represent pixels that the ICCA+ estimate believes are less correlated. We plot the ICCA, orthogonal, and ICCA+ estimates on the same scale, but plot the CCA estimates on its own scale because they vary widely (i.e. are inaccurate).

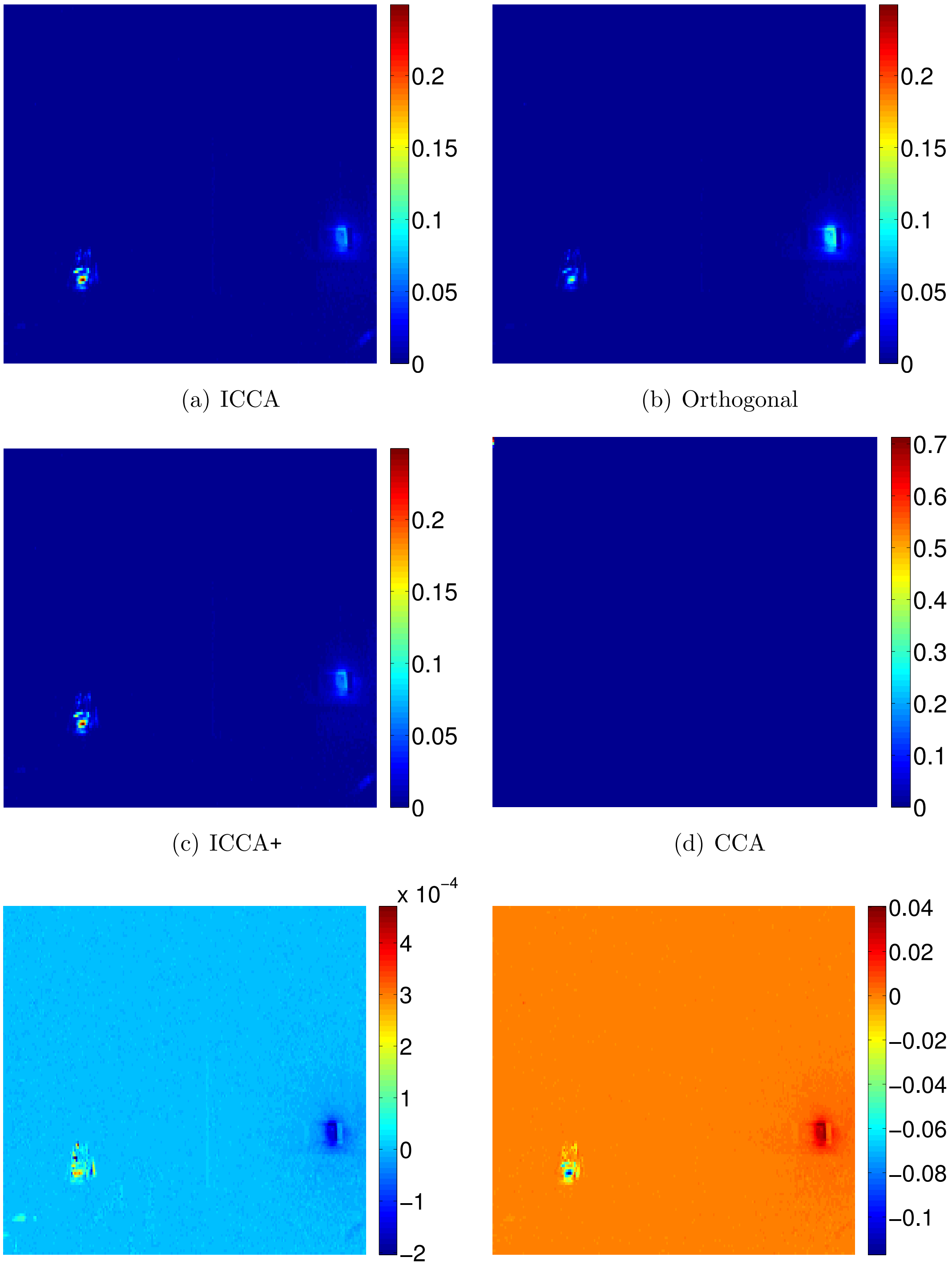
We can draw a number of conclusions from Figures 5.9 - 5.11. First, the empirical CCA canonical vector estimates are meaningless. This gives credence to Conjecture 5.5.1 because we are in the sample deficient regime where the number of pixels is much larger than the number of frames that we have. Second, the first population canonical vector identifies the shared camera PH2, which is the rightmost source in the left camera. As we get more frames, these canonical vector estimates become more “accurate”, i.e. identify only that source. Third, the ICCA, orthogonal, and

ICCA+ canonical vectors estimates are all very similar. Each estimate places a large weight on pixels around the shared source, PH2. However, there are slight differences between the estimates as seen in the sub-figures (e) and (f). We note that the scale of these differences in on the order of 10^{-3}, which is fairly small compared to the magnitude of the pixels. First we see that the ICCA estimate places more weight on the middle source PH1 than the ICCA+ estimate. This is not desirable as source PH1 is not correlated with any source in the right camera. Second, we see that the orthogonal estimate places less weight on source PH1 and less weight on source BPL than the ICCA+ estimate. This is desirable as these sources are not correlated with the shared source PH2.

Therefore, we can conclude for this first canonical vector, the orthogonal estimate performs the best and that the ICCA+ estimate performs better than the plug-in estimate. We attribute this behavior to the fact that there is no mixing of principle components in this example, as each source is identified as a principle component (see Chapter IV). This results in a $U_{\widetilde{K}}$ very close to identity, which is when the orthogonal estimate is known to perform well.

Figures 5.12 - 5.14 plot the second canonical vector estimates for the left camera after frame 5, 30, and 600, respectively. Each figure again plots the absolute value of the ICCA, orthogonal, ICCA+, and empirical CCA canonical vector estimates. Each figure also plots the difference between the ICCA and ICCA+ canonical vectors and the orthogonal and ICCA+ canonical vectors. In these figures, pixels with negative values represent pixels that the ICCA+ estimate believes are more correlated while positive values represent pixels that the ICCA+ estimate believes are less correlated.

Similar to the estimates of the first canonical vector, we see that the empirical CCA canonical vector estimate is just simply noise. The ICCA, orthogonal, and ICCA+ estimates are all very similar and all identify source BPL, which is correlated to source RPL in the right camera. Again, these estimates improve as we get more samples (frames). Examining the difference plots in (e) and (f), we once again see that the ICCA estimate is suboptimal as it places more weight on the independent source PH1 than the ICCA+ estimate. The difference between the orthogonal and ICCA+ estimates is fairly interesting. The orthogonal estimate places more weight on source PH2, while the ICCA+ estimate places more weight on source PH1. Both of these sources are not correlated with the police lights and so we conclude that the orthogonal and ICCA+ estimates do equally well estimating the second canonical vector.



(a) ICCA

(b) Orthogonal

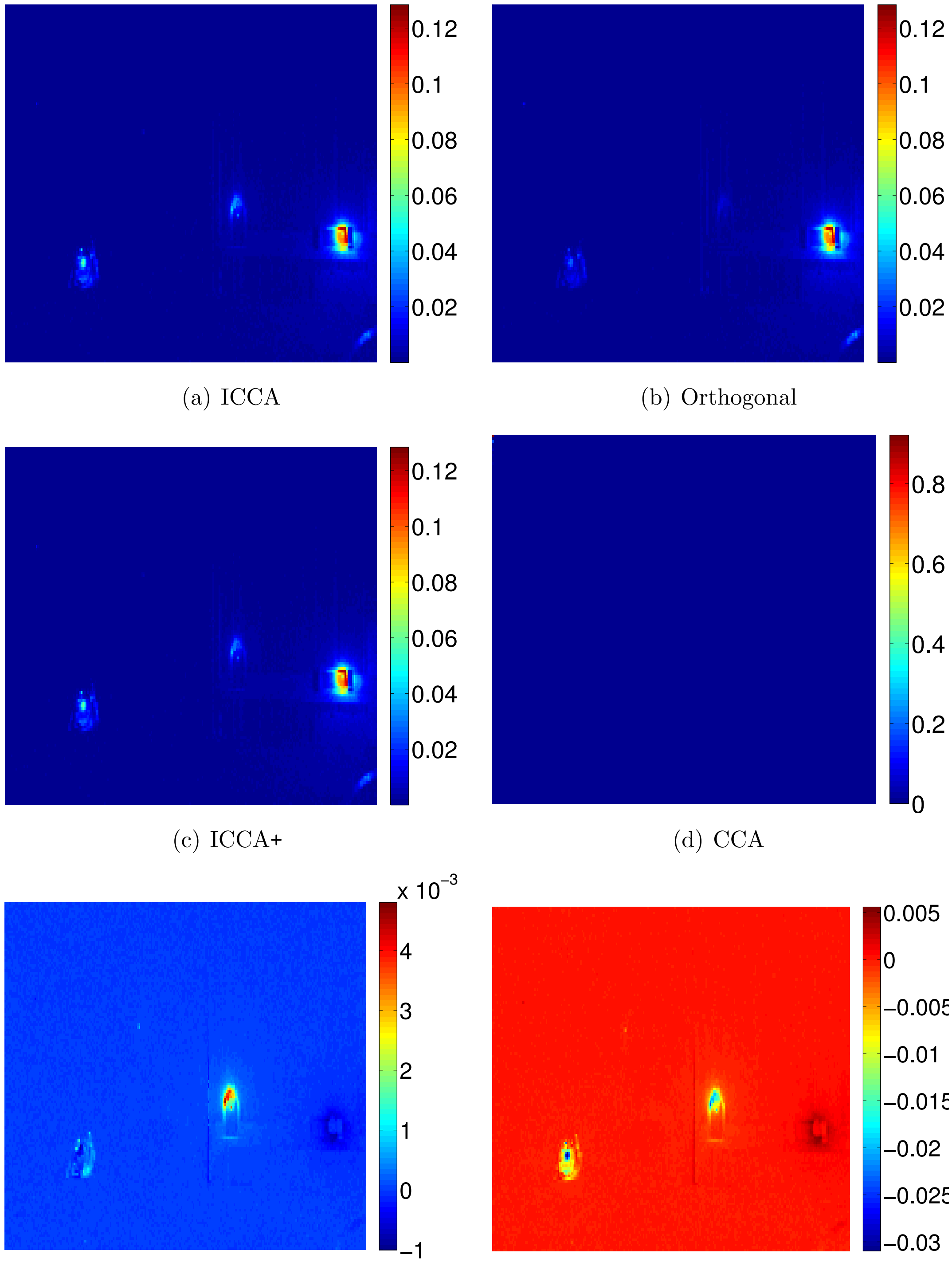
(c) ICCA+

(d) CCA

(e) ICCA minus ICCA+

(f) Orthogonal minus ICCA+

Figure 5.9: First canonical vector estimates for the left camera at frame 5. This corresponds to a total capture time of 1/6 of a second. (a)-(d) show the absolute value of the vectors displayed in an image so that large values indicate correlated pixels. (e)-(f) plot the difference between the ICCA estimate and the ICCA+ estimate and the orthogonal estimate and the ICCA+ estimate. Positive values indicate pixels that the ICCA+ estimate thinks are less correlated while negative values indicate pixels that the ICCA+ estimate thinks are more correlated.



(a) ICCA

(b) Orthogonal

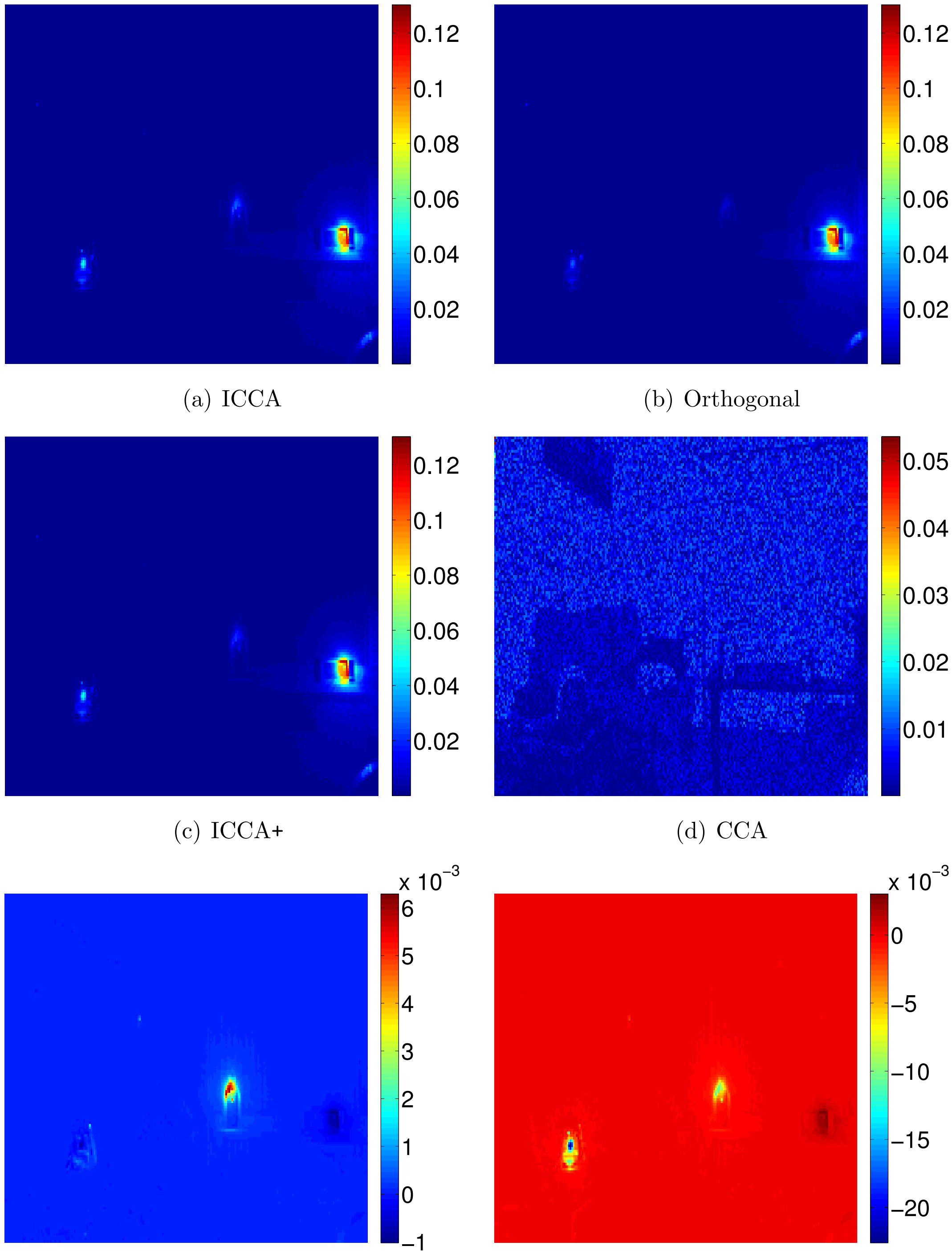
(c) ICCA+

(d) CCA

(e) ICCA minus ICCA+

(f) Orthogonal minus ICCA+

Figure 5.10: First canonical vector estimates for the left camera at frame 30. This corresponds to a total capture time of 1 second. (a)-(d) show the absolute value of the vectors displayed in an image so that large values indicate correlated pixels. (e)-(f) plot the difference between the ICCA estimate and the ICCA+ estimate and the orthogonal estimate and the ICCA+ estimate. Positive values indicate pixels that the ICCA+ estimate thinks are less correlated while negative values indicate pixels that the ICCA+ estimate thinks are more correlated.



(a) ICCA

(b) Orthogonal

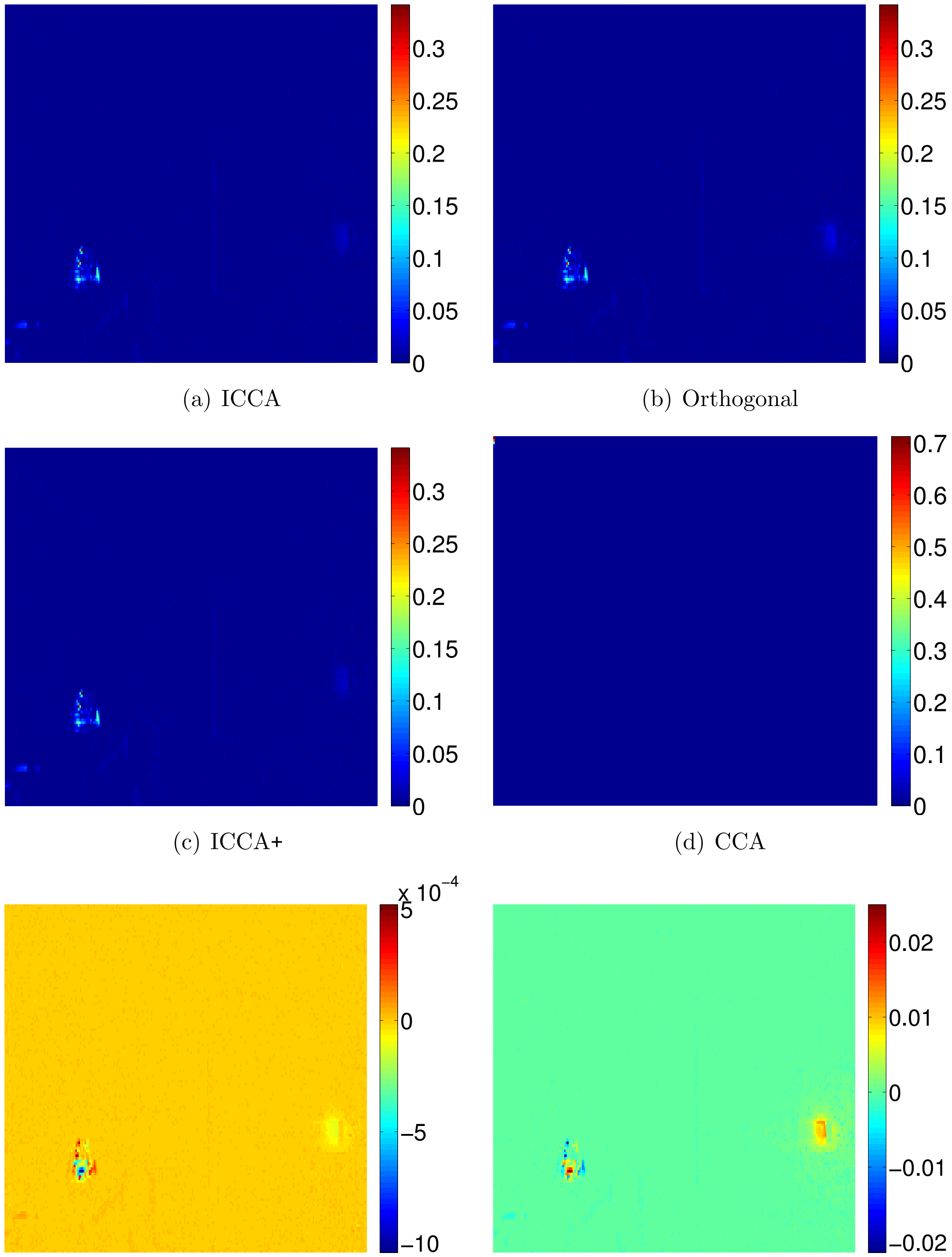
(c) ICCA+

(d) CCA

(e) ICCA minus ICCA+

(f) Orthogonal minus ICCA+

Figure 5.11: First canonical vector estimates for the left camera at frame 600. This corresponds to a total capture time of 20 seconds. (a)-(d) show the absolute value of the vectors displayed in an image so that large values indicate correlated pixels. (e)-(f) plot the difference between the ICCA estimate and the ICCA+ estimate and the orthogonal estimate and the ICCA+ estimate. Positive values indicate pixels that the ICCA+ estimate thinks are less correlated while negative values indicate pixels that the ICCA+ estimate thinks are more correlated.



(a) ICCA

(b) Orthogonal

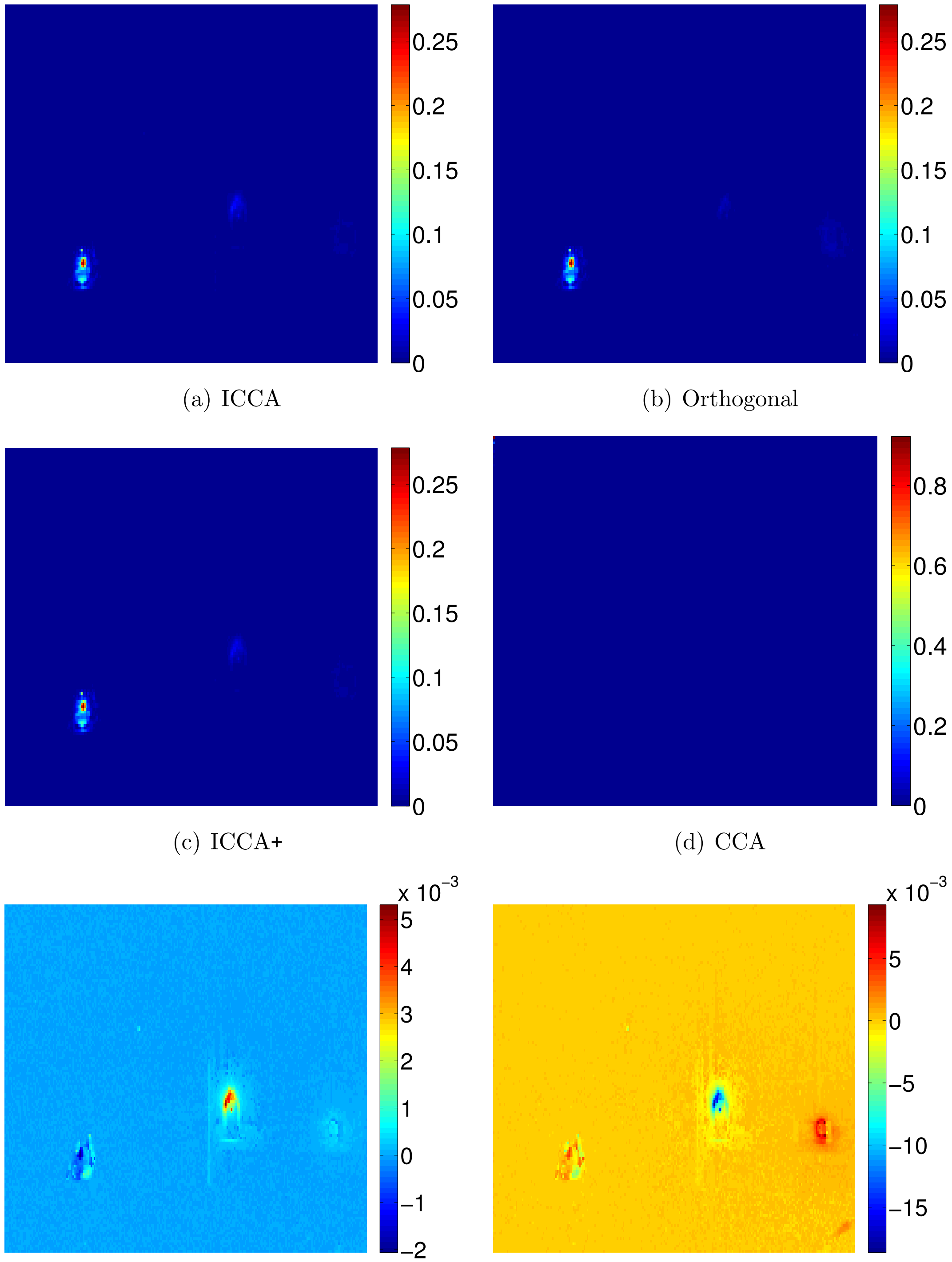
(c) ICCA+

(d) CCA

(e) ICCA minus ICCA+

(f) Orthogonal minus ICCA+

Figure 5.12: Second canonical vector estimates for the left camera at frame 5. This corresponds to a total capture time of 1/6 of a second. (a)-(d) show the absolute value of the vectors displayed in an image so that large values indicate correlated pixels. (e)-(f) plot the difference between the ICCA estimate and the ICCA+ estimate and the orthogonal estimate and the ICCA+ estimate. Positive values indicate pixels that the ICCA+ estimate thinks are less correlated while negative values indicate pixels that the ICCA+ estimate thinks are more correlated.



(a) ICCA

(b) Orthogonal

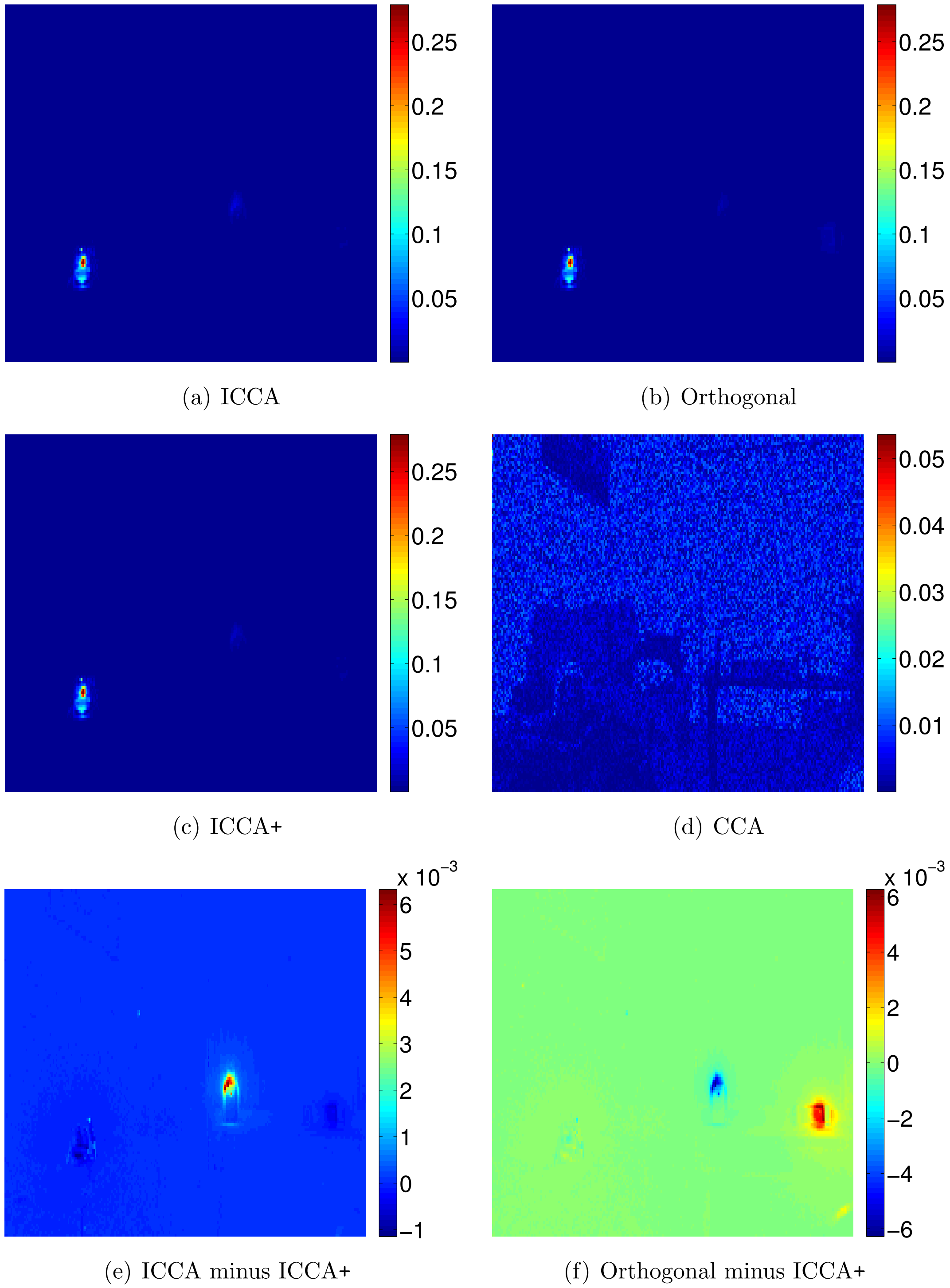
(c) ICCA+

(d) CCA

(e) ICCA minus ICCA+

(f) Orthogonal minus ICCA+

Figure 5.13: Second canonical vector estimates for the left camera at frame 30. This corresponds to a total capture time of 1 second. (a)-(d) show the absolute value of the vectors displayed in an image so that large values indicate correlated pixels. (e)-(f) plot the difference between the ICCA estimate and the ICCA+ estimate and the orthogonal estimate and the ICCA+ estimate. Positive values indicate pixels that the ICCA+ estimate thinks are less correlated while negative values indicate pixels that the ICCA+ estimate thinks are more correlated.



(a) ICCA

(b) Orthogonal

(c) ICCA+

(d) CCA

(e) ICCA minus ICCA+

(f) Orthogonal minus ICCA+

Figure 5.14: Second canonical vector estimates for the left camera at frame 600. This corresponds to a total capture time of 20 seconds. (a)-(d) show the absolute value of the vectors displayed in an image so that large values indicate correlated pixels. (e)-(f) plot the difference between the ICCA estimate and the ICCA+ estimate and the orthogonal estimate and the ICCA+ estimate. Positive values indicate pixels that the ICCA+ estimate thinks are less correlated while negative values indicate pixels that the ICCA+ estimate thinks are more correlated.

5.8.2 Audio-Audio Experiment

We also explore the accuracy of the canonical vectors estimates on the audio-audio experiment created in Chapter IV. In this experiment, we generate two 30 second audio sequences. Each sequence contains two pure-tones, which are amplitude modulated (AM) at different frequencies. In addition we add uncorrelated coffee shop noise, which is independent between each audio sequence. One pure-tone in each sequence is amplitude modulated at a shared rate, inducing correlation between the audio sequences. The remaining pure-tones are amplitude modulated at different rates, making them independent of the shared AM tones. Our waveforms are

$$\begin{aligned} a_1(t) &= \frac{1}{3}s_1(t) + \frac{1}{3}s_2(t) + \frac{1}{3}n_1(t) \\ a_2(t) &= \frac{1}{3}s_3(t) + \frac{1}{3}s_4(t) + \frac{1}{3}n_2(t) \end{aligned}$$

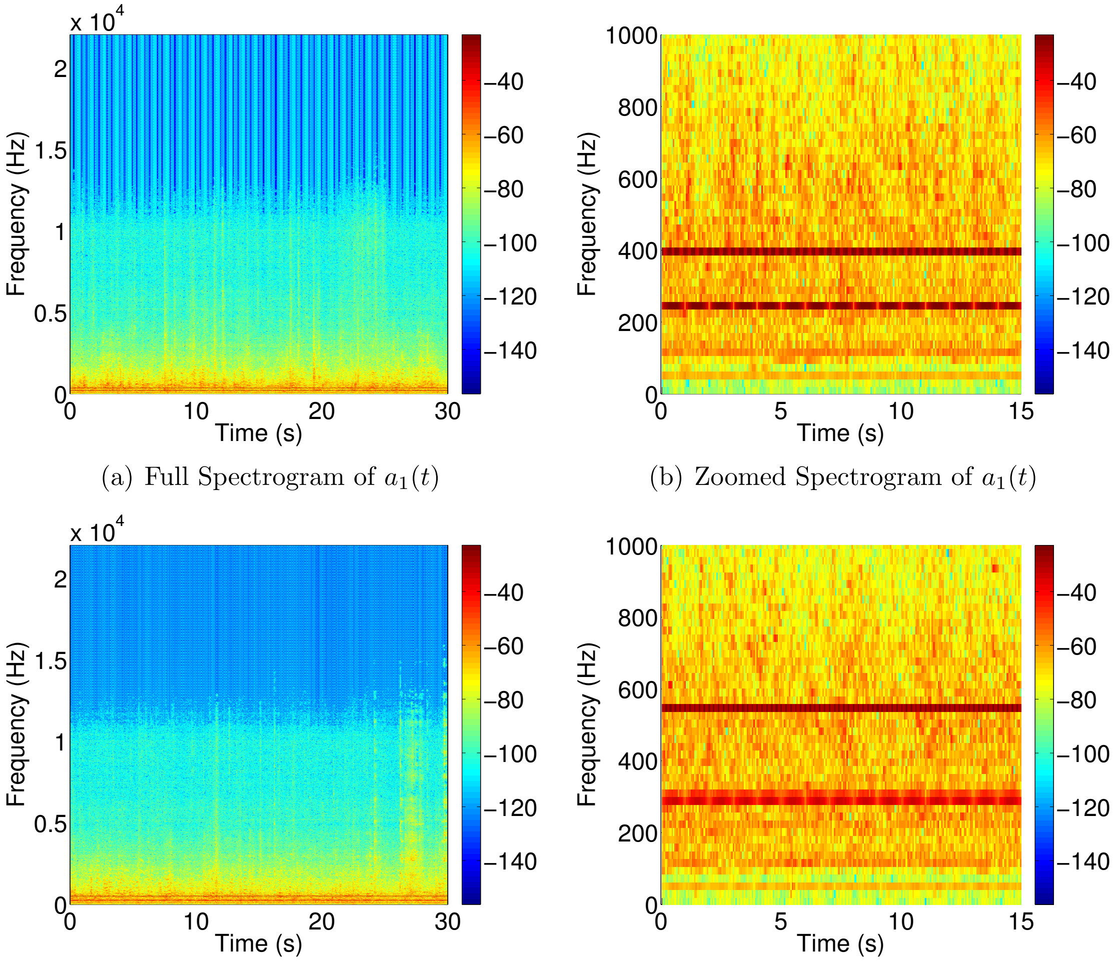
where

$$\begin{aligned} s_1(t) &= \frac{(1+\sin(2\pi t))}{2}\sin\left(2\pi\left(250t\right)\right) \\ s_2(t) &= \frac{(1+\cos(2\pi(3t))}{2}\sin\left(2\pi\left(400t\right)\right) \\ s_3(t) &= \frac{(1+\sin(2\pi t))}{2}\sin\left(2\pi\left(300t\right)\right) \\ s_4(t) &= \frac{(1+\cos(2\pi(5t))}{2}\sin\left(2\pi\left(550t\right)\right) \\ n_1(t) &= \text{independent coffee shop noise.} \\ n_2(t) &= \text{independent coffee shop noise.} \end{aligned}$$

All time sequences are generated with a sample rate of 44.1 kHz. Figure 5.15 plots the spectrogram of each sequence and zooms in on a smaller portion of the spectrum to see the AM sequences. Table 5.1 summarizes each of our signals in each audio sequences.

To post-process the data, we separate the audio streams into equal window sizes of 2940 time points, corresponding to a time interval of 1/15 second. On each window, we run a 4096 point FFT and take the magnitude of the first 2049 points as a feature vector. We then stack the feature vectors for all windows into a matrix and subtract the mean, resulting in 2049 × 450 matrices X_{a_1} and Y_{a_2}.

Figures 5.16 and 5.17 plot the first canonical vector estimates for the first and second audio steams, respectively. Each figure plots the absolute value of the canonical vectors, whose entries correspond to frequencies. Thus, large weights correspond to frequencies that are correlated between the two audio streams. Each figures plots the



(a) Full Spectrogram of $a_1(t)$

(b) Zoomed Spectrogram of $a_1(t)$

(c) Full Spectrogram of $a_2(t)$

(d) Zoomed Spectrogram of $a_2(t)$

Figure 5.15: (a) Full spectrogram of $a_1(t)$. (b) Zoomed in spectrogram of $a_1(t)$ to see the 2 sources at 250 Hz and 400 Hz. (c) Full spectrogram of $a_2(t)$ (d) Zoomed in spectrogram of $a_2(t)$ to see the 2 sources at 300 Hz and 550 Hz. The 250 Hz signal in $a_1(t)$ is amplitude modulated at the same frequency as the 300 Hz signal in $a_2(t)$.

View	Source	Frequency
$a_1(t)$	250 Hz pure tone 400 Hz pure tone coffee shop noise 1	1 Hz 3 Hz
$a_2(t)$	300 Hz pure tone 550 Hz pure tone coffee shop noise 2	1 Hz 5 Hz

Table 5.1: Summary of the audio sources. The 250 Hz pure tone in Audio 1 is amplitude modulated at the same frequency as the 300 Hz pure tone in Audio 2 and is thus correlated with it.

canonical vector estimates for 3 different frames, corresponding to 1/6 of a second, 1 second, and 20 seconds.

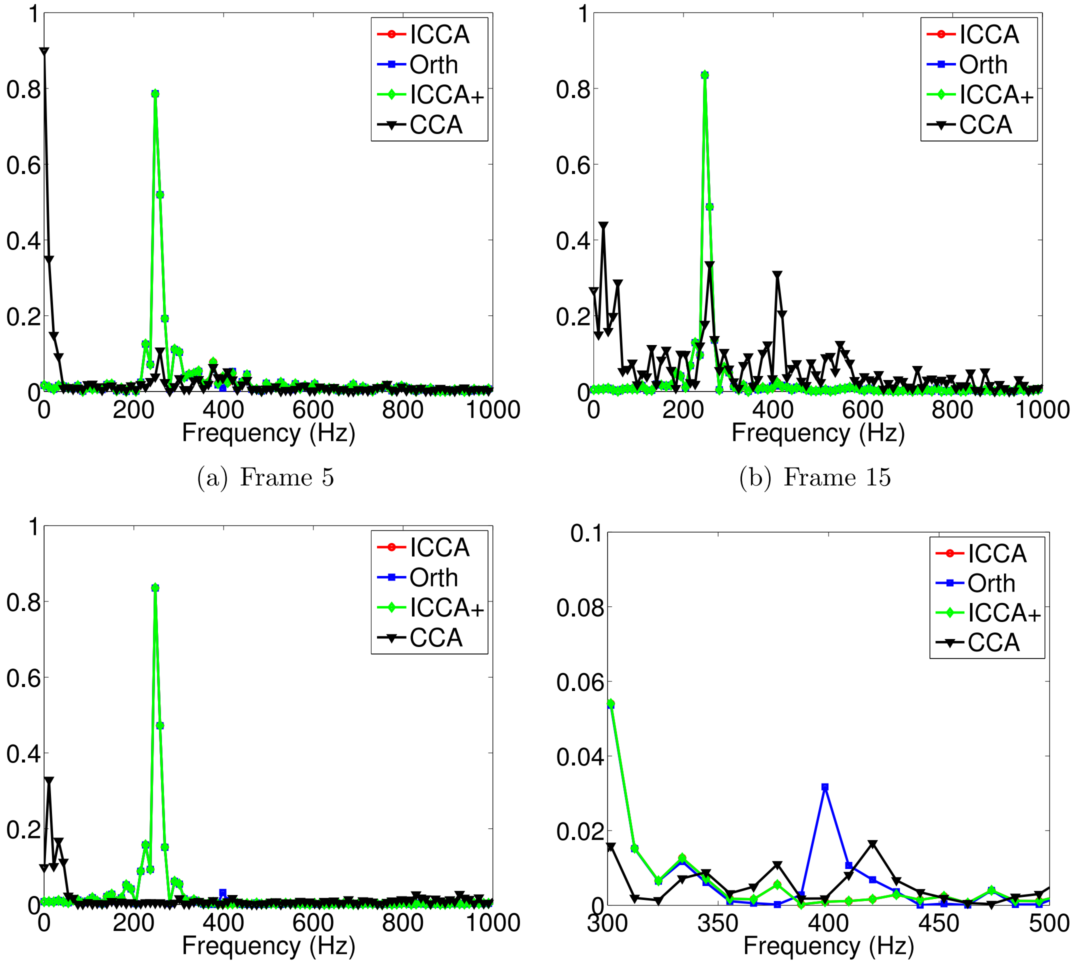
From these figures, we once again see that the empirical CCA estimates are very inaccurate in the low-sample regime, lending credence to Conjecture 5.5.1. We also observe that the ICCA, orthogonal, and ICCA+ estimates are all very similar for this experiment. Each identifies the correlated AM signal at 250 HZ (Figure 5.16) and 300 Hz (Figure 5.17). In each figure, we plot a zoomed in version of canonical vector estimates at the independent AM frequencies of 400 Hz (Figure 5.16(d)) and 550 Hz (Figure 5.17(d)). In both figures we observe a slight difference in the orthogonal estimate. In both cases, it places a larger weight on this independent AM signal than the ICCA and ICCA+ estimates. This is not desirable as this signal is not correlated across the audio streams.

Therefore, in this application, the orthogonal estimate performs the worst while the ICCA and ICCA+ estimates perform equally. This is due to the fact that the principle components for the audio streams contain frequency components from both present signals (see Figure 4.32). Therefore, $U_{\widetilde{K}}$ is not identity, and we empirically see the sub-optimality of the orthogonal estimate that we theoretically predicted.

5.9 Proof of Theorem 5.5.1, Theorem 5.5.2, Corollary 5.5.1, and Theorem 5.5.4

We will prove Theorem 5.5.1 for λ_x^{opt} and by a similar argument assume the result for λ_y^{opt}. By the unitary invariance of the Frobenius norm, we have

$$\|W_x - \widehat{U}_x \,\mathbf{diag}(\lambda_x)\widehat{U}_{\widetilde{K}}\|_F = \|\widehat{U}_x^H W_x \widehat{U}_{\widetilde{K}}^H - \mathbf{diag}(\lambda_x)\|_F.$$



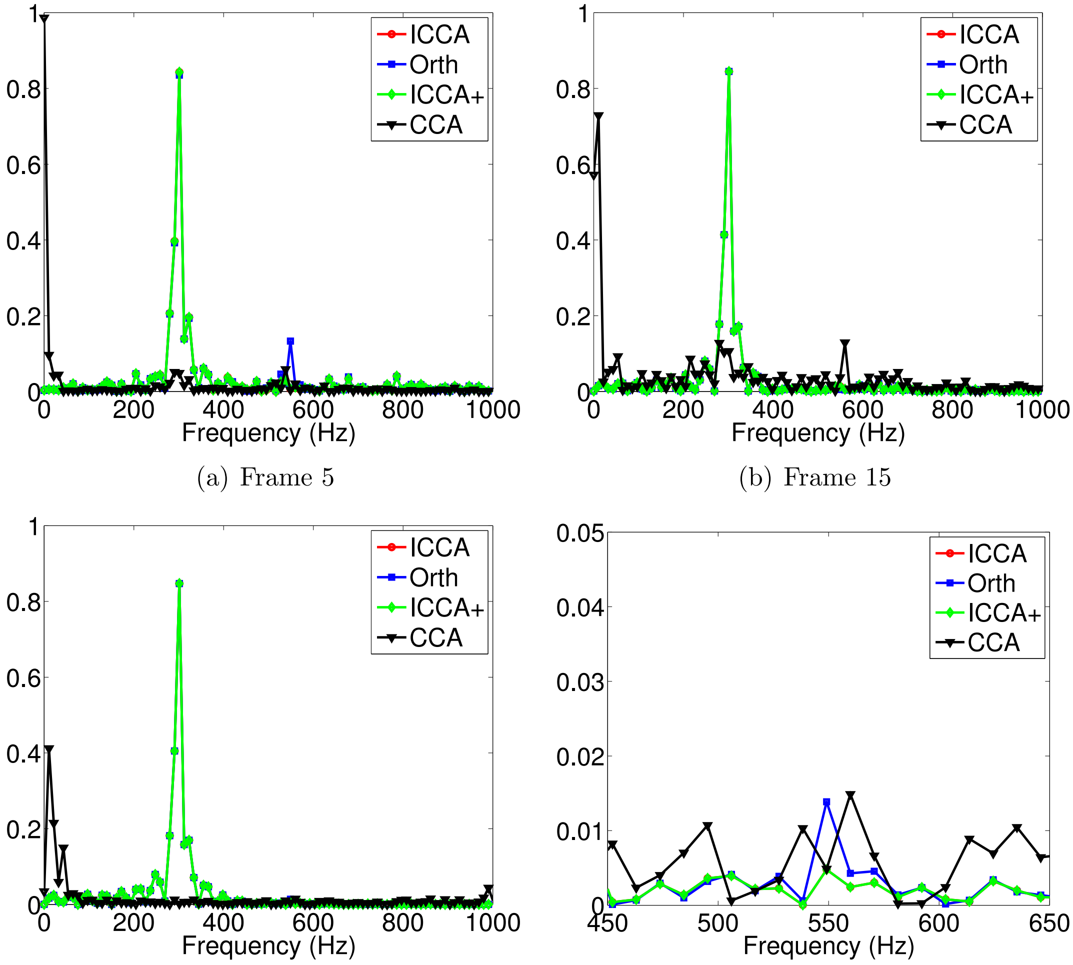
(a) Frame 5

(b) Frame 15

(c) Frame 300

(d) Frame 300 - Zoomed

Figure 5.16: Canonical vectors estimates for the first audio stream at 3 different frames. One frame corresponds to 1/15 seconds. Frequencies with large weights are those that the algorithms mark as correlated with frequencies with large weights in Figure 5.17.



(a) Frame 5

(b) Frame 15

(c) Frame 300

(d) Frame 300 - Zoomed

Figure 5.17: Canonical vectors estimates for the second audio stream at 3 different frames. One frame corresponds to 1/15 seconds. Frequencies with large weights are those that the algorithms mark as correlated with frequencies with large weights in Figure 5.16.

Substituting the definition of the population canonical vectors in (5.8), we have

$$\|\widehat{U}_x^H W_x \widehat{U}_{\widetilde{K}}^H - \mathbf{diag}(\lambda_x)\|_F = \|\widehat{U}_x^H U_x \left(\Theta_x + I_{k_x}\right)^{-1/2} U_{\widetilde{K}} \widehat{U}_{\widetilde{K}}^H - \mathbf{diag}(\lambda_x)\|_F.$$

Let $A = \widehat{U}_x^H U_x \left(\Theta_x + I_{k_x}\right)^{-1/2} U_{\widetilde{K}} \widehat{U}_{\widetilde{K}}^H$. By assumption, we have that $\widehat{U}_{\widetilde{K}}$ is a consistent estimator of $U_{\widetilde{K}}$ so that $U_{\widetilde{K}} \widehat{U}_{\widetilde{K}}^H = I_{k_x}$. Therefore $A = \widehat{U}_x^H U_x \left(\Theta_x + I_{k_x}\right)^{-1/2}$. Therefore, our optimization problem is

$$\lambda_x^{\text{opt}} = \underset{\lambda_x}{\operatorname{argmin}} \|A - \mathbf{diag}(\lambda_x)\|_F.$$

By Lemma 4.1 and Corollary 4.1 from [117], we have that

$$\lambda_x^{\text{opt}} = \mathbf{diag}(A),$$

which completes the proof of Theorem 5.5.1.

We next turn toward Theorem 5.5.2. First, Theorem 5.5.2 b) follows immediately from Theorem 2.9 in [116]. To prove part a), we note that by Theorem 5.5.1, we have that

$$\lambda_{x,\text{opt}}^{(i)} = A_{ii},$$

where A is defined above. Examining the diagonal entries of this matrix, we have

$$A_{ii} = \frac{\widehat{u}_x^{(i)H} u_x^{(i)}}{\sqrt{\left(\theta_i^{(x)}\right)^2 + 1}}.$$

To complete the proof, we must characterize the limiting behavior of these quantities. Let

$$\sigma_x^{(i)} = D_{\mu_{Z_x}}^{-1}\left(1/\left(\theta_1^{(x)}\right)^2\right). \tag{5.20}$$

Theorem 2.10 a) of [116] showed that

$$\left|\langle \widehat{u}_x^{(i)}, u_x^{(i)} \rangle\right|^2 \xrightarrow{\text{a.s.}} \frac{-2\varphi_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)}{\left(\theta_i^{(x)}\right)^2 D'_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)},$$

where for any probability measure μ,

$$\varphi_\mu(z) = \int \frac{z}{z^2 - t^2} d\mu(t).$$

We note that there is a ambiguity in the sign (or phase, when complex value) of these singular vectors. While we use the consistency assumption to get $U_{\widetilde{K}}\widehat{U}_{\widetilde{K}}^H = I_{k_x}$, we note that the sign of $\widehat{U}_x$ is coupled with the sign of $\widehat{U}_{\widetilde{K}}$ and so we may take the positive square root of the above expression. Finally, we know that by definition in (5.20),

$$1/\left(\theta_u^{(x)}\right)^2 = D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right).$$

Substituting these expressions into the diagonal elements of A, we arrive at our theorem conclusion,

$$\begin{aligned}\lambda_{x,\text{opt}}^{(i)} &\xrightarrow{\text{a.s.}} \frac{\sqrt{\frac{-2D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)\varphi_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)}{D'_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)}}}{\sqrt{\frac{1}{D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)}+1}} \\ &= D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)\sqrt{\frac{-2\varphi_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)}{D'_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)\left(D_{\mu_{Z_x}}\left(\sigma_x^{(i)}\right)+1\right)}}\end{aligned}$$

A similar argument proves the result for $\lambda_{y,\text{opt}}^{(i)}$.

Next, we prove Corollary 5.5.1 by providing the explicit forms of the D transform and its derivative when we have Gaussian data as in (5.1). From [84, 107], we have that

$$\left|\langle\widehat{u}_x^{(i)}, u_x^{(i)}\rangle\right|^2 \xrightarrow{\text{a.s.}} \begin{cases}\frac{\left(\theta_i^{(x)}\right)^4 - c_x}{\left(\theta_1^{(x)}\right)^4 + \left(\theta_i^{(x)}\right)^2 c_x} & \text{if } \theta_i^{(x)} > c_x^{1/4} \\ 0 & \text{o.w.}\end{cases}, \tag{5.21}$$

and

$$\left(\widehat{\theta}_i^{(x)}\right)^2 \xrightarrow{\text{a.s.}} \begin{cases}\left(\theta_i^{(x)}\right)^2 + c_x + \frac{c_x}{\left(\theta_i^{(x)}\right)^2} & \text{if } \theta_i^{(x)} > c_x^{1/4} \\ c_x + 2\sqrt{c_x} & \text{o.w.}\end{cases}. \tag{5.22}$$

Substituting these expressions into A_{ii} and the plug-in weights and performing some minor algebra yields the result. A similar argument proves the result for the y weights.

Finally, we prove Theorem 5.5.4. To prove this theorem, we follow the same steps as the proof for Theorem 4.7.1 of Chapter 4. We omit the steps here to save space as they are exactly the same. The main result of these proof steps is that we replace Θ_x with $\gamma_x\Theta_x$ and Θ_y with $\gamma_y\Theta_y$. We additionally notes that when we are below the phase transition, this analysis shows that

$$\begin{aligned} X &\to \gamma_x Z_x \\ Y &\to \gamma_y Z_y \end{aligned}$$

and so the estimates of $\theta_i^{(x)}$ and $\theta_i^{(y)}$ below the phase transition change by a factor of γ_x and γ_y, respectively. Making these substitutions in Corollary 5.5.1 yields the expressions in Theorem 5.5.4.

5.10 Proof of Theorem 5.5.3, Corollary 5.5.2, and Theorem 5.6.1

We begin with our definition of accuracy in (5.16)

$$\mathrm{ACC}^{(i)}(\lambda) = \frac{\left(U_{\widetilde{K}}^{(i)H}\left(\Theta_x + I_{k_x}\right)^{-1/2} U_x^H \mathring{U}_x \Lambda \widehat{U}_{\widetilde{K}}^{(i)}\right)^2}{\left(U_{\widetilde{K}}^{(i)H}\left(\Theta_x + I_{k_x}\right)^{-1} U_{\widetilde{K}}^{(i)}\right)\left(\widehat{U}_{\widetilde{K}}^{(i)H}\Lambda^2 \widehat{U}_{\widetilde{K}}^{(i)}\right)}.$$

Using the assumption that $\widehat{U}_{\widetilde{K}}$ is a consistent estimator of $U_{\widetilde{K}}$, we have that in the considered asymptotic regime, $\widehat{U}_{\widetilde{K}} \to \widehat{U}_{\widetilde{K}}$. Therefore, the numerator of the above expression becomes

$$\left[\sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \langle u_x^{(j)}, \widehat{u}_x^{(j)}\rangle \lambda_j}{\sqrt{\left(\theta_j^{(x)}\right)^2 + 1}} + \sum_{j\neq\ell}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j \left(U_{\widetilde{K}}^{(i)}\right)_\ell \langle u_x^{(j)}, \widehat{u}_x^{(\ell)}\rangle \lambda_\ell}{\sqrt{\left(\theta_j^{(x)}\right)^2 + 1}}\right]^2.$$

In [116], it was shown that for $j \neq \ell$ when $\left(\theta_i^{(x)}\right)^2 > 1/D_{\mu_{Z_x}}(b_x)$ that $\langle u_x^{(j)}, \widehat{u}_x^{(\ell)}\rangle \xrightarrow{\text{a.s.}} 0$. Therefore, the numerator becomes

$$\left[\sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \langle u_x^{(j)}, \widehat{u}_x^{(j)}\rangle \lambda_j}{\sqrt{\left(\theta_j^{(x)}\right)^2 + 1}}\right]^2.$$

In this regime, we have that from our Proof of Theorem 5.5.1

$$\langle u_x^{(j)}, \widehat{u}_x^{(j)}\rangle = \alpha_j \xrightarrow{\text{a.s.}} \sqrt{\frac{-2\phi_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right) D_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}{D'_{\mu_{Z_x}}\left(\sigma_x^{(j)}\right)}}. \tag{5.23}$$

Again using the assumption that $\widehat{U}_{\widetilde{K}}$ is consistent, we have that terms in the denominator are

$$U_{\widetilde{K}}^{(i)H}\left(\Theta_x + I_{k_x}\right)^{-1} U_{\widetilde{K}}^{(i)} = \sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2}{\sqrt{\left(\theta_j^{(x)}\right)^2 + 1}}$$

$$\widehat{U}_{\widetilde{K}}^{(i)H}\Lambda^2 \widehat{U}_{\widetilde{K}}^{(i)} = \sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \lambda_j^2.$$

Combining all of these terms we have that

$$\mathrm{ACC}^{(i)}(\lambda) \xrightarrow{\text{a.s.}} \frac{\left(\sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \alpha_j \lambda_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)\left(\sum_{j=1}^{k_x}\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \lambda_j^2\right)}.$$

Theorem 5.5.3 a) and b) immediately follow from substituting the limiting values of $\lambda_{x,\text{opt}}$ and $\lambda_{x,\text{icca}}$ given in Theorem 5.5.1. Analogous expressions for the accuracy of the canonical vectors for Y may be derived in a similar fashion.

To prove Corollary 5.5.2, we use (5.21) and (5.22) to substitute the necessary quantities such as α. This gives the general form for arbitrary λ. We then may substitute the closed form expressions derived for Corollary 5.5.1 to complete the proof. Specifically, we note that we can write

$$\lambda_{x,\text{opt}}^{(j)} = \frac{\alpha_j}{\sqrt{\left(\theta_i^{(x)}\right)^2 + 1}},$$

and

$$\lambda_{x,\text{icca}}^{(j)} = \frac{\theta_j^{(x)}}{\sqrt{\left(\left(\theta_i^{(x)}\right)^2 + 1\right)\left(c + \left(\theta_j^{(x)}\right)^2\right)}},$$

which simplifies the expressions for $\mathrm{ACC}^{(i)}(\lambda_{x,\mathrm{opt}})$ and $\mathrm{ACC}^{(i)}(\lambda_{x,\mathrm{icca}})$.

Finally, we prove Theorem 5.6.1. This proof is straightforward by noting that the weights of the orthogonal approximation are $\lambda_j = 1$. Substituting this into the result from Theorem 5.5.3 we get

$$\mathrm{ACC}^{(i)}(\lambda_{x,\mathrm{orth}}) \xrightarrow{\text{a.s.}} \frac{\left(\sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2 \alpha_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x} \frac{\left(U_{\widetilde{K}}^{(i)}\right)_j^2}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}}\right)\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2\right)}$$

where α_j is either the general form in (5.23) or the specific form by taking the square root of the expression in (5.21). Again, we note that the derivation may be repeated to obtain the accuracy for the canonical vectors associated with the Y dataset.

CHAPTER VI

The Top Singular Values of XY^H

6.1 Introduction

Correlation analysis is a ubiquitous problem in statistical signal processing. In many applications, we have access to multiple datasets each using a different feature space to describe a system. In image annotation [34] and image retrieval [28] we assume that both image features and textual captions describe the depicted scene. In speaker identification, we assume that the video of a speaker is correlated with his/her audio [35]. In medical signal processing, we assume that different modalities such as EEG, MEG, MRI, fMRI, and genetic data (SNPs, QTs) capture a shared signal of interest in a patient [36, 40, 119, 44, 120, 19, 47, 121, 48]. With the increasing ability to collect such a variety of data, multimodal datasets even make appearances in non-classical statistical signal processing applications such as economics [58, 122], climatology [60, 61], and psychology [123, 124].

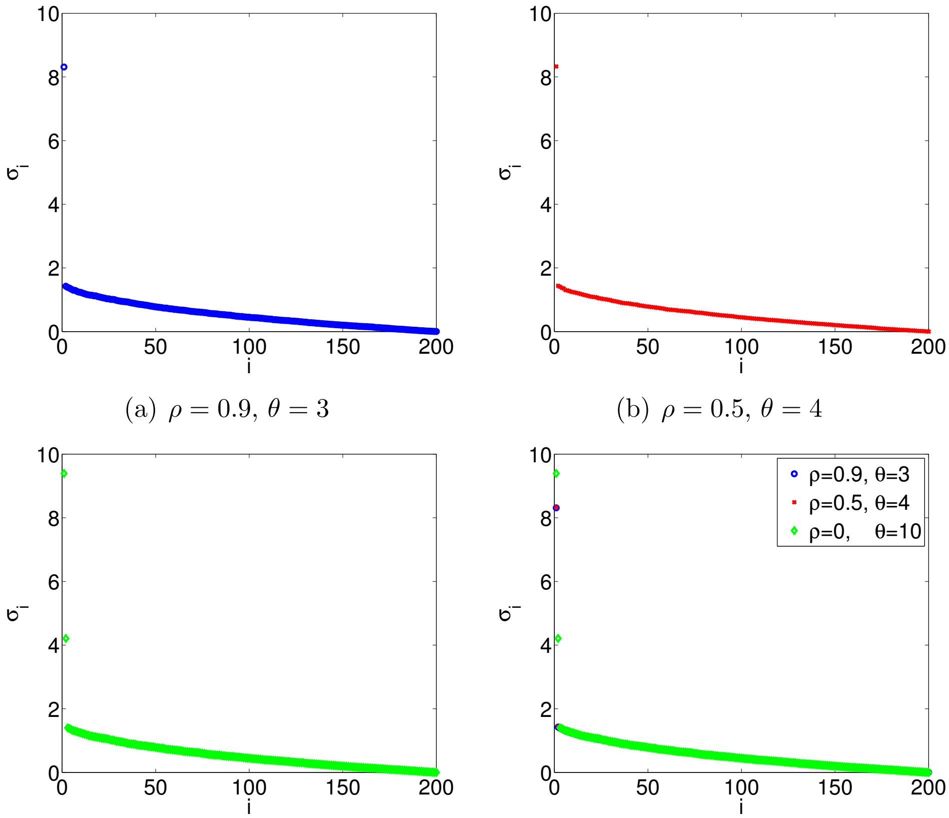
In many of these applications, we model observations using a low rank signal-plus-noise model. This model assumes that observations from a dataset lie in an unknown low-rank subspace and that the signal vectors between datasets are correlated. We obtain estimates of the unknown signal subspace and signal-to-noise ratios (SNR) of signals in a particular dataset via the eigenvalue decomposition of its sample covariance matrix. The accuracy of this eigenvalue decomposition has been extensively studied [84, 85] and applied to applications such as matched subspace detection [107]. This analysis has been extended to examine the accuracy of the singular values and singular vectors of the original rectangular data matrix [116].

When given multiple datasets, one hopes to leverage the correlations existing between the datasets. Canonical Correlation Analysis (CCA) is a dimensionality reduction algorithm for exactly two datasets that finds a linear transformation for each dataset such that the datasets are maximally correlated in their reduced dimensional representations [4]. These linear transformations are found through a SVD of a matrix product involving each dataset's covariance matrix and the cross covariance matrix. However, these covariance matrices are typically unknown and estimated from data. When the number of training samples is relatively small compared to the dimension of the datasets, the correlations and linear transformations returned by empirical CCA are very inaccurate [6, 8]. We are interested in this sample deficient regime that is becoming increasingly present with increased dimensions of datasets.

In this chapter, we begin by exploring the performance of regularized CCA (RCCA). RCCA is a common variant of CCA that adds a multiple of the identity to each sample covariance matrix to better condition these matrices in the sample deficient regime. We explore the empirical performance of RCCA in the signal-plus-noise model and observe that the performance of RCCA improves with increased regularization parameter. We then prove that, when setting the regularization parameter to infinity, the solution of RCCA may be found by simply taking the SVD of the sample cross covariance matrix between the two datasets. We name this algorithm limit RCCA (LRCCA). This algorithm is more desirable than RCCA not only because it offers better performance but because it does not have a tunable parameter.

We then use random matrix theory proof techniques to derive the almost sure convergence of the top singular values of the matrix product used in LRCCA. We show the existence of a phase transition below which the largest singular values of LRCCA behave exactly as if the matrices were simply noise, i.e. containing no signal. This critical threshold is dependent on the dimensionality of each dataset, the number of observations, the SNRs of each dataset, and the correlation between the datasets.

The SVD of the sample covariance matrix is often used to determine the presence of correlated signals between datasets. This technique arises in direction of arrival (DOA) [125, 126, 127, 128], nerual network models [129], and brain connectivity analysis using fMRI [130]. We are motivated by Figure 6.1, which plots the singular value spectra of the cross-covariance matrix for three different rank-1 data matrices. Figure 6.1(a) has a high correlation between low-SNR signals; Figure 6.1(b) has a medium correlation between medium SNR signals; Figure 6.1(c) has no correlation between high SNRS signals. We see that in the first two settings, one singular value separates from the bulk and that this singular value is about the same for both settings. In the last setting, two singular values separate from the bulk of the singular values. From these settings, we see the difficulty in using the spectrum of the cross covariance matrix to detect correlations. Our analysis throughout this chapter will explore this in more detail concluding with the observation that it is better to use



(a) $\rho = 0.9$, $\theta = 3$

(b) $\rho = 0.5$, $\theta = 4$

(c) $\rho = 0$, $\theta = 10$

(d) Combined

Figure 6.1: Motivational example of the singular value spectra of $\frac{1}{n}XY^H$ for three different sets of parameters. In all figures $p = q = 200$, $n = 500$, and $\theta = \theta_x = \theta_y$. In the settings in (a) and (b), the singular value spectra are very similar, with one singular value separating from the bulk of the singular values. In the setting in (c) where there is no correlation between the datasets, two singular values separate from the bulk of the singular values.

informative CCA (ICCA), which we presented in Chapter IV.

This chapter is organized as follows. In Section 6.2 we provide the signal-plus-noise data model that we use throughout the chapter and provide the optimization problems used by CCA, RCCA, and ICCA. In Section 6.3, we present our two main theorems describing the performance of LRCCA. We then empirically demonstrate the performance of RCCA as a function of its regularization parameter, showcase the singular value prediction accuracy, and compare LRCCA to ICCA in Section 6.4. We provide the proofs of our main results in Section 6.5.

6.2 Data Model and Background

In this section we provide the data model that we will use throughout the chapter. This model emulates the linear signal-plus-noise model used in many applications. We then provide the optimization problems and solutions to canonical correlation analysis (CCA), regularized CCA (RCCA), informative CCA (ICCA), and limit RCCA (LRCCA).

6.2.1 Data Model

Let $\widetilde{X}_n = [\widetilde{x}_1, \ldots, \widetilde{x}_n]$ and $\widetilde{Y}_n = [\widetilde{y}_1, \ldots, \widetilde{y}_n]$ be two datasets with observations $\widetilde{x}_i \in \mathbb{R}^p$ and $\widetilde{y}_i \in \mathbb{R}^q$. Throughout, we model the datasets as

$$\begin{aligned} \widetilde{X}_n &= U_x \Theta_x V_x^T + X_n, \\ \widetilde{Y}_n &= U_y \Theta_y V_y^T + Y_n \end{aligned} \tag{6.1}$$

where $U_x \in \mathbb{R}^{p\times r}$, $U_y \in \mathbb{R}^{q\times r}$ are independent orthonormal matrices, $V_x \in \mathbb{R}^{n\times r}$ and $V_y \in \mathbb{R}^{n\times r}$ are orthogonal matrices such that $\mathbb{E}\left[V_x^T V_y\right] = P = \mathbf{diag}\left(\rho_1, \ldots, \rho_r\right)$ with $0 \leq \rho_i \leq 1$, and $\Theta_x = \mathbf{diag}(\theta_{x1}, \ldots, \theta_{xr})$ and $\Theta_y = \mathbf{diag}(\theta_{y1}, \ldots, \theta_{yr})$. We denote p as the dimension of the first dataset, q as the dimension of the second dataset, n as the number of training samples, and r as the maximum number of signals in either dataset. $X_n \in \mathbb{R}^{p\times n}$ and $Y_n \in \mathbb{R}^{p\times n}$ model the system noise and are assumed to be independent. In this regard, our model accounts for different dimensional signal subspaces in $\widetilde{X}$ and $\widetilde{Y}$ by setting the appropriate θ and ρ to zero. Finally, define $\widehat{R}_{xx} = \frac{1}{n}\widetilde{X}_n\widetilde{X}_n^T$, $\widehat{R}_{yy} = \frac{1}{n}\widetilde{Y}_n\widetilde{Y}_n^T$, and $\widehat{R}_{xy} = \frac{1}{n}\widetilde{X}_n\widetilde{Y}_n^T$ as the sample covariance matrices.

6.2.2 Empirical CCA

The goal of CCA is to find a linear transformation for each dataset that maximizes the correlation between the datasets in the projected spaces. We represent the linear transformations with the canonical vectors $w_x \in \mathbb{R}^{p\times 1}$ and $w_y \in \mathbb{R}^{q\times 1}$ and the projection with the canonical variates $z_x = w_x^H x$ and $z_y = w_y^H y$. The objective is to find the canonical vectors w_x and w_y that maximize the correlation between the canonical variates z_x and z_y. Formally, the optimization problem is

$$\begin{aligned} \underset{w_x, w_y}{\text{argmax}} \quad & \rho = \mathbb{E}\left[z_x z_y\right] \\ \text{subject to} \quad & \mathbb{E}\left[z_x^2\right] = 1, \mathbb{E}\left[z_y^2\right] = 1. \end{aligned} \tag{6.2}$$

We may obtain a closed form solution for (6.2) through the SVD of

$$\widehat{C}_{\text{cca}} = \widehat{R}_{xx}^{-1/2}\widehat{R}_{xy}\widehat{R}_{yy}^{-H/2},$$

which relies on estimates of the unknown covariance matrices. Let FKG^H be the SVD of $\widehat{C}_{\text{cca}}$ where $F = [f_1, \ldots, f_p]$, $K = \mathbf{diag}(k_1, \ldots, k_{\min(p,q)})$, and $G = [g_1, \ldots, g_q]$. Then the solution for the canonical vector pair corresponding to the largest canonical correlation is

$$\begin{aligned} \rho &= k_1 \\ w_x &= \widehat{R}_{xx}^{-1/2} f_1 \\ w_y &= \widehat{R}_{yy}^{-1/2} g_1. \end{aligned} \tag{6.3}$$

To find higher order canonical vector and correlation pairs we take successive singular vector and value pairs of $\widehat{C}_{\text{cca}}$.

6.2.3 RCCA

When $n < p+q$, CCA reports a perfect correlation of $\rho = 1$ regardless of the true correlation [6]. To overcome this performance loss, RCCA introduces a regularization parameter, η, that adds a multiple of the identity matrix to the sample covariance matrix of each dataset. Formally, the RCCA optimization problem is

$$\begin{aligned} \underset{w_x, w_y}{\text{argmax}} \quad & \rho = E[z_x z_y] \\ \text{subject to} \quad & E[z_x^2] + \eta w_x^H w_x \leq 1 \\ & E[z_y^2] + \eta w_y^H w_y \leq 1. \end{aligned} \tag{6.4}$$

We solve (6.4) by taking the SVD of $C_{\text{reg}} = \left(\widehat{R}_{xx} + \eta I_p\right)^{-1/2} \widehat{R}_{xy} \left(\widehat{R}_{yy} + \eta I_q\right)^{-1/2}$. Let FKG^H be the SVD of C_{reg} where $F = [f_1, \ldots, f_p]$, $K = \mathbf{diag}(k_1, \ldots, k_{\min(p,q)})$, and $G = [g_1, \ldots, g_q]$. The solution to RCCA is

$$\begin{aligned} \rho &= k_1 \\ w_x &= (\widehat{R}_{xx} + \eta I_p)^{-1/2} f_1 \\ w_y &= (\widehat{R}_{yy} + \eta I_q)^{-1/2} g_1. \end{aligned} \tag{6.5}$$

Higher order canonical vector and correlation pairs are again computed using successive singular value and vector pairs of C_{reg}.

6.2.4 ICCA

We repeat the informative CCA (ICCA) algorithm presented in Chapter IV, first proposed by Nadakuditi [8]. ICCA can avoid the performance loss in the sample deficient regime by first trimming the individual data matrices to only include informative subspace components. Let $\widetilde{X} = F_x K_x G_x^T$ and $\widetilde{Y} = F_y K_y G_y^T$ be the data SVDs for our data matrices. Define the trimmed data matrices

$$\begin{aligned} \widetilde{F}_x &= F_x(:, 1 : r_x) \quad \widetilde{G}_x = G_x(:, 1 : r_x) \\ \widetilde{F}_y &= F_y(:, 1 : r_y) \quad \widetilde{G}_y = G_y(:, 1 : r_y) \end{aligned}$$

where r_x and r_y are the number of informative components in the first and second datasets, respectively. To determine the number of informative components one may employ techniques in [107, 83]. Using these trimmed data matrices, we form the matrix used for ICCA,

$$\widetilde{C} = \widetilde{F}_x \widetilde{G}_x^T \widetilde{G}_y \widetilde{F}_y^T, \tag{6.6}$$

with SVD $\widetilde{C} = \widetilde{F}\widetilde{K}\widetilde{G}^H$, where $\widetilde{F} = [\widetilde{f}_1, \ldots, \widetilde{f}_{r_x}]$, $\widetilde{K} = \mathbf{diag}(\widetilde{k}_1, \ldots, \widetilde{k}_{\min(r_x, r_y)})$, and $\widetilde{G} = [\widetilde{g}_1, \ldots, \widetilde{g}_{r_y}]$. ICCA returns the following informative correlation estimate and canonical vectors

$$\begin{aligned} \rho &= \widetilde{k}_1 \\ w_x &= \widehat{R}_{xx}^{-1/2} \widetilde{f}_1 \\ w_y &= \widehat{R}_{yy}^{-1/2} \widetilde{g}_1 \end{aligned} \tag{6.7}$$

Higher order canonical vector and correlation pairs are computed using successive singular value and vector pairs of $\widetilde{C}$.

6.3 Main Results

Figure 6.2 shows the empirical performance of RCCA for various regularization parameters. Evident in this figure, we observe that increasing the regularization parameter increases the performance of RCCA. The following theorem gives the solution of RCCA when taking $\eta \to \infty$.

Theorem 6.3.1. *Let $\widetilde{X}_n$ and $\widetilde{Y}_n$ be modeled as in (6.1). Let $C_{lrcca} = \frac{1}{n}\widetilde{X}_n \widetilde{Y}_n^T$ have SVD FKG^T where $F = [f_1, \ldots, f_p]$, $K = \mathbf{diag}(k_1, \ldots, k_{\min(p,q)})$, and $G = [g_1, \ldots, g_q]$.*

When $\eta \to \infty$, the solution to the RCCA optimization problem in (6.4) is

$$\begin{aligned} \rho &\propto k_1 \\ x_1 &\propto f_1 \\ x_2 &\propto g_1. \end{aligned} \tag{6.8}$$

Proof. See Section 6.5.1. □

We call the above algorithm limit RCCA (LRCCA), which is preferred over RCCA as it both offers better performance and has no tuning parameter. Next we characterize the asymptotic limit of the top singular values of C_{lrcca}.

Theorem 6.3.2. *Let $\widetilde{X}_n$ and $\widetilde{Y}_n$ be modeled as in (6.1) and define $C_n = \frac{1}{n}\widetilde{X}_n\widetilde{Y}_n^T$. Let $p \to \infty$, $q \to \infty$, and $n \to \infty$ such that $\frac{p}{n} \to c_x$ and $\frac{q}{n} \to c_y$. Given the noise matrices X_n and Y_n, define $R_n = \frac{1}{n}X_n^TX_n$ and $S_n = \frac{1}{n}Y_n^TY_n$. Let μ_{R_n} and μ_{S_n} be the respective empirical eigenvalue distributions and assume that each converges almost surely weakly, as $n, p, q \to \infty$ as above, to the non-random compactly supported probability measures μ_R and μ_S, respectively. Similarly, let $M_1 = \frac{1}{n^2}X_nY_n^TY_nX_n^T$, $M_2 = \frac{1}{n^2}Y_nX_n^TX_nY_n^T$, and $M_3 = M_1\left(\sigma_i^2 - M_1\right)^{-1}$ have limiting eigenvalue distributions μ_{M_1}, μ_{M_2}, and μ_{M_3} respectively. For $i = 1, \ldots, r$, let σ_i be the larest singular values of C_n. Then, almost surely, σ_i are the solutions to the following equation*

$$0 = \prod_{i=1}^r \left(\varphi_H(\sigma_i)\varphi_F(\sigma_i) - \frac{1}{\theta_{yi}^2}\right)\left(\varphi_J(\sigma_i)\varphi_G(\sigma_i) - \frac{1}{\theta_{xi}^2}\right) - \rho_i^2\varphi_H(\sigma_i)\varphi_G(\sigma_i)\left(1 + \varphi_K(\sigma_i)\right)^2 \tag{6.9}$$

where

$$\begin{aligned} \varphi_F(\sigma_i) &= -\sigma_i\mathbb{E}\left[xm_{\mu_{RS|R}}\left(\sigma_i^2, x\right)\right]_{\mu_R} \\ \varphi_J(\sigma_i) &= -\sigma_i\mathbb{E}\left[xm_{\mu_{RS|S}}\left(\sigma_i^2, x\right)\right]_{\mu_S} \\ \varphi_G(\sigma_i) &= -\sigma_i m_{\mu_{M_1}}(\sigma_i^2) \\ \varphi_H(\sigma_i) &= -\sigma_i m_{\mu_{M_2}}(\sigma_i^2) \\ \varphi_K(\sigma_i) &= c_x\mathbb{E}\left[x\right]_{\mu_{M_3}} \end{aligned}$$

and

$$m_{\mu_M}(z) = \int \frac{1}{t - z} d\mu_M(t)$$

is the Stieltjes transform of μ_M and

$$m_{\mu_{XY|X}}(x, y) = \int \frac{1}{y - z} k_{XY|X}(x, z) dz$$

where $k_{XY|X}$ is the Markov transition kernel density function.

Proof. See Section 6.5.2. □

6.4 Empirical Simulations

In this section we first motivate the need for LRCCA by exploring the performance of RCCA for various regularization parameters. We then explore the accuracy of Theorem 6.3.2. While this theorem gives an asymptotic limit, we show that the finite-sized approximation holds for moderately sized systems.

For all of the following simulations, we generate correlated signal datasets by

$$\begin{aligned}\widetilde{X}^{\text{signal}} &= \left[\widetilde{x}_1^{\text{signal}}, \ldots, \widetilde{x}_n^{\text{signal}}\right]\\ \widetilde{Y}^{\text{signal}} &= \left[\widetilde{x}_1^{\text{signal}}, \ldots, \widetilde{y}_n^{\text{signal}}\right]\end{aligned}$$

where

$$\begin{aligned}\widetilde{x}_i^{\text{signal}} &= U_x \Theta_x v_x^{(i)} + x_i\\ \widetilde{y}_i^{\text{signal}} &= U_y \Theta_y v_y^{(i)} + y_i,\end{aligned}$$

where $x_i \sim \mathcal{N}(0, I_p)$ and $y_i \sim \mathcal{N}(0, I_q)$ and

$$\begin{bmatrix} v_x^{(i)} \\ v_y^{(i)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} I_r & P \\ P^T & I_r \end{bmatrix}\right).$$

$P = \mathbf{diag}(\rho_1, \ldots, \rho_r)$ with $0 \leq \rho_i \leq 1$ and $\Theta_x = \mathbf{diag}(\theta_{x1}, \ldots, \theta_{xr})$ and $\Theta_y = \mathbf{diag}(\theta_{y1}, \ldots, \theta_{yr})$ with $\theta_{xi} \geq 0$ and $\theta_{yi} \geq 0$. We generate U_x by taking the eigenvectors corresponding to the top r eigenvalues of a random $p \times p$ matrix with $\mathcal{N}(0, 1)$ entries. We generate U_y independently in a similar manner.

We then generate noise only datasets

$$\begin{aligned}\widetilde{X}^{\text{noise}} &= \left[\widetilde{x}_1^{\text{noise}}, \ldots, \widetilde{x}_n^{\text{noise}}\right]\\ \widetilde{Y}^{\text{noise}} &= \left[\widetilde{x}_1^{\text{noise}}, \ldots, \widetilde{y}_n^{\text{noise}}\right]\end{aligned}$$

where

$$\begin{aligned}\widetilde{x}_i^{\text{noise}} &\sim \mathcal{N}(0, I_p)\\ \widetilde{y}_i^{\text{noise}} &\sim \mathcal{N}(0, I_q).\end{aligned}$$

6.4.1 Performance of RCCA

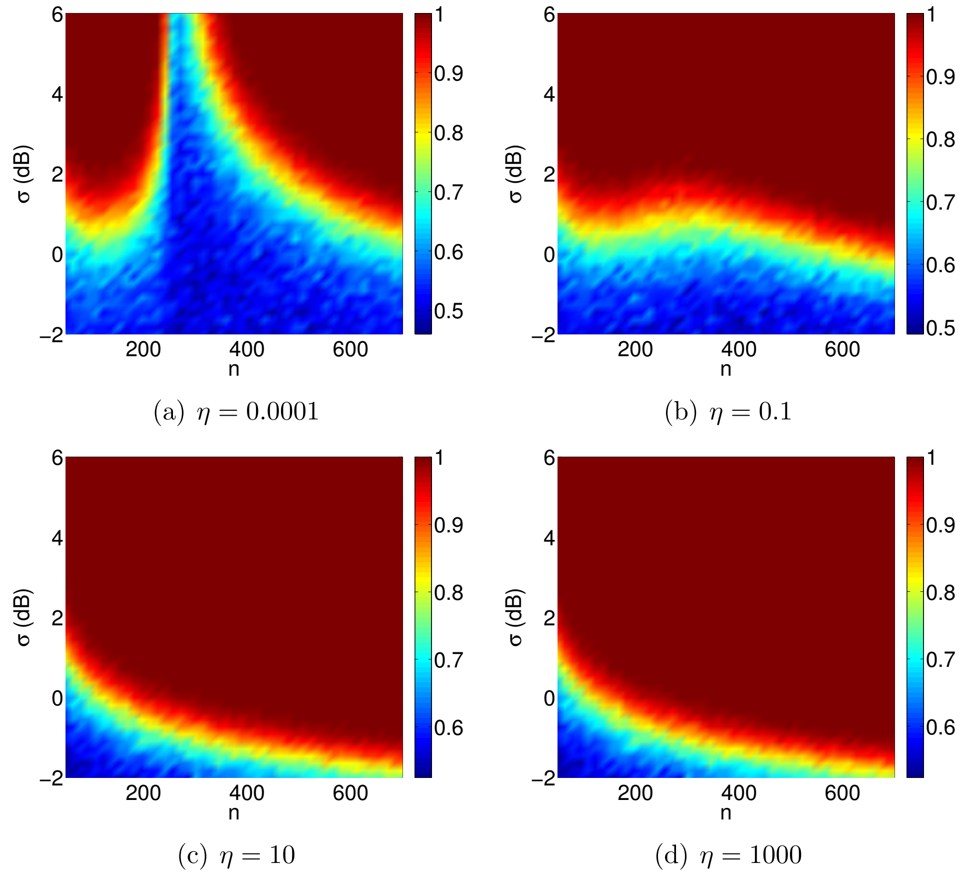
First we explore the effect of the regularization parameter in RCCA. For the above simulation setup we generate both correlated signal data matrices and with $r = 1$ and noise only data matrices. We are interested the distribution of the correlation estimate, $\widehat{\rho}$, returned by RCCA when there is a correlation present ($\widetilde{X}^{\text{signal}}$ and $\widetilde{Y}^{\text{signal}}$), and when there is no correlation present ($\widetilde{X}^{\text{noise}}$ and $\widetilde{Y}^{\text{noise}}$).

For a fixed $p = 100$, $q = 150$, and $\rho_1 = 0.9$ we compute this RCCA correlation estimate under each hypothesis for 500 trials giving $\left[\widehat{\rho}_1^{\text{signal}}, \ldots, \widehat{\rho}_{500}^{\text{signal}}\right]$ and $\left[\widehat{\rho}_1^{\text{noise}}, \ldots, \widehat{\rho}_{500}^{\text{noise}}\right]$. We then compute the empirical ROC (receiver operating characteristic) curve for these two statistics and the resulting AUC (area under the ROC curve). We repeat this process by varying $\theta = \theta_{x1} = \theta_{y1}$ and n. We plot AUC heatmaps for four different values of the RCCA regularization parameter in Figure 6.2. AUC values close to 0.5 indicate the distributions of $\widehat{\rho}^{\text{signal}}$ and $\widehat{\rho}^{\text{noise}}$ are not separable while values close to 1 indicate that they are perfectly separable.

As evident in this figure, the ability of RCCA to detect the presence of a signal increases with the regularization parameter. This is a non-intuitive result as typical regularized algorithms‘ have an optimal regularization parameter that maximizes performance. The non-monotonicity of the AUC heatmaps evident in Figures 6.5(a) and 6.5(b) also give credence to the difficulty in selecting an appropriate regularization parameter. In certain regimes, increasing the number of samples reduces performance, which is a very undesirable property. Based on these empirical observations about the effect of the regularization parameter in RCCA, we conclude that setting $\eta \to \infty$ results in optimal performance of RCCA. As is stated in Theorem 6.3.1, in this regime the solution RCCA is found by simply taking the SVD of $\frac{1}{n}\widetilde{X}\widetilde{Y}^T$.

6.4.2 Numerical Accuracy of Theorem 6.3.2

For $p = 200$, $q = 400$, $n = 400$, and $\rho = 1$ we compute the largest singular value returned by LRCCA for various $\theta = \theta_{x1} = \theta_{y1}$. This is repeated and for 100 trials and compared to the theoretical prediction. Results are shown in Figure 6.3 and confirm the accuracy of our theoretical prediction. As evident in Figure 6.3, if θ is below a critical value, the largest singular value does not change and remains constant. This phase transition phenomenon arises in similar analyses of eigenvalue decomposition and SVDs of signal-plus-noise models [85, 116, 84]. This limiting value is the largest singular value of the noise matrix $\frac{1}{n}XY^T$, which we define as



(a) $\eta = 0.0001$

(b) $\eta = 0.1$

(c) $\eta = 10$

(d) $\eta = 1000$

Figure 6.2: AUC performance of RCCA for various regularization parameters. For all figures, $p = 100$, $q = 150$, $r = 1$, and $\rho_1 = 0.9$. Each figure plots an AUC heatmap while sweeping over $\theta = \theta_{x1} = \theta_{y1}$ and n. AUC points are generated from an ROC formed from 500 points of each distribution. Increasing the regularization parameter increases the performance of CCA. This gives rise to LRCCA, which sets $\eta \to \infty$.

$b = \sigma_1\left(XY^T\right)$. Substituting b into (6.9) we have the following equality

$$0 = \left(\varphi_H(b)\varphi_F(b) - \frac{1}{\theta_{y1}^2}\right)\left(\varphi_J(b)\varphi_G(b) - \frac{1}{\theta_{x1}^2}\right) - \rho\varphi_H(b)\varphi_G(b)\left(1 + \varphi_K(b)\right)^2. \quad (6.10)$$

This equation may be solved for any desired parameter $c_x, c_y, \theta_{x1}, \theta_{y1}, \rho$, while keeping the rest fixed. We note that the φ functions and b are implicitly dependent on c_x and c_y.

Figure 6.4 plots the top singular value returned by LRCCA when the datasets contain $r = 1$ signal each, empirically averaged over 500 trials. Each heatmap sweeps over two parameters while keeping the rest constant. We then solve (6.10) by substituting our constant parameters to achieve a function of the two parameters that we sweep. We overlay this line in each heatmap. Below this line the top singular value

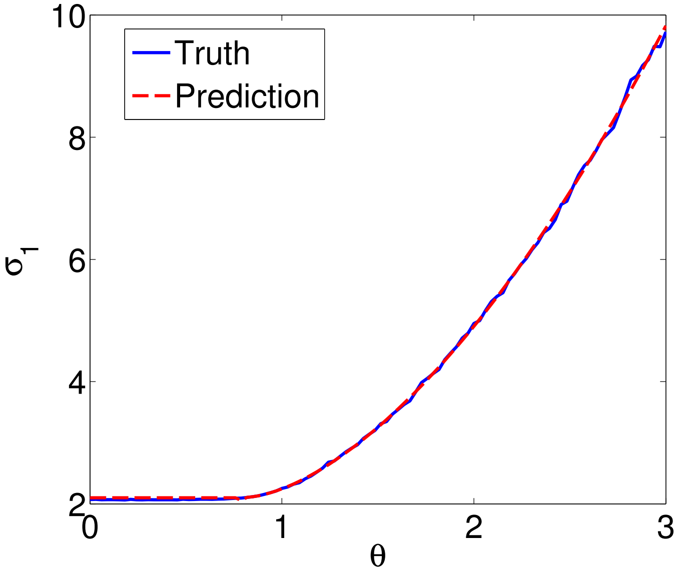


Figure 6.3: Top singular value prediction for the rank-1 case for $p = 200$, $q = 400$, $n = 400$, and $\rho = 1$.

is indistinguishable from that returned by LRCCA with noise only datasets.

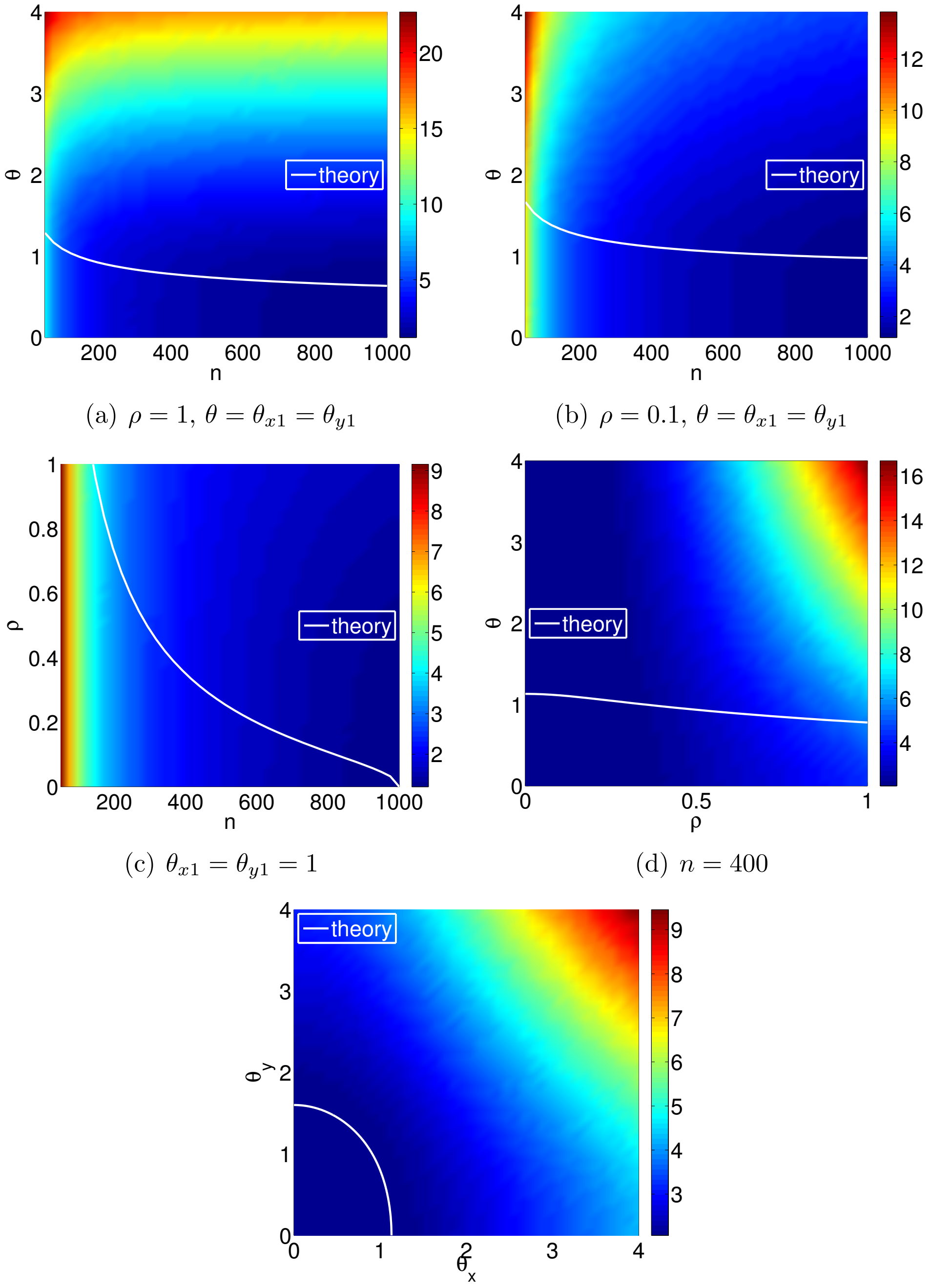
Next we explore the phase transition in Figure 6.4(e) for a fixed $n = 400$ and numerous ρ. Instead of plotting the top singular value returned by LRCCA, we instead plot the log of the KS-statistic between the singular values in the signal bearing case and the singular values in the noise bearing case. A KS statistic of 1 represents perfectly distinct distributions while a KS statistic of 0 represents the same distribution. We plot the results in Figure 6.5 and it is evident that our theoretical phase transition in (6.10), which relies on Theorem 6.3.2, is very accurate even though we apply the asymptotic result to the finite dimensional setting.

6.4.3 Comparison to ICCA

We now compare the performance of LRCCA to that of ICCA. As shown in [8] and presented in Theorem 4.6.2, the correlation coefficient does not affect the performance of ICCA. The consistency phase transition of ICCA for the rank-1 setting is

$$\theta_x > c_x^{1/4} \text{ and } \theta_y > c_y^{1/4}.$$

We plot this phase transition against the phase transitions of LRCCA for a variety of ρ in Figure 6.6.



(a) $\rho = 1$, $\theta = \theta_{x1} = \theta_{y1}$

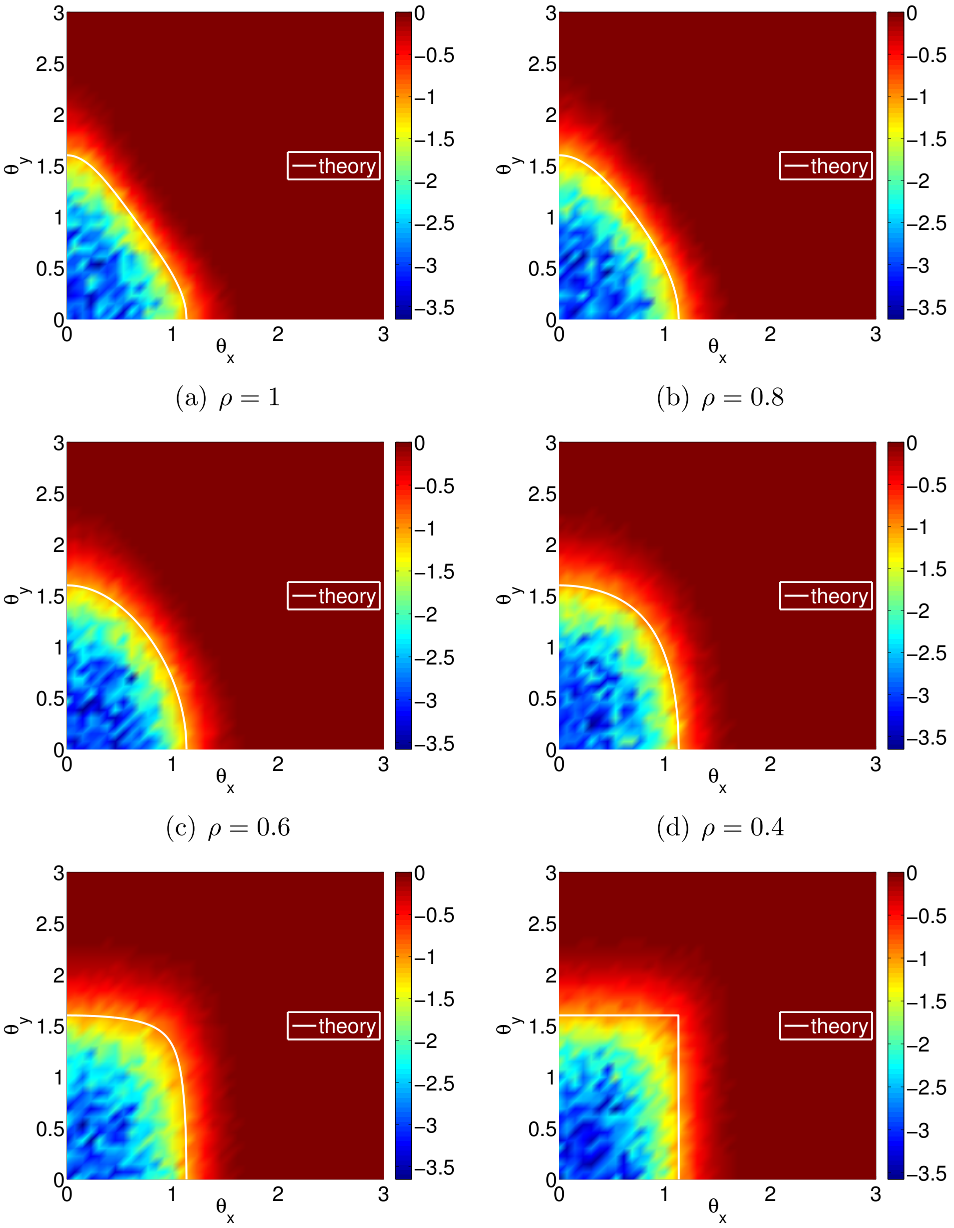
(b) $\rho = 0.1$, $\theta = \theta_{x1} = \theta_{y1}$

(c) $\theta_{x1} = \theta_{y1} = 1$

(d) $n = 400$

(e) $\rho = 0.5$, $n = 400$

Figure 6.4: Top singular value of LRCCA plotted for pairs of parameter sweeps. In all plots, $p = 200$ and $q = 400$. The theoretical boundary where the top singular value is indistinguishable from a noise only setting is plotted for each. Below this line, the top singular value is asymptotically identical to the noise only setting. Above this line, the top singular value is asymptotically different from that of the noise only setting.



(a) $\rho = 1$

(b) $\rho = 0.8$

(c) $\rho = 0.6$

(d) $\rho = 0.4$

(e) $\rho = 0.2$

(f) $\rho = 0$

Figure 6.5: KS statistic between the top singular value of LRCCA in signal bearing and noise only settings. In all plots, $p = 200$, $q = 400$, and $n = 400$. The theoretical boundary where the top singular value is indistinguishable from a noise only setting is plotted for each.

We begin our discussion by comparing what these phase transition boundaries represent for each algorithm. The ICCA phase transition boundary represents when we reliably detect the presence of a correlated signal. Above this boundary, the largest singular value of $\widetilde{C}$ used in ICCA is used to statistically detect the presence of a correlated signal. We direct the reader to Chapter IV for a discussion of this process. However, if either SNR drops below its individual phase transition, ICCA is not able to detect a correlated signal. The LRCCA phase transition boundaries, on the other hand, represent when the largest singular value of C_{lrcca} represents a signal, not necessarily a correlated signal. As we saw in Figure 6.1(c), even uncorrelated datasets will cause the largest singular value to separate from the rest of the singular value. Therefore, these phase transition boundaries represent different boundaries.

One may incorrectly conclude from Figure 6.6 that for LRCCA is superior to ICCA since the boundary of LRCCA includes the regime when θ_x is very small but θ_y is large, and vice versa. However, this singular value only indicates the presence of a signal, not that it is correlated. We saw in Figure 6.1 that different values of θ_x, θ_y, ρ can result in the same largest singular value. Thus, simply using the largest singular value of $\frac{1}{n}XY^H$ to determine whether correlation exists between the dataset is incorrect. As Figure 6.1(c) shows, the cross covariance matrix will have a large singular value even if the individual datasets are independent.

One may then want to use the relative individual SNRs of X and Y to determine whether this leading singular value is large because of correlation or individual large SNRs. However, this process of pre-whitening the data matrices X and Y is exactly the process used in CCA and ICCA. Therefore, to use the cross-covariance matrix XY^H to detect the presence of correlation between the datasets, one would perform the equivalent analysis as CCA, which is suboptimal to ICCA. Therefore, we urge users to reconsider using the cross covariance matrix to screen for correlation and instead use ICCA.

6.5 Proofs of Theorems 6.3.1 and 6.3.2

6.5.1 Proof of Theorem 6.3.1

We begin with the RCCA matrix $C_{\text{reg}} = \left(R_{xx} + \eta I_p\right)^{-1/2} R_{xy} \left(R_{yy} + \eta I_q\right)^{-1/2}$. Recall the data SVDs $\widetilde{X} = F_x K_x G_x^H$ and $\widetilde{Y} = F_y K_y G_y^H$. Substituting these into C_{reg}

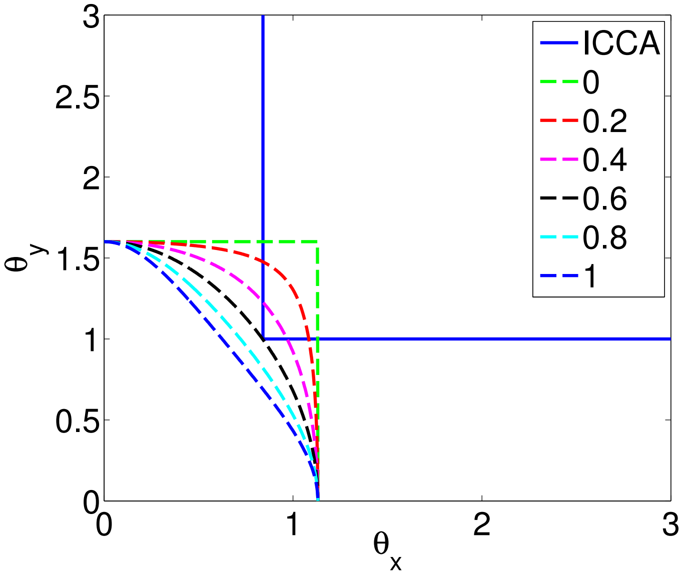


Figure 6.6: Phase transition for LRCCA (dahsed lines) for various ρ and ICCA. The performance of ICCA is independent of ρ. The setting shown in for $c_x = 0.5$ and $c_y = 1$.

yields

$$\begin{aligned} C_{\text{reg}} &= \left(F_x K_x K_x^H F_x^H + \eta I_p\right)^{-1/2} F_x K_x G_x^H G_y K_y^H F_y^H \left(F_y K_y K_y^H F_y^H + \eta I_q\right)\right)^{-1/2} \\ &= F_x \left(K_x K_x^H + \eta I_p\right)^{-1/2} K_x G_x^H G_y K_y^H \left(K_y K_y^H + \eta I_q\right)^{-1/2} \end{aligned}$$

Define $\widetilde{F}_x = F_x(:, 1 : \min(p,n))$, $\widetilde{F}_y = F_y(:, 1 : \min(q,n))$, $\widetilde{G}_x = G_x(:, 1 : \min(p,n))$, and $\widetilde{G}_y = G_y$:, 1 : $\min(q,n))$. Then

$$\widehat{C}_{\text{reg}} = \widetilde{F}_x \, \mathbf{diag}\left(\frac{k_{xi}}{\sqrt{k_{xi}^2 + \eta}}\right) \widetilde{G}_x^H \widetilde{G}_y \, \mathbf{diag}\left(\frac{k_{yi}}{\sqrt{k_{yi}^2 + \eta}}\right) \widetilde{F}_y^H. \tag{6.11}$$

Clearly, as $\eta \to \infty$, this matrix becomes the zero matrix. However, the ratio of the diagonal entries as $\eta \to \infty$ dictates the limiting form of C_{reg}. Examining this ratio of adjacent diagonal elements yields

$$\lim_{\eta \to \infty} \frac{\sqrt{\frac{k_{xi}^2}{\sigma_{xi}^2 + \eta}}}{\sqrt{\frac{k_{x(i+1)}^2}{k_{x(i+1)}^2 + \eta}}} = \lim_{\eta \to \infty} \sqrt{\frac{k_{xi}^2 \left(k_{x(i+1)}^2 + \eta\right)}{k_{x(i+1)}^2 \left(k_{xi}^2 + \eta\right)}} = \frac{k_{xi}}{k_{x(i+1)}}$$

Thus, as $\eta \to \infty$, the ratio of entries along the diagonal matrix approaches the ratio between the singular values. Therefore,

$$\lim_{\eta\to\infty} \mathbf{diag}\left(\frac{k_{xi}}{\sqrt{k_{xi}^2+\eta}}\right) \propto \left(K_x K_x^H\right)^{1/2}.$$

A similar analysis yields an analogous results for the diagonal matrix of the singular values of $\widetilde{Y}$. Therefore,

$$\lim_{\eta\to\infty} \widehat{C}_{\text{reg}} \propto \widetilde{F}_x \left(K_x K_x^H\right)^{1/2} \widetilde{G}_x^H \widetilde{G}_y \left(K_y K_y^H\right)^{1/2} \widetilde{F}_y^H = \widetilde{X}\widetilde{Y}^H.$$

Therefore, as $\eta \to \infty$, the largest singular value of $\widehat{C}_{\text{reg}}$ is proportional to the largest singular value of $\frac{1}{n}\widetilde{X}\widetilde{Y}^H$.

To complete the proof, we must show that the canonical vectors are proportional to the singular vectors of $\frac{1}{n}\widetilde{X}\widetilde{Y}^H$. The top canonical vector for dataset X returned by RCCA is $w_x = \left(R_{xx} + \eta I_d\right)^{-1/2} f_1$, where f_1 is the top left singular vector of $\widehat{C}_{\text{reg}}$. As $\eta \to \infty$, $\left(R_{xx} + \eta I_d\right)^{-1/2} \to \frac{1}{\sqrt{\eta}} I_d$. Therefore, $x_1 \propto f_1$. Similarly, $x_2 \propto g_1$. Therefore, when $\eta \to \infty$, the solution to RCCA is

$$\begin{aligned} &\rho \propto k_1 \\ &x_1 \propto f_1 \\ &x_2 \propto g_1, \end{aligned}$$

where k_1 is the top singular value of $\frac{1}{n}\widetilde{X}\widetilde{Y}^H$ with corresponding left and right singular vectors f_1 and g_1. Successive canonical correlation and vector pairs are found via successive singular value-vector pairs.

6.5.2 Proof of Theorem 6.3.2

We remove the scaling $\frac{1}{n}$ for proof simplicity as this only scales the singular value. The singular values of $C_{\text{lrcca}} = \widetilde{X}\widetilde{Y}^H$ are the positive eigenvalues of

$$\begin{aligned} C_n &= \begin{bmatrix} 0 & \widetilde{X}_n \widetilde{Y}_n^H \\ \widetilde{Y}_n \widetilde{X}_n^H & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \left(U_x \Theta_x V_x^H + X_n\right)\left(U_y \Theta_y V_y^H + Y_n\right)^H \\ \left(U_y \Theta_y V_y^H + Y_n\right)\left(U_x \Theta_x V_x^H + X_n\right)^H & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & X_n Y_n^H \\ Y X^H & 0 \end{bmatrix} + U_n \Lambda U_n^H, \end{aligned}$$

where

$$U_n = \left[\begin{array}{cccc} U_x & X_nV_y & 0 & 0 \\ 0 & 0 & Y_nV_x + U_y\Theta_y P & U_y \end{array}\right], \ \Lambda = \left[\begin{array}{cccc} 0 & 0 & \Theta_x & 0 \\ 0 & 0 & 0 & \Theta_y \\ \Theta_x & 0 & 0 & 0 \\ 0 & \Theta_y & 0 & 0 \end{array}\right].$$

If σ is an eigenvalue of C_n, it must satisfy $\det\left(\sigma I_{p+q} - C_n\right) = 0$. Using our expression above, this is

$$\det\left(\sigma I_{p+q} - \left[\begin{array}{cc} 0 & X_nY_n^H \\ Y_nX_n^H & 0 \end{array}\right] - U_n\Lambda U_n^H\right) = 0. \tag{6.12}$$

Define

$$B_n = \left(\sigma I_{p+q} - \left[\begin{array}{cc} 0 & X_nY_n^H \\ Y_nX_n^H & 0 \end{array}\right]\right).$$

Using properties of determinants, we may re-write (6.12) as

$$\begin{aligned} \det\left(B_n - U_n\Lambda U_n^H\right) &= \det\left(\Lambda\right)\det\left(\Lambda^{-1}\right)\det\left(B - U_n\Lambda U_n^H\right) \\ &= \det(\Lambda)\det\left(\left[\begin{array}{cc} \Lambda^{-1} & U_n^H \\ U_n & B_n \end{array}\right]\right) \\ &= \det\left(B_n\right)\det(\Lambda)\det\left(\Lambda^{-1} - U_n^HB_n^{-1}U_n\right). \end{aligned}$$

If $\sigma > 0$ is an eigenvalue of C_n, it is not a singular value of $X_nY_n^H$ as by assumption $\Lambda \neq 0$. Therefore, B_n is not singular and its inverse exists and it has a nonzero determinant. Therefore for (6.12) to hold,

$$\det\left(\Lambda^{-1} - U_n^HB_n^{-1}U_n\right) = 0. \tag{6.13}$$

Expanding B_n^{-1} yields

$$\begin{aligned} B_n^{-1} &= \left[\begin{array}{cc} \sigma I_p & -X_nY_n^H \\ -Y_nX_n^H & \sigma I_q \end{array}\right]^{-1} \\ &= \left[\begin{array}{cc} \left(\sigma I_p - \frac{1}{\sigma}X_nY_n^HY_nX_n^H\right)^{-1} & \frac{1}{\sigma}\left(\sigma I_p - \frac{1}{\sigma}X_nY_n^HY_nX_n^H\right)^{-1}X_nY_n^H \\ \frac{1}{\sigma}Y_nX_n^H\left(\sigma I_p - \frac{1}{\sigma}X_nY_n^HY_nX_n^H\right)^{-1} & \left(\sigma I_q - \frac{1}{\sigma}Y_nX_n^HX_nY_n^H\right)^{-1} \end{array}\right] \\ &= \left[\begin{array}{cc} \sigma\left(\sigma^2 I_p - X_nY_n^HY_nX_n^H\right)^{-1} & \left(\sigma^2 I_p - X_nY_n^HY_nX_n^H\right)^{-1}X_nY_n^H \\ Y_nX_n^H\left(\sigma^2 I_p - X_nY_n^HY_nX_n^H\right)^{-1} & \sigma\left(\sigma^2 I_q - Y_nX_n^HX_nY_n^H\right)^{-1} \end{array}\right]. \end{aligned}$$

Define $A_n = \left(\sigma^2 I_p - X_n Y_n^H Y_n X_n^H\right)^{-1}$ and $\widetilde{A}_n = \left(\sigma^2 I_q - Y_n X_n^H X_n Y_n^H\right)^{-1}$. Next, we explore $Q_n = U_n^H B_n^{-1} U_n$, which is a $4r \times 4r$ matrix. Denote its block-columns $Q_n = [q_1, \ldots, q_4]$. These block-columns are

$$q_1 = \begin{bmatrix} \sigma U_x^H A_n U_x \\ \sigma V_y^H X_n^H A_n U_x \\ V_x^H Y^H Y_n X_n^H A_n U_x + P\Theta_y U_y^H Y_n X_n^H A_n U_x \\ U_y^H Y_n X_n^H A_n U_x \end{bmatrix}$$

$$q_2 = \begin{bmatrix} \sigma U_x^H A_n X V_y \\ \sigma V_y^H X_n^H A_n X_n V_y \\ V_x^H Y_n^H Y_n X_n^H A_n X_n V_y + P\Theta_y U_y^H Y_n X_n^H A_n X_n V_y \\ U_y^H Y_n X_n^H A_n X_n V_y \end{bmatrix}$$

$$q_3 = \begin{bmatrix} U_x^H A_n X_n Y_n^H Y_n V_x + U_x^H A_n X_n Y_n^H U_y \Theta_y P \\ V_y^H X_n^H A_n X_n Y_n^H Y_n V_x + V_y^H X_n^H A_n X_n Y_n^H U_y \Theta_y P \\ \sigma\left(V_x^H Y_n^H + P\Theta_y U_y^H\right) \widetilde{A}_n \left(V_x^H Y_n^H + P\Theta_y U_y^H\right)^H \\ \sigma U_y^H \widetilde{A}_n Y_n V_x + \sigma U_y^H \widetilde{A}_n U_y \Theta_y P \end{bmatrix}$$

$$q_4 = \begin{bmatrix} U_x^H A_n X_n Y_n^H U_y^H \\ V_y^H X_n^H A_n X_n Y_n^H U_y^H \\ \sigma V_x^H Y_n^H \widetilde{A}_n U_y^H + \sigma P\Theta_y U_y^H \widetilde{A}_n U_y^H \\ \sigma U_y^H \widetilde{A}_n U_y^H \end{bmatrix}.$$

Define

$$\begin{aligned} G_n &= U_x^H A_n U_x \\ F_n &= V_y^H X_n^H A_n X_n V_y^H \\ H_n &= U_y^H \widetilde{A}_n U_y \\ K_n &= V_x^H Y_n^H Y_n X_n^H A_n X_n V_y \\ J_n &= V_x^H Y_n^H \widetilde{A}_n Y_n V_x. \end{aligned}$$

Note that in the large matrix limit $(n, p, q \to \infty)$, matrices of the form $U_x^H M U_y$, $V_x^H M U_y$, $U_x^H M V_y$, $U_x^H M V_x$, $U_y^H M V_y$ are zero in the large matrix limit because U_x, U_y, V_x, and V_y are pairwise independent except for V_x and V_y. Therefore in the large matrix limit,

$$Q_n = \begin{bmatrix} \sigma G_n & 0 & 0 & 0 \\ 0 & \sigma F_n & K_n^H & 0 \\ 0 & K_n & \sigma J_n + \sigma P \Theta_y H_n \Theta_y P & \sigma P \Theta_y H_n \\ 0 & 0 & \sigma H_n \Theta_y P & \sigma H_n \end{bmatrix}.$$

Then define

$$M_n(\sigma) = Q_n - \Lambda^{-1} = \begin{bmatrix} \sigma G_n & 0 & -\Theta_x^{-1} & 0 \\ 0 & \sigma F_n & K_n^H & -\Theta_y^{-1} \\ -\Theta_x^{-1} & K_n & \sigma J_n + \sigma P \Theta_y H_n \Theta_y P & \sigma P \Theta_y H_n \\ 0 & -\Theta_y^{-1} & \sigma H_n \Theta_y P & \sigma H_n \end{bmatrix},$$

which is a $4r \times 4r$ matrix. Then the solution to (6.13) is

$$\det\left(M(\sigma)\right) = 0.$$

We would like to compute this determinant in closed form. To do so, we first note that in the large matrix limit,

$$\begin{aligned} \sigma G_n &\to \varphi_G I_r, & \varphi_G &= \frac{\sigma}{p}\,\mathbf{tr}(A_n) \\ \sigma F_n &\to \varphi_F I_r, & \varphi_F &= \frac{\sigma}{n}\,\mathbf{tr}(X_n^H A_n X_n) \\ \sigma H_n &\to \varphi_H I_r, & \varphi_H &= \frac{\sigma}{q}\,\mathbf{tr}(\widetilde{A}_n) \\ \sigma J_n &\to \varphi_J I_r, & \varphi_J &= \frac{\sigma}{n}\,\mathbf{tr}(Y_n^H \widetilde{A}_n Y_n) \\ K_n &\to \varphi_K P, & \varphi_K &= \frac{1}{n}\,\mathbf{tr}(Y_n^H Y_n X_n^H A_n X_n) \end{aligned}$$

Note that the expressions for φ are implicitly dependent on σ. To simplify this determinant we will rely on the following property of determinants of block matrices,

$$\det\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \det(A)\det(D - CA^{-1}B).$$

Using this,

$$\begin{aligned}
\det(M) \quad =& \det\left(\begin{bmatrix} \varphi_G I_r & 0 \\ 0 & \varphi_F I_r \end{bmatrix}\right) \cdot \\
& \det\left(\begin{bmatrix} \varphi_J + P\Theta_y\varphi_H\Theta_y P & P\Theta_y\varphi_H \\ \varphi_H\Theta_y P & \varphi_H \end{bmatrix}\right. \\
& - \quad \left.\begin{bmatrix} -\Theta_x^{-1} & \varphi_K P \\ 0 & \Theta_y^{-1} \end{bmatrix} \begin{bmatrix} \frac{1}{\varphi_G} I_r & 0 \\ 0 & \frac{1}{\varphi_F} I_r \end{bmatrix} \begin{bmatrix} -\Theta_x^{-1} & 0 \\ \varphi_K P & -\Theta_y^{-1} \end{bmatrix}\right) \\
=& (\varphi_G\varphi_F)^r \det\begin{pmatrix} \underbrace{\varphi_J I_r + \varphi_H P\Theta_y^2 P - \frac{1}{\varphi_G}\Theta_x^{-2} - \frac{\varphi_K^2}{\varphi_F}P^2}_{a} & \underbrace{\varphi_H P\Theta_y + \frac{\varphi_K}{\varphi_F}P\Theta_y^{-1}}_{b} \\ \underbrace{\varphi_H\Theta_y P + \frac{\varphi_K}{\varphi_F}\theta_y^{-1}P}_{c} & \underbrace{\varphi_H I_r - \frac{1}{\varphi_F}\Theta_y^{-2}}_{d} \end{pmatrix} \\
=& (\varphi_G\varphi_F)^r \prod_{i=1}^{r}\left(\varphi_H - \frac{1}{\varphi_F\theta_{yi}^2}\right)\det\left(a - bd^{-1}c\right) \\
=& (\varphi_G\varphi_F)^r \prod_{i=1}^{r}\left(\frac{\varphi_H\varphi_F\theta_{yi}^2 - 1}{\varphi_F\theta_{yi}^2}\right)\prod_{i=1}^{r}\left(\frac{\varphi_J\varphi_G\theta_{xi}^2 - 1}{\varphi_G\theta_{xi}^2} - \rho_i^2\frac{\varphi_H\theta_{yi}^2\left(1+\varphi_{K_i}\right)^2}{\varphi_H\varphi_F\theta_{yi}^2 - 1}\right) \\
=& (\varphi_G\varphi_F)^r \prod_{i=1}^{r}\left(\frac{\varphi_H\varphi_F\theta_{yi}^2 - 1}{\varphi_F\theta_{yi}^2}\right)\left(\frac{\varphi_J\varphi_G\theta_{xi}^2 - 1}{\varphi_G\theta_{xi}^2} - \rho_i^2\frac{\varphi_H\theta_{yi}^2\left(1+\varphi_{K_i}\right)^2}{\varphi_H\varphi_F\theta_{yi}^2 - 1}\right) \\
=& (\varphi_G\varphi_F)^r \prod_{i=1}^{r}\left[\frac{\left(\varphi_H\varphi_F\theta_{yi}^2 - 1\right)\left(\varphi_J\varphi_G\theta_{xi}^2 - 1\right)}{\varphi_F\varphi_G\theta_{xi}^2\theta_{yi}^2} - \frac{\rho_i^2\varphi_H\varphi_G\theta_{xi}^2\theta_{yi}^2(1+\varphi_K)^2}{\varphi_F\varphi_G\theta_{xi}^2\theta_{yi}^2}\right] \\
=& \prod_{i=1}^{r}\left(\varphi_H\varphi_F - \frac{1}{\theta_{yi}^2}\right)\left(\varphi_J\varphi_G - \frac{1}{\theta_{xi}^2}\right) - \rho_i^2\varphi_H\varphi_G\left(1+\varphi_K\right)^2
\end{aligned}$$

This is the form of (6.9). To evaluate $\det(M(\sigma)$ and to complete the theorem proof, we need closed form expressions for φ_H, φ_F, φ_J, φ_G, and φ_K that do not rely on the noise matrices X_n and Y_n. To accomplish this, we use proposition 10.11 in [131].

6.5.2.1 Expression for φ_F

By definition,

$$\begin{aligned}\varphi_F &= \frac{\sigma}{n}\,\mathbf{tr}\left(X_n^H A_n X_n\right)\\ &= \frac{\sigma}{n}\,\mathbf{tr}\left(A_n X_n X_n^H\right)\\ &= \frac{\sigma}{n}\,\mathbf{tr}\left(\left(\sigma^2 I_p - X_n Y_n^H Y_n X_n^H\right)^{-1} X_n X_n^H\right).\end{aligned}$$

Let $U_X\Sigma_X V_X^H$ be the SVD of X_n. Using this definition,

$$\begin{aligned}\varphi_F &= \frac{\sigma}{n}\,\mathbf{tr}\left(\left(\sigma^2 I_p - U_X\Sigma_X V_X^H Y_n^H Y_n V_X \Sigma_X^H U_X^H\right)^{-1} U_X\Sigma_X\Sigma_X^H U_X^H\right)\\ &= \frac{\sigma}{n}\,\mathbf{tr}\left(\left(\sigma^2 I_p - \Sigma_X V_X^H Y_n^H Y_n V_X \Sigma_X^H\right)^{-1} \Sigma_X\Sigma_X^H\right)\end{aligned} \tag{6.14}$$

Now define $R_n = X_n^H X_n$ and $S_n = Y_n^H Y_n$ and the functions $h(L_n) = \left(\sigma^2 I_n - L_n\right)^{-1}$ and $g(L_n) = L_n$. Let μ_{R_n} and μ_{S_n} be the empirical eigenvalue distribution and assume that each converges almost surely weakly, as $n, p, q \to \infty$ as above, to the non-random compactly supported probability measures μ_R and μ_S, respectively.With these definitions and the SVD of X_n, note that

$$\mathcal{E} = \mathbf{tr}(h(R_n^{1/2} S_n R_n^{1/2}) g(R_n))$$

is equivalent to

$$\begin{aligned}\mathcal{E} &= \mathbf{tr}\left(\left(\sigma^2 I_n - \left(X_n^H X_n\right)^{1/2} Y_n^H Y_n \left(X_n^H X_n\right)^{1/2}\right)^{-1} X_n^H X_n\right)\\ &= \mathbf{tr}\left(\left(\sigma^2 I_n - V_X\left(\Sigma_X^H\Sigma_X\right)^{1/2} V_X^H Y_n^H Y_n V_X \left(\Sigma_X^H\Sigma_X\right)^{1/2} V_X^H\right)^{-1} V_X\Sigma_X^H\Sigma_X V_X^H\right)\\ &= \mathbf{tr}\left(\left(\sigma^2 I_n - \left(\Sigma_X^H\Sigma_X\right)^{1/2} V_X^H Y^H Y V_X \left(\Sigma_X^H\Sigma_X\right)^{1/2}\right)^{-1} \Sigma_X^H\Sigma_X\right)\end{aligned} \tag{6.15}$$

We now show that (6.14) and (6.15) are equivalent. We break this into two cases, one where $n > p$ and one where $p \geq n$. Define $\widetilde{\Sigma}_X$ to be the $\min(n,p) \times \min(n,p)$ diagonal matrix of the non-zero singular values found along the diagonal of Σ_X. Also define $\widetilde{V}_X$ to be the corresponding $\min(n,p)$ right singular vectors of X.

In case 1, $n > p$. Here

$$\left(\Sigma_X^H\Sigma_X\right)^{1/2} = \begin{bmatrix} \widetilde{\Sigma}_X & 0 \\ 0 & 0_{n-p} \end{bmatrix}$$

and $\Sigma_X \Sigma_X^H = \widetilde{\Sigma}_X^2$. Using these expressions, (6.15) becomes

$$\begin{aligned}
&\mathbf{tr}\left(\left(\begin{bmatrix} \sigma^2 I_p & 0 \\ 0 & \sigma^2 I_{n-p} \end{bmatrix} - \begin{bmatrix} \widetilde{\Sigma}_X \widetilde{V}_X^H Y_n^H Y_n \widetilde{V}_X \widetilde{\Sigma}_X & 0 \\ 0 & 0 \end{bmatrix}\right)^{-1} \begin{bmatrix} \widetilde{\Sigma}_X^2 & 0 \\ 0 & 0 \end{bmatrix}\right) \\
&= \mathbf{tr}\left(\left(\sigma^2 I_p - \widetilde{\Sigma}_X \widetilde{V}_x^H Y_n^H Y_n \widetilde{V}_X \widetilde{\Sigma}_X\right)^{-1} \widetilde{\Sigma}_X^2\right).
\end{aligned}$$

Because $n > p$, $\Sigma_X V_X^H = \widetilde{\Sigma}_X \widetilde{V}_X^H$ and therefore (6.14)=(6.15).

In case 2, $n \leq p$. Here

$$\left(\Sigma_X \Sigma_X^H\right)^{1/2} = \begin{bmatrix} \widetilde{\Sigma}_X & 0 \\ 0 & 0_{n-p} \end{bmatrix}$$

and $\Sigma_X^H \Sigma_X = \widetilde{\Sigma}_X^2$. In this setting, $\Sigma_X V_X^H = \widetilde{\Sigma}_X V_X^H$. Using these expressions, (6.15) becomes

$$\begin{aligned}
&\frac{\sigma}{n}\mathbf{tr}\left(\left(\begin{bmatrix} \sigma^2 I_n & 0 \\ 0 & \sigma^2 I_{p-n} \end{bmatrix} - \begin{bmatrix} \widetilde{\Sigma}_X V_X^H Y_n^H Y_n V_X \widetilde{\Sigma}_X^H & 0 \\ 0 & 0 \end{bmatrix}\right)^{-1} \begin{bmatrix} \widetilde{\Sigma}_X & 0 \\ 0 & 0_{n-p} \end{bmatrix}\right) \\
= \quad &\frac{\sigma}{n}\mathbf{tr}\left(\left(\sigma^2 I_n - \widetilde{\Sigma}_X V_X^H Y_n^H Y_n V_X \widetilde{\Sigma}_X^H\right)^{-1} \widetilde{\Sigma}_X^2\right)
\end{aligned}$$

Therefore, in this second setting, (6.14)=(6.15) as well. Therefore,

$$\varphi_F = \frac{\sigma}{n}\mathbf{tr}(h(S_n^{1/2} B_n S_n^{1/2}) g(S_n)).$$

By Proposition 10.11 of [131], as $n \to \infty$,

$$\varphi_F \to \sigma \int g(x) h(y) \rho_{RS}(x, y) dx dy$$

where $\rho_{RS}(x, y) = k_{RS|R}(x, y) f_R(x)$, where $f_R(x)$ is the limiting eigenvalue density function of R and $k_{RS|R}(x, y)$ is the Markov transition kernel density function. Using these definitions yields

$$\begin{aligned}
\varphi_F \quad &= \frac{\sigma}{n}\,\mathbf{tr}\left(h\left(R_n^{1/2}S_nR_n^{1/2}\right)g(R_n)\right) \\
&\to \sigma \int g(x)h(y)\rho_{RS}(x,y)dxdy \\
&= \sigma \int g(x)h(y)k_{RS|R}(x,y)f_R(x)dxdy \\
&= \sigma \int\int \frac{x}{\sigma^2-y}k_{RS|R}(x,y)f_R(x)dxdy \\
&= \sigma \int xf_R(x)\left(\frac{1}{\sigma^2-y}k_{RS|R}(x,y)dy\right)dx \\
&= -\sigma \int xm_{\mu_{RS|R}}\left(\sigma^2,x\right)f_R(x)dx \\
&= -\sigma\mathbb{E}\left[xm_{\mu_{RS|R}}\left(\sigma^2,x\right)\right]_{\mu_R}
\end{aligned}$$

6.5.2.2 Expression for φ_J

Using an analogous derivation,

$$\varphi_J \to -\sigma\mathbb{E}\left[xm_{\mu_{RS|S}}\left(\sigma^2,x\right)\right]_{\mu_S}.$$

6.5.2.3 Expression for φ_G

Let μ_{M_1} be the limiting eigenvalue distribution of $X_nY_n^HY_nX_n^H$. By definition,

$$\begin{aligned}
\varphi_G \quad &= \frac{\sigma}{p}\,\mathbf{tr}\left(\left(\sigma^2I_p - X_nY_n^HY_nX_n^H\right)^{-1}\right) \\
&\to \sigma \int \frac{1}{\sigma^2-x}\mu_{M_1} \\
&= -\sigma m_{M_1}(\sigma^2)
\end{aligned}$$

6.5.2.4 Expression for φ_H

Let μ_{M_2} be the limiting eigenvalue distribution of $Y_nX_n^HX_nY_n^H$. By definition,

$$\begin{aligned}
\varphi_H \quad &= \frac{\sigma}{p}\,\mathbf{tr}\left(\left(\sigma^2I_p - Y_nX_n^HX_nY_n^H\right)^{-1}\right) \\
&\to \sigma \int \frac{1}{\sigma^2-x}\mu_{M_2} \\
&= -\sigma m_{M_2}(\sigma^2)
\end{aligned}$$

6.5.2.5 Expression for φ_K

Let μ_{M_2} be the limiting eigenvalue distribution of $XY^HYX^H\left(\sigma^2 I_p - XY^HYX^H\right)^{-1}$. Note that μ_{M_3} is a mobius transform of μ_{M_1}. By definition,

$$\begin{aligned}\varphi_K &= \frac{1}{n}\,\mathbf{tr}\left(Y^HYX^HAX\right)\\ &= \frac{1}{n}\,\mathbf{tr}\left(XY^HYX^H\left(\sigma^2 I_p - XY^HYX^H\right)^{-1}\right)\\ &\to \frac{p}{n}\int x\mu_{M_3}\\ &= c_x\mathbb{E}\left[x\right]_{\mu_{M_3}}.\end{aligned}$$

This establishes the proof of Theorem 6.3.2.

CHAPTER VII

The Largest Singular Values of a Random Projection of a Low-Rank Perturbation of a Random Matrix

7.1 Introduction

In Chapters II-VI, we stack observations in a data matrix that is assumed low-rank plus noise, modeled as

$$\widetilde{X}_n = \sum_{i=1}^{r} \theta_i u_i v_i^T + X_n. \tag{7.1}$$

In the above equation, for $i = 1, \dots, r$, $u_i \in \mathbb{C}^{n \times 1}$ and $v_i \in \mathbb{C}^{N \times 1}$ are independent unit norm signal vectors, $\theta_i > 0$ are the associated signal values and X_n is a noise-only matrix. Assume that $u_i^H u_j = \delta_{\{i=j\}}$ and $v_i^H v_j = \delta_{\{i=j\}}$. Let $X_n \in \mathbb{C}^{n \times N}$ be a real or complex random matrix. Let $\sigma_1, \dots, \sigma_{\min(n,N)}$ be the singular values of X_n. Let μ_{X_n} be the empirical singular value distribution, i.e, the probability measure defined as

$$\mu_{X_n} = \frac{1}{\min(n,N)} \sum_{i=1}^{\min(n,N)} \delta_{\sigma_i}.$$

Assume that as $n \to \infty$, $n/N \to c_1$.

In many signal processing applications, we treat the columns of $\widetilde{X}_n$ as noisy observations of a desired target signal lying in the span of $\{u_1, \dots, u_r\}$. In this light, we treat θ_i as the signal-to-noise ratio (SNR) for its corresponding subspace component, n as the intrinsic dimension of the problem, and N as the number of samples (or snapshots or observations) we have at our disposal. To recover the underlying signal subspace, **span** $\{u_1, \dots, u_r\}$, it is common to take the left singular vectors of $\widetilde{X}_n$ corresponding to the largest r singular values. The accuracy of this estimate is well studied (see [84, 85, 107, 116]). Specifically, when X_n has independent $\mathcal{CN}(0,1)$ entries, the individual subspace component estimates are known to have a non-random estimate when $\theta_i > \left(\frac{n}{N}\right)^{1/4}$.

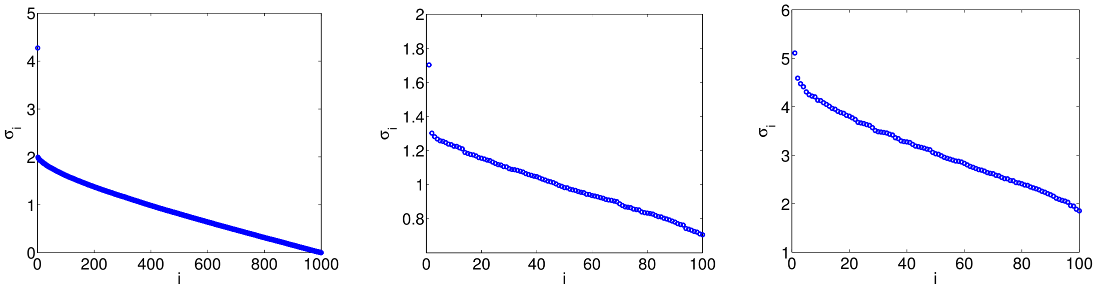
However, in many such applications the intrinsic dimension, n, of the system is so large that taking the SVD of $\widetilde{X}_n$ may not be tractable. In this chapter, we explore the performance of signal detection when randomly projecting $\widetilde{X}$ into a lower dimensional space using either a Gaussian or unitary projection. Specifically for $m < n$, let $G_n \in \mathbb{C}^{n\times m}$ be a random matrix with independent $\mathcal{CN}(0,1)$ entries and let $Q_n \in \mathbb{C}^{n\times m}$ be a unitary matrix such that $Q_n^H Q_n = I_m$. Define the $m \times N$ complex matrices

$$\begin{aligned} Y_n^G &= G_n^H \widetilde{X}_n \\ Y_n^Q &= Q_n^H \widetilde{X}_n \end{aligned} \tag{7.2}$$

Since $m < n$, taking the SVD of Y_n^G and Y_n^Q is more tractable than taking the SVD of $\widetilde{X}_n$. Since $m < n$, taking the SVD of Y_n^G and Y_n^Q is more tractable than taking the SVD of $\widetilde{X}_n$. Such compressed sensing strategies for both unitary [132, 133, 134] and Gaussian [135, 136, 137] sensing matrices have been extensively studied. These algorithms have been extended to include Gaussian-like strategies that employ matrices with partially observed entries [138, 139] as well as unitary-like strategies that use a discrete Fourier transform matrix [140] or discrete cosine transform [141]. For excellent reviews of such compressed sensing algorithms please see [142, 143, 144], for example.

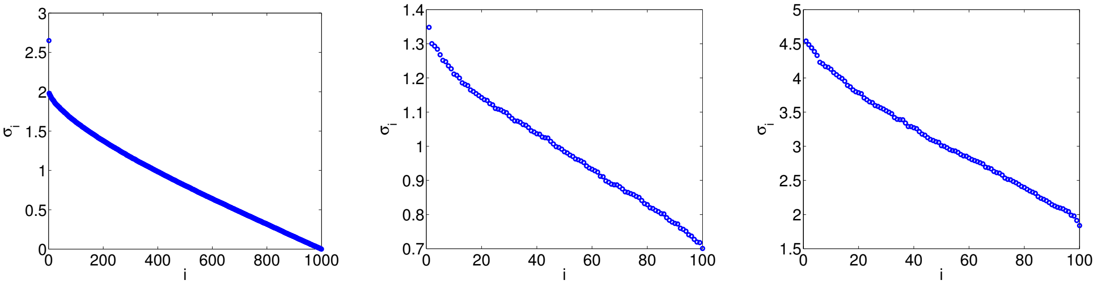
These works examine the ability of such matrices to approximate the original data matrix as low rank. In this chapter, we consider the fundamental limits of the resulting singular values when used to detect low-rank signals. We quantify how the dimensions of our matrices, m, n, N, and the SNR θ affect the behavior of the largest singular values of Y_n^G and Y_n^Q. Finally, we compare the detection performance of these two specific choices of the projection matrix and show that a unitary projection matrix can more reliably detect low-rank signals than a Gaussian projection matrix.

Our main conclusion is summarized in Figures 7.1 and 7.2. In both figures, we consider a rank-1 setting where $r = 1$. In Figure 7.1, the SNR of the lone signal is large enough so that the largest singular value of both Y_n^G and Y_n^Q separate from the bulk of the singular values. The top singular values of these matrices detect the presence of our lone signal. In Figure 7.2, we decrease the SNR of the lone signal. In this simulation, the largest singular value of Y_n^Q continues to separate from the bulk distribution but the largest singular value of Y_n^G no longer separates from the bulk distribution. The unitary projection can reliably detect the presence of signal vectors



(a) $\widetilde{X}$ (b) $Q^H\widetilde{X}$ (c) $G^H\widetilde{X}$

Figure 7.1: Singular value spectra for the full matrix (a), orthogonal projection matrix (b), and Gaussian projection matrix (c). This example uses a rank-1 setting where $n = 1000$, $m = 100$, $N = 1000$, $\theta = \theta_x = \theta_y = 4$.



(a) $\widetilde{X}$ (b) $Q^H\widetilde{X}$ (c) $G^H\widetilde{X}$

Figure 7.2: Singular value spectra for the full matrix (a), orthogonal projection matrix (b), and Gaussian projection matrix (c). This example uses a rank-1 setting where $n = 1000$, $m = 100$, $N = 1000$, $\theta = \theta_x = \theta_y = 2.2$.

at a lower SNR than the Gaussian projection.

This chapter is organized as follows. In Section 7.2, we provide the main results of this chapter including the almost sure limit of the top singular values of the projection matrices in (7.2). We then provide corollaries to the main result that highlight a phase transition below which signal detection is impossible and a closed form expression of our main theorem for unitary projections. We provide the proof of our main theorem in Section 7.3 and the proof of the main corollary in Section 7.4. In Section 7.5, we verify our asymptotic results on finite sized systems and highlight the accuracy of our predictions. We make the following assumptions and definitions about the random matrices needed throughout the rest of the chapter.

Assumption 7.1.1. *The probability measures μ_{X_n}, μ_{G_n}, and μ_{Q_n} converge almost surely weakly to a non-random compactly supported probability measures μ_X,μ_G, and*

μ_Q, respectively.

Definition 7.1.1. *Let $M_n^G = G_n^H X_n$ be the product of the random matrices G_n and X_n and let $M_n^Q = Q_n^H X_n$ be the product of the random matrices Q_n and X_n.*

Assumption 7.1.2. *The probability measure $\mu_{M_n^G}$ converges almost surely weakly to a non-random compactly supported probability measure μ_{M_G}. The probability measure $\mu_{M_n^Q}$ converges almost surely weakly to a non-random compactly supported probability measure μ_{M_Q}.*

Assumption 7.1.3. *Let a_G be the infimum of the support μ_{M_G}. The smallest singular value of M_n^G converges almost surely to a_G. Let a_Q be the infimum of the support μ_{M_Q}. The smallest singular value of M_n^Q converges almost surely to a_Q.*

Assumption 7.1.4. *Let b_G be the supremum of the support μ_{M_G}. The largest singular value of M_n^G converges almost surely to b_G. Let b_Q be the supremum of the support μ_{M_Q}. The largest singular value of M_n^Q converges almost surely to b_Q.*

7.2 Main Results

Our main result of this chapter characterizes the asymptotic behavior of the largest singular values of the projection matrices defined in (7.2).

Theorem 7.2.1. *Let Y_n be the projection of $\widetilde{X}_n$ onto either G_n or Q_n as in (7.2). The largest r singular values of the $m \times N$ matrix Y_n exhibit the following behavior as $n, m, N \to \infty$ with $n/N \to c_1$ and $m/N \to c_2$. We have that for each fixed $1 \leq i \leq r$, $\sigma_i(Y_n)$ solves*

$$\sigma_i^2 \varphi_F(\sigma_i)\varphi_H(\sigma_i) = \frac{1}{\theta_i^2}, \tag{7.3}$$

where

$$\varphi_F(\sigma_i) \xrightarrow{a.s.} -\mathbb{E}\left[x m_{\mu_{RS|R}}\left(\sigma_i^2, x\right)\right]_{\mu_R}$$
$$\varphi_H(\sigma_i) \xrightarrow{a.s.} -\frac{n}{N} m_{M_3}(\sigma_i^2) - \frac{1}{\sigma_i^2}\frac{n-N}{N}$$

where m_{μ_M} is the Stieltjes transform of a matrix M defined as

$$m_{\mu_M}(z) \int \frac{1}{x-z}\mu_M(x),$$

and μ_R is the limiting eigenvalue density of either $G_n G_n^H$ or $Q_n Q_n^H$, μ_S is the limiting eigenvalue density of $X_n X_n^H$, $m_{\mu_{RS|S}}$ is the Stieltjes transform of the limiting conditional density and $m_{\mu_{M_3}}$ is the Stieltjes transform of $G_n G_n^H X_n X_n^H$ or $Q_n Q_n^H X_n X_n^H$.

When using G_n, $m_{\mu_{RS|S}}(z,x)$ solves the following equation

$$\begin{aligned}0 &= \left(-n^2z^2\right)\left(m_{\mu_{RS|S}}(z,x)\right)^3 + \left(Nnz + mnz - 2n^2z\right)\left(m_{\mu_{RS|S}}(z,x)\right)^2 \\ &\quad + \left(Nn + mn + Nmz - n^2 - Nm\right) m_{\mu_{RS|S}}(z,x) + Nm.\end{aligned} \tag{7.4}$$

When using the Gaussian projection matrix, G_n, we do not get a closed form of the top singular values. Solving (7.4) for $m_{\mu_{RS|S}}(z,x)$ is unwieldy as we must solve a cubic polynomial. Furthermore, we must take the expectation of the resulting solution with respect to the distribution μ_R. To solve the expressions φ_F and φ_H when using a Gaussian projection matrix, we use RMTool [3]. We discuss this process in Section 7.5 but note here that this process still yields an analytic solution, although not closed form. However, when using a unitary projection matrix, we do get a closed form expression for the largest singular values.

Corollary 7.2.1. *When Y_n is a generated using a unitary matrix Q_n, we have that for each fixed $1 \leq i \leq r$,*

$$\sigma_i \xrightarrow{a.s.} \begin{cases} \sqrt{\frac{c_1}{\theta_i^2} + c_2\theta_i^2 + 1 + c_1c_2} & \text{if } \theta_i \geq \left(\frac{c_1}{c_2}\right)^{1/4} \\ \sqrt{c_1c_2} + 1 & \text{if } \theta_i < \left(\frac{c_1}{c_2}\right)^{1/4} \end{cases}.$$

This corollary nicely gives the almost sure limit of the top singular values as a function of the system parameters n, m, N, and θ_i. This corollary also makes contact with a natural phase transition. When the SNR of a component is below a critical value depending only on n, m, N, the corresponding top singular value behaves as if Y_n is a noise only matrix. Such phase transitions appear in other matrix analyses (see [84, 85, 107, 116]). Similarly, we may solve for the phase transition when using a Gaussian matrix, although we do not get a closed form expression as we do in the unitary case.

Corollary 7.2.2. *When*

$$\theta_i \leq \theta_{crit} = \frac{1}{b\sqrt{\varphi_F(b)\varphi_H(b)}}$$

then

$$\sigma_i \xrightarrow{a.s.} b,$$

where b is either b_Q or b_G depending on our projection matrix.

7.3 Proof of Theorem 7.2.1

To simplify the notation, we use the matrix G_n to represent both G_n and Q_n. We break this notation only where we need to differentiate between the two. Define the matrices

$$\Theta = \mathbf{diag}(\theta_1, \ldots, \theta_r), \;\; U_n = [u_1, \ldots, u_r], \;\; V_N = [v_1, \ldots, v_r].$$

The singular values of Y_n are the positive eigenvalues of

$$\begin{aligned} C_n &= \begin{bmatrix} 0 & Y_n \\ Y_n^H & 0 \end{bmatrix} = \begin{bmatrix} 0 & G_n^H U_n \Theta V_N^H + G_n^H X_n \\ V_N \Theta U_n^H G_n + X_n^H G_n & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & G_n^H X_n \\ X_n^H G_n & 0 \end{bmatrix} + Q_n \Lambda Q_n^H, \end{aligned}$$

where

$$Q_n = \begin{bmatrix} G_n^H U_n & 0 \\ 0 & V_m \end{bmatrix}, \;\; \Lambda = \begin{bmatrix} 0 & \Theta \\ \Theta & 0 \end{bmatrix}.$$

If σ_i is an eigenvalue of C_n, it must satisfy $\det(\sigma_i I_{N+m} - C_n) = 0$. Using our expression above, this is

$$\det\left(\sigma_i I_{N+m} - \begin{bmatrix} 0 & G_n^H X_n \\ X_n^H G_n & 0 \end{bmatrix} - Q_n \Lambda Q_n^H\right) = 0. \tag{7.5}$$

Define

$$B_n = \left(\sigma_i I_{N+m} - \begin{bmatrix} 0 & G_n^H X_n \\ X_n^H G_n & 0 \end{bmatrix}\right).$$

Using properties of determinants, we may re-write (7.5) as

$$\begin{aligned} \det\left(B_n - Q_n \Lambda Q_n^T\right) &= \det(\Lambda) \det\left(\Lambda^{-1}\right) \det\left(B_n - Q_n \Lambda Q_n^H\right) \\ &= \det(\Lambda) \det\left(\begin{bmatrix} \Lambda^{-1} & Q_n^H \\ Q_n & B_n \end{bmatrix}\right) \\ &= \det(B_n) \det(\Lambda) \det\left(\Lambda^{-1} - Q_n^H B_n^{-1} Q_n\right). \end{aligned}$$

If $\sigma_i > 0$ is an eigenvalue of C_n, it is not a singular value of $G_n^T X_n$ as by assumption $\Lambda \neq 0$. Therefore, B_n is not singular and its inverse exists and it has a nonzero determinant. Therefore for (7.5) to hold,

$$\det\left(\Lambda^{-1} - Q_n^H B_n^{-1} Q_n\right) = 0. \tag{7.6}$$

Expanding B_n^{-1} using the properties of block diagonal matrices yields

$$\begin{aligned} B_n^{-1} &= \begin{bmatrix} \sigma_i I_m & -G_n^H X_n \\ -X_n^H G_n & \sigma I_N \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \sigma_i \left(\sigma_i^2 I_p - G_n^H X_n X_n^H G_n\right)^{-1} & \left(\sigma_i^2 I_p - G_n^H X_n X_n^H G_n\right)^{-1} G_n^H X_n \\ X_n^H G_n \left(\sigma_i^2 I_p - G_n^H X_n X_n^H G_n\right)^{-1} & \sigma_i \left(\sigma_i^2 I_q - X_n^H G_n G_n^H X_n\right)^{-1} \end{bmatrix}. \end{aligned}$$

Define $A_n = \left(\sigma_i^2 I_m - G_n^H X_n X_n^H G_n\right)^{-1}$ and $\widetilde{A}_n = \left(\sigma_i^2 I_N - X_n^H G_n G_n^H X_n\right)^{-1}$. Therefore

$$B_n^{-1} = \begin{bmatrix} \sigma_i A_n & A_n G_n^H X_n \\ X_n^H G_n A_n & \sigma_i \widetilde{A}_n \end{bmatrix}.$$

Therefore

$$Q_n^H B_n^{-1} Q_n = \begin{bmatrix} \sigma_i U_n^H G_n A_n G_n^H U_n & U_n^H A_n G_n^H X_n V_N \\ V_N^H X_n^H G_n A_n G_n^H U_n & \sigma_i V_N^H \widetilde{A}_n V_N \end{bmatrix}.$$

By Proposition 10.11 of [131], for smooth functions, h and g, on $\mathbb{R}$ and asymptotically free random matrices A_n and B_n,

$$\frac{1}{n}\,\mathbf{tr}\left(h\left(A_n^{1/2} B_n A_n^{1/2}\right) g\left(A_n\right)\right) \to \int g(x) h(y) \rho_{AB}(x,y)\,dx dy$$

where $\rho_{AB}(x,y)$ is a bivariate probability density function of $\mathbb{R}^2$ that may be decomposed

$$\rho_{AB}(x,y) = k_{AB|A}(x,y) f_A(x),$$

where $f_A(x)$ is the limiting eigenvalue density function of A and $k_{AB}(x,y)$ is the Markov transition kernel density function.

Armed with this proposition, define $R_n = G_n G_n^H$ and $S_n = X_n X_n^H$ and the functions $h(L_n) = \left(\sigma_i^2 I_m - L_n\right)^{-1}$ and $g(L_n) = L_n$. With these definitions,

$$\mathbf{tr}\left(G_n A_n G_n^H\right) = \mathbf{tr}(h(R_n^{1/2} S_n R_n^{1/2}) g(R_n)).$$

Therefore by the above proposition and Assumption 7.1.1,

$$\frac{1}{n}\,\mathbf{tr}\left(G_n A_n G_n^H\right) \to \int g(x) h(y) \rho_{RS}(x,y)\,dx dy,$$

where $\rho_{RS}(x,y) = k_{RS|R}(x,y)f_R(x)$, where $f_R(x)$ is the limiting eigenvalue density function of R and $k_{RS|R}(x,y)$ is the Markov transition kernel density function. Therefore almost surely we have that

$$\sigma_i U_n^H G_n A_n G_n^H U_n \to \sigma_i \underbrace{\int g(x)h(y)\rho_{RS}(x,y)dxdy}_{\varphi_F(\sigma_i)} \cdot I_r.$$

In a similar manner, by Assumption 7.1.2

$$\frac{\sigma_i}{N}\,\mathbf{tr}(\widetilde{A}_n) \to \int \frac{\sigma_i}{\sigma_i^2 - t^2} d_M(t).$$

Therefore, almost surely we have that

$$\sigma_i V_N^H \widetilde{A}_n V_n \to \sigma_i \underbrace{\left(\int \frac{1}{\sigma_i^2 - t^2} d_M(t)\right)}_{\varphi_H(\sigma_i)} \cdot I_r.$$

In the same way, almost surely,

$$\begin{aligned} &U_n^H A_n G_n^H X_n V_N \to 0 \\ &V_N^H X_n^H G_n A_n G_n^H U_n \to 0 \end{aligned}$$

Therefore, it follows that almost surely,

$$Q_n^H B_N^{-1} Q_n \to \begin{bmatrix} \sigma\varphi_F(\sigma_i)I_r & 0 \\ 0 & \sigma\varphi_H(\sigma_i)I_r \end{bmatrix}.$$

Then

$$\begin{aligned} \det(\Lambda^{-1} - U_n^H B_n^{-1} U_n) \quad &\xrightarrow{\text{a.s.}} \det\left(\begin{bmatrix} -\sigma\varphi_F(\sigma_i)I_r & \Theta^{-1} \\ \Theta^{-1} & -\sigma\varphi_H(\sigma_i)I_r \end{bmatrix}\right) \\ &= \det\left(-\sigma_i\varphi_F(\sigma_i)I_r\right) \cdot \\ &\quad \det\left(-\sigma_i\varphi_H(\sigma_i)I_r - \Theta^{-1}\left(-\sigma_i\varphi_F(\sigma_i)I_r\right)^{-1}\Theta^{-1}\right). \end{aligned}$$

Then the solution to (7.6) is

$$0 = \prod_{j=1}^{r}\left(\frac{1}{\theta_j^2\sigma_i\varphi_F(\sigma_i)} - \sigma_i\varphi_H(\sigma_i)\right),$$

which implies that σ_i must solve

$$\frac{1}{\theta_i^2} = \sigma_i^2 \varphi_F(\sigma_i)\varphi_H(\sigma_i).$$

This completes the general statement of the theorem. We now next develop the expressions for $\varphi_F(\sigma_i)$ and $\varphi_H(\sigma_i)$ stated in the theorem.

7.3.1 Expression for φ_F

Using these definitions for R, S, h, and g above, we have

$$\begin{aligned}
\varphi_F(\sigma_i) \quad &= \int g(x)h(y)\rho_{RS}(x,y)dxdy \\
&= \int g(x)h(y)k_{RS|R}(x,y)f_R(x)dxdy \\
&= \int\int \frac{x}{\sigma_i^2 - y} k_{RS|R}(x,y)f_R(x)dxdy \\
&= \int xf_R(x)\left(\frac{1}{\sigma_i^2 - y}k_{RS|R}(x,y)dy\right)dx \\
&= -\int xm_{\mu_{RS|R}}\left(\sigma_i^2, x\right)f_R(x)dx \\
&= -\mathbb{E}\left[xm_{\mu_{RS|R}}\left(\sigma_i^2, x\right)\right]_{\mu_R}.
\end{aligned}$$

7.3.2 Expression for φ_H

By Assumption 7.1.2, the matrix product $M_n = G_n^H X_n$ has the limiting distribution μ_M. Define $M_n^{(2)} = X_n^H G_n G_n^H X_n$, which by the same assumption has limiting distribution, which we denote μ_{M_2}. By definition,

$$\begin{aligned}
\varphi_H(\sigma_i) \quad &= \int \frac{1}{\sigma_i^2 - x}\mu_{M_2}(x)dx \\
&= -m_{\mu_{M_2}}(\sigma_i^2)
\end{aligned}$$

where m_{μ_M} is the Stieltjes transform of a matrix M.

To compute φ_H, we consider the matrix $M_n^{(3)} = G_n G_n^H X_n X_n^H = R_n S_n$, which has the same non-zero eigenvalues as $M_n^{(2)} = X_n^H G_n G_n^H X_n$ and depending on whether n or N is larger, one has $|n - N|$ extra zero eigenvalues. Therefore,

$$f_{\mu_{M_2}}(x) = \frac{n}{N} f_{\mu_{M_3}}(x) - \frac{n-N}{N}\mathbb{1}_{\{x=0\}}.$$

Using this relationship, we can rewrite

$$\begin{aligned}\varphi_H(\sigma_i) &= \int \frac{1}{\sigma_i^2 - x} f_{\mu_{M_2}}(x)dx \\ &= \int \frac{1}{\sigma_i^2 - x}\left(\frac{n}{N} f_{\mu_{M_3}}(x) - \frac{n-N}{N}\mathbb{1}_{\{x=0\}}\right)dx \\ &= \frac{n}{N}\int \frac{1}{\sigma_i^2 - x} f_{\mu_{M_3}}(x)dx - \frac{1}{\sigma_i^2}\frac{n-N}{N} \\ &= -\frac{n}{N} m_{M_3}(\sigma_i^2) - \frac{1}{\sigma_i^2}\frac{n-N}{N}.\end{aligned}$$

7.3.3 Proof of Corollary 7.2.2

Equation (7.3) gives the relationship to find the largest singular value of Y_n. We can also use this equation to derive a boundary, below which this largest singular value behaves exactly as the noise-only case where $\theta = 0$. By Assumption 7.1.4, b is the supremum of the largest singular value of the noise only matrix. For a given $c_1 = \frac{n}{N}$ and $c_2 = \frac{m}{n}$, we must compute θ_{crit} such that (7.3) has the solution $\sigma = b$. Thus

$$\theta_{\text{crit}} = \frac{1}{b\sqrt{\varphi_F(b)\varphi_H(b)}}. \tag{7.7}$$

This is a function of c_1 and c_2 and changes depending on the type of random matrix G used.

7.4 Proof of Corollary 7.2.1

In this section, we develop closed form expressions for φ_F, φ_H when using a unitary projection. To determine these expressions, we rely on free probability theory and the utility RMTool [3]. This allows us to expression a random matrix, A, as bivariate polynomials $L^A_{mz}(m, z)$ such that the Stieltjes transform of A, $m_A(z)$, is the solution to the equation $L^A_{mz}(m, z) = 0$. This representation is extremely convenient because it allows us to perform standard matrix operations, such as addition and multiplication, in the polynomial space.

To compute φ_F and φ_H, we need the Stieltjes transform of $M_2 = QQ^H XX^H$, the Stieltjes transform of the kernel function of $R = QQ^H$ and $S = XX^H$ and the eigenvalue distribution of R. We consider X to be an appropriately scaled random Gaussian matrix whose entries are independent standard Gaussian random variables. The scaling is such that S is a Wishart random matrix with parameter c_1. The bivariate polynomial for this matrix is

$$L^S_{mz}(m,z) = -c_1 z m^2 + (1 - z - c_1)m - 1.$$

For the orthogonal setting, we assume that Q is a unitary matrix such that $Q^H Q = I_m$. With this formulation of Q, $R = QQ^H$ has a simple atomic eigenvalue distribution,

$$f_{\mu_R}(x) = \frac{n-m}{n}\mathbb{1}_{\{x=0\}} + \frac{m}{n}\mathbb{1}_{\{x=1\}},$$

where $\mathbb{1}$ is an indicator function. With this, we can easily compute the expected value needed for φ_F.

$$\begin{aligned} \varphi_F(\sigma_i) &= -\mathbb{E}\left[x m_{\mu_{RS|R}}\left(\sigma_i^2, x\right)\right]_{\mu_R} \\ &= \frac{m}{n} m_{\mu_{RS|R}}\left(\sigma_i^2, 1\right). \end{aligned}$$

We may use the RMTool function `AtimesBkernel` to compute $m_{\mu_{RS|R}}$. Doing so results in the following expression

$$\varphi_F(\sigma_i) = \frac{\sigma_i^2 - \ell(\sigma_i) + c_1 c_2 - 1}{2 c_1 \sigma_i^2}, \tag{7.8}$$

where

$$\ell(\sigma_i) = \sqrt{c_1^2 c_2^2 - 2c_1 c_2 \sigma_i^2 - 2 c_1 c_2 + \sigma_i^4 - 2\sigma_i^2 + 1}. \tag{7.9}$$

Similarly, we can use the RMTool function `AtimesB` to compute $m_{\mu_{M_2}}$, needed for φ_H. Doing so results in the following expression,

$$\begin{aligned} \varphi_H(\sigma_i) &= \frac{2c_1 + \sigma^2 - \ell(\sigma) - c_1 c_2 - 1}{2\sigma^2} - \frac{c_1 - 1}{\sigma_i^2} \\ &= \frac{\sigma_i^2 - \ell(\sigma_i) + c_1 c_2 - 1}{2\sigma_i^2}, \end{aligned} \tag{7.10}$$

where $\ell(\sigma)$ is defined in (7.9). Substituting (7.10) and (7.8) into (7.3) and performing the necessary algebra to solve for σ_i results in

$$\sigma_i \xrightarrow{\text{a.s.}} \sqrt{\frac{c_1}{\theta^2} + c_2\theta^2 + 1 + c_1 c_2}.$$

Next we solve for b_Q, the largest singular value of $Q^H X$, so that we may compute the phase transition. First we note that the largest singular of $Q^H X$ is the square root of the largest eigenvalue of M_2. This is convenient because we can compute the Stieltjes transform of M_2 using RMTool. The command

```
solve(feval(symengine,'polylib::discrim',lmzM2, m),'z')
```

gives the possible largest eigenvalues of M_2, the largest of which is the correct solution. This results in

$$b_Q = \sqrt{c_1 c_2} + 1. \tag{7.11}$$

Substituting b into φ_F and φ_H and using algebra to simplify results in

$$\varphi_H(b_Q) = 1, \quad \varphi_F(b_Q) = \sqrt{\frac{c_2}{c_1}}.$$

Substituting these expressions into (7.7), results in the phase transition

$$\theta_{\text{crit}} = \left(\frac{c_1}{c_2}\right)^{1/4}. \tag{7.12}$$

We may summarize all results via

$$\sigma_i \xrightarrow{\text{a.s.}} \begin{cases} \sqrt{\frac{c_1}{\theta^2} + c_2\theta^2 + 1 + c_1 c_2} & \text{if } \theta \geq \left(\frac{c_1}{c_2}\right)^{1/4} \\ \sqrt{c_1 c_2} + 1 & \text{if } \theta < \left(\frac{c_1}{c_2}\right)^{1/4} \end{cases}. \tag{7.13}$$

7.5 Empirical Results

In this section we verify the singular value prediction given in (7.3) that relies on the asymptotic approximations φ_F and φ_H. We consider two different types of projection matrices. In the first setting, we use a matrix G_n with independent $\mathcal{N}(0, 1)$ entries. In the second setting, we use a unitary matrix Q_n such that $Q_n^H Q_n = I_m$. In both settings, we let the noise matrix X_n be an appropriately scaled random Gaussian matrix whose entries are independent standard Gaussian random variables. In MATLAB, we generate X_n with

```
X = randn(n,N)/sqrt(N).
```

We first provide the necessary MATLAB code to solve for φ_F and φ_H in the Gaussian case. We then provide empirical results that showcase the accuracy of Theorem 7.2.1 for both the Gaussian and unitary matrices. Finally we compare the performance of each to showcase that the unitary projection matrix is uniformly better.

```
syms m;
lmzX = wishartpol(n/N);
lmzG = wishartpol(n/m_param);
lmzP = AtimesB(lmzX,lmzG);
kerA = AtimesBkernel(lmzG,lmzX);
m_kerA = solve(kerA,'m');
num_points = 2500;
max_g_pdf_point = (sqrt(n/m_param) + 12)^2+1;
pdfA = Lmz2pdf(lmzG,linspace(0,max_g_pdf_point,num_points));
spacing = max_g_pdf_point/num_points;
yintA = -subs(m_kerA,'z',(sig_lim^2));
yintxA = real(subs(yintA,pdfA.range));
yintxA(isnan(yintxA)) = 0;
yintxA(isinf(yintxA)) = 0;
phiF = yintxA*((pdfA.range').*(pdfA.density))*(spacing);
phiF = real(phiF(3));
```

Figure 7.3: MATLAB code to compute φ_F and φ_H for a Gaussian projection matrix and Gaussian noise matrix. This relies on function provided in RMTool [3].

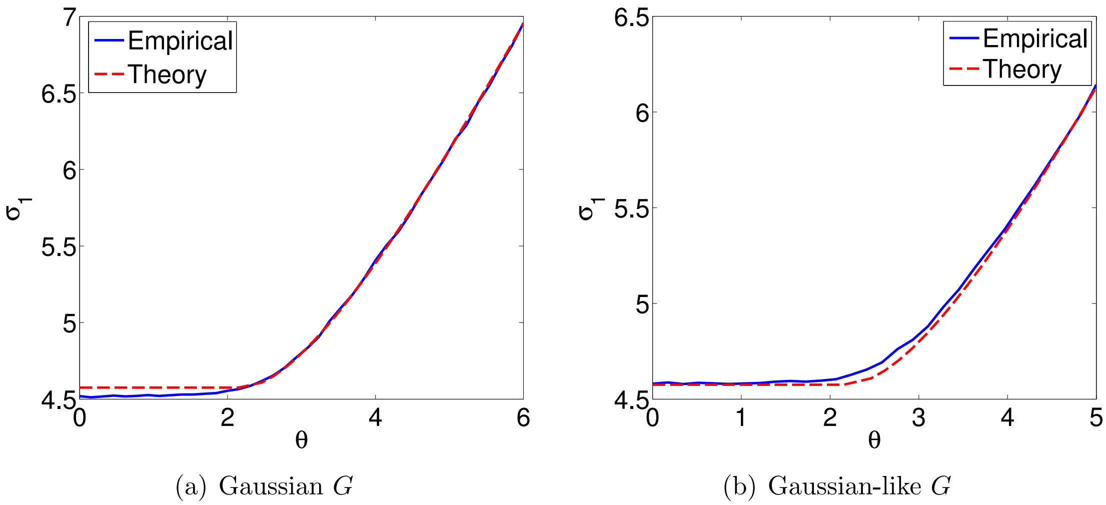
7.5.1 Gaussian Projection, G

In this setting, we generate G in the same way that we generate X. In MATLAB, this is accomplished with

```
G = randn(n,m)/sqrt(m).
```

With G defined this way, $R = GG^H$ and $S = XX^H$ are independent Wishart random matrices with parameters $c_1 = \frac{n}{m}$ and $c_2 = nN$. To compute φ_F and φ_H, we use RMTool. We numerically approximate the expected value using RMTool to approximate the density of R and to compute the Stieltjes transform of the kernel. We use 2500 points in the approximation. To compute φ_H, We consider the matrix $M_2 = GG^H XX^H = RS$, which is a product of Wishart matrices. This is desirable as M_2 is a product of Wishart random matrices and we can use RMTool to compute the Stieltjes transform as above for φ_H. The MATLAB code to for these approximations is given in Figure 7.3.

Figure 7.4(a) shows the performance of our theoretical prediction when using a



(a) Gaussian G

(b) Gaussian-like G

Figure 7.4: (a) Singular value prediction for Gaussian G and X for a rank-1 setting with fixed $n = 1000$, $N = 1220$ and $m = 100$. The theoretical prediction uses (7.3) with approximations from Figure 7.3. Empirical results are averaged over 500 trials. (b) Singular value prediction for Gaussian-like G and Gaussian X for a rank-1 setting with fixed $n = 1000$, $N = 1220$ and $m = 100$. Here, the entries of G are either ± 1 with equal probability. The theoretical prediction is the same for (a). Empirical results are again averaged over 500 trials.

Gaussian projection matrix, G, for a rank-1 setting with a fixed $n = 1000$, $N = 1220$, $m = 100$. In our empirical setup, we generate 500 matrices from (7.1) and 500 noise only matrices. We then generate a random G selected as above. The figure plots the empirical and theoretically predicted top singular value for a number of $\theta_1 = \theta$. The theoretical prediction does a good job except for one inaccurate point, which we attribute to the numerical instability of the process outlined in Figure 7.3 around the phase transition.

In Figure 7.4(b) we consider a Gaussian-like projection matrix for the same rank-1 setting as Figure 7.4(a). For this figure, the entries of G are

$$G_{ij} = \begin{cases} 1 & \text{w.p. } 1/2 \\ -1 & \text{w.p. } 1/2 \end{cases}$$

so that they have zero mean and unit variance. We see that the theoretical prediction from (7.3) for the Gaussian setting is still valid for this Gaussian-like projection matrix.

We then explore the accuracy of the phase transition boundary for the Gaussian setting in Figure 7.5. In the first row of this Figure, we plot the KS-statistic between the largest singular values from these 500 signal and noise only matrices. In the second row we plot the average top singular value of the signal matrix. All figures set $n = 1000$. The left column sweeps over θ and N while the right column sweeps over θ and m. In all figures, we plot our theoretical phase transition prediction in solid white line. Using a dashed white line, we plot the theoretical phase transition when no projection is used; this is $\theta = \left(\frac{n}{N}\right)^{1/4}$.

From this figure, we observe that the phase transition prediction is very accurate. Similarly we notice that the phase transition when using the Gaussian projection is significantly worse that that when not projecting. The figures in the left column set $m = 100$ so that we reduce our SVD dimension by one order of magnitude. Interestingly, and perhaps most importantly, when using a Gaussian projection matrix, setting $m = n = 1000$ results in worse performance than the non-projecting case even though we aren't reducing the dimension of the problem. This is evident in the right column.

7.5.2 Unitary Projection, Q

In this setting, we a unitary projection matrix Q using a QR decomposition of a random matrix. In MATLAB this is accomplished with

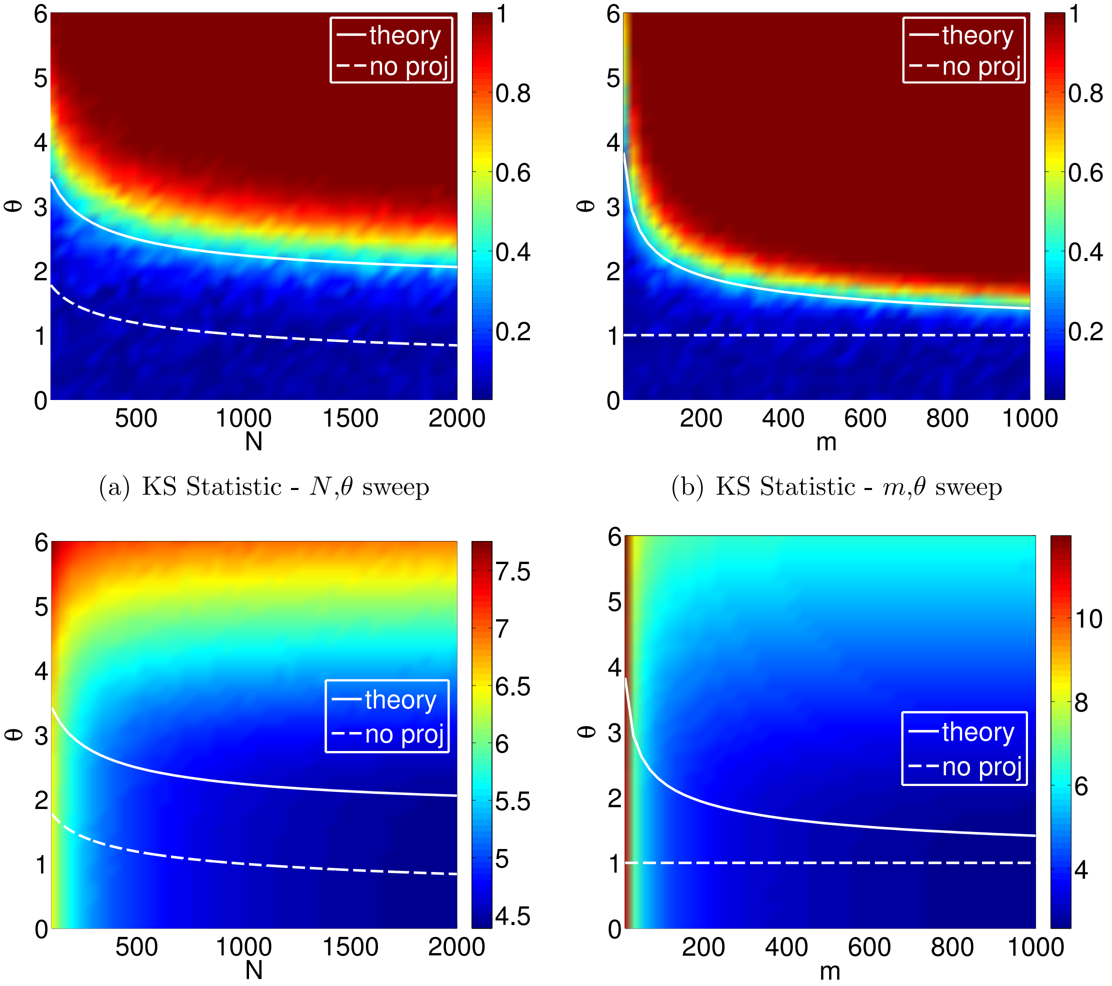
```
[Q,~] = qr(randn(n)); Q=Q(:,1:m).
```

Figure 7.6(a) plots the performance of our theoretical prediction when using a unitary projection matrix, Q, for a rank-1 seeting with a fixed $n = 1000$, $N = 1220$, $m = 100$. In our empirical setup, we generate 500 matrices from (7.1) and 500 noise only matrices. We then generate a random Q selected as above. The figure plots the empirical and theoretically predicted top singular value for a number of $\theta_1 = \theta$. The theoretical prediction uses the result from Corollary 7.2.1 and does an excellent job at the singular value prediction.

In Figure 7.6(b), we consider a specific choice of unitary matrix. Here, we randomly select columns from the $n \times n$ discrete Fourier matrix F with entries

$$F_{kj} = \frac{1}{\sqrt{n}} \exp\left\{\frac{-2\pi i(k-1)(j-1)}{n}\right\} \tag{7.14}$$

for $k = 1 \ldots, n$ and $j = 1, \ldots, n$. To generate Q we then select m columns from F. We see that the theoretical prediction from Corollary 7.2.1 still does an excellent job at the singular value prediction for this specific choice of unitary matrix.



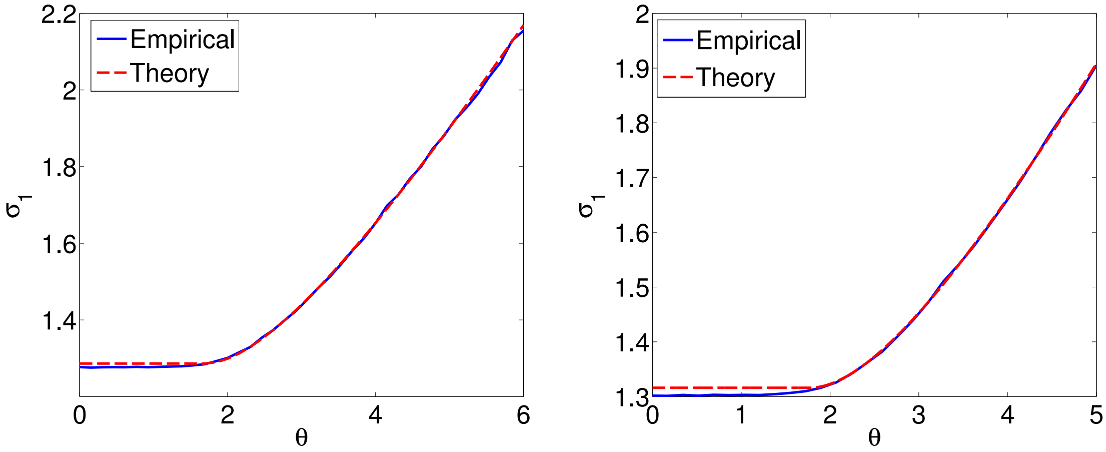
(a) KS Statistic - N,θ sweep

(b) KS Statistic - m,θ sweep

(c) Maximum singular value - N,θ sweep

(d) Maximum singular value - m, θ sweep

Figure 7.5: Performance of theoretical phase transition prediction for Gaussian G and X for a rank-1 setting with fixed $n = 1000$. The theoretical prediction uses (7.3) with approximations from Figure 7.3. The first row plots the KS statistic between singular values generated from 500 signal bearing and 500 noise only matrices. The bottom row plots the average empirical singular value averaged over 500 trials. The left column sweeps over both θ and N for a fixed $m = 100$ while the right column sweeps over θ and m for a fixed $N = 1000$.



(a) Unitary Q

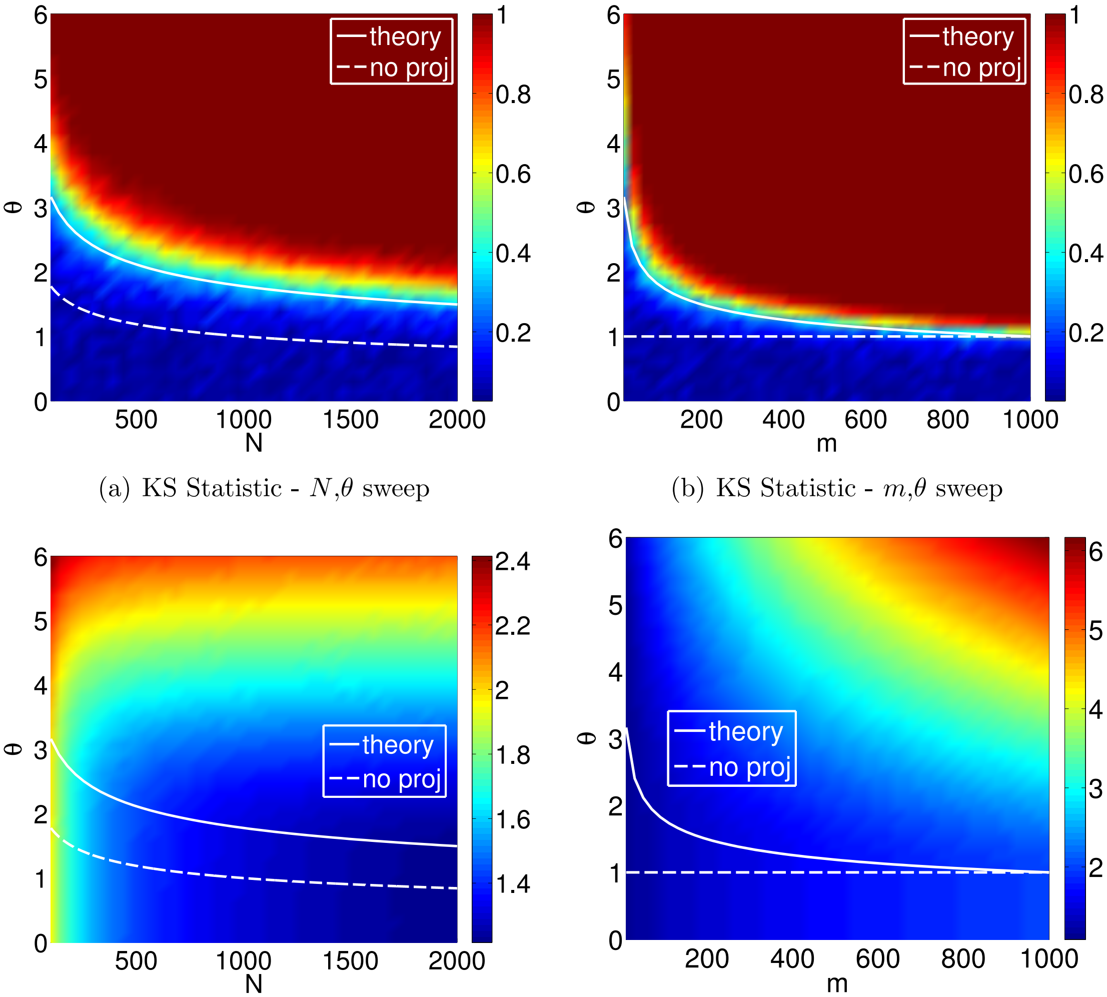
(b) Fourier Q

Figure 7.6: (a) Singular value prediction for unitary projection matrix Q and Gaussian noise matrix X for a rank-1 setting with fixed $n = 1000$, $N = 1220$ and $m = 100$. The theoretical prediction uses Corollary 7.2.1. Empirical results are averaged over 500 trials. (b) Singular value prediction for unitary-like matrix Q and Gaussian noise matrix X for a rank-1 setting with fixed $n = 1000$, $N = 1220$ and $m = 100$. Here, the columns of Q are sampled from the $n \times n$ discrete Fourier matrix defined in (7.14). The theoretical prediction is the same as (a) and uses Corollary 7.2.1. Empirical results are averaged over 500 trials.

Figure 7.7 plots the performance of our theoretical prediction when using a unitary projection matrix, Q. Our parameter sweep is the same as described for Figure 7.5, except that here we use the phase transition prediction given in Corollary 7.2.1. Again, we notice that our phase transition prediction is very accurate. A key observation is that for a unitary projection matrix, as $m \to n$, the phase transition approaches that of not using a projection matrix. This is very desirable as we don't want to suffer much performance loss for only slightly reducing the dimension of the problem.

7.5.3 Comparison

Here we discuss the difference between the two choices of projection matrices. Figure 7.8 shows the empirical difference of the KS statistic plots from Figures 7.5 and 7.7. On each plot we overlay the theoretical phase transition lines. The solid white is the prediction for a Gaussian projection matrix from (7.3) using the method in Figure 7.3; the dashed white is the prediction for an unitary projection matrix from Corollary 7.2.1; the solid black is the prediction when not using a projection ($\theta = c^{1/4}$). Positive values in these plots indicate that the top singular value using a Gaussian



(a) KS Statistic - N,θ sweep

(b) KS Statistic - m,θ sweep

(c) Maximum singular value - N,θ sweep

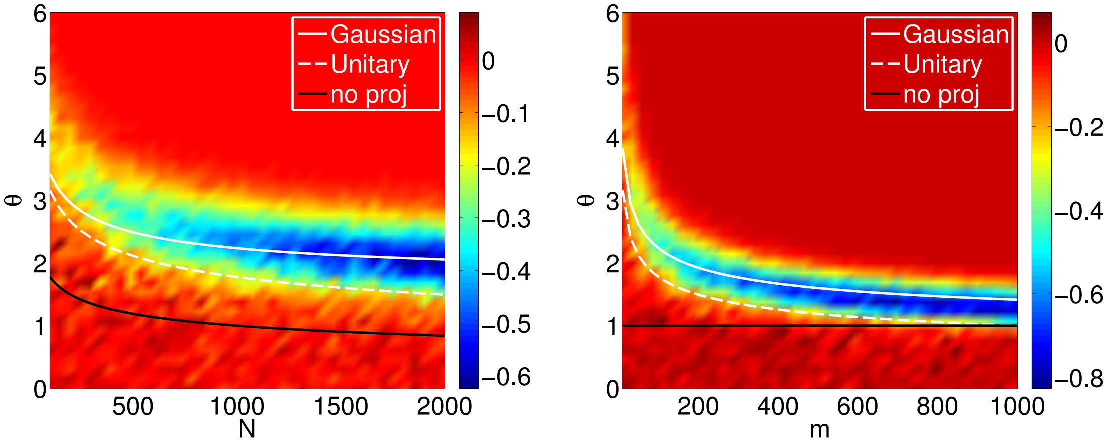
(d) Maximum singular value - m, θ sweep

Figure 7.7: Performance of theoretical phase transition prediction for unitary projection matrix Q and Gaussian noise matrix X for a rank-1 setting with fixed $n = 1000$. The theoretical prediction uses Corollary 7.2.1. The first row plots the KS statistic between singular values generated from 500 signal bearing and 500 noise only matrices. The bottom row plots the average empirical singular value averaged over 500 trials. The left column sweeps over both θ and N for a fixed $m = 100$ while the right column sweeps over θ and m for a fixed $N = 1000$.

projection matrix can more reliably detect our one signal; negative values indicate that the top singular value using a unitary projection matrix can more reliably detect our signal. The first column sweeps over N and θ for a fixed $m = 100$ and $n = 1000$ while the second column sweeps over m and θ for a fixed $N = 1000$ and $n = 1000$.

We see that the unitary projection matrix performs uniformly better than the Gaussian projection matrix above the phase transition. Below their respective phase transitions, all methods fail. Importantly, even when setting $m = n$ so that the projection doesn't reduce the dimension, the unitary projection keeps the same phase transition while the Gaussian projection changes the phase transition so that it is harder to detect the presence of a signal. This allows us to conclude that in terms of detection performance, the unitary projection matrix is better than the Gaussian projection matrix.

However, we do note that generating these projection matrices, particularly for large dimensions, is important. Generating the Gaussian projection matrix G is very easy as every entry is an independent Gaussian random variable. However, generating a $n \times m$ unitary matrix Q for high dimensions may be prohibitive. The analysis in this chapter gives the practitioner the ability to choose the projection matrix that best fits his or her needs. Given system parameters, the practitioner can select the projection dimension m to achieve a certain detection ability. The decision may be driven by the ease of creating each projection matrix.



(a) KS Statistic difference - N,θ sweep

(b) KS Statistic difference - m,θ sweep

Figure 7.8: Performance difference between using a Gaussian projection matrix, G, and a unitary projection matrix, Q, for a rank-1 setting with fixed $n = 1000$ and Gaussian noise matrix X. Positive values indicate that the Gaussian projection can more reliably detect the signal while negative values indicate that the unitary projection can more reliably detect the signal. We observe that the unitary projection outperforms the Gaussian projection.

CHAPTER VIII

CCA and ICCA for Regression and Detection

8.1 Introduction

In this chapter, we consider the classical problems of detection and regression in the multi-modal data setting. In such a setting, we assume that each dataset embeds signals in a low-rank subspace, but that the observations reside in a much higher dimensional space and are corrupted with noise. In this chapter, we show that when we know all parameters in the data model, the classical solutions to the detection and regression problems may be written in terms of the CCA canonical vectors and correlations. However, we show that empirical CCA, which relies on sample covariance estimates, fails to solve these problems in the low-sample, low-SNR regime. We then showcase that the ICCA solution to the detection and regression problems are equivalent to the standard plug-in solutions.

When there is only one dataset present, the detection problem reduces to the classical matched subspace detector (MSD). MSDs are used in fields such as array processing [69, 68], radar detection [71, 70], and handwriting recognition [78]. The performance of matched subspace detectors (MSDs) has been studied extensively when the signal subspace is known [76, 75, 74, 77] and in this thesis when the signal subspace is unknown. Here we explore the theory of MSDs when two multi-modal sets of observations are available. Since these datasets both describe the same system, one would hope that theoretically fusing feature vectors to account for correlations will result in better detection ability.

We are motivated by the work in [52], which shows that the canonical basis is the right basis to use in low-rank detection and estimation. Here, Pezeshki et al. consider the signal plus noise model where an observation from one dataset is available. This observation is a sum of a unknown low rank signal and Gaussian noise. They apply CCA using the observation as the first modality and the unknown signal as the second modality. We are interested in the different setting where we are presented with two datasets, each possibly containing a low rank signal buried in high dimensional noise. Their work on estimation is directly applicable and we extend their result to analyze the performance for a low-rank signal plus noise model when the model parameters are unknown.

We begin by providing the data model used throughout the rest of the thesis and the classical (maximum likelihood) estimates of unknown parameters. We then first consider the case when we know all of the parameters to show that classical regression and detection solutions may be written in terms of the CCA basis. We then predict mean squared error for different estimators when using parameter estimates. Finally, we use numerical simulations to demonstrate the extreme sub-optimality of empirical CCA compared to the plug-in LRT detector and estimator. Instead, using the ICCA basis in such algorithms results in the same performance as the plug-in LRT algorithm, giving credence to the previous idea that using only the informative components in data fusion is extremely important.

8.2 Data Model and Parameter Estimation

8.2.1 Training Data

Similar to the previous chapters in this thesis, we model our multi-modal data via

$$\begin{aligned} x_i &= U_x s_{x,i} + z_{x,i} \\ y_i &= U_y s_{y,i} + z_{y,i} \end{aligned} \tag{8.1}$$

where $U_x^H U_x = I_{k_x}$, $U_y^H U_y = I_{k_y}$, $z_{x,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_p)$ and $z_{y,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_q)$. Furthermore, assume that

$$\begin{aligned} s_{x,i} &\sim \mathcal{CN}(0, \Theta_x) \\ s_{y,i} &\sim \mathcal{CN}(0, \Theta_y), \end{aligned}$$

where $\Theta_x = \mathbf{diag}\left(\left(\theta_1^{(x)}\right)^2, \dots, \left(\theta_{k_x}^{(x)}\right)^2\right)$ and $\Theta_y = \mathbf{diag}\left(\left(\theta_1^{(y)}\right)^2, \dots, \left(\theta_{k_y}^{(y)}\right)^2\right)$. Assume that $z_{x,i}$ and $z_{y,i}$ are mutually independent and independent from both $s_{x,i}$ and $s_{y,i}$. Finally, assume that

$$\mathbb{E}\left[s_{x,i} s_{y,i}^H\right] =: K_{xy} = \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}$$

where the entries of P_{xy} are $-1 \leq \rho_{kj} \leq 1$ and represent the correlation between $s_{x,i}^{(k)}$ and $s_{y,i}^{(j)}$. For reasons to be made clear later, define

$$\widetilde{K}_{xy} = \left(\Theta_x + I_{k_x}\right)^{-1/2} K_{xy} \left(\Theta_y + I_{k_y}\right)^{-1/2}$$

and define the singular values of $\widetilde{K}_{xy}$ as $\kappa_1, \ldots, \kappa_{\min(k_x,k_y)}$. Under this model, we define the following covariance matrices

$$\begin{aligned} \mathbb{E}\left[x_i x_i^H\right] &= U_x \Theta_x U_x^H + I_p =: R_{xx} \\ \mathbb{E}\left[y_i y_i^H\right] &= U_y \Theta_y U_y^H + I_q =: R_{yy} \\ \mathbb{E}\left[x_i y_i^H\right] &= U_x K_{xy} U_y^H =: R_{xy}. \end{aligned} \tag{8.2}$$

Let $w_i = \left[x_i^H \, y_i^H\right]^H$ be the joint observation vector, $d = p + q$ be the dimension of w_i, and $k = k_x + k_y \ll d$ be the combined rank of the two low rank signal subspaces.

8.2.2 Parameter Estimation

Assume that we are given n observations of each dataset, $x_1, \ldots, x_n$, and $y_1, \ldots, y_n$. We stack these observations into two training data matrices $X = [x_1, \ldots, x_n]$, and $Y = [y_1, \ldots, y_n]$. We assume that k_x and k_y are known. Let $Q_x D_x V_x^H$ be the SVD of $\frac{1}{\sqrt{n}}X$ and let $Q_y D_y V_y^H$ be the SVD of $\frac{1}{\sqrt{n}}Y$. The maximum likelihood (ML) estimates of our unknown parameters are

$$
\begin{aligned}
\widehat{U}_x \quad &= Q_x(:, 1:k_x) \\
\widehat{U}_y \quad &= Q_y(:, 1:k_y) \\
\widehat{U} \quad &= \begin{bmatrix} \widehat{U}_x & 0 \\ 0 & \widehat{U}_y \end{bmatrix} \\
\widehat{\Theta}_x \quad &= D_x^2(1:k_x, 1:k_x) - I_{k_x} \\
\widehat{\Theta}_y \quad &= D_y^2(1:k_y, 1:k_y) - I_{k_y} \\
\widehat{\Theta} \quad &= \begin{bmatrix} \widehat{\Theta}_x & 0 \\ 0 & \widehat{\Theta}_y \end{bmatrix} \\
\widehat{P}_{xy} \quad &= \widehat{\Theta}_x^{-1/2} \widehat{U}_x^H \frac{1}{n} X Y^H \widehat{U}_y \widehat{\Theta}_y^{-1/2} \\
&= \widehat{\Theta}_x^{-1/2} D_x(1:k_x, :) V_x^H V_y D_x(1:k_y, :)^H \widehat{\Theta}_y^{-1/2} \\
&= \widehat{\Theta}_x^{-1/2} \left(\widehat{\Theta}_x + I_{k_x} \right)^{1/2} V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y} \right)^{1/2} \widehat{\Theta}_y^{-1/2} \\
\widehat{\widehat{P}} \quad &= \begin{bmatrix} I_{k_x} & \widehat{P}_{xy} \\ \widehat{P}_{xy}^H & I_{k_y} \end{bmatrix}.
\end{aligned}
\tag{8.3}
$$

8.2.3 Testing Data

For the regression problem, we generate testing observations x and y from (8.1). However, we only observe y and our goal is to estimate x from y. In the detection problem, we generate testing observations x and y from either a noise only or signal plus noise model

$$
\begin{aligned}
&\text{Noise only,} \qquad && H_0 : \begin{cases} x_i = z_{x,i} \\ y_i = z_{y,i} \end{cases} \\
&\text{Signal plus noise,} \qquad && H_1 : \begin{cases} x_i = U_x s_{x,i} + z_{x,i} \\ y_i = U_y s_{y,i} + z_{y,i} \end{cases}
\end{aligned}
\tag{8.4}
$$

where the parameters are modeled the same as the training model in (8.1).

8.3 Standard Regression Techniques

In this section, we will explore standard regression techniques using the prior information of our data model and using canonical correlation analysis (CCA) and informative CCA (ICCA). Specifically, we will compute the theoretical mean squared error (MSE) of each method. A key observation of these derivations is that the MSE computation relies on insights from random matrix theory even if the predictor does not. In the Gaussian setting of our testing data, the maximum likelihood estimator of x given y is

$$\widehat{x} = R_{xy}R_{yy}^{-1}y. \tag{8.5}$$

Based on data model in (8.1), this estimator is

$$\begin{aligned}\widehat{x} &= U_xK_{xy}U_y^H\left(U_y\Theta_yU_y^H + I_{k_y}\right)^{-1}y \\ &= U_xK_{xy}\left(\Theta_y + I_{k_y}\right)^{-1}U_y^Hy.\end{aligned} \tag{8.6}$$

8.3.1 Plug-in Predictor

By substituting the parameter estimates in (8.3) into the ML estimator in (8.6) we arrive at the standard plug-in predictor

$$\widehat{x}_{\text{plugin}} = \widehat{U}_x\widehat{\Theta}_x^{1/2}\widehat{P}_{xy}\widehat{\Theta}_y^{1/2}\left(\widehat{\Theta}_y + I_{k_y}\right)^{-1}\widehat{U}_yy. \tag{8.7}$$

8.3.2 Prediction using CCA, empirical CCA, and ICCA

We first use basic definitions of Canonical Correlation Analysis (CCA) to rewrite (8.5) in terms of canonical vectors and canonical correlations. Recall from previous chapters that CCA takes the SVD of the matrix

$$C_{\text{cca}} = R_{xx}^{-1/2}R_{xy}R_{yy}^{-1/2},$$

and we write this SVD as $C_{\text{cca}} = FKG^H$. The singular values of the matrix are exactly the canonical correlations. To recover the corresponding canonical vectors, we make the transformations

$$\begin{aligned}W_x &= R_{xx}^{-1/2}F \\ W_y &= R_{yy}^{-1/2}G,\end{aligned}$$

where $W_x \in \mathbb{C}^{p\times p}$ and $W_y \in \mathbb{C}^{q\times q}$. The columns of these matrices are exactly the canonical vectors. Therefore, we may estimate x using the canonical basis by observing that

$$\begin{aligned}
\widehat{x} &= R_{xy}R_{yy}^{-1}y \\
R_{xx}^{-1/2}\widehat{x} &= R_{xx}^{-1/2}R_{xy}R_{yy}^{-1/2}R_{yy}^{-1/2}y \\
R_{xx}^{-1/2}\widehat{x} &= C_{\text{cca}}R_{yy}^{-1/2}y \\
R_{xx}^{-1/2}\widehat{x} &= FKG^{H}R_{yy}^{-1/2}y \\
F^{H}R_{xx}^{-1/2}\widehat{x} &= KG^{H}R_{yy}^{-1/2}y \\
W_x^{H}\widehat{x} &= KW_y^{H}y.
\end{aligned}$$

From this derivation, we see that the canonical correlation basis is the correct basis to use for regression. Essentially, CCA gives a linear prediction model for each of the canonical variates

$$w_x^{(i)H}x = \rho^{(i)}w_y^{(i)H}y.$$

We know from our model in (8.1) that the datasets have at most $r =: \min(k_x, k_y)$ correlated components. With this observation and abusing notation, redefine $K = K(1:r, 1:r)$, $W_x = W_x(:, 1:r)$, and $W_y = W_y(:, 1:r)$. Therefore, the CCA predictor when the canonical correlations and canonical vectors are known *a priori* is

$$\widehat{x}_{\text{cca}} = \left(W_x^H\right)^{\dagger} KW_y^H y.$$

However, we do not know the canonical correlations and canonical vectors *a priori* and must estimate them from data. This results in the following empirical predictors,

$$\begin{aligned}
\widehat{x}_{\text{cca}} &= \left(\widehat{W}_{x,\text{cca}}^{H}\right)^{\dagger} \widehat{K}_{\text{cca}}\widehat{W}_{y,\text{cca}}^{T}y \\
\widehat{x}_{\text{icca}} &= \left(\widehat{W}_{x,\text{icca}}^{H}\right)^{\dagger} \widehat{K}_{\text{icca}}\widehat{W}_{y,\text{icca}}^{H}y,
\end{aligned}$$

where the estimated canonical correlations and vectors for CCA and ICCA are found in a similar manner as in Chapters IV and V.

8.4 Random Matrix Theory Preliminaries

In this section, we state previous results in random matrix theory to help us characterize the accuracy of our parameter estimates in (8.3). Our first proposition characterizes the accuracy of our subspace estimates $\widehat{U}_x$ and $\widehat{U}_y$.

Proposition 8.4.1. *Given our training data model in (8.1), as $n, p, q \to \infty$ with*

$p/n \to c_x$ and $q/n \to c_y$,

$$\begin{aligned} \left|\langle u_x^{(i)}, \widehat{u}_x^{(i)} \rangle\right|^2 &\xrightarrow{a.s.} \begin{cases} \frac{\theta_x^{(i)4} - c_x}{\theta_x^{(i)4} + \theta_x^{(i)2} c_x} & \textit{if } \left(\theta_x^{(i)}\right)^2 > \sqrt{c_x} \\ 0 & \textit{otherwise} \end{cases} \\ \left|\langle u_y^{(i)}, \widehat{u}_y^{(i)} \rangle\right|^2 &\xrightarrow{a.s.} \begin{cases} \frac{\theta_y^{(i)4} - c_y}{\theta_y^{(i)4} + \theta_y^{(i)2} c_y} & \textit{if } \left(\theta_y^{(i)}\right)^2 > \sqrt{c_y} \\ 0 & \textit{otherwise} \end{cases} . \end{aligned} \tag{8.8}$$

Proof. See Theorem 4 of [84] and Theorem 2.2 of [85]. □

A key observation in the proposition is that if the SNR governing a subspace component drops below a critical value, dependent only on the dimension of the dataset and the number of samples, then that estimated subspace component in *uninformative.* A similar results characterizes the accuracy of the SNR estimates $\widehat{\Theta}_x$ and $\widehat{\Theta}_y$.

Proposition 8.4.2. *Given our training data model in (8.1), as $n, p, q \to \infty$ with $p/n \to c_x$ and $q/n \to c_y$,*

$$\begin{aligned} \widehat{\theta}_x^{(i)} &\xrightarrow{a.s.} \begin{cases} \sqrt{\sigma_x^{(i)2} + c_x + \frac{c_x}{\theta_x^{(i)2}}} & \textit{if } \theta_x^{(i)2} > \sqrt{c_x} \\ \sqrt{c_x + 2\sqrt{c_x}} & \textit{otherwise} \end{cases} \\ \widehat{\theta}_y^{(i)} &\xrightarrow{a.s.} \begin{cases} \sqrt{\theta_y^{(i)2} + c_y + \frac{c_y}{\theta_y^{(i)2}}} & \textit{if } \theta_y^{(i)2} > \sqrt{c_y} \\ \sqrt{c_y + 2\sqrt{c_y}} & \textit{otherwise} \end{cases} . \end{aligned} \tag{8.9}$$

Proof. See Theorems 1 and 2 in [84] for the real setting for $c_x < 1$ and $c_y < 1$. See Theorem 2.6 in [86] for the complete result. □

Propositions 8.4.1 and 8.4.2 both reveal a phase transition in our estimates. When the SNR is below a critical value, our estimates behave truly randomly and our signal subspace estimates contain no information and our SNR estimates behave as the largest singular value of a noise-only matrix. Next we present two propositions to characterize the limit of the CCA and ICCA canonical correlations.

Proposition 8.4.3. *Let $n, p, q \to \infty$ such that $p/n \to c_x$ and $q/n \to c_y$. Assume that $p + q < n$. For $i = 1, \ldots, \min(k_x, k_y)$ let $\widehat{\rho}_{cca}^{(i)}$ be the largest singular singular values of*

$\widehat{C}_{cca}$ *generated from data modeled in (8.1). Then these singular values behaves as*

$$\widehat{\rho}_{cca}^{(i)} \xrightarrow{a.s.} \begin{cases} \sqrt{\kappa_i^2 \left(1 - c_x + \frac{c_x}{\kappa_i^2}\right)\left(1 - c_y + \frac{c_y}{\kappa_i^2}\right)} & \kappa_i^2 \geq r_c \\ \sqrt{d_r} & \kappa_i^2 < r_c \end{cases} \tag{8.10}$$

where κ_i *are the singular values of* $\widetilde{K}_{xy}$ *and*

$$\begin{aligned} r_c &= \frac{c_x c_y + \sqrt{c_y c_y (1-c_x)(1-c_y)}}{(1-c_x)(1-c_y) + \sqrt{c_x c_y (1-c_x)(1-c_y)}} \\ d_r &= c_x + c_y - 2c_x c_y + 2\sqrt{c_x c_y (1-c_x)(1-c_y)}. \end{aligned} \tag{8.11}$$

Proof. Bao et al. [2] proved this result for a slightly simplified model. See Chapter 4 for a short derivation and Appendix C for a lengthy derivation using our own notation. □

The estimated empirical canonical correlations also exhibit a phase transition that is dependent on the dimensions of both datasets and the number of samples available. An important consequence, first shown by [6] shows that when $n < p + q$, $\widehat{\rho}_{\text{cca}}^{(i)} = 1$. Next we characterize the ICCA canonical correlation estimates.

Proposition 8.4.4. *Let* $p, q, n \to \infty$ *with* $p/n \to c_x$ *and* $q/n \to c_y$*. Define*

$$\varphi_x^{(i)} = \begin{cases} \sqrt{1 - \left(c_x + \theta_x^{(i)2}\right) / \left(\theta_x^{(i)4} + \theta_x^{(i)2}\right)} & \text{if } \left(\theta_x^{(i)}\right)^2 > \sqrt{c_x} \\ 0 & \text{otherwise} \end{cases}$$

$$\varphi_y^{(i)} = \begin{cases} \sqrt{1 - \left(c_y + \theta_y^{(i)2}\right) / \left(\theta_y^{(i)4} + \theta_y^{(i)2}\right)} & \text{if } \left(\theta_y^{(i)}\right)^2 > \sqrt{c_y} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\left[V_x^H V_y\right]_{ij} \xrightarrow{a.s.} \varphi_x^{(i)} \left[P_{xy}\right]_{ij} \varphi_y^{(i)}.$$

The singular values of this matrix limit are the almost sure limit of the ICCA canonical correlation estimates $\widehat{\rho}_{icca}^{(i)}$*.*

Proof. See [8] for a derivation of this result for the rank 1 case. See Corollary 4.6.1 of this thesis for a complete result. □

Both ICCA and CCA exhibit a phase transition where the canonical correlation is deterministic. In this regime, the ICCA correlation estimate is 0 while the CCA correlation estimate is non-zero. Finally, we characterize the accuracy of the canonical vectors in ICCA and CCA. We derive the CCA accuracy in Appendix C, but do not have a closed form expression for this result yet. Below is the accuracy for the ICCA vector as derived in Chapter 5.

Proposition 8.4.5. *Let $p, n \to \infty$ with $p/n \to c_x$. Then*

$$\left|\left\langle \frac{w_x^{(i)}}{\|w_x^{(i)}\|_2}, \frac{\widehat{w}_x^{(i)}}{\|\widehat{w}_x^{(i)}\|_2} \right\rangle\right|^2 \xrightarrow{a.s.} \frac{\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{\alpha_j}{\sqrt{\left(\theta_j^{(x)}\right)^2+1}\sqrt{\left(\widehat{\theta}_x^{(j)}\right)^2+1}}\right)^2}{\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\theta_j^{(x)}\right)^2+1}\right)\left(\sum_{j=1}^{k_x} \left(U_{\widetilde{K}}^{(i)}\right)_j^2 \frac{1}{\left(\widehat{\theta}_x^{(j)}\right)^2+1}\right)},$$

where $\widehat{\theta}_x^{(j)}$ is defined above and

$$\alpha_i = \left|\langle u_x^{(i)}, \widehat{u}_x^{(i)} \rangle\right|$$

A similar results holds for the accuracy of $\widehat{w}_y$.

Proof. See Chapter 5 for a derivation of this result □

8.5 Theoretical MSE Derivations

In this section, we derive the theoretical mean squared error (MSE) for the plug-in, CCA, and ICCA predictors derived in Section 8.3. These derivations rely on the expressions presented in Section 8.4. Given a predictor, the MSE is

$$\begin{aligned} \text{MSE} &= \mathbb{E}\left[(x-\widehat{x})^T (x-\widehat{x})\right] \\ &= \mathbb{E}\left[x^T x\right] - 2\mathbb{E}\left[x^T \widehat{x}\right] + \mathbb{E}\left[\widehat{x}^T \widehat{x}\right]. \end{aligned} \tag{8.12}$$

The first term above is only dependent on our data model,

$$\begin{aligned} \mathbb{E}\left[x^T x\right] &= \mathbb{E}\left[(U_x s_x + z_x)^H (U_x s_x + z_x)\right] \\ &= \mathbb{E}\left[s_x^H s_x\right] + 2\mathbb{E}\left[s_x^H U_x^H z_x\right] + \mathbb{E}\left[z_x^H z_x\right] \\ &= \sum_{i=1}^{k_x} \left(\theta_i^{(x)}\right)^2 + 0 + p \\ &= p + \sum_{i=1}^{k_x} \left(\theta_i^{(x)}\right)^2. \end{aligned} \tag{8.13}$$

The other terms are dependent on the individual predictors and we compute them individually next. To do so, we make the following definitions to ease notation.

$$\begin{aligned} A_x^u &= \mathbf{diag}\left(\left|\left\langle u_x^{(i)}, \widehat{u}_x^{(i)}\right\rangle\right|\right) \\ A_y^u &= \mathbf{diag}\left(\left|\left\langle u_y^{(i)}, \widehat{u}_y^{(i)}\right\rangle\right|\right) \\ A_x^v &= \mathbf{diag}\left(\varphi_x^{(i)}\right) \\ A_y^v &= \mathbf{diag}\left(\varphi_y^{(i)}\right). \end{aligned}$$

Also, where it is clear, we drop the ICCA and CCA subscripts.

8.5.1 ICCA

Before computing the necessary terms in MSE, we note that

$$V_x^H V_y = \widehat{U}_{\widetilde{K}} K \widehat{V}_{\widetilde{K}},$$

and our canonical vectors are

$$\begin{aligned} W_x &= \widehat{R}_{xx}^{-1/2} \widehat{U}_x \widehat{U}_{\widetilde{K}} = \widehat{U}_x \left(\widehat{\Theta}_x + I_{k_x}\right)^{-1/2} \widehat{U}_{\widetilde{K}} \\ W_y &= \widehat{R}_{yy}^{-1/2} \widehat{U}_y \widehat{V}_{\widetilde{K}} = \widehat{U}_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{V}_{\widetilde{K}} \end{aligned}$$

so that

$$\begin{aligned} \left(W_x^H\right)^\dagger &= W_x \left(W_x^H W_x\right)^{-1} \\ &= \widehat{U}_x \left(\widehat{\Theta}_x + I_{k_x}\right)^{-1/2} \widehat{U}_{\widetilde{K}} \left(\widehat{U}_{\widetilde{K}}^H \left(\widehat{\Theta}_x + I_{k_x}\right)^{-1} \widehat{U}_{\widetilde{K}}\right)^{-1} \\ &= \widehat{U}_x \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} \widehat{U}_{\widetilde{K}} \end{aligned}$$

Therefore, the first ICCA MSE component is

$$\begin{aligned}
\mathbb{E}\left[x^H \widehat{x}_{\text{icca}}\right] &= \mathbb{E}\left[\left(U_x s_x + z_x\right)^H \left(W_x^H\right)^\dagger K W_y^H \left(U_y s_y + z_y\right)\right] \\
&= \mathbb{E}\left[s_x^H U_x^H \left(W_x^H\right)^\dagger K W_y^H U_y s_y\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(s_x^H U_x^H \left(W_x^H\right)^\dagger K W_y^H U_y s_y\right)\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(U_x^H \left(W_x^H\right)^\dagger K W_y^H U_y s_y s_x^H\right)\right] \\
&= \mathbf{tr}\left(U_x^H \left(W_x^H\right)^\dagger K W_y^H U_y \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}\right) \\
&= \mathbf{tr}\left(U_x^H \widehat{U}_x \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} \widehat{U}_{\widetilde{K}} K \widehat{V}_{\widetilde{K}}^H \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H U_y \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}\right) \\
&= \mathbf{tr}\left(A_x^u \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^u \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}\right).
\end{aligned}$$

Similarly, we have the following equivalencies for $\mathbb{E}\left[\widehat{x}_{\text{icca}}^H \widehat{x}_{\text{icca}}\right]$,

$$\begin{aligned}
&= \mathbb{E}\left[\left(\left(W_x^H\right)^\dagger K W_y^H \left(U_y s_y + z_y\right)\right)^H \left(\left(W_x^H\right)^\dagger K W_y^H \left(U_y s_y + z_y\right)\right)\right] \\
&= \mathbb{E}\left[z_y^H W_y K^H \left(W_x^H\right)^{\dagger H} \left(W_x^H\right)^\dagger K W_y^H z_y\right] + \\
&\quad \mathbb{E}\left[s_y^H U_y^H W_y K^H \left(W_x^H\right)^{\dagger H} \left(W_x^H\right)^\dagger K W_y^H U_y s_y\right] \\
&= \mathbf{tr}\left(W_y K^H \left(W_x^H\right)^{\dagger H} \left(W_x^H\right)^\dagger K W_y^H\right) + \\
&\quad \mathbf{tr}\left(U_y^H W_y K^H \left(W_x^H\right)^{\dagger H} \left(W_x^H\right)^\dagger K W_y^H U_y \Theta_y\right) \\
&= \mathbf{tr}\left(W_y K^H \widehat{U}_{\widetilde{K}}^H \left(\widehat{\Theta}_x + I_{k_x}\right) \widehat{U}_{\widetilde{K}} K W_y^H\right) + \\
&\quad \mathbf{tr}\left(U_y^H W_y K^H \widehat{U}_{\widetilde{K}}^H \left(\widehat{\Theta}_x + I_{k_x}\right) \widehat{U}_{\widetilde{K}} K W_y^H U_y \Theta_y\right) \\
&= \mathbf{tr}\left(\widehat{U}_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{V}_{\widetilde{K}} K^H \widehat{U}_{\widetilde{K}}^H \left(\widehat{\Theta}_x + I_{k_x}\right) \widehat{U}_{\widetilde{K}} K \widehat{V}_{\widetilde{K}}^H \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H\right) + \\
&\quad \mathbf{tr}\left(U_y^H \widehat{U}_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{V}_{\widetilde{K}} K^H \widehat{U}_{\widetilde{K}}^H \left(\widehat{\Theta}_x + I_{k_x}\right) \widehat{U}_{\widetilde{K}} K \widehat{V}_{\widetilde{K}}^H \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H U_y \Theta_y\right) \\
&= \mathbf{tr}\left(\left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^v P_{xy} A_x^v \left(\widehat{\Theta}_x + I_{k_x}\right) A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2}\right) + \\
&\quad \mathbf{tr}\left(A_y^u \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^v P_{xy} A_x^v \left(\widehat{\Theta}_x + I_{k_x}\right) A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^u \Theta_y\right).
\end{aligned}$$

We substitute these expressions and (8.13) into (8.12) to arrive at the ICCA MSE prediction.

8.5.2 CCA

CCA does not first trim data matrices and so our canonical vectors are more complicated as they contain all components of our original data matrices. In this setting, our canonical vectors are

$$\begin{aligned} W_x &= Q_x D_x \widehat{U}_{\widetilde{K}} \\ W_y &= Q_y D_y \widehat{V}_{\widetilde{K}} \end{aligned}$$

where here $\widehat{U}_{\widetilde{K}} \in \mathbb{C}^{p\times k}$ and $\widehat{V}_{\widetilde{K}} \in \mathbb{C}q \times k$ are the first k left and right singular vectors of the $\min(n,p) \times \min(n,q)$ matrix $V_x^H V_y$. Unlike in ICCA, this matrix is large and contains all right singular vectors of our original data matrix. In Appendix C, we derive (non-closed form) expressions for the accuracy of the CCA canonical vectors. The expressions derived in the appendix are for unit norm vectors, but the steps in the derivations solve for the expressions $W_x^H \widehat{W}_x$ and $\widehat{W}_x^H \widehat{W}_x$, and similarly for the canonical vectors of dataset Y. Therefore, in the following derivations, we leave these expressions in this form and refer the reader to the appendix. Again, we note that we do not have closed form expressions for these types of terms and leave this to future work. We have the following equivalencies for $\mathbb{E}\left[x^H \widehat{x}_{\text{cca}}\right]$

$$\begin{aligned}
&= \mathbb{E}\left[(U_x s_x + z_x)^H \left(\widehat{W}_x^H\right)^{\dagger} K \widehat{W}_y^H \left(U_y s_y + z_y\right)\right] \\
&= \mathbb{E}\left[s_x^H U_x^H \left(\widehat{W}_x^H\right)^{\dagger} K \widehat{W}_y^H U_y s_y\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(U_x^H \left(\widehat{W}_x^H\right)^{\dagger} K \widehat{W}_y^H U_y s_y s_x^H\right)\right] \\
&= \mathbf{tr}\left(U_x^H \left(\widehat{W}_x^H\right)^{\dagger} K \widehat{W}_y^H U_y \Theta_y^{1/2} P_{xy}^H \Theta_x^{1/2}\right) \\
&= \mathbf{tr}\left(U_x^H \widehat{W}_x \left(\widehat{W}_x^H \widehat{W}_x\right)^{-1} K \widehat{W}_y^H U_y \Theta_y^{1/2} P_{xy}^H \Theta_x^{1/2}\right) \\
&= \mathbf{tr}\left(\left(\Theta_x + I_{k_x}\right)^{1/2} U_{\widetilde{K}} W_x^H \widehat{W}_x \left(\widehat{W}_x^H \widehat{W}_x\right)^{-1} K \widehat{W}_y^H W_y \widehat{V}_{\widetilde{K}}^H \left(\Theta_y + I_{k_y}\right)^{1/2} \Theta_y^{1/2} P_{xy}^H \Theta_x^{1/2}\right).
\end{aligned}$$

The above expression relies on the model parameters Θ_x, Θ_y, P_{xy}, $U_{\widetilde{K}}$, $V_{\widetilde{K}}$ and CCA expressions that we know by proposition (K) or appendix derivations (canonical vector accuracy). Next we derive an expression for the last term needed in the CCA

MSE derivation. We have the following equivalencies for $\mathbb{E}\left[\widehat{x}_{\text{cca}}^H \widehat{x}_{\text{cca}}\right]$

$$\begin{aligned}
&= \mathbb{E}\left[\left(\left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H \left(U_y s_y + z_y\right)\right)^H \left(\left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H \left(U_y s_y + z_y\right)\right)\right] \\
&= \mathbb{E}\left[s_y^H U_y^H \widehat{W}_y K \left(\widehat{W}_x^H\right)^{\dagger H} \left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H U_y s_y\right] \\
&\quad + \mathbb{E}\left[z_y^H \widehat{W}_y K \left(\widehat{W}_x^H\right)^{\dagger H} \left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H z_y\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(U_y^H \widehat{W}_y K \left(\widehat{W}_x^H\right)^{\dagger H} \left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H U_y s_y s_y^H\right)\right] + \\
&\quad \mathbb{E}\left[\mathbf{tr}\left(\widehat{W}_y K \left(\widehat{W}_x^H\right)^{\dagger H} \left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H z_y z_y^H\right)\right] \\
&= \mathbf{tr}\left(U_y^H \widehat{W}_y K \left(\widehat{W}_x^H\right)^{\dagger H} \left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H U_y \Theta_y\right) + \\
&\quad \mathbf{tr}\left(\widehat{W}_y K \left(\widehat{W}_x^H\right)^{\dagger H} \left(\widehat{W}_x^H\right)^{\dagger} K\widehat{W}_y^H\right) \\
&= \mathbf{tr}\left(U_y^H \widehat{W}_y K \left(\widehat{W}_x^H \widehat{W}_x\right)^{-1} K\widehat{W}_y^H U_y \Theta_y\right) + \\
&\quad \mathbf{tr}\left(K \left(\widehat{W}_x^H \widehat{W}_x\right)^{-1} K\widehat{W}_y^H \widehat{W}_y\right) \\
&= \mathbf{tr}\left(\left(\Theta_y + I_{k_y}\right)^{1/2} V_{\widetilde{K}} W_y^H \widehat{W}_y K \left(\widehat{W}_x^H \widehat{W}_x\right)^{-1} K\widehat{W}_y^H W_y V_{\widetilde{K}}^H \left(\Theta_y + I_{k_y}\right)^{1/2} \Theta_y\right) + \\
&\quad \mathbf{tr}\left(K \left(\widehat{W}_x^H \widehat{W}_x\right)^{-1} K\widehat{W}_y^H \widehat{W}_y\right).
\end{aligned}$$

We substitute these expressions and (8.13) into (8.12) to arrive at the CCA MSE prediction.

8.5.3 Plug-in

For the first expression needed in the MSE derivation, we have the following equivalencies for $\mathbb{E}\left[x^H \widehat{x}_{\text{plugin}}\right]$,

$$\begin{aligned}
&= \mathbb{E}\left[(U_x s_x + z_x)^H \widehat{U}_x \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H \left(U_y s_y + z_y\right)\right] \\
&= \mathbb{E}\left[s_x^H U_x^H \widehat{U}_x \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H U_y s_y\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(s_x^H A_x^u \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^u s_y\right)\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(A_x^u \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^u s_y s_x^H\right)\right] \\
&= \mathbf{tr}\left(A_x^u \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^u \Theta_y^{1/2} P_{xy}^H \Theta_x^{1/2}\right).
\end{aligned}$$

For the second expression, we have the following equivalencies for $\mathbb{E}\left[\widehat{x}_{\text{plugin}}^H \widehat{x}_{\text{plugin}}\right]$,

$$\begin{aligned}
&= \mathbb{E}\left[s_y^H U_y^H \widehat{U}_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} V_y^H V_x \left(\widehat{\Theta}_x + I_{k_x}\right) V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H U_y s_y\right] + \\
&\quad \mathbb{E}\left[z_y^H \widehat{U}_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} V_y^H V_x \left(\widehat{\Theta}_x + I_{k_x}\right) V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} \widehat{U}_y^H z_y\right] \\
&= \mathbf{tr}\left(A_y^u \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^v P_{xy}^H A_x^v \left(\widehat{\Theta}_x + I_{k_x}\right) A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^u \Theta_y\right) + \\
&\quad \mathbf{tr}\left(\left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2} A_y^v P_{xy}^H A_x^v \left(\widehat{\Theta}_x + I_{k_x}\right) A_x^v P_{xy} A_y^v \left(\widehat{\Theta}_y + I_{k_y}\right)^{-1/2}\right)
\end{aligned}$$

We substitute these expressions and (8.13) into (8.12) to arrive at the plugin MSE prediction. Most importantly, when examining these expressions, we see that the ICCA predictor achieves the same MSE as the plug-in predictor.

8.6 Rank-1 Empirical Results

To verify our theoretical MSE predictions, we generate data from (8.1) in a rank-1 setting where $k_x = k_y = 1$. We set $\theta_1^{(x)} = 3$, $\theta_1^{(y)} = 4$, $p = 100$, $q = 200$ and $\rho = 0.9$. In the rank-1 setting, $U_{\widetilde{K}} = V_{\widetilde{K}} = 1$. We then sweep over various values of the number of training samples we are given, n. For each value of n, we use the training samples from (8.1) to train the plug-in, CCA, and ICCA parameters. Then, for 1000 testing points, we generate both x and y from (8.1) and use y and our estimated parameters to predict x. We repeat this process 250 times for each value of n to average over noise and different subspaces. We plot the empirical and theoretical MSE curves in Figure 8.1.

Our predictions do a decent job at predicting the MSE for all three estimators. First, we note that empirically and theoretically, the ICCA estimator achieves the same performance as the plug-in estimator. This is wonderful as it matched our theoretical observation. Second, we note that the CCA estimator is extremely suboptimal, not only because it achieves a larger MSE than the other two estiamtors, but also because its MSE does not monotonically decrease with increased n. This non-monotonicity is centered around $n = q = 200$. Examining the second term in the expression for $\widehat{x}_{\text{cca}}^H \widehat{x}_{\text{cca}}$, we see that we need the expression for $\widehat{W}_y^H \widehat{W}_y$. When $n < p+q$, the singular vectors, f and g, of $U_x V_x^H V_y U_y^H$ used in the CCA computation of $\widehat{W}_y$ are random. Therefore, in this regime

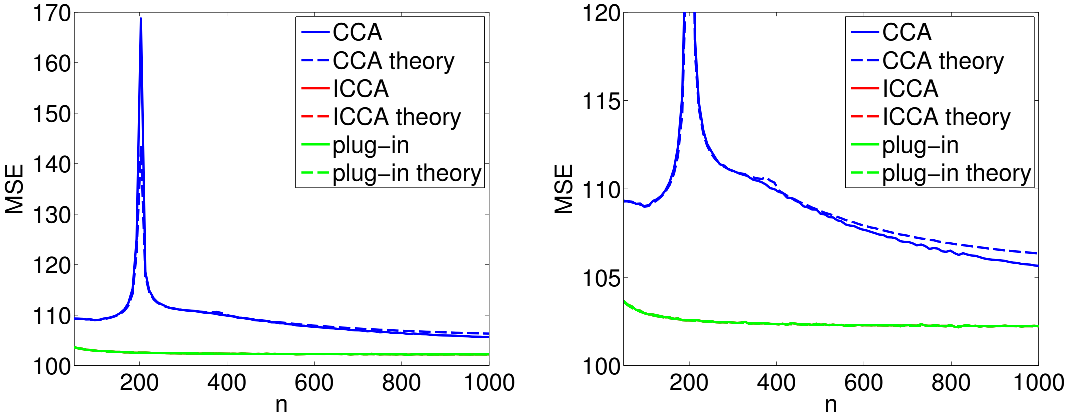
$$\begin{aligned}
\widehat{W}_y^H \widehat{W}_y &= g^H \widehat{R}_{yy}^{-1} g \\
&= \frac{1}{q} \sum_{i=1}^{n} \sigma_i \left(\left(\frac{1}{n} Y Y^H \right)^{-1} \right) \\
&\to \mathbb{E} \left[\sigma_i \left(\left(\frac{1}{n} Y Y^H \right)^{-1} \right) \right] \\
&= \max \left(\frac{1}{1 - c_y}, \frac{1}{c_y - 1} \right).
\end{aligned}$$

As $n \to q$, $c_y \to 1$ and this above expression tends to infinity. We empirically observe this singularity, which we are able to predict. Finally, we note that in certain regimes our empirical CCA prediction is not entirely accurate. We attribute this to the fact that for empirical CCA, we do not have closed form expressions for the accuracy of the canonical vectors. The appendix makes a number of approximations for the unknown quantities and for these plots, we simulated random matrices to generate these unknown quantities.

8.7 LRT Detector Derivation

Formally, we are given two observation vectors, x and y, of different modalities (having different features). The goal is to design a detector to distinguish between the H_1 hypothesis that the observations contain a target signal and the H_0 hypothesis that the observations are purely noise.

We consider the Neyman-Pearson setting for detection (see [93]) where, given



(a) Empirical and Theoretical MSE

(b) Zoomed-in

Figure 8.1: Empirical and theoretically predicted MSE for the plug-in, CCA, and ICCA estimators. We use a rank-1 setting where $k_x = k_y = 1$, $\theta_1^{(x)} = 3$, $\theta_1^{(y)} = 4$, $p = 100$, $q = 200$ and $\rho = 0.9$. In the rank-1 setting, $U_{\widetilde{K}} = V_{\widetilde{K}} = 1$. The plot on the right is a zoomed in version of the plot on the left. The ICCA and plug-in curves lie on top of each other, as we showed that the two are equivalent.

a test observations from (8.4), we form $w = [x^H y^H]^H$ by stacking the individual observations in a column vector. The Neyman-Pearson lemma states that such a detector takes the form of a LRT

$$\Lambda(w) := \frac{f\left(w \,|\, H_1\right)}{f\left(w \,|\, H_0\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \tag{8.14}$$

where $\Lambda(w)$ is a test statistic, γ is a threshold set to achieve a desired false alarm rate, and f is the appropriate conditional density of the observation.

The conditional distributions of w under each hypothesis are

$$\begin{aligned} w|H_0 &\sim \mathcal{N}(0, I_d) \\ w|H_1 &\sim \mathcal{N}(0, R_w), \end{aligned}$$

where $R_w = \mathbb{E}\left[ww^H\right]$. Substituting these conditional distributions in (8.14) , the LRT statistic is

$$\Lambda(w) = \frac{\mathcal{N}(0, R_w)}{\mathcal{N}(0, I_d)},$$

which can be simplified to

$$\Lambda(w) = w^H \left(I_d - R_w^{-1}\right) w. \tag{8.15}$$

The covariance matrix of the observation vector is

$$\begin{aligned}
R_w &= \begin{bmatrix} R_{xx} & R_{xy} \\ R_{xy}^H & R_{yy} \end{bmatrix} = \begin{bmatrix} U_x\Theta_x U_x^H + I_p & U_x K_{xy} U_y^H \\ U_y K_{xy}^H U_x^H & U_y \Theta_y U_y^H + I_q \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} U_x & 0 \\ 0 & U_y \end{bmatrix}}_{U} \underbrace{\underbrace{\begin{bmatrix} \Theta_x^{1/2} & 0 \\ 0 & \Theta_y^{1/2} \end{bmatrix}}_{\Theta^{1/2}} \underbrace{\begin{bmatrix} I_{k_x} & P_{xy} \\ P_{xy}^H & I_{k_y} \end{bmatrix}}_{\widetilde{P}} \underbrace{\begin{bmatrix} \Theta_x^{H/2} & 0 \\ 0 & \Theta_y^{H/2} \end{bmatrix}}_{\Theta^{H/2}}}_{\widetilde{\Theta}} \underbrace{\begin{bmatrix} U_x^H & 0 \\ 0 & U_y^H \end{bmatrix}}_{U^H} + I_d \\
&= U\Theta^{1/2}\widetilde{P}\Theta^{H/2}U^H + I_d \\
&= U\widetilde{\Theta}U^H + I_d.
\end{aligned}$$

Substituting this covariance matrix into the LRT statistic in (8.15) yields (using the matrix inversion lemma)

$$\begin{aligned}
\Lambda_{\text{lrt}}(w) &= w^H\left(I_d - R_w^{-1}\right)w \\
&= w^H\left(I_d - \left(U\widetilde{\Theta}U^H + I_d\right)^{-1}\right)w \\
&= w^H\left(I_d - \left(I_d - U\left(\widetilde{\Theta}^{-1} + U^H U\right)\right)^{-1}U^H\right)w \\
&= w^H U\left(\widetilde{\Theta}^{-1} + I_k\right)^{-1}Uw \\
&= w^H U\left(\Theta^{-1/2}\widetilde{P}^{-1}\Theta^{-1/2} + I_k\right)^{-1}U^H w \\
&= w^H U\Theta^{1/2}\left(\widetilde{P}^{-1} + \Theta\right)^{-1}\Theta^{H/2}U^H w.
\end{aligned}$$

The LRT detector is

$$\Lambda_{\text{lrt}}(w) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{lrt}} \tag{8.16}$$

where γ_{lrt} is a threshold set to satisfy

$$\mathbb{P}\left(\Lambda_{\text{lrt}}(w) > \gamma_{\text{lrt}} \,|\, H_0\right) = \alpha,$$

where α is a desired false alarm rate.

Writing the LRT statistic in this form is desirable for computational reasons. Instead of inverting R_w, which is a $d \times d$ matrix of high dimension, we only need to invert $k \times k$ matrices.

8.8 CCA Detector Equivalency

In this section, we will show that the LRT derived above in (8.16) can be written using the canonical vectors and correlation coefficients found by CCA.Recall that the matrix of interest in CCA is $C_{\text{cca}} = R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2}$ and that the canonical vectors and correlation coefficients are found by solving the SVD of $C_{\text{cca}} = FKG^H$. We begin by manipulating the covariance matrix of w.

$$\begin{aligned} R_w &= \begin{bmatrix} R_{xx} & R_{xy} \\ R_{xy}^H & R_{yy} \end{bmatrix} = \begin{bmatrix} R_{xx}^{1/2} & 0 \\ 0 & R_{yy}^{1/2} \end{bmatrix} \begin{bmatrix} I_p & C_{\text{cca}} \\ C_{\text{cca}}^H & I_q \end{bmatrix} \begin{bmatrix} R_{xx}^{H/2} & 0 \\ 0 & R_{yy}^{H/2} \end{bmatrix} \\ &= \begin{bmatrix} R_{xx}^{1/2} & 0 \\ 0 & R_{yy}^{1/2} \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} I_p & K \\ K^H & I_q \end{bmatrix} \begin{bmatrix} F^H & 0 \\ 0 & G^H \end{bmatrix} \begin{bmatrix} R_{xx}^{H/2} & 0 \\ 0 & R_{yy}^{H/2} \end{bmatrix}. \end{aligned}$$

Using this decomposition, the inverse of the covariance matrix of w is

$$R_w^{-1} = \begin{bmatrix} R_{xx}^{-1/2} & 0 \\ 0 & R_{yy}^{-1/2} \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} I_p & K \\ K^H & I_q \end{bmatrix}^{-1} \begin{bmatrix} F^H & 0 \\ 0 & G^H \end{bmatrix} \begin{bmatrix} R_{xx}^{-H/2} & 0 \\ 0 & R_{yy}^{-H/2} \end{bmatrix}.$$

Recall that the i-th canonical vectors returned by CCA are

$$\begin{aligned} w_x^{(i)} &= R_{xx}^{-1/2} f_i \\ w_y^{(i)} &= R_{yy}^{-1/2} g_i \end{aligned}$$

where f_i and g_i are the left and right singular vectors of C corresponding to the i-th largest singular value, k_i, respectively. Define the matrices

$$\begin{aligned} W_x &= \left[w_x^{(1)}, \ldots, w_x^{(p)}\right] = R_{xx}^{-1/2} F \\ W_y &= \left[w_y^{(1)}, \ldots, w_y^{(q)}\right] = R_{yy}^{-1/2} G \end{aligned}$$

to be the matrices of canonical vectors returned by CCA. Using this notation and substituting the expression for R_y^{-1} in the LRT statistic in (8.15), we arrive at

$$\begin{aligned} \Lambda(w) &= w^H \left(I_d - R_w^{-1}\right) w \\ &= w^H \left(I_d - \begin{bmatrix} W_x & 0 \\ 0 & W_y \end{bmatrix} \begin{bmatrix} I_p & K \\ K^H & I_q \end{bmatrix}^{-1} \begin{bmatrix} W_x^H & 0 \\ 0 & W_y^H \end{bmatrix} \right) y. \end{aligned}$$

The above expression is written in terms of the observation w, the canonical vectors W_x and W_y and the correlation coefficients K returned by CCA. This statistic is exactly equivalent to the LRT statistic derived earlier. Therefore, we conclude that the CCA basis is the correct basis to use in such low-rank Gauss-Gauss detection with two datasets.

We can write this detector slightly differently by recalling that the canonical variates are $\xi_x^{(i)} = w_x^{(i)H}x$ and $\xi_y^{(i)} = w_y^{(i)H}y$. Let $\xi_x = \left[\xi_x^{(1)}, \ldots, \xi_x^{(p)}\right]^H$, $\xi_y = \left[\xi_y^{(1)}, \ldots, \xi_y^{(q)}\right]^H$, and define

$$\xi = \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} = \begin{bmatrix} W_x^H & 0 \\ 0 & W_y^H \end{bmatrix} w.$$

Using this definition and defining

$$W = \begin{bmatrix} W_x & 0 \\ 0 & W_y \end{bmatrix},$$

the above detector may be written

$$\Lambda_{\text{cca}}(\xi) = \xi^H \left(\left(W^H W\right)^{-1} - \begin{bmatrix} I_p & K \\ K^H & I_q \end{bmatrix}^{-1} \right) \xi. \tag{8.17}$$

In conclusion, we derived a detector that takes the canonical variates as inputs and uses only the canonical vectors X and the canonical correlation coefficients K in its test statistic. This detector is

$$\Lambda_{\text{cca}}(\xi) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{cca}} \tag{8.18}$$

where $\Lambda_{\text{cca}}(\xi)$ is defined in (8.17) and γ_{cca} is a threshold set to satisfy

$$\mathbb{P}\left(\Lambda_{\text{cca}}(\xi) > \gamma_{\text{cca}} \,|\, H_0\right) = \alpha,$$

where α is the desired false alarm rate. The CCA detector in (8.18) is equivalent to the LRT detector in (8.16). This is a general proof and is independent of any data models placed on w. That is, in this proof, we did not refer to the data model in (8.4) that motivated the problem.

8.8.1 CCA Detector for Data Model (8.4)

The above CCA detector was derived for a generic data model. Here we find the canonical vectors and correlation coefficients for the data model described in (8.4). Under this model, the data covariance matrices are defined in (8.2) and their inverses are

$$
\begin{aligned}
R_{xx}^{-1} &= \begin{bmatrix} U_x & U_x^{\perp} \end{bmatrix} \begin{bmatrix} (\Theta_x + I_{k_x})^{-1} & 0 \\ 0 & I_{p-k_x} \end{bmatrix} \begin{bmatrix} U_x^H \\ U_x^{H\perp} \end{bmatrix} \\
R_{yy}^{-1} &= \begin{bmatrix} U_y & U_y^{\perp} \end{bmatrix} \begin{bmatrix} \left(\Theta_y + I_{k_y}\right)^{-1} & 0 \\ 0 & I_{q-k_y} \end{bmatrix} \begin{bmatrix} U_y^H \\ U_y^{H\perp} \end{bmatrix}.
\end{aligned}
$$

It follows that the CCA matrix C_{cca} is

$$
\begin{aligned}
C_{\text{cca}} &= R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2} \\
&= U_x \left(\Theta_x + I_{k_x}\right)^{-1/2} \Theta_x^{1/2} K_{xy} \Theta_y^{1/2} \left(\Theta_y + I_{k_y}\right)^{-1/2} U_y^H \qquad (8.19) \\
&= U_x \widetilde{K}_{xy} U_y^H.
\end{aligned}
$$

Clearly, when expressed in (8.19), C_{cca} is a $\min\left(k_x, k_y\right)$ rank matrix. This implies that there are only $r := \min\left(k_x, k_y\right)$ non-zero correlation coefficients. Therefore, there are only r canonical vectors that should be used in a detector. Define

$$
\begin{aligned}
\widetilde{W}_x &= W_x(:, 1:r) \\
\widetilde{W}_y &= W_y(:, 1:r) \\
\widetilde{K} &= K(1:r, 1:r)
\end{aligned}
$$

as the trimmed canonical vectors and correlation coefficients. Finally define

$$
\widetilde{W} = \begin{bmatrix} \widetilde{W}_x & 0 \\ 0 & \widetilde{W}_y \end{bmatrix}
$$

and $\widetilde{\xi} = \widetilde{W}^H w$. Then the CCA detector is

$$
\Lambda_{\text{cca}}(\widetilde{\xi}) = \widetilde{\xi}^H \left(\left(\widetilde{W}^H \widetilde{W} \right)^{-1} - \begin{bmatrix} I_r & \widetilde{K} \\ \widetilde{K}^H & I_r \end{bmatrix}^{-1} \right) \widetilde{\xi}, \qquad (8.20)
$$

which only uses the r nonzero CCA correlation coefficients and corresponding canonical vectors. The matrix inverses here are also much easier to compute as the matrices are only $2r \times 2r$ instead of $d \times d$.

8.9 Empirical Detectors

In many applications, the signal matrices U_x, U_y, their SNR matrices Θ_x, Θ_y, and the correlation matrix between datasets P_{xy} are unknown and thus the resulting data covariance matrices are unknown. Therefore, neither the LRT statistic in (8.16) or the CCA statistic in (8.20), which relies on C_{cca} in (8.19), can be computed. In such settings, we are given training data to estimate any unknown parameters. This section will describe how to estimate these unknown parameters and use these estimates in the previously derived detectors. We then will describe how to use ICCA for detection and show its equivalence to the plug-in LRT detector. Finally, we close with numerical simulations demonstrating that the ICCA detector achieves the same performance as the plug-in LRT and that the empirical CCA detector is extremely suboptimal.

8.9.1 Plug-in Detector

To form a realizable LRT detector, we plug-in the parameter estimates in (8.3) into the statistic in (8.15). This results in the plug-in LRT statistic

$$\Lambda_{\text{plugin}}(w) = w^H \widehat{U}\widehat{\Theta}^{1/2}\left(\widehat{\widetilde{P}}^{-1} + \widehat{\Theta}\right)^{-1}\widehat{\Theta}^{H/2}\widehat{U}w. \tag{8.21}$$

8.9.2 Empirical CCA Detector

Similarly, we create a realizable CCA detector by performing empirical CCA as described in Section 8.3.2 by forming

$$\widehat{C}_{\text{cca}} = \widehat{R}_{xx}^{-1/2}\widehat{R}_{xy}\widehat{R}_{yy}^{-1/2} = Q_x I_{p\times n} V_x^H V_y I_{n\times q} Q_y^H.$$

We then use the largest r singular values and corresponding left and right singular vectors of $\widehat{C}_{\text{cca}}$ to form estimates of the canonical vectors and correlation coefficients. Specifically, let $\widehat{F} = [\widehat{f}_1, \ldots \widehat{f}_r]$ and $\widehat{G} = [\widehat{g}_1, \ldots, \widehat{g}_r]$ be the left and right singular vectors corresponding to the largest r singular values $\widehat{\kappa}_1, \ldots, \widehat{\kappa}_r$. Then the estimates of the canonical vectors and correlation coefficient are

$$
\begin{aligned}
\widehat{\widetilde{K}} &= \mathbf{diag}(\widehat{\kappa}_1, \ldots, \widehat{\kappa}_r) \\
\widehat{\widetilde{W}} &= \left[\begin{array}{cc} \widehat{R}_{xx}^{-1/2}\widehat{F} & 0 \\ 0 & \widehat{R}_{yy}^{-1/2}\widehat{G} \end{array}\right] \\
\widehat{\xi} &= \widehat{\widetilde{W}}^H w.
\end{aligned}
\tag{8.22}
$$

We then substitute these estimates into the CCA detector in (8.20). This results in the empirical CCA detector statistic

$$
\Lambda_{\text{cca}}(\widehat{\xi}) = \widehat{\xi}^H \left(\left(\widehat{\widetilde{W}}^H \widehat{\widetilde{W}} \right)^{-1} - \left[\begin{array}{cc} I_r & \widehat{\widetilde{K}} \\ \widehat{\widetilde{K}} & I_r \end{array}\right]^{-1} \right) \widehat{\xi}. \tag{8.23}
$$

8.9.3 ICCA Detector

We saw in Chapter 4 that empirical CCA is suboptimal and that we can avoid much of the performance loss of CCA by informatively trimming data components before computing the canonical vectors and correlations. We apply these insights here to form an ICCA detector. We instead form the matrix

$$
\widehat{C}_{\text{icca}} = Q_x(:, 1:k_x)V_x(:, 1:k_x)^H V_y(:, 1:k_y)Q_y(:, 1:k_y)^H
$$

and take the top r singular values $\widetilde{\kappa}_1, \ldots, \widetilde{\kappa}_r$ and corresponding singular vectors $\widetilde{F} = [\widetilde{f}_1, \ldots, \widetilde{f}_r]$ and $\widetilde{G} = [\widetilde{g}_1, \ldots, \widetilde{g}_r]$. Using this rank-$r$ SVD, we form informative canonical vectors and correlation coefficient similarly as in (8.22). Substituting these informative parameters into the CCA detector in (8.20) results in the ICCA detector statistic

$$
\Lambda_{\text{icca}}(\widetilde{\xi}) = \widetilde{\xi}^H \left(\left(\widehat{\widetilde{W}}^H \widehat{\widetilde{W}} \right)^{-1} - \left[\begin{array}{cc} I_r & \widehat{\widetilde{K}} \\ \widehat{\widetilde{K}} & I_r \end{array}\right]^{-1} \right) \widetilde{\xi}. \tag{8.24}
$$

8.9.4 Proof that $\Lambda_{\textbf{icca}}(\widetilde{\xi}) \equiv \Lambda_{\textbf{plug-in}}(w)$

In this section, we prove that the ICCA detector statistic in (8.24) is equivalent to the plug-in LRT statistic in (8.21). We begin by manipulating the plug-in detector, relying heavily on the Woodbury matrix inversion lemma. First we re-write the plug-in detector statistic

$$\begin{aligned}
\Lambda_{\text{plugin}}(w) &= w^H \widehat{U} \widehat{\Theta}^{1/2} \left(\widehat{\widetilde{P}}^{-1} + \widehat{\Theta} \right)^{-1} \widehat{\Theta}^{H/2} \widehat{U} w \\
&= w^H \widehat{U} \left(\widehat{\Theta}^{-1/2} \widehat{\widetilde{P}}^{-1} \widehat{\Theta}^{-1/2} + I_{2r} \right)^{-1} \widehat{U} w \\
&= w^H \widehat{U} \left(I_{2r} - \left(\widehat{\Theta}^{1/2} \widehat{\widetilde{P}} \widehat{\Theta}^{1/2} + I_{2r} \right)^{-1} \right) \widehat{U} w.
\end{aligned}$$

Next we manipulate the ICCA detector statistic. To do this, we need expressions for the canonical vectors and correlations in terms of our estimated parameters. First define

$$\widetilde{C}_{\text{icca}} = V_x(:, 1 : k_x)^H V_y(:, 1 : k_y)$$

and its SVD $\widetilde{C}_{\text{icca}} = \widehat{U}_{\widetilde{K}} \widehat{\widetilde{K}} \widehat{V}_{\widetilde{K}}$. Note that these singular values are exactly the singular values of $\widehat{C}_{\text{icca}}$. Therefore, we have that

$$\widehat{\widetilde{W}} = \widehat{U} \left(\widehat{\Theta} + I_{2k} \right)^{-1/2} \underbrace{\begin{bmatrix} \widehat{U}_{\widetilde{K}} & 0 \\ 0 & \widehat{V}_{\widetilde{K}} \end{bmatrix}}_{Q_{\widetilde{K}}}.$$

Therefore

$$\widehat{\widetilde{W}}^H \widehat{\widetilde{W}} = Q_{\widetilde{K}}^H \left(\widehat{\Theta} + I_{2k}) \right)^{-1} Q_{\widetilde{K}}$$

and

$$\left(\widehat{\widetilde{W}}^H \widehat{\widetilde{W}} \right)^{-1} = Q_{\widetilde{K}}^H \left(\widehat{\Theta} + I_{2k}) \right) Q_{\widetilde{K}}.$$

Also note that

$$\widetilde{\xi} = \widehat{\widetilde{W}}^H w.$$

Substituting these expressions into (8.24), we have

$$\begin{aligned}
\Lambda_{\text{icca}}(w) \quad &= \widetilde{\xi}^H \left(\left(\widehat{\widetilde{W}}^H \widehat{\widetilde{W}} \right)^{-1} - \left[\begin{array}{cc} I_r & \widehat{\widetilde{K}} \\ \widehat{\widetilde{K}} & I_r \end{array} \right]^{-1} \right) \widetilde{\xi}. \\
&= w^H \widehat{\widetilde{W}} \left(\left(\widehat{\widetilde{W}}^H \widehat{\widetilde{W}} \right)^{-1} - \left[\begin{array}{cc} I_r & \widehat{\widetilde{K}} \\ \widehat{\widetilde{K}} & I_r \end{array} \right]^{-1} \right) \widehat{\widetilde{W}}^H w. \\
&= w^H \widehat{U} \left(\widehat{\Theta} + I_{2r} \right)^{-1/2} Q_{\widetilde{K}} \left(Q_{\widetilde{K}}^H \left(\widehat{\Theta} + I_{2r} \right) \right) Q_{\widetilde{K}} - \left[\begin{array}{cc} I_r & \widehat{\widetilde{K}} \\ \widehat{\widetilde{K}} & I_r \end{array} \right]^{-1} \Bigg) \cdot \\
&\quad Q_{\widetilde{K}}^H \left(\widehat{\Theta} + I_{2r} \right)^{-1/2} \widehat{U}^H w \\
&= w^H \widehat{U} \left(\widehat{\Theta} + I_{2r} \right)^{-1/2} \left(\left(\widehat{\Theta} + I_{2r} \right) - Q_{\widetilde{K}} \left[\begin{array}{cc} I_r & \widehat{\widetilde{K}} \\ \widehat{\widetilde{K}} & I_r \end{array} \right]^{-1} Q_{\widetilde{K}}^H \right) \cdot \\
&\quad \left(\widehat{\Theta} + I_{2r} \right)^{-1/2} \widehat{U}^H w \\
&= w^H \widehat{U} \left(I_{2r} - \left(\widehat{\Theta} + I_{2r} \right)^{-1/2} \left[\begin{array}{cc} I_r & \widetilde{C}_{\text{icca}} \\ \widetilde{C}_{\text{icca}}^H & I_r \end{array} \right]^{-1} \left(\widehat{\Theta} + I_{2r} \right)^{-1/2} \right) \widehat{U}^H w \\
&= w^H \widehat{U} \left(I_{2r} - (M + I_{2r})^{-1} \right) \widehat{U}^H w
\end{aligned}$$

where

$$M = \left[\begin{array}{cc} \widehat{\Theta}_x & \left(\widehat{\Theta}_x + I_{2r} \right)^{-1/2} \widetilde{C}_{\text{icca}} \left(\widehat{\Theta}_y + I_{2r} \right)^{1/2} \\ \left(\widehat{\Theta}_y + I_{2r} \right)^{-1/2} \widetilde{C}_{\text{icca}}^H \left(\widehat{\Theta}_x + I_{2r} \right)^{1/2} & \widehat{\Theta}_y \end{array} \right].$$

Therefore, we must show that $M = \widehat{\Theta}^{1/2} \widehat{\widetilde{P}} \widehat{\Theta}^{1/2}$. The block diagonal entries of $\widehat{\Theta}^{1/2} \widehat{\widetilde{P}} \widehat{\Theta}^{1/2}$ are exactly $\widehat{\Theta}_x$ and $\widehat{\Theta}_y$. Therefore, to complete the proof, we must show that

$$\widehat{\Theta}_x^{1/2} \widehat{P}_{xy} \widehat{\Theta}_y^{1/2} = \left(\widehat{\Theta}_x + I_{2r} \right)^{1/2} \widetilde{C}_{\text{icca}} \left(\widehat{\Theta}_y + I_{2r} \right)^{1/2}$$

Substituting the definition of $\widehat{P}_{xy}$, we have

$$\begin{aligned}\widehat{\Theta}_x^{1/2}\widehat{P}_{xy}\widehat{\Theta}_y^{1/2} &= \widehat{\Theta}_x^{1/2}\left(\widehat{\Theta}_x^{-1/2}\left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{1/2}\widehat{\Theta}_y^{-1/2}\right)\widehat{\Theta}_y^{1/2}\\ &= \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} V_x^H V_y \left(\widehat{\Theta}_y + I_{k_y}\right)^{1/2}\\ &= \left(\widehat{\Theta}_x + I_{k_x}\right)^{1/2} \widetilde{C}_{\text{icca}} \left(\widehat{\Theta}_y + I_{k_y}\right)^{1/2},\end{aligned}$$

which completes the proof.

8.9.5 Rank 1 Numerical Simulations

We now use numerical simulations to explore the performance of the plug-in LRT detector in (8.21), the empirical CCA detector in (8.23), and the ICCA detector in (8.24) in the rank-1 setting where $k_x = k_y = 1$. Specifically, we wish to empirically verify that the plug-in LRT detector is equivalent to the ICCA detector. We also wish to explore how the performance of the CCA detector in compares to that of the plug-in LRT detector.

To compare the performance of these detectors, we compute empirical ROC curves. To compute an empirical ROC curve, we first generate two random signal vectors, $U_x = u_x$ and $U_y = u_y$, by taking the first left singular vector of two appropriately sized random matrices with i.i.d. $\mathcal{N}(0, 1)$ entries. In this simulation we make the simplifying assumption that $\theta_1^{(x)} = \theta_1^{(y)} = \theta$. Given a desired SNR, correlation $\rho = P_{xy}$, and random u_x and u_y, we generate n training samples of x_i and y_i from the H_1 hypothesis in (8.4). Using these training samples, we form estimates $\widehat{U}$, $\widehat{\Theta}$, $\widehat{\rho}$, $\widehat{R}_{xx}$, $\widehat{R}_{yy}$, and $\widehat{R}_{xy}$ as described in Section 8.2.2.

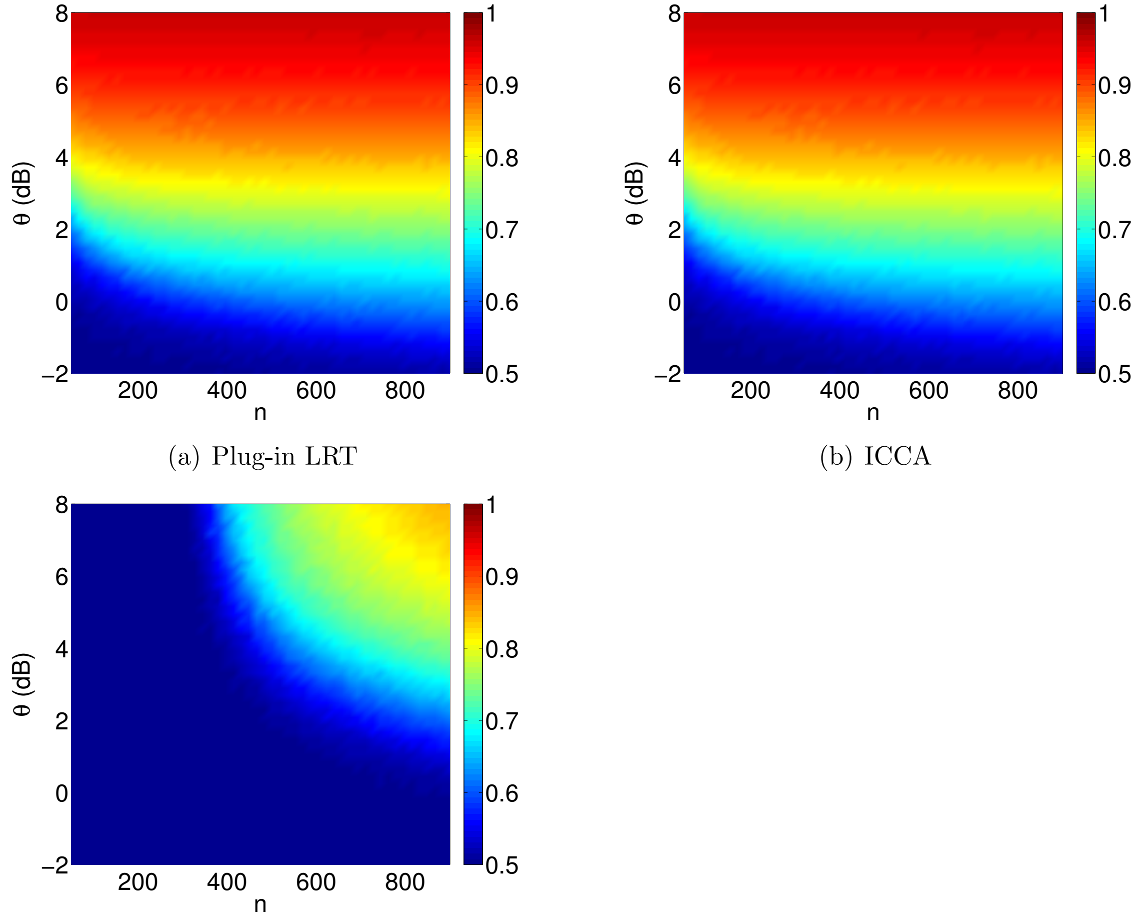
We then generate a desired number of test samples from each hypothesis using (8.4). For each test sample, we compute the test statistic for the plug-in LRT, empirical CCA, and ICCA detectors in (8.21), (8.23), and (8.24), respectively. Using Fawcett's [1] 'Algorithm 2', we compute an empirical ROC curve by first sorting the test statistics for a given detector. At each statistic, we log a (P_F, P_D) pair by counting the number of lower scores generated from each hypothesis. This is repeated for multiple realizations of u_x and u_y, generating multiple empirical ROC curves for each detector. We refer to a single empirical ROC curve corresponding to a realization of u_x and u_y as a trial. We then average the empirical ROC curves for a detector over multiple trials using Fawcett's [1] 'Algorithm 4'. This performs threshold averaging by first uniformly sampling the sorted list of all test scores of ROC curves and then computing (P_F, P_D) pairs in the same way as 'Algorithm 2'.

To compare the ROC curves of different detectors, we use the area under the ROC curve (AUC) statistic. The AUC statistic ranges between 0.5, which represents a random guessing detector, and 1.0, which represents a detector that can perfectly distinguish between the two hypotheses. We compute the ROC curves and their respective AUC for many values of the number of training samples, n, and SNR $\theta = \theta_1^{(x)} = \theta_1^{(y)}$. We present the AUC results in the form of a heatmap for two different values of ρ for each of the detectors. Figure 8.2 presents results for $\rho = 0.8$ and Figure 8.3 presents results for $\rho = 0.2$.

Evident in both Figures 8.2 and 8.3, the ICCA detector exhibits the same AUC performance as the plug-in LRT for both values of ρ. This confirms the derivation in the above section. In Figure 8.2, we observe that the CCA detector is extremely suboptimal in the sample and SNR regime presented. When $n < 350 = p + q$, the CCA detector degrades to random guessing, evident in an AUC of 0.5. The results presented in Chapter 4 show that in this sample poor regime, the correlation coefficient estimate returned by CCA is deterministically 1. It is of no surprise that the subsequent CCA detector is useless in this regime. Even when $n > p + q$, the CCA detector achieves a lower AUC than the ICCA detector. The ICCA detector can tolerate a much lower SNR to achieve the same AUC performance as the CCA detector.

When decreasing ρ in Figure 8.3, the CCA detector observes an even further performance loss. In the training sample and SNR parameter regime presented, the CCA detector achieves an AUC of 0.5, indicating it is useless in detection. We plot the difference between the ICCA AUC heatmaps for the two choices of ρ in Figure 8.4. For small values of θ, the larger value of ρ results in the better performance while for large values of θ, the smaller value of ρ results in better performance. Decreasing ρ makes the observations x and y more independent, thereby containing more information and increasing detection performance. Therefore, this observation is intuitive. When the SNR is large, we have more reliable information for larger ρ. When the SNR is small, the correlation between the datasets helps to better detect the signal. We can think of this as SNR boosting.

These results are particularly surprising because we began this chapter by deriving the fact that the LRT detector is equivalent to the CCA detector. However, when using parameter estimates, the empirical CCA detector no longer is equivalent to the plug-in detector. As many applications require estimating the covariance matrices used in CCA, this is an extremely undesirable property of CCA. However, using

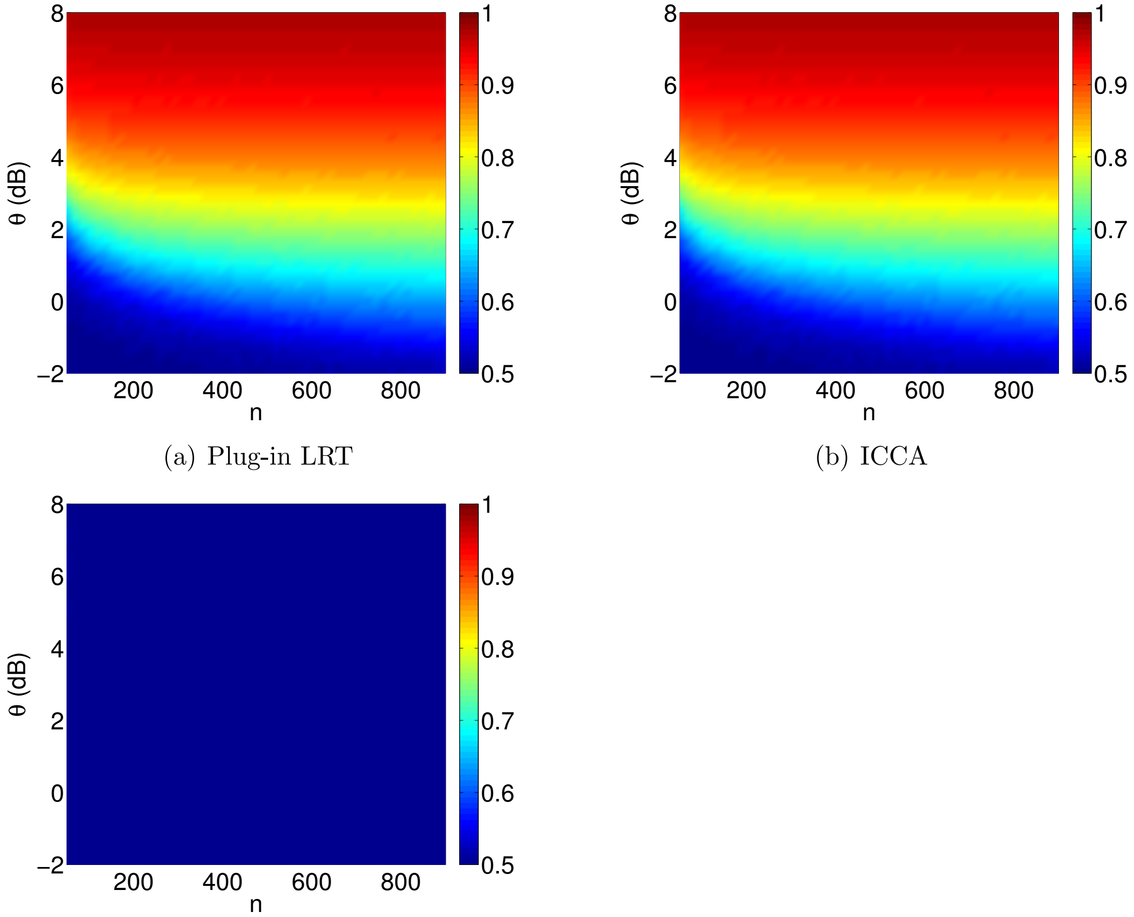


(a) Plug-in LRT

(b) ICCA

(c) Empirical CCA

Figure 8.2: AUC results for the plug-in LRT, empirical CCA, and ICCA detectors in (8.21), (8.23), and (8.24), respectively. Empirical ROC curves were simulated using 2000 test samples for each hypothesis and averaged over 50 trials using algorithms 2 and 4 of [1]. Simulations parameters were $p = 200$, $q = 150$, and $\rho = 0.8$. Each figure plots the AUC for the average ROC curve at a different values of SNR, $\theta = \theta_1^{(x)} = \theta_1^{(y)}$, and training samples, n.



(a) Plug-in LRT

(b) ICCA

(c) CCA

Figure 8.3: AUC results for the plug-in LRT, empirical CCA, and ICCA detectors in (8.21), (8.23), and (8.24), respectively. Empirical ROC curves were simulated using 2000 test samples for each hypothesis and averaged over 50 trials using algorithms 2 and 4 of [1]. Simulations parameters were $p = 200$, $q = 150$, and $\rho = 0.2$. Each figure plots the AUC for the average ROC curve at a different value of SNR, $\theta = \theta_1^{(x)} = \theta_1^{(y)}$, and training samples, n.

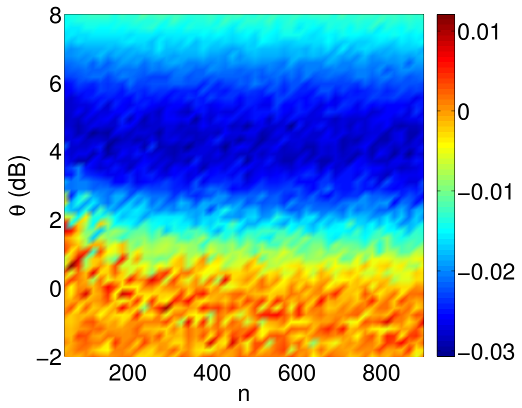


Figure 8.4: Difference between ICCA AUC heatmaps in Figures 8.3(c) and 8.2(c). Positive values indicate when the setting of $\rho = 0.8$ achieves a higher AUC. Negative values indicate when the setting of $\rho = 0.2$ achieves a higher AUC.

only the informative components from our training data, as ICCA does, results in equivalent performance as the plug-in LRT detector. This performance loss of the empirical CCA detector can be avoided by instead using ICCA.

CHAPTER IX

Content Based Image Retrieval and Automatic Image Annotation Using Correlation Methods

9.1 Introduction

In this chapter we evaluate the performance of linear correlation methods based on eigen-analysis when applied to the related problems of image retrieval and image annotation. In both problems, we have a corpus of images with corresponding text captions or text keywords. We may also have a text document or article associated with each image-caption pair. Image retrieval allows a user to enter a text query to retrieve relevant images from the corpus. Automatic image annotation allows a user to enter an image as a query to retrieve relevant keywords about that image. These interrelated machine learning problems require a way of transforming both words and images into objects that are understandable by machines [145]. For such problems, we naturally assume that the words in a caption are correlated with features of the image and also representative of the (possibly) associated text document. Therefore, correlation detection algorithms are natural candidates to help solve these problems. Using regression techniques relying on these correlated components, we can predict relevant image features given a set of words. Similarly, given a set of image features, we can predict relevant keywords.

Canonical Correlation Analysis (CCA) is a dimensionality reduction algorithm for two datasets. For each dataset, CCA learns a linear transformation such that the transformed datasets are maximally correlated. Representing data in a lower-dimensional, maximally correlated space allows learning algorithms to more easily and more efficiently exploit such naturally existing correlations. Image retrieval and image annotation are natural applications for CCA as there are exactly two datasets, images (or text documents) and text captions, that are assumed to have correlated features.

CCA and various modifications have been previously applied to information retrieval problems. In [35], the authors use CCA to cluster Wikipedia articles based on their text and incoming and outgoing links. In [28], the authors implemented a CCA based image retrieval system on a limited set of image types (sports, aviation, and paintball). The authors in [34] used a kernel version of CCA for automatic image annotation on a more varied set of images.

While these papers report moderate performance when using CCA for information retrieval tasks, CCA is fundamentally flawed. When the number of image-caption pairs is less than the combined dimension of the textual and image features, CCA returns random linear transformations and a perfect correlation between datasets [6]. For this reason, CCA is often overlooked when considering appropriate algorithms for machine learning problems. However, as was shown in Chapter 4 in this thesis, Informative CCA (ICCA) overcomes the massive performance loss of CCA in the low-sample regime. By applying insights from random matrix theory to eliminate noisy subspaces, ICCA can detect correlations in the low-sample low-SNR regime where CCA would otherwise fail to do so.

The purpose of this chapter is to showcase that intelligent correlation algorithms are indeed worthy of further investigation by the information retrieval community. In Section 9.2, we provide a brief overview of CCA and ICCA including their mathematical formulations and solutions. In Section 9.3, we outline how to use linear correlation algorithms for image retrieval and automatic image annotation. For captions and articles/documents, we create feature vectors using tf-idf weightings. For images, we construct visual word feature vectors based on SIFT features. We describe how to train a correlation model on a corpus and how to use the model to predict one modality from the other. In Section 9.4, we apply our image retrieval and annotation system on four datasets. First, we visually compare the performance of CCA and ICCA based image retrieval and image annotation on the Pascal dataset. Second, we specifically consider the image annotation problem on the University of Washington Ground Truth dataset, the Gold Standard Web dataset, and the BBC News dataset. These datasets allow us to compare the performance of eigen-based correlation algorithms to standard NLP and IR techniques on three datasets of varying difficulties. We provide a discussion of the benefits and limitations of eigen-based correlation algorithms in Section 9.5 and concluding remarks in Section 9.6.

9.2 Correlation Methods

Canonical correlation analysis (CCA) is a popular algorithm to identify features of maximal correlation between exactly two multi-modal datasets. CCA is a dimensionality reduction algorithm that finds a linear transformation for each dataset such that the correlation between the two transformed feature sets is maximized [4]. These linear transformations are easily found by solving a quadratic optimization problem; this solution is a closed form expression relying on the singular value decomposition (SVD) of a matrix product involving the covariance matrices of each dataset and the cross-covariance matrix between the two datasets. As these covariance matrices are rarely known *a priori*, practical uses of CCA rely on substituting sample covariance matrices formed from training data; we call this algorithm empirical CCA.

While we will apply CCA to image retrieval and annotation, CCA is commonly used in a variety of other disciplines. In [28], CCA is used to learn semantics of multimedia content by fusing image and text data. CCA is applied to to the common communications problem of blind equalization of single-input multiple-output (SIMO) channels in [51]. In the field of medical imaging, CCA is used to determine interactions, or connectivities, between brain areas in fMRI data [36] and used to fuse fMRI, sMRI, and EEG data [40]. CCA has also been applied to clustering speakers given an audio-video dataset [35]. In the more abstract problems of Gauss-Gauss detection and estimation, [52] shows that standard detectors and estimators can be written in terms of the solution to CCA.

9.2.1 Mathematical Formulation of CCA

Assume that observations $y_1 \in \mathbb{R}^{d_x \times 1}$, $y_2 \in \mathbb{R}^{d_y \times 1}$ are drawn from two distributions $y_1 \sim \mathcal{X}$, $y_2 \sim \mathcal{Y}$. Furthermore, assume that the distributions have zero mean, *i.e.* $\mathbb{E}\left[y_1\right] = \mathbb{E}\left[y_2\right] = 0$. We will use the following notation for the covariance matrices of the distributions: $\mathbb{E}\left[y_1 y_1^T\right] = R_{11}$, $\mathbb{E}\left[y_2 y_2^T\right] = R_{22}$, $\mathbb{E}\left[y_1 y_2^T\right] = R_{12}$.

The goal of CCA is to find a linear transformation for each dataset that maximizes the correlation between the datasets in the projected spaces. We represent the linear transformations with the canonical vectors $x_1 \in \mathbb{R}^{d_x \times 1}$ and $x_2 \in \mathbb{C}^{d_y \times 1}$ and the projection with the canonical variates $w_1 = x_1^T y_1$ and $w_2 = x_2^T y_2$. The objective is to find the canonical vectors x_1 and x_2 that maximize the correlation between the canonical variates w_1 and w_2. Formally, the optimization problem is

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = \mathbb{E}\left[w_1 w_2\right] \\ \text{subject to} \quad & \mathbb{E}\left[w_1^2\right] = 1, \mathbb{E}\left[w_2^2\right] = 1. \end{aligned} \tag{9.1}$$

Substituting the expressions for the canonical variates and the correlation matrices, this optimization problem may be written as

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = x_1^T R_{12} x_2 \\ \text{subject to} \quad & x_1^T R_{11} x_1 = 1, x_2^T R_{22} x_2 = 1. \end{aligned} \tag{9.2}$$

Standard Lagrange multiplier techniques can be used to solve (9.2). The proof is omitted here but please reference [55, 8, 14, 64, 65] if interested. The solution is the following eigenvalue system

$$R_{11}^{-1} R_{12} R_{22}^{-1} R_{12}^T x_1 = \rho^2 x_1 \tag{9.3}$$

with the relationship

$$x_2 = \frac{1}{\rho} R_{22}^{-1} R_{12}^T x_1. \tag{9.4}$$

Solving (9.3) for the eigenvector corresponding to the largest eigenvalue solves (9.2). Substituting this eigenvalue/eigenvector pair in (9.4) gives the complete solution (x_1, x_2, ρ) for the transformations and maximum correlation between the datasets. Multiple canonical basis vectors may be found by recursively finding the next largest eigenvalue and corresponding eigenvector in (9.3). In many learning applications, it is common to project onto multiple canonical basis vectors.

Using a similarity transform, we can frame the eigen-system in (9.3) as an SVD problem. Define $f = R_{11}^{1/2} x_1$ and $g = R_{22}^{1/2} x_2$. Then (9.3) may be rewritten as

$$R_{11}^{-1/2} R_{12} R_{22}^{-1} R_{12}^T R_{11}^{-T/2} f = \rho^2 f. \tag{9.5}$$

Defining $C = R_{11}^{-1/2} R_{12} R_{22}^{-T/2}$, (9.5) can be rewritten as

$$CC^T f = \rho^2 f. \tag{9.6}$$

Clearly, from (9.6), we may obtain a closed form solution for f, g, and ρ through the SVD of C. Let FKG^T be the SVD of C where $F = [f_1, \ldots, f_{d_x}]$, $K \in \mathbb{C}^{d_x \times d_y} =$

$\mathbf{diag}(k_1, \ldots, k_{\min(d_x, d_y)})$, and $G = [g_1, \ldots, g_{d_y}]$. Then the solution for the canonical vector pair corresponding to the largest canonical correlation is

$$\begin{aligned} \rho &= k_1 \\ x_1 &= R_{11}^{-1/2} f_1 \\ x_2 &= R_{22}^{-1/2} g_1 \end{aligned} \tag{9.7}$$

with successive pairs of canonical vectors obtained from successive singular vectors and singular value pairs.

9.2.2 Empirical CCA

The above derivation assumes that the covariance matrices R_{11}, R_{22}, and R_{12} are all known. However, in most applications these covariance matrices are unknown and must be estimated from data. In such an empirical setting, we assume that we are given n observations, or samples, from each dataset $y_1^{(i)}$ and $y_2^{(i)}$ for $i = 1, \ldots, n$ such that $y_1^{(i)}$ and $y_2^{(i)}$ each represent the same object. In the application for this chapter, $y_1^{(i)}$ will be a text caption and $y_2^{(i)}$ will be the corresponding image or article features. We may stack these observations in training data matrices

$$X = \left[y_1^{(1)}, \ldots, y_1^{(n)} \right], \text{ and } Y = \left[y_2^{(1)}, \ldots, y_2^{(n)} \right]$$

and use these training data matrices to estimate the unknown covariance matrices via

$$\begin{aligned} \widehat{R}_{11} &= \frac{1}{n} X X^T \\ \widehat{R}_{22} &= \frac{1}{n} Y Y^T \\ \widehat{R}_{12} &= \frac{1}{n} X Y^T. \end{aligned} \tag{9.8}$$

We may then substitute these covariance matrix estimates in the expression for C, resulting in the estimator

$$\widehat{C} = \widehat{R}_{11}^{-1/2} \widehat{R}_{12} \widehat{R}_{22}^{-1/2}. \tag{9.9}$$

Defining $\widehat{C} = \widehat{F} \widehat{K} \widehat{G}^T$ as the SVD of $\widehat{C}$, the solution to empirical CCA is

$$\begin{aligned} \widehat{\rho} &= \widehat{k}_1 \\ \widehat{x}_1 &= \widehat{R}_{11}^{-1/2} \widehat{f}_1 \\ \widehat{x}_2 &= \widehat{R}_{22}^{-1/2} \widehat{g}_1. \end{aligned} \tag{9.10}$$

9.2.3 Informative CCA

A main drawback to CCA is that when then number of samples is less than the combined dimension of the two datasets ($n < d_x + d_y$), CCA always reports a perfect correlation, no matter how correlated the datasets actually are [6].

In earlier chapters, we demonstrated that informative CCA (ICCA) avoids this drastic performance loss associated with CCA. Assuming a linear signal-plus-noise model, ICCA uses random matrix theory insights to trim data SVDs to only include informative subspaces. Formally, let $U_x \Sigma_x V_x^T$ be the SVD of X and $U_y \Sigma_y V_y^T$ be the SVD of Y. Define

$$\begin{aligned} \widetilde{U}_x &= U_x(:, 1:k_x) \quad \widetilde{V}_x = V_x(:, 1:k_x) \\ \widetilde{U}_y &= U_y(:, 1:k_y) \quad \widetilde{V}_y = V_y(:, 1:k_y) \end{aligned}$$

where k_x and k_y are the number of informative components in the first and second datasets, respectively. These may be estimated using standard random matrix theory principles, as we proposed in earlier chapters. Using these trimmed data matrices, we form the matrix used for ICCA,

$$\widetilde{C} = \widetilde{U}_x \widetilde{V}_x^T \widetilde{V}_y \widetilde{U}_y^T. \tag{9.11}$$

Let $\widetilde{C} = \widetilde{F} \widetilde{K} \widetilde{G}^H$ be the SVD of this matrix. ICCA returns the following informative correlation estimate and canonical vectors

$$\begin{aligned} \widetilde{\rho} &= \widetilde{k}_1 \\ \widetilde{x}_1 &= \widehat{R}_{11}^{-1/2} \widetilde{f}_1 \\ \widetilde{x}_2 &= \widehat{R}_{22}^{-1/2} \widetilde{g}_1. \end{aligned} \tag{9.12}$$

As we theoretically showed in earlier chapters, ICCA is able to detect correlations in the low-sample and low-SNR regimes where CCA would not. In these regimes, the linear transformations that CCA returns are random and contain no information, while the linear transformations returned by ICCA contain information. With these observations, ICCA may return meaningful results in image retrieval and annotation where CCA was previously ignored due to random performance.

9.3 System Implementation

In this section we describe how to use CCA and ICCA for content based image retrieval and automatic image annotation. A common training stage learns the correlation model necessary to perform both tasks. Figure 9.1 outlines the training phase when the two datasets are captions and associated images. Figure 9.2 outlines the training phase when the two datasets are captions and associated text documents. Each block is a sub-task that will be described in more detail below. Depending on our dataset, one pipeline will be more appropriate. For example, the University of Washington Ground Truth dataset only has images and captions so we will use the pipeline in Figure 9.1. However, in the Gold Standard Web dataset, we have access to captions, images, and associated documents. For this dataset, we only have a few samples of a large variety of images and so we will used the pipeline in Figure 9.2. The output of both pipelines are canonical bases W_x and W_y and the corresponding canonical correlations, P. After training, we may use the correlation model to retrieve images, which is outlined in Figure 9.4, or annotate images, which is outlined in Figure 9.5. These sub-tasks will also be described below.

9.3.1 Text Processing

Both training pipelines form feature vectors from text (either captions or documents). To transform text into a machine understandable object, we create a feature vector whose length is the size of the vocabulary of the training data and whose entries are tf-idf weights. Once these vectors are created for each caption, we have a caption dataset, $X \in \mathbb{R}^{d_x \times n}$ where d_x is the size of the vocabulary and n is the number of training captions associated with images. Each vector is then normalized to have an ℓ_2 norm of 1, so as not to penalize shorter documents. With this type of processing, feature vector entries represent the importance that the corresponding word carries in the document. When processing a text query for image retrieval, the text query is transformed into a $d_x \times 1$ vector using the same tf-idf weighted scheme used to generate the training dataset. When generating the vocabulary, we used stopword removal and Porter stemming [1].

9.3.2 Image Processing

The training system in Figure 9.1 takes an image database as its second input. To transform an image into a machine understandable object, we propose to use visual words, which is an extension of the vector space model for text documents to images. Here, a feature vector for an image has entries corresponding to the total occurrences of a “visual word” in that image. Each visual word is a cluster of image features

[1] http://tartarus.org/ martin/PorterStemmer/python.txt

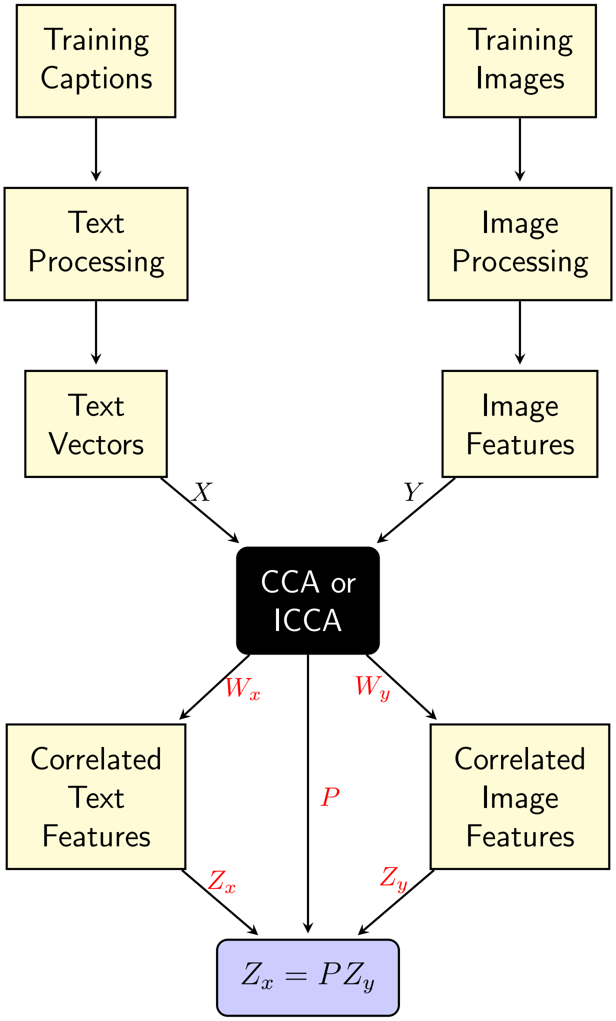


Figure 9.1: Shared training pipeline for image retrieval and annotation when using raw images as the second dataset. The system takes training images and captions as inputs and returns the canonical bases W_x and W_y and the correlation coefficients P.

extracted across all training data [146]. For our implementation, we use SIFT image features vectors as the feature vectors to cluster.

The Scale Invariant Feature Transform (SIFT) was first introduced in [147]. This algorithm transforms an image into a collection of local feature vectors such that each feature vector is invariant to translation, scaling, and rotation. The algorithm may be broken down into two parts. First, keypoints (pixels) are identified using a difference-of-Gaussian function. See Figure 9.3 for an example of SIFT keypoint generation. Second, a descriptor (feature vector) is generated for each keypoint using

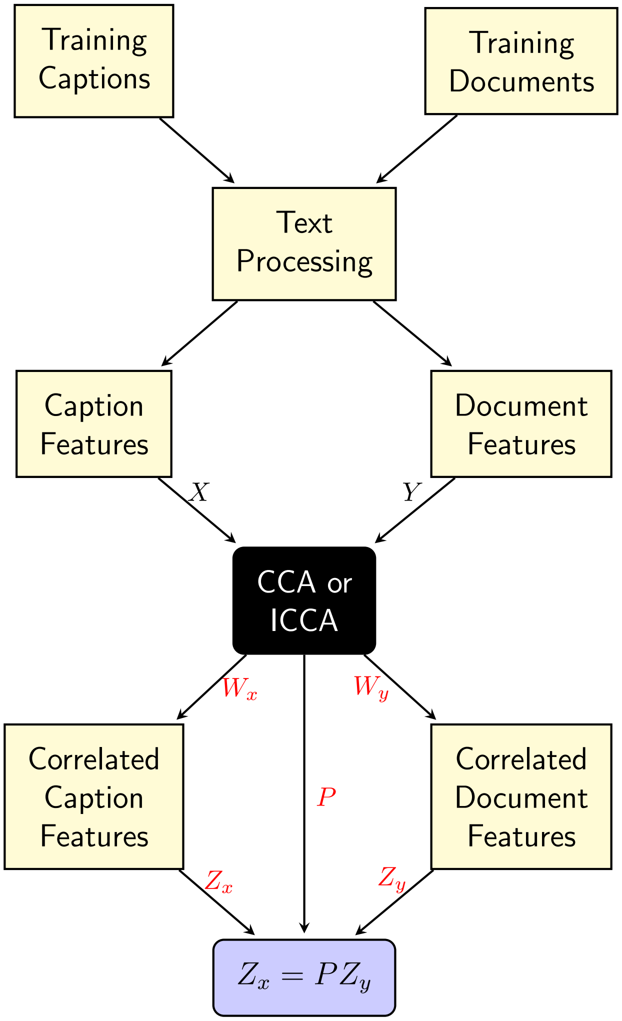
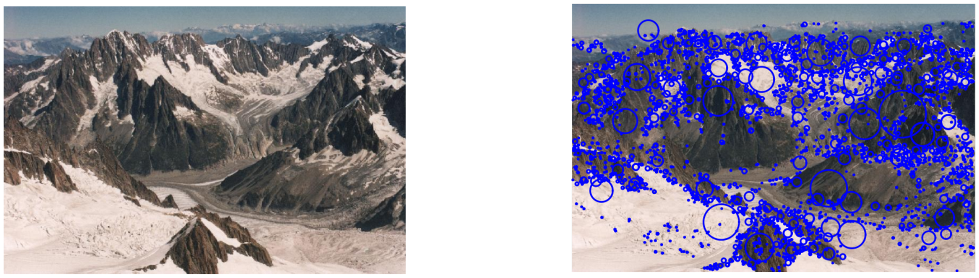


Figure 9.2: Shared training pipeline for image retrieval and annotation when the second dataset is associated text documents. The system takes an training captions and associated documents as inputs and returns the canonical bases W_x and W_y and the correlation coefficients P.

weighted magnitude and orientation histograms in pixel neighborhoods in a region around each keypoint. The visual word training process can be broken down into the following steps:

1. Create SIFT features for all training images
2. Use k-means to cluster all SIFT features into 1000 “visual words”
3. Assign each SIFT keypoint in every image to the closest (ℓ_2 distance) visual



(a) Original Image

(b) SIFT Keypoints

Figure 9.3: (a) Original image. (b) Original image with SIFT keypoint identification.

word

4. Count the number of occurrences of each visual word in each image

Given the training images, the visual word image processing returns a $d_y \times n$ matrix Y, where d_y is the number of visual words and n is the number of images. Each vector in Y is then normalized to have an ℓ_2 norm of 1. When processing a test image for automatic image annotation, the image's feature vector is created using the same method as the training data.

When creating a feature vector for a query image, we generate the SIFT features for that image and then assign each feature to the closest visual word in the training set. Since we create 1000 clusters, the dimension of our image feature vectors for visual words is $d_y = 1000$. We note that the visual words feature vector is highly dependent on the training data. Each image's feature vector is dependent on the clusters found in the training images. Therefore, we cannot pre-compute each image's feature vector without knowing all training images.

We follow the implementation of visual words provided in [148] using some of the code. The main changes we made were applying tf-idf weight to the visual words and changing how we represent the vocabulary of the visual words. We also make each visual word feature vector unit norm. For the SIFT implementation, we use the publicly available C code at http://www.vlfeat.org/install-shell.html. We made some minor changes to how the visual word implementation interfaced with the SIFT feature creation.

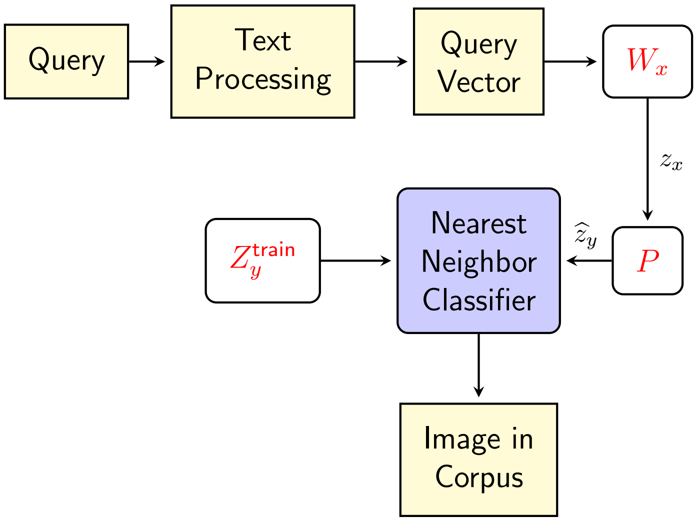


Figure 9.4: Image retrieval pipeline. This system takes a text query as input and the correlation model from the training pipeline and will return relevant images.

9.3.3 Correlation Algorithm

After the datasets have been processed into a caption data matrix X and an image or document data matrix Y, we train the CCA or ICCA as described in Section 9.2. Regardless of the correlation algorithm, the output at this stage in the training pipeline are two linear transformations $W_x \in \mathbb{R}^{d_x \times k_x}$ and $W_y \in \mathbb{R}^{d_y \times k_y}$ and a diagonal matrix of canonical correlations, $P \in \mathbb{R}^{k_x \times k_y}$. The parameters k_x and k_y are the number of canonical vectors to use for each dataset, respectively. As any uncorrelated canonical vectors carry no prediction power, we simply set the number of parameters to $k = k_x = k_y$ and so P is a square matrix.

Once we learn the canonical basis vectors W_x and W_y, we then form the dimensionality reduced datasets of canonical variates. This is accomplished with the simple linear transformations

$$Z_x = W_x^T X \qquad Z_y = W_y^T Y$$

where $Z_x \in \mathbb{R}^{k_x \times n}$ and $Z_y \in \mathbb{R}^{k_y \times n}$. These are dimensionality reduced datasets that are maximally correlated. The beauty of these correlation algorithms is that by solving for W_x, W_y and P, we automatically solve a regression problem in the domain of the canonical variates. This relationship is given by

$$\mathbb{E}\left[w_x \,|\, w_y\right] = P w_y \;\text{ and }\; \mathbb{E}\left[w_y \,|\, w_x\right] = P w_x.$$

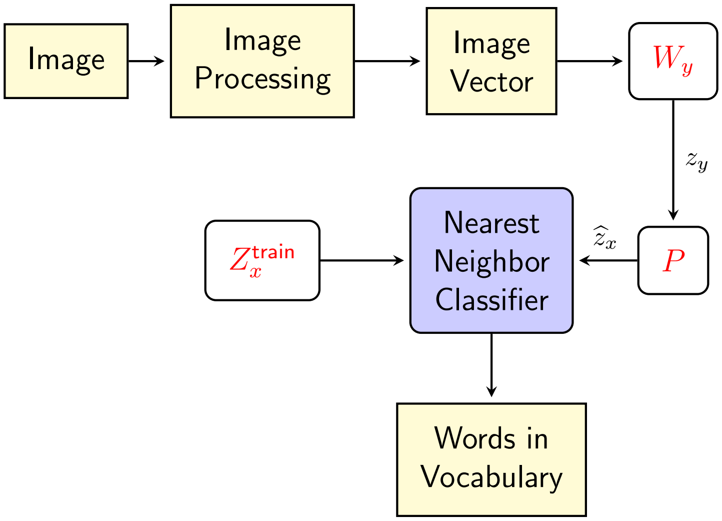


Figure 9.5: Image annotation pipeline when using the raw image as input. This system takes an image and correlation model from the training system as inputs and will return a list of words that are most relevant for the image.

Notice that there is no linear offset needed as the datasets are zero mean. We also note that this above equation is correct and that in both predictions we scale down the known canonical variate. This makes sense by considering the following example. If $\rho_i = 1$, then the variates w_x^i and w_y^i are perfectly correlated and predict each other: $w_x = w_y$. However if $\rho_i = 0$, then the variates contain no information and the best guess that we have is the mean, which is zero. For any $0 < \rho_i < 1$, we scale the prediction toward zero depending on the strength of the correlation.

9.3.4 Image Retrieval

After training the system on a corpus, a user may perform image retrieval. Given a text query, we first process it using the same tf-idf weighting scheme used in the training model, resulting in the vector $q \in \mathbb{R}^{d_x \times 1}$. To obtain an estimate of our image feature, we perform the following sequence of linear transformations, learned by one of the correlation algorithms,

$$\widehat{z}_y = P W_x^T q$$

To return relevant images, we use a nearest neighbor classifier in the canonical variate domain. We can pre-compute all possible images to return via $Z_y^{\text{train}} = W_y^T Y$. The output of the search is

$$y_{\text{guess}} = \underset{z_y \in Z_y^{\text{train}}}{\operatorname{argmin}} \|z_y - \widehat{z}_y\|_2^2, \tag{9.13}$$

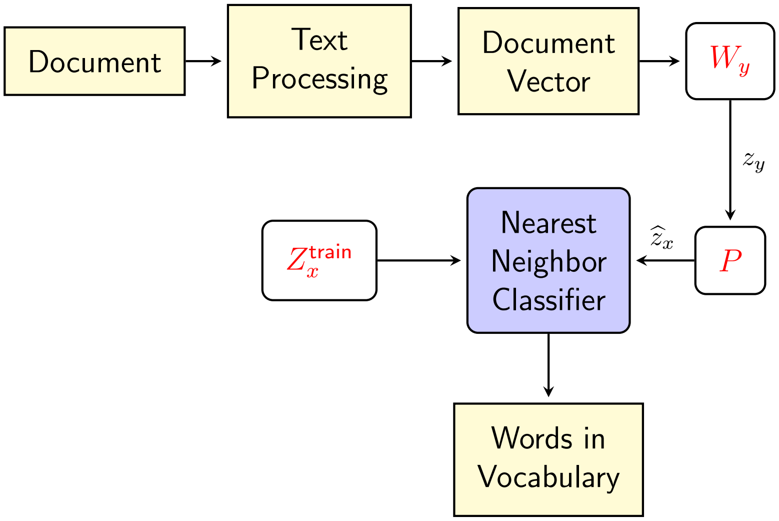


Figure 9.6: Image annotation pipeline when using associated text documents. This system takes a text document and correlation model from the training system as inputs and will return a list of words that are most relevant for the associated image.

which is repeated for as many results the user desires, excluding previously returned results from the search set each time. A nice benefit of using correlation methods is that additional images may be added to the set of returnable images, even if they do not have a caption associated with them. All we need is the low dimensional representation of these additional images using the transformation learned from the training set, whether that be image features or document vectors.

9.3.5 Image Annotation

The trained system can also handle automatic image annotation. Given a query image or associated document, we first process it using the same image or text processing that was used in training, resulting in a query vector $q \in \mathbb{R}^{d_y \times 1}$. To obtain an estimate of our text features, we perform the following linear transformation

$$\widehat{z}_x = P W_y^T q$$

using the correlation model learned in the training phase. To return relevant words, we use a nearest neighbor classifier in the canonical variate domain. However, instead of using the captions as the vectors to compare against, we consider d_x documents, each of which contains exactly one of the words in the vocabulary. Let $D \in \mathbb{R}^{d_x \times d_x}$ be a diagonal matrix with entries equal to the tf-idf score of that word. Define

$Z_x^{\text{train}} = W_x^T D$ to be the canonical variate vectors of each word. Then the output of the nearest neighbor classifier is

$$x_{\text{guess}} = \underset{z_x \in Z_x^{\text{train}}}{\operatorname{argmin}} \|z_D - \widehat{z}_x\|_2^2. \tag{9.14}$$

We then return the words corresponding to the vectors returned by the nearest neighbor classifier.

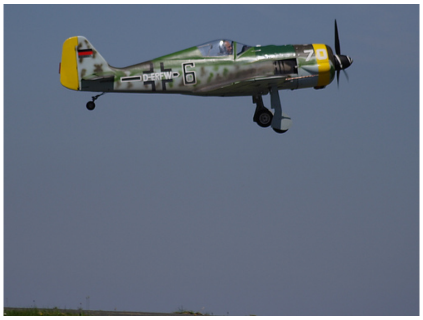
9.4 Experiments

To compare the performance of CCA and ICCA in image retrieval and annotation, we test our system on four different datasets. For the Pascal Image Dataset, we wrote a command line interface to perform both image retrieval and image annotation. This allows for us to qualitatively determine how each method works. For the University of Washington Ground Truth dataset, we run the training and testing pipelines in Figures 9.1 and 9.5 and compare the R-precision for the image annotation task. For the Gold Standard Web dataset, we only have a few image-caption pairs but also have associated text documents and so we use the training pipelines in Figures 9.2 and 9.6. We compare our eigen-based annotation methods to previous NLP methods using the evaluation framework in [149]. Finally, we use the training pipelines in Figures 9.2 and 9.6 to test the image annotation performance of CCA and ICCA on the BBC News dataset. This dataset is particularly challenging first because the images are extremely varied and may only be tangentially related to the associated document, and second because the accompanying caption is often extremely nuanced, containing few keywords. These challenges present many latent variables, which CCA and ICCA will not handle very well.

9.4.1 Pascal Image Dataset

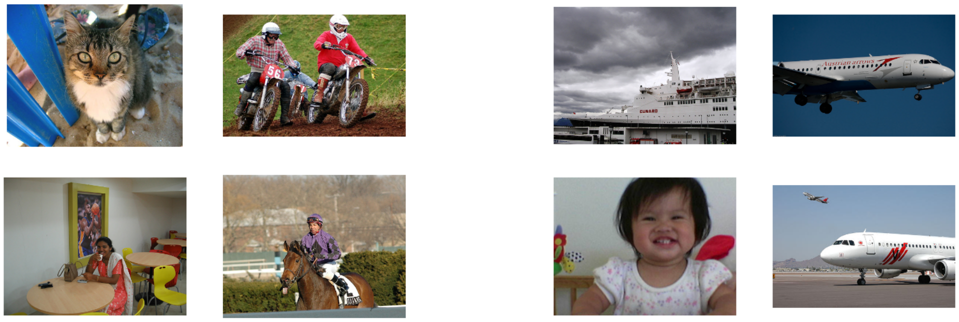
The Pascal Image dataset [2] was created using Amazon's Mechanical Turk [150]. The dataset consists of 1000 images, each with 5 captions. The average image has 26.67 caption words and the total vocabulary size of the corpus is 2393. The 5 captions for each image are unique, but they may repeat keywords or use synonyms. For example, airplanes in the dataset can be described as airplanes, planes, fighter planes, jets, and even their model. See Figure 9.7 for an example image-caption pair.

[2] http://nlp.cs.illinois.edu/HockenmaierGroup/pascal-sentences/



- A D-ERFW-6 in flight.
- An army green plane flying in the sky.
- An old fighter plane flying with German military markings.
- A small green and yellow plane in the sky.
- A WWII fighter plane with its landing gear down.

Figure 9.7: Example of an image and its captions in the Pascal dataset



(a) CCA Results

(b) ICCA Results

Figure 9.8: (a) CCA results for query ”airplane”. (b) ICCA results for query ”airplane”.

To qualitatively evaluate the performance of CCA and ICCA using visual words, we implemented the image retrieval and image annotation systems described in Section 9.3. Figure 9.8 shows the first four images retrieved for the search query “airplane” using both CCA and ICCA correlation models. We plot the scores used to return these images in Figure 9.9. The largest four scores correspond to the images in Figure 9.8 as computed via (9.13). Figure 9.10 shows the first ten words returned for the image in Figure 9.10(a), which was taken from the Pascal dataset. Figure 9.11 plots the corresponding scores for all words in the database as computed via (9.14). The top scores correspond to the words returned in Figure 9.10. For both tasks, we used all 1000 image-caption pairs in the Pascal dataset for training. Thus, any image in the dataset may be returned in the image retrieval task. Similarly, any Porter-stemmed vocabulary word in the entire caption dataset may be returned in the image annotation task.

	Mean R-precision	Mean k
CCA	0.024	63
ICCA	0.232	20

Table 9.1: Average R-precision values and correlation basis dimension, k, for image annotation of the University of Washington Ground Truth Dataset

9.4.2 Ground Truth Image Dataset

To quantitatively compare the performance of CCA and ICCA, we used the University of Washington Ground Truth Image Dataset [3]. This dataset contains 1109 images with an average of 5.57 keywords per image and a total of 346 unique words. We randomly split the dataset into 3 sets: a training set of 550 images, a validation set of 250 images and a testing set of 309 images. This was repeated to obtain 10 such partitions.

For each partitioning, the training dataset was used to learn the correlation model for CCA, and ICCA. This model was learned for values of k =10,25,50,75,100. The validation dataset was then used to determine the best value of k for each algorithm, using R-precision as our evaluation metric. In this setting, R-precision is equal to the percentage of correctly predicted keywords for an image. Once the best k was determined, that partition's R-precision value for each algorithm was determined on the testing datasets. The R-precision values were then averaged across partitions. Figure 9.12 plots the probability density function of R-precision for CCA and ICCA and Table 9.1 shows the mean R-precision and average k values.

9.4.3 Gold Standard Web Dataset

Next, we evaluate our eigen-based image annotation methods on the Gold Standard Web Dataset [149]. This dataset contains 300 image-text pairs that was collected from the web. The average text document length is 278 tokens and the vocabulary size is 8,409 words. Each image also has a gold standard of manually assigned tags labeled by five human annotators. We consider these manually assigned tags as the caption. However, as we only have 300 images and these images contain many different objects, we use the associated text document as the second modality. Hence, our goal is to predict the captions given the text document, which is essentially keyword identification.

We use the same four evaluation metrics as [149] to be able to make direct comparisons with previous methods. As CCA and ICCA can return an arbitrary number

[3] http://www.cs.washington.edu/research/imagedatabase/groundtruth/

Metric	Expression
Best Normal	$\frac{1}{\lvert\mathcal{I}\rvert}\sum_{i\in\mathcal{I}}\frac{1}{r_iR_i}\sum_{j=1}^{r_i} f_i^j$
Best Mode	$\frac{1}{\lvert\mathcal{IM}\rvert}\sum_{i\in\mathcal{IM}}\mathbb{1}_{\{\text{top tag = mode}\}}$
oot Normal	$\frac{1}{\lvert\mathcal{I}\rvert}\sum_{i\in\mathcal{I}}\frac{1}{R_i}\sum_{j=1}^{r_i} f_i^j$
oot Mode	$\frac{1}{\lvert\mathcal{IM}\rvert}\sum_{i\in\mathcal{IM}}\mathbb{1}_{\{\text{any tag = mode}\}}$

Table 9.2: Image annotation for the Web Image-Article dataset.

Models	Best Normal	Best Mode	oot Normal	oot Mode
CCA	0.01	0.00	1.49	35.23
ICCA	0.19	9.09	16.11	76.7
Flickr picturability	6.32	78.57	35.61	92.86
Wikipedia Salience	6.40	7.14	35.19	92.86
Topic modeling	5.99	42.86	37.13	85.71
Doc Title	6.40	75.00	18.97	82.14
tf*idf	5.94	14.29	38.40	78.57

Table 9.3: Image annotation for the Web Image-Article dataset.

of keywords, we choose to always predict keywords equal to the number of true keywords. Therefore, for our methods, these metrics are variations on R-precision. Some images are considered Mode images when there is a unique keyword selected by more annotators than any other keyword. The metrics are given in Table 9.2. Let $\mathcal{I}$ be the set of all images, $\mathcal{IM}$ be the set of mode images, f_i^j be the number annotators who labeled image i with keyword j, r_i be the number of keywords selected for image i, and $R_i = \sum_{j=1}^{r_i} f_i^j$. Table 9.3 reports the performance of CCA and ICCA in generating keywords given these 4 metrics using leave-one-out testing. We also provide the performance of models used in [149] for comparison.

9.4.4 BBC News Dataset

Finally, we evaluate CCA and ICCA based image annotation on the BBC News dataset [151]. Similar to the Gold Standard Web dataset, this dataset contains image-caption-document tuples that are separated into 3121 training examples and 240 testing examples. The images in this dataset are again very varied and the captions are sometimes nuanced and very specific to the image and not very related to the accompanying document. For the CCA and ICCA annotations, we again will use the accompanying documents to predict keywords in the caption. We report the precision and recall when returning the top 10, 15 and 20 predicted keywords in Table 9.4. In

	Top 10		Top 15		Top 20	
Models	P	R	P	R	P	R
CCA	0.08	0.11	0.08	0.22	0.08	0.28
ICCA	0.79	1.48	0.72	1.95	0.73	2.63
tf*idf	4.37	7.09	3.57	8.12	2.65	8.89
DocTitle	9.22	7.03	9.22	7.03	9.22	7.03
Lavrenko03	9.05	16.01	7.73	17.87	6.55	19.38
ExtModel	14.72	27.95	11.62	32.99	9.72	36.77
Flickr picturability	12.13	22.82	9.52	26.82	8.23	29.80
Wikipedia Salience	11.63	21.89	9.28	26.20	7.81	29.41
Topic Modeling	11.42	21.49	9.28	26.20	7.86	29.57

Table 9.4: Image annotation for the Web Image-Article dataset.

the table we also report methods from [151] and [149] for comparison. We follow implementations reported in [151] and [149] to not Porter stem the words but instead use Tree Tagger[152] to include only nouns, verbs, and adjectives.

9.5 Discussion

9.5.1 Pascal Results

For the Pascal dataset, we can qualitatively see the performance increase that ICCA gives. Examining Figure 9.8(a), we see that CCA returns random images associated with the query "airplane". This is the case with any other query entered. However this is expected with CCA as it returns a correlation of 1 between all images and tokens as we are operating in the sample deficient regime. Any positive results returned by CCA can be attributed to random luck. On the other hand, using ICCA to train a retrieval system results in better overall performance than CCA. This can be seen specifically in Figure 9.8(b), which shows images for the same query of "airplane". Using ICCA, two of the first four results are planes. By only using *informative* singular vectors, ICCA is able to return more relevant images.

Specifically, if we examine Figure 9.9, we can compare the scores returned for each image for CCA and ICCA. The top scores for ICCA seem to separate from the bulk of the others, while for CCA, the top scores seem to be part of the bulk distribution. This gives support to the notion that ICCA is able to identify images that are relevant to the desired keyword.

Image annotation using CCA also returns random keywords for any image query.

An example of this is shown in Figure 9.10. The query image is an airplane but the top 10 words returned by CCA are all irrelevant. This once again reinforces the idea that CCA only returns random results in the sample deficient regime. However, examining the top 10 words returned by ICCA for the same image, we observe meaningful annotations such as "plane", "blue", "fly". Examining Figure 9.11, we see the corresponding scores to the words returned in Figure 9.10. Again we notice that a majority of the words fall into the bulk distribution for CCA and ICCA. The words in the bulk part of the distribution are not informative. However, in ICCA, there are a number of words that separate from the bulk distribution, which we show in Figure 9.10. However, the scores for CCA do not separate as nicely from the distribution. This further gives support that CCA randomly returned words/images in the sample deficient regime.

A naïve image retrieval system may perform a search using only the captions and then return the image associated with the most notable caption. This produces very good results for the Pascal dataset because the captions are very clean and noise-free. Every caption with the word sheep will have a sheep in the image. However, correlation based approaches have a few main advantages over such a naïve method. First, correlation methods solve both the image retrieval and image annotation problem simultaneously. The naïve image retrieval method cannot solve the image annotation problem. Second, correlation methods can handle adding images to the corpus post training, even if it does not have an associated caption. For the image retrieval problem, the correlation methods will return images with low-dimensional representation that are close to the predicted vector. Adding additional images (even without captions) requires transforming the images into the low-dimensional representation using the trained transformation and then adding them to the set of possible images to be returned. The naïve method needs a caption for every image and if a new image-caption pair was added, the entire inverted index and vocabulary would need to be recomputed.

9.5.2 Image Annotation of Ground Truth Dataset

We use the Ground Truth Dataset to provide a more quantitative comparison of CCA and ICCA based image annotation. The captions for this dataset only consist of keywords and is thus easier to assess the quality of the image annotations. We did not clean up the dataset by removing misspelled words or words appearing only once. Therefore, the reported results are a nice lower-bound that one could expect by doing such clever pre-processing steps. As evident in Table 9.1, CCA performs very poorly on the image annotation task. This R-precision corresponds to random guessing, which matches the results of [6] stating that in the sample starved regime, the CCA bases are random projections. However, using ICCA to informatively trim data matrices results in improved annotations. Examining the figures in Figure 9.12, we see that when using CCA, approximately 85% of the images result is an R-precision of 0. However, ICCA returns zero relevant images only 35% of the time. Clearly, ICCA is able to uncover true correlations to return relevant annotations. This gives credence to using correlation methods for image annotation tasks.

9.5.3 Annotation of Gold Standard Web Dataset

The Gold Standard Web dataset is a more difficult dataset than the Ground Truth dataset. First, there are only 300 examples in the dataset. Using leave-one-out testing gives a training dataset of only 299 examples. Second, the dimensions of our dataset increases because we use tf-idf weights from the associated documents instead of visual words from the image. Therefore, we are in a very sample deficient regime where our dimension of each dataset is on the order of 8000 and we only have 299 samples. Compounding issues, these vectors are very sparse as documents only have a subset of the total words in the vocabulary. However, ICCA is indeed able to recover meaningful annotations even in this regime.

Examining the difference between the Best Mode and oot Mode performance metrics, we see that ICCA has a very large gap. This indicates that the ICCA retrieval method is very often able to retrieve the mode annotation, just not label it as the best annotation.

9.5.4 Annotation of BBC News Dataset

The BBC News dataset is the most difficult dataset we consider in this chapter. Here, we have a large number of training data, however, our captions are very "noisy". Unlike the Gold Standard Web dataset, the BBC captions may contain words that do not appear in the accompanying document. These captions are also very nuanced and may describe an image that is only tangentially related to the main article. For example, consider Figure 9.13. The image is of two men shaking hands, however, the caption describes this very abstractly. In addition, the caption highlights a very subtle point of the main article, as we can see by the title. Similar to the Gold Standard Web dataset, our feature vectors are both very high dimensional and sparse. We see from the results in Table 9.4 that both CCA and ICCA do a poor job at retrieving relevant annotations given the BBC article. However, we do observe the behavior that CCA returns completely random results and ICCA is able to perform slightly better (non-randomly).

The BBC News dataset breaks many of the assumptions of ICCA and therefore causes its performance to decrease. First, the dataset breaks the linear correlation assumption of ICCA. As we saw in Figure 9.13, captions are can be very nuanced, indicating many latent variables interacting in most likely nonlinear ways. Due to the size of the vocabulary of the BBC dataset, our text feature vectors are incredibly sparse, which as we saw in previous chapters, requires a larger SNR to detect correlations. The nonlinear correlations most certainly decrease the SNR in our dataset, which, coupled with the sparse data, stretches ICCA based image annotation to the limit of decent operation.

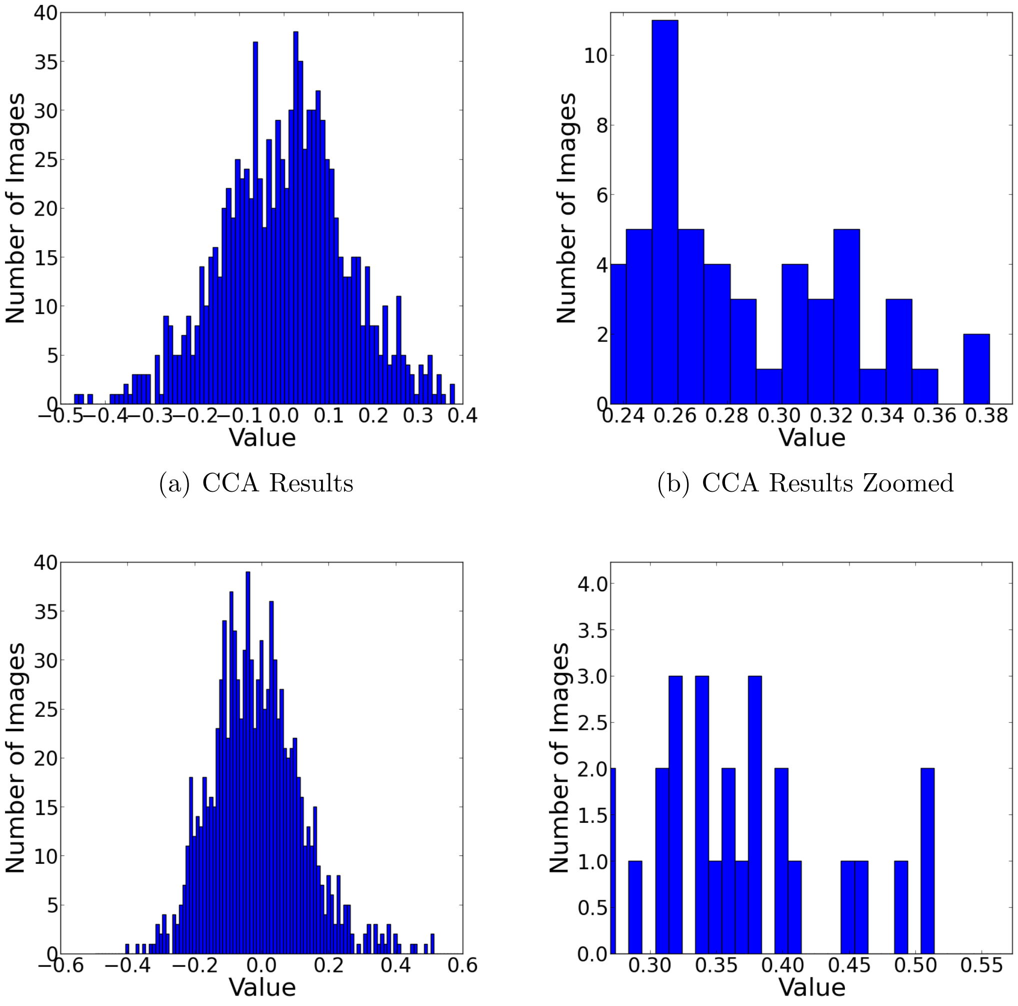
Clever feature engineering and alterations of ICCA could yield potential new avenues to improve the performance on difficult datasets such as the BBC News dataset. One possible extension is to use a kernel version of ICCA to account for nonlinear correlations. Similarly, extending ICCA to better handle sparse vectors could improve performance. Finally, using more intelligent IR and NLP techniques to create more informative feature vectors than tf*idf weights could increase the relative SNR high enough to allow for more reliable correlation detection.

9.6 Conclusion

In this chapter, we applied CCA and ICCA based correlation detection methods to image retrieval and annotation. By trimming data matrices to only include informative subspace components, ICCA is able to avoid the performance loss of CCA in the sample deficient regime. We demonstrated through multiple datasets that ICCA is able to outperform CCA on both image retrieval and image annotation tasks, both qualitatively and quantitatively.

For all datasets, CCA failed completely while ICCA was able to return meaningful results. Depending on the difficulty of the dataset, the performance of ICCA ranged from acceptable (Ground Truth dataset) to poor (BBC News Dataset). The more difficult datasets tend to break many of the assumptions that ICCA makes. The vectors in these datasets are very sparse, which ICCA does not account for directly. ICCA is also a linear method and so any nonlinear correlations will not be detected. For these more difficult datasets, the captions contained very nuanced language or words not even used in the main article.

The purpose of this chapter is to spark a discussion for using eigen-based correlation methods for image annotation, image retrieval, and possibly other NLP problems. While ICCA performs worse than the current NLP techniques it is able to capture underlying meaning between words and images. By applying NLP techniques to create better feature vectors than tf*idf weights and extending ICCA to allow for non-linear sparse correlations, one could hope for improved performance. While CCA was rightfully overlooked as a possible solution to such information retrieval problems, we hope that practitioners will reconsider eigen-based correlation approaches in the future.



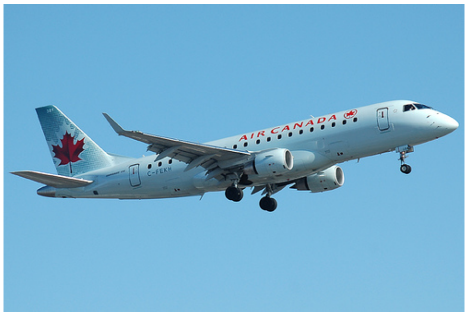
(a) CCA Results

(b) CCA Results Zoomed

(c) ICCA Results

(d) ICCA Results Zoomed

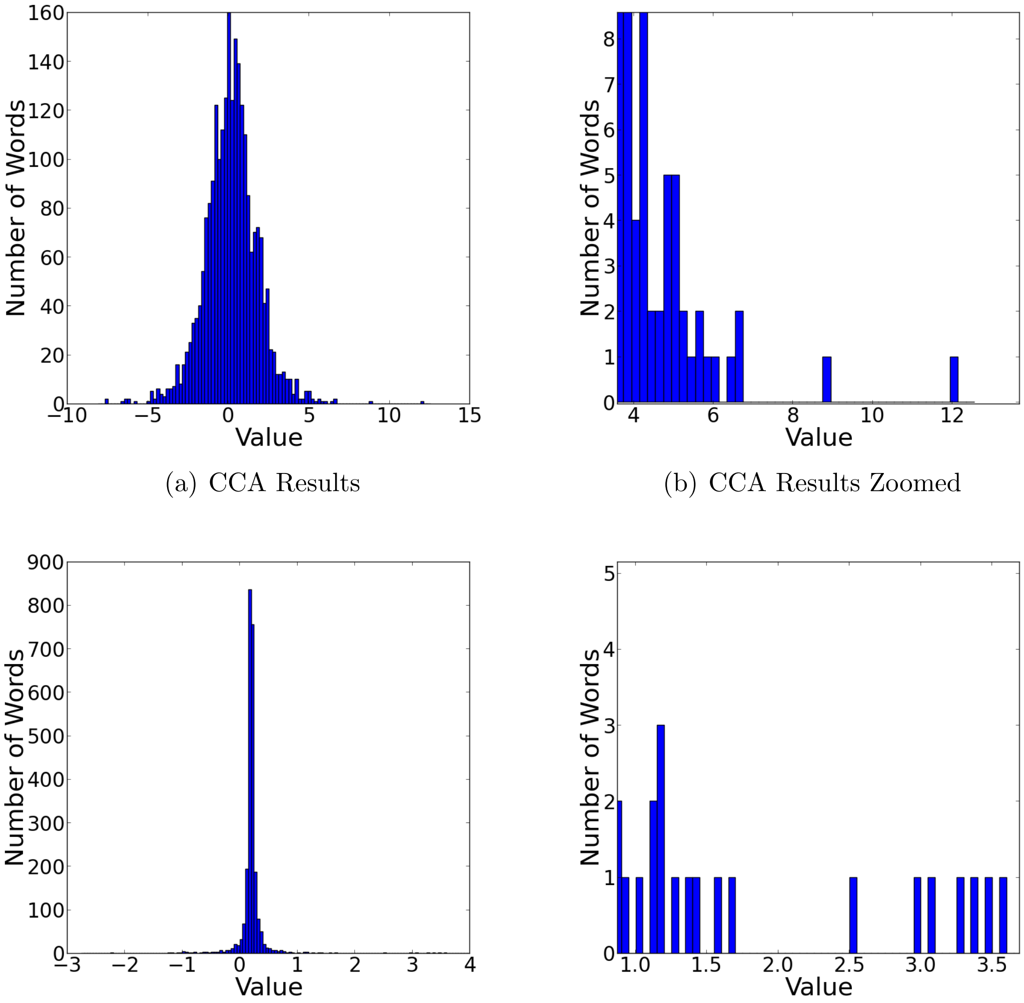
Figure 9.9: (a) Scores for all 1000 Pascal images for the query "airplane" for CCA. (b) Zoomed in version of (a) to highlight the top scores returned in Figure 9.8(a). (c) Scores for all 1000 Pascal images for the query "airplane" for ICCA. (d) Zoomed in version of (c) to highlight the top scores which are returned in Figure 9.8(b). All scores are the norm in (9.13).



(a) Image Query

CCA Annotation	**ICCA Annotation**
1. hairless	1. plane
2. buddi	2. ship
3. swan	3. cruis
4. leaf-less	4. fly
5. bnsf	5. blue
6. desert	6. jet
7. fluffi	7. airplan
8. salad	8. dock
9. majest	9. fighter
10. memorabilia	10. through

Figure 9.10: CCA vs ICCA annotation results for the image query shown in 9.10(a).



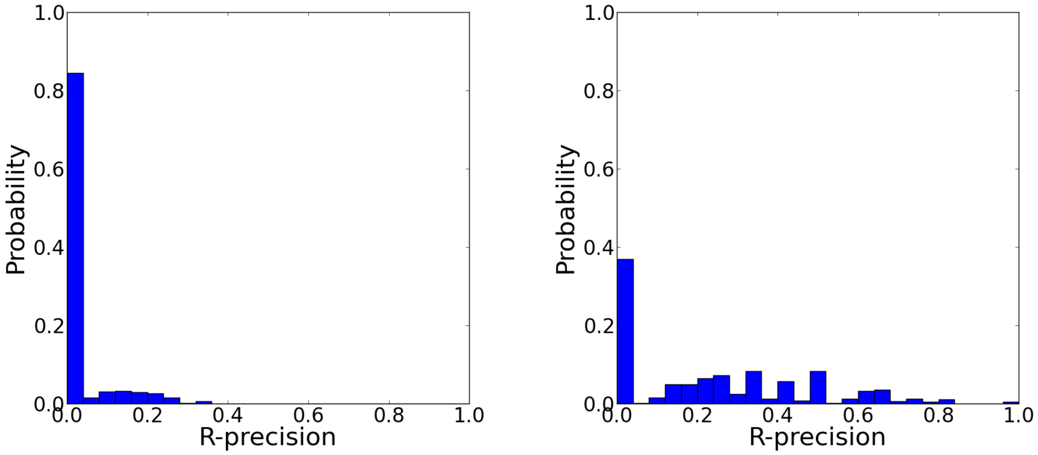
(a) CCA Results

(b) CCA Results Zoomed

(c) ICCA Results

(d) ICCA Results Zoomed

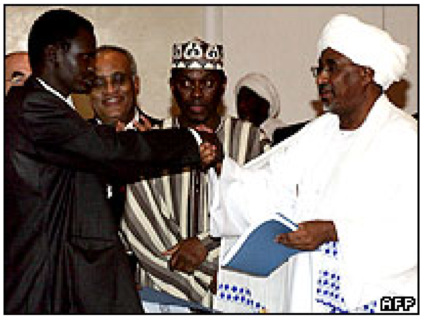
Figure 9.11: (a) CCA scores for all words in the Pascal database for the image query in Figure 9.10(a). (b) Zoomed in version of (a) to highlight the top scores returned in Figure 9.10. (c) ICCA scores for all words in the Pascal database for the image query in Figure 9.10(a). (d) Zoomed in version of (c) to highlight the top scores which are returned in Figure 9.10. All scores are the norm in (9.14).



(a) CCA

(b) ICCA

Figure 9.12: Empirical probability density functions of R-precision of image annotation of the University of Washington Ground Truth Dataset.



Caption

- Agreement came despite reservations on both sides

Title

- UN Secretary General Kofi Annan has called on the Sudanese government to allow a UN assessment team into the war-torn region of Darfur

Figure 9.13: Example of an image, its caption, and title in the BBC News dataset

CHAPTER X

Multiset CCA (MCCA)

10.1 Introduction

The correlation algorithms considered thus far are useful only when there are exactly two datasets. However, in many applications, we may have access to multiple datasets of high dimensional features that we believe contain correlated signals. Access to more than two datasets arises in applications such as handwritten digit classification [111], multi-temporal hyperspectral imaging [55], and medical imaging [40, 36].

The theory of multiset canonical correlation analysis (MCCA) has evolved over the past decades. The earliest work on extending CCA to three datasets was conducted by Vinograde [153]. This work found the canonical form of the three dataset correlation matrix but made no attempts at finding the canonical vectors. In [154], Steel considers the particular objective function of minimizing the generalized variance between the canonical variates of an arbitrary number of datasets. In 1961, Horst first considered the practical problem of fusing features from multiple datasets [155, 156]. He provides a solution for two particular objective functions originally called the "maximum correlation method", which is now called the sum of correlations method, and the "rank one approximation method", which is now called the maximum variance method. A decade later, Kettenring [64] considered a more general extension of Hotellings's [4] original CCA work. He considers five objective functions that extend CCA to multiple datasets. Each objective function represents some notion of multiset correlation. All five formulations of multiset CCA return canonical vectors for each dataset and correlation coefficients and each reduce to CCA when only two datasets are present. Two decades later, Nielsen [65] extended Kettenring's analysis by also considering four constraint functions placed on the canonical vectors in the optimization problem.

The five objective functions posed by Kettenring and four constraint functions posed by Nielsen give rise to twenty different optimization problems and thus twenty different formulations of MCCA. In this section, we consider all twenty such optimization problems. We begin by deriving the theoretical solution to each of these, unifying the works above and completing any formulations previously unsolved. As the performance of empirical MCCA has not previously been studied, we also derive empirical versions of each MCCA formulation using training data SVDs of each dataset. In Appendix B, we derive a solution to each of the twenty optimization problems formed by choosing one objective function and one constraint function. We then consider empirical version of each MCCA formulation.

We then consider the performance of one particular optimization problem, MAXVAR. We show that, similar to empirical CCA, the solution to this problem is a SVD of matrix with block entries of the pairwise product of right singular vectors of the individual datasets. We then apply the same principles used in ICCA to develop an informative version of MAXVAR, which we call IMCCA. Using the idea of trim-then-fuse, we propose to trim all data SVDs to include only the *informative* singular vectors. We provide some analysis of the behavior of these algorithms and provide a test statistic to use to determine the number of correlations present in multiple datasets. We discuss why multi-dataset correlation analysis is difficult but showcase on a real world dataset that IMCCA greatly outperforms MAXVAR and can robustly identify sources of correlation.

10.2 Mathematical Formulation of MCCA

Let $y_1, y_2, \dots, y_m$ be observations drawn from m distributions $y_i \sim \mathcal{Y}_i$ with $y_i \in \mathbb{C}^{d_i}$. Assume, without loss of generality, that y_i is zero mean. Define the covariance between distributions as $\mathbb{E}\left[y_i y_j^T\right] = R_{ij}$ for $i, j = 1, \dots, m$. Define the joint observation vector y and its covariance $R = \mathbb{E}\left[yy^H\right]$ as

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{C}^{d \times 1}, \quad R = \begin{bmatrix} R_{11} & \dots & R_{1m} \\ \vdots & \ddots & \vdots \\ R_{m1} & \dots & R_{mm} \end{bmatrix} \in \mathbb{C}^{d \times d}$$

where $d = \sum_{i=1}^m d_i$.

The goal of MCCA is to find canonical coefficient vectors, $x_i \in \mathbb{C}^{d_i \times 1}$ for $i = 1, \dots, m$, such that the canonical variates, $w_i = x_i^H y_i$, are optimal with respect to an objective function $J(\cdot)$ and constraint function $h(\cdot)$. We consider five objective functions [64] in Section 10.2.2 and four constraints functions [65] in Section 10.2.1. Define the vector of canonical vectors as $x = \left[x_1^H, \ldots, x_m^H\right]^H \in \mathbb{C}^{d\times 1}$ and the vector of canonical variates as $w = [w_1, \ldots, w_m]^H \in \mathbb{C}^{m\times 1}$. The covariance matrix of w is

$$\Phi(x) = \mathbb{E}\left[ww^H\right] = \begin{bmatrix} x_1^H R_{11} x_1 & \ldots & x_1^H R_{1m} x_m \\ \vdots & \ddots & \vdots \\ x_m^H R_{m1} x_1 & \ldots & x_m^H R_{mm} x_m \end{bmatrix}.$$

Using this notation, the MCCA optimization problem is

$$\begin{aligned} \underset{x}{\text{optimize}} \quad & J(\Phi(x)) \\ \text{subject to} \quad & h(x, R). \end{aligned} \tag{10.1}$$

10.2.1 Constraint Functions, $h(x, R)$

In [55, 65], Nielsen describes four constraints placed on the canonical vectors that are natural to use in MCCA. Using our notation and new naming scheme, these constraint functions are:

a) NORM - The canonical coefficient vectors each have unit norm.

$$h(x, R) = x_i^H x_i = 1,\ 1 \le i \le m$$

This objective function has the same flavor as other machine learning algorithms such as PCA.

b) AVGNORM - The vector of canonical vectors, x, has unit norm.

$$h(x, R) = x^H x = \sum_{i=1}^{m} x_i^H x_i = 1$$

c) VAR - The canonical variates each have unit variance.

$$x_i^H R_{ii} x_i = 1,\ 1 \le i \le m.$$

This is the natural extension of the CCA constraint functions.

d) AVGVAR - The canonical variates have average variance of $1/m$.

$$\sum_{i=1}^{m} x_i^H R_{ii} x_i = 1.$$

This may be written $\mathbf{tr}(X^H R X) = 1$, where $X = \mathbf{blkdiag}(x_1, \ldots, x_m)$.

10.2.2 Objective Functions, $J(\Phi(x))$

In [64], Kettenring describes five objective functions, each used to detect a different form of linear relationship among the datasets. Under the VAR and AVGVAR constraints above, each of the objective functions reduces to the standard CCA formulation and thus the standard CCA solution. Using our notation, these objective functions are:

1. **SUMCORR** - Maximize the sum of the correlations between each of the canonical variates.

$$J(\Phi(x)) = \max_{x_1,\ldots,x_m} \sum_{i=1}^{m} \sum_{i=1}^{m} x_i^H R_{ij} x_j = \max_{x_1,\ldots,x_m} \mathbf{1}^H \Phi(x) \mathbf{1}$$

This is the natural extension of the CCA objective function. It was first proposed by Horst in [155].

2. **SSQCORR** - Maximize the sum of the squares of the correlations between each of the canonical variates.

$$J(\Phi(x)) = \max_{x_1,\ldots,x_m} \sum_{i=1}^{m} \sum_{j=1}^{m} (x_i^H R_{ij} x_j)^2 = \max_{x_1,\ldots,x_m} \|\Phi(x)\|_F^2 = \max_{x_1,\ldots,x_m} \sum_{i=1}^{m} \lambda_i^2(\Phi(x)).$$

where λ_i are the eigenvalues of $\phi(x)$. This is very similar to SUMCORR except that it penalizes small pairwise correlations more than SUMCORR does. Under the VAR constraint, the $m \times m$ identity matrix is the least informative $\Phi(x)$ as this denotes no correlation between any of the canonical variates. Therefore, we want $\Phi(x)$ to be as different as possible from the identity matrix. Under the VAR constraint, this is what the SSQCORR objective function accomplishes. It was first proposed in 1971 by Kettenring [64].

3. **MAXVAR** - Maximize the largest eigenvalue of Φ, $\lambda_1(\Phi(x))$.

$$J(\Phi(x)) = \max_{x_1,\ldots,x_m} \lambda_1(\Phi(x))$$

   MAXVAR was created by Horst in [155] to find the canonical vectors that give $\Phi(x)$ the best approximation (in the Frobeneus norm) to a rank-1 matrix. Horst's original name for this method was the "rank one approximation method". The corresponding largest eigenvalue, $\lambda_1(\Phi(x))$ is a notion of variance and thus the new name.

4. **MINVAR** - Minimize the smallest eigenvalue of Φ, $\lambda_m(\Phi(x))$.

$$J(\Phi(x)) = \min_{x_1,\ldots,x_m} \lambda_d(\Phi(x))$$

   Instead of maximizing the energy in the top eigenvalue, we wish to minimize the energy in the last eigenvalue. In [157], MINVAR is shown to have the desired property that the minimal eigenvalue has a fixed range in $[0, 1]$ whereas the maximal eigenvalue found by MAXVAR has a range dependent on the dimensions of the variables. It was first proposed in 1971 by Kettenring [64].

5. **GENVAR** - Minimize the generalized variance of w, which is equivalent to minimizing the determinant of the correlation matrix of w.

$$J(\Phi(x)) = \min_{x_1,\ldots,x_m} |\Phi(x)| = \min_{x_1,\ldots,x_m} \prod_{i=1}^{m} \lambda_i(\Phi(x))$$

   This is the oldest of the five criterion and was proposed by Steel in 1951 [154]. This seems to involve a tradeoff between choosing x to have large leading eigenvalues and small tail eigenvalues.

10.3 Theoretical and Empirical MCCA Derivations

In this section, we provide a solution for each of the twenty MCCA formulations based on the five objection functions described in Section 10.2.2 and four constraint functions described in Section 10.2.1. Some of these solutions have been previously reported in [64, 65]. We complete the analysis and unify the results. We provide the empirical solution for each algorithm provided training data matrices $Y_1, \ldots, Y_m$. In such a setting, we are given n samples (observations) from each data distribution.

y_i	Observation from dataset i
y	$[y_1^H, \ldots, y_m^H]^H$
d_i	Dimension of y_i
$d = \sum_{i=1}^m d_i$	Dimension of y
m	Number of datasets
n	Number of observations
$x_i \in \mathbb{C}^{d_i}$	Canonical coefficient vector
$x \in \mathbb{C}^d$	$[x_1^H, \ldots, x_m^H]^H$
$w_i \in \mathbb{C}$	Canonical variate
$w \in \mathbb{C}^m$	$[w_1, \ldots, w_m]^H$
$X \in \mathbb{C}^{d \times m}$	$\mathbf{blkdiag}(x_1, \ldots, x_m)$
$\Phi(x)$	Correlation matrix of w
$R_D \in \mathbb{C}^{d \times d}$	$\mathbf{blkdiag}(R_{11}, \ldots, R_{mm})$
$R \in \mathbb{C}^{d \times d}$	Matrix of $[R_{ij}]_{ij}$
$\widetilde{R}(x) \in \mathbb{C}^{d \times d}$	Matrix of$[(x_i^H R_{ij} x_j) R_{ij}]_{ij}$
$Y_i \in \mathbb{C}^{d_i \times n}$	Training data matrix
$U_i \in \mathbb{C}^{d_i \times d_i}$	Left singular vectors of Y_i
$U \in \mathbb{C}^{d \times m}$	$\mathbf{blkdiag}(U_1, \ldots, U_m)$
$\Sigma_i \in \mathbb{C}^{d_i \times n}$	Singular values matrix of Y_i
$\Sigma \in \mathbb{C}^{d \times nm}$	$\mathbf{blkdiag}(\Sigma_1, \ldots, \Sigma_m)$
$V_i \in \mathbb{C}^{n \times n}$	Right singular vectors of Y_i
$V \in \mathbb{C}^{n \times nm}$	$[V_1, \ldots, V_m]$
$\Lambda \in \mathbb{C}^{m \times m}$	Diag matrix of Lagrange multipliers
$\Lambda_D \in \mathbb{C}^{d \times d}$	$\mathbf{blkdiag}\,(\lambda_1 I_{d_1}, \ldots, \lambda_m I_{d_m})$
$\widetilde{\Sigma} \in \mathbb{C}^{d \times d}$	$\mathbf{blkdiag}(\Sigma_1(:, 1 : d_1), \ldots \Sigma_m(:, 1 : d_m))$
$\widetilde{V} \in \mathbb{C}^{n \times d}$	$[V_1(:, 1 : d_1), \ldots, V_m(:, 1 : d_m)]$
$\mathbf{1}$	$[1, \ldots, 1]$

Table 10.1: Notation used in MCCA

Using these n samples, we form m training data matrices by stacking the observations as columns in a matrix. We denote these training data matrices $Y_1 = \left[y_1^{(1)}, \ldots y_1^{(n)}\right] \in \mathbb{C}^{d_1 \times n}, \ldots, Y_m = \left[y_m^{(1)}, \ldots, y_m^{(n)}\right] \in \mathbb{C}^{d_m \times n}$.

For all empirical derivations, we assume that we are given n samples in each training dataset. We denote the SVD of each training dataset as $Y_i = U_i \Sigma_i V_i^H$ and form the matrices $U \in \mathbb{C}^{d \times d} = \mathbf{blkdiag}(U_1, \ldots, U_m)$, $\Sigma \in \mathbb{C}^{d \times nm} = \mathbf{blkdiag}(\Sigma_1, \ldots, \Sigma_m)$, and $V \in \mathbb{C}^{n \times nm} = [V_1, \ldots, V_m]$. Using these data SVDs, we form sample covariance matrices, $\widehat{R}_{ij} = \frac{1}{n} Y_i Y_j^H = \frac{1}{n} U_i \Sigma_i V_i^H V_j \Sigma_j^H U_j^H$ with which we form $\widehat{R} = U \Sigma V^H V \Sigma^H U^H$ and $\widehat{R}_D = U \Sigma \Sigma^H U^H$. Please refer to Table 10.1 for a summary of the notation used throughout the MCCA derivations.

The derivations are provided in Appendix B. Table 10.2 in the following section summarizes the solution to each problem. It assigns a number-letter pair to each MCCA optimization problem (1-5 for the objective function, a-d for the constraint function). This label can be used to look up the appropriate derivation in Appendix B. The table provides the appropriate eigen-system used to solve the problem if all the covariance matrices are known. The table also provides the appropriate eigen-system used to solve the problem in the empirical setting where we are given training datasets to estimate unknown covariance matrices. The last column in the table provides references that use, discuss, or derive the MCCA formulation.

10.3.1 Manopt Software for Optimization on Manifolds

Many of the problems discussed in Appendix B do not yield closed form solutions because either the cost function is unwieldy or because the constraint functions complicate the derivations. For these problems we use the Manopt software provided at *www.manopt.org*. The Manopt software specializes in solving constrained optimization problems when the constraints are manifolds. This software package is able to solve nonlinear optimization problems. For reference, see [158]. To use the solvers, we must provide a cost function and its associated gradient.

All of our constraints will be of the form $\|x\| = 1$ where $x \in \mathbb{R}^p$. The associated manifold that we use for this constraint is the sphere manifold called via `spherefactory(p,1)`. If we have multiple of such constraints, then we use the `productmanifold` to ensure all constraints are satisfied. See the Manopt documentation and provided code for an example.

After selecting the appropriate manifold and providing the cost and gradient functions, we use the `trustregions` solver to find a solution for our problems. This returns the minimized cost and the point that achieved the minimum cost. If our objective function has a cost function that seeks a maximum, we provide the negative of the true cost function and the gradient is computed from this negative cost.

10.3.2 Successive Canonical Vectors

The derivations in Appendix B show how to compute the first stage canonical vectors and canonical correlation. We may compute $r = \min(d_1, \ldots, d_m, n)$ canonical vector and correlation pairs. We use the standard constraint on successive canonical variates

$$\mathbb{E}\left[w_i^{(k)} w_i^{(k-j)}\right] = 0, \quad \text{for } j = 1, \ldots, k-1, \ \forall i.$$

Here, the subscript i indexes the canonical variates and the superscript (k) indexes the stage of the canonical vector and correlation pair. This requires the next stage canonical variates to be uncorrelated to all previous canonical variates for a given dataset. Using the definition for canonical variates, this constraint becomes

$$\mathbb{E}\left[x_i^{(k)H} y_i y_i^H x_i^{(k-j)}\right] = x_i^{(k)H} R_{ii} x_i^{(k-j)} = 0, \quad \text{for } j = 1, \dots, k-1, \ \forall i.$$

To enforce this constraint, we run the following algorithm

1. Form $X \in \mathbb{C}^{d \times mk} = \textbf{blkdiag}(X_1, \dots, X_m)$ where $X_i = [x_i^{(1)}, \dots, x_i^{(k)}]$
2. Project the canonical vectors onto R_D via $B = R_D X \in \mathbb{C}^{d \times mk}$
3. Compute a basis for the span of B via the rank-k SVD, $B = U_B \Sigma_B V_B^H$
4. Form the projection matrix onto the orthogonal completment of this basis $P = I - U_B U_B^H$
5. Project the training data onto P via $\widetilde{Y} = PY$
6. Recompute covariance matrices used in optimization using $\widetilde{Y}$

10.3.3 MCCA Summary

Table 10.2 summarizes the solution to each of the twenty optimization problems and shows for which ones we must use the Manopt software package and which ones we have closed form solutions in terms of eigen-systems. All of the empirical eigenvalue systems rely on the matrix product $V^H V$. This is wonderful news because it directly makes contact with the similar $V_x^H V_y$ matrix used in CCA. In Chapter 4 we saw that by trimming this matrix to only include informative singular vectors of the individual datasets we can greatly improve correlation detection. This observation will drive our derivation of IMCCA. We note that in this thesis we only do so for MAXVAR, but this $V^H V$ matrix appears in many of the optimization problems and other such informative versions of these algorithms are within reach. Many of the SUMCORR and SSQCORR theoretical eigen-systems are non-normal, using multiple Lagrange multipliers. Some of these problems can be solved with Manopt, however, some result in non-unique solutions. Obviously, such formulations of MCCA should be avoided.

#	$J(x)$	$h(x,R)$	Eigenvalue Prob	Empirical Prob	Ref
1a	SUMCORR	NORM	$R\widetilde{x} = \Lambda_D\widetilde{x}$ $x = \Lambda_{\widetilde{x}}\widetilde{x}$	Manopt	[55, 65]
1b	SUMCORR	AVGNORM	$Rx = \rho x$	$\widehat{R}\widehat{x} = \widehat{\rho}\widehat{x}$	[65]
1c	SUMCORR	VAR	$R\widetilde{x} = \Lambda_D R_D\widetilde{x}$ $x = R_D^{-1/2}\Lambda_{\widetilde{x}}\widetilde{x}$	Manopt	[55, 64, 65]
1d	SUMCORR	AVGVAR	$R_D^{-1/2}RR_D^{-1/2}\widetilde{x} = \rho\widetilde{x}$ $x = R_D^{-1/2}\widetilde{x}$	$\widetilde{V}^T\widetilde{V}\widehat{f} = \widehat{\rho}\widehat{f}$ $\widehat{x} = U\widetilde{\Sigma}^{-1}\widehat{f}$	[36, 55, 51] [65, 111]
2a	SSQCORR	NORM	$\widetilde{R}(x)x = \Lambda_D x$	Manopt	[65]
2b	SSQCORR	AVGNORM	$\widetilde{R}(x)x = \lambda x$	Manopt	[65]
2c	SSQCORR	VAR	$\widetilde{R}(x)x = \Lambda_D R_D x$	Manopt	[40, 64, 65]
2d	SSQCORR	AVGVAR	$\widetilde{R}(x)x = \lambda R_D x$	Manopt	[65]
3a	MAXVAR	NORM	$R\tilde{a} = \rho\tilde{a}$ $x = \Lambda_{\tilde{a}}^{-1}\tilde{a}$	$\widehat{R}\widehat{f} = \widehat{\rho}\widehat{f}$ $\widehat{x} = \Lambda_{\widehat{f}}^{-1}\widehat{f}$	[65]
3b	MAXVAR	AVGNORM	$x_i = u_{1i}$	$\widehat{x}_i = u_{1i}$	[65]
3c	MAXVAR	VAR	$R_D^{-1/2}RR_D^{-1/2}\tilde{a} = \rho\tilde{a}$ $x = R_D^{-1/2}\Lambda_{\widetilde{a}}^{-1}\widetilde{a}$	$\tilde{V}^H\tilde{V}\widehat{f} = \widehat{\rho}\widehat{f}$ $\widehat{x} = U\widetilde{\Sigma}^{-1}\Lambda_{\widehat{f}}^{-1}\widehat{f}$	[64, 65]
3d	MAXVAR	AVGVAR	Non-unique $x = u_i/\sigma_i$	Non-unique $\widehat{x} = u_i/\sigma_i$	[36, 51, 65]
4a	MINVAR	NORM	$R\tilde{a} = \rho_{\min}\widetilde{a}$ $x = \Lambda_{\widetilde{a}}^{-1}\widetilde{a}$	$\widehat{R}\widehat{a} = \widehat{\rho}_{\min}\widehat{a}$ $\widehat{x} = \Lambda_{\widehat{a}}^{-1}\widehat{a}$	[65]
4b	MINVAR	AVGNORM	Non-unique $x_i = u_{1i}$	Non-unique $\widehat{x}_i = u_{1i}$	[65]
4c	MINVAR	VAR	$R_D^{-1/2}RR_D^{-1/2}\widetilde{a} = \rho_{\min}\widetilde{a}$ $x = R_D^{-1/2}\Lambda_{\widetilde{a}}^{-1}\widetilde{a}$	$\widetilde{V}^H\widetilde{V}\widehat{f} = \widehat{\rho}_{\min}\widehat{f}$ $\widehat{f} = U\widetilde{\Sigma}^{-1}\Lambda_{\widehat{f}}\widehat{f}$	[64, 65]
4d	MINVAR	AVGVAR	Non-unique $x = u_i/\sigma_i$	Non-unique $\widehat{x} = u_i/\sigma_i$	[65, 157]
5a	GENVAR	NORM	Non-eigen prob	Manopt	[65]
5b	GENVAR	AVGNORM	Non-eigen prob	Manopt	[65]
5c	GENVAR	VAR	Non-eigen prob	Manopt	[64, 65]
5d	GENVAR	AVGVAR	Non-eigen prob	Manopt	[65]

Table 10.2: Summary of MCCA optimization problems. The objective functions are described in Section 10.2.2. The constraints are described in section 10.2.1. The eigenvalue problem column is the theoretical solution while the Empirical problem column describes how to solve the problem given empirical data. All eigenvalue problems solve for the maximum eigenvalue-eigenvector pair except for the MINVAR problems, which solves for the minimum eigenvalue-eigenvector pair. The final column lists references which describe the MCCA optimization problem.

10.4 Proposed Informative MCCA Algorithm

We choose to focus our attention on two of the above problems, 3c and 4c, MAXVAR and MINVAR with the VAR constraint. We choose to examine these because they are a very natural extension from CCA. One can show that each of the objective functions is equivalent to the CCA objective function when $m = 2$. To draw a natural connection from CCA to MAXVAR, we first recall that the canonical correlations of empirical CCA are exactly the singular values of

$$C = V_1^H V_2, \tag{10.2}$$

where V_1 and V_2 are the right singular vectors of the data matices Y_1 and Y_2, respectively. Define the SVD of $C = FKG^H$. Consider the matrix

$$R_{\text{cca}} = \begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} I_{d_1} & V_1^H V_2 \\ V_2^H V_1 & I_{d_2} \end{bmatrix}. \tag{10.3}$$

As shown in [157], R has eigenvalues that come in pairs

$$\{1 + k_i, 1 - k_i, \}.$$

More specifically, the eigenvalue decomposition of R_{cca} is

$$R_{\text{cca}} = \begin{bmatrix} F & -F \\ G & G \end{bmatrix} \begin{bmatrix} I + \left(KK^H\right)^{-1/2} & 0 \\ 0 & I - \left(K^H K\right)^{-1/2} \end{bmatrix} \begin{bmatrix} F & -F \\ G & G \end{bmatrix}^H.$$

From the eigenvalue decomposition of this matrix, we can exactly recover the canonical correlations k_i and the needed transformations F and G. These transformations appear in a very specific block structure. Each eigenvector contains the corresponding block components of the transformation for each dataset. In addition, just as the eigenvalues come in pairs, the eigenvectors come in pairs. Comparing the eigenvectors corresponding to the eigenvalues $1 + k_i$ and $1 - k_i$ we see that the component corresponding to the second dataset is the same while the component of the first dataset simply changes sign.

Therefore, the CCA solution by taking the SVD of C in 10.2 is equivalent to maximizing the largest $\min(d_1, d_2)$ eigenvalues of R in (10.3). However, we can also uncover the same solution by minimizing the smallest $\min(d_1, d_2)$ eigenvalues of R as well. We note that the CCA optimization problem explicitly uses the VAR constraint function.

From this discussion it is clear that MAXVAR and MINVAR using the VAR constraint are extremely natural extensions for the CCA optimization problem. We simply concatenate the right singular vectors of any additional datasets to the matrix product in (10.3) to form

$$C_{\text{mcca}} = \begin{bmatrix} V_1^H \\ V_2^H \\ \vdots \\ V_m^H \end{bmatrix} \begin{bmatrix} V_1 & V_2 & \cdots & V_m \end{bmatrix} = \begin{bmatrix} I_{d_1} & V_1^H V_2 & \cdots & V_1^H V_m \\ V_2^H V_1 & I_{d_2} & \cdots & V_2^H V_m \\ \vdots & \vdots & \ddots & \vdots \\ V_m^H V_1 & V_m^H V_2 & \cdots & I_{d_m} \end{bmatrix}. \qquad (10.4)$$

If our datasets are completely noise free, then examining the eigenvalues of C_{mcca} directly gives us the correlation structure. Any eigenvalues $k_i \neq 1$ represent correlated components. We can have at most

$$r = \left\lfloor \frac{\sum_{i=1}^m d_i}{2} \right\rfloor$$

correlations. Having this many number of correlated components would require all correlations to be pair-wise.

Interpreting the eigenvalues and eigenvectors of C_{mcca} is much more complicated than CCA. While any eigenvalue not equal to 1 conveys a correlation between datasets, the strength of this correlation is coupled with the eigenvector structure. Consider the following examples both for $m = 3$ and $d_1 = d_2 = d_3 = 1$

$$C_{\text{mcca}}^{(1)} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C_{\text{mcca}}^{(2)} = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix}.$$

$C_{\text{mcca}}^{(1)}$ corresponds to a setting where the only component of dataset 1 and 2 are perfectly correlated and dataset 3 is independent. $C_{\text{mcca}}^{(2)}$ corresponds to the setting where all three components are weakly mutually correlated. However, the largest eigenvalue of both of these matrices is 2. This eigenvalue is not one so it represents correlation in our datasets. However, the structure of our correlation is very different.

To determine the structure of our correlation, we must examine the associated eigenvectors of C_{mcca}. The eigenvectors corresponding to the largest eigenvalues of these matrices are

$$u^{(1)} = \frac{1}{\sqrt{2}} [1, 1, 0]^T, \quad u^{(2)} = \frac{1}{\sqrt{3}} [1, 1, 1]^T.$$

From these eigenvectors, we directly can see the correlation structure revealed by the eigenvector. This ambiguity in correlation structure is never a problem in vanilla CCA as there are only two datasets. Components are either correlated or they are not and so the eigenvalue in CCA directly gives our correlation structure and we can determine our necessary transformations directly from the eigenvectors. In MAXVAR, after determining that our eigenvalue represents a correlation, we then need to inspect the corresponding eigenvector to determine the correlation structure.

Similar to the result by [6], we would like to determine when the eigenvalues returned by empirical MAXVAR erroneously represent false correlation. In CCA, we had the result that when $n < d_1 + d_2$, $k_1 = 1$ deterministically. For MAXVAR and MINVAR, we provide the following two Theorems and Conjecture.

Theorem 10.4.1. *If $2n < \min_{i \neq j \neq k}(d_i + d_j + d_k)$ then the largest eigenvalue of C_{mcca} is equal to m.*

Proof. By definition, V_i is a $n \times d_i$ matrix. Without loss of generality, let V_i be ordered such that $d_1 \leq d_2 \leq d_m$. When $n < d_1 + d_2$, then V_1 and V_2 must span a shared subspace of dimension $d_1 + d_2 - n$. By a similar geometric argument, this shared subspace of dimension $d_1 + d_2 - n$ will intersect the span of V_3 if

$$(d_1 + d_2 - n) + d_3 > n.$$

Re-arranging terms means that

$$d_1 + d_2 + d_3 > 2n$$

implies that V_1, V_2 and V_3 all span a common subspace of dimension $(d_1 + d_2 + d_3) - 2n$. In this setting we have that $d_1 + d_2 > n$ and $d_1 + d_2 + d_3 > 2n$ which implies that $d_3 > n$. Therefore for any $i > 3$, $d_i > n$. By induction, we have that for any $i > 3$, V_i will intersect the common subspace of dimension $(d_1 + d_2 + d_3) - 2n$ if

$$(d_1 + d_2 + d_3 - 2n) + d_i > n,$$

which holds if $d_i > n$, which we just showed was true. Therefore, when

$$2n < \min_{i \neq j \neq k}(d_i + d_j + d_k)$$

all V_i span a common subspace of at least dimension 1.

With this observation in mind, we may write

$$V_i = \begin{bmatrix} q & q_i^{\perp} \end{bmatrix}$$

where q is a basis vector for this shared subspace and $q_i^{\perp}$ is the orthogonal complement representing the rest of V_i. With this definition, we have that for any $i \neq j$,

$$V_i^H V_j = \begin{bmatrix} 1 & 0 \\ 0 & Q_{ij} \end{bmatrix},$$

where $Q_{ij} = \left(q_i^{\perp}\right)^H q_j^{\perp}$. With this block structure we can observe that the first column and row of C_{mcca} is

$$w = \left[e_1^H, e_2^H, \ldots, e_m^H\right]^H,$$

where $e_i^H = [1, 0, \ldots, 0] \in \mathbb{C}^{1\times d_i}$. Therefore, it is clear that $\frac{1}{m}w$ is an eigenvector of C_{mcca} associated with the eigenvalue m. □

Theorem 10.4.2. *If $n < \sum_{i=1}^m d_i$ then the smallest eigenvalue of C_{mcca} is zero.*

Proof. We provide a short proof by simply geometric and rank arguments. Recall that $V_i \in \mathbb{C}^{n\times d_i}$. Then

$$V = \begin{bmatrix} V_1^H \\ V_2^H \\ \vdots \\ V_m^H \end{bmatrix}$$

is a $\sum_{i=1}^m d_i \times n$ matrix. When $n < \sum_{i=1}^m d_i$, this matrix can have a maximum rank of n. Therefore

$$\text{rank}(C_{\text{mcca}}) = \text{rank}(V^H V) \leq n < \sum_{i=1}^m d_i.$$

Therefore, C_{mcca} is not full rank and has at least 1 zero-eigenvalue. As C_{mcca} is symmetric positive semi-definite, the smallest eigenvalue is therefore zero. □

Conjecture 10.4.1. *We conjecture that when $n < \sum_{i=1}^m d_i$, the largest eigenvalue of C_{mcca} is determined entirely based on n and $\sum_{i=1}^m d_i$ and not on the underlying correlation.*

The intuition behind Conjecture 10.4.1 is based on the observation from CCA that the eigenvalues of R_{cca} come in pairs $\{1 + k_i, 1 - k_i\}$. MAXVAR returns the eigenvalues above 1 and MINVAR returns the eigenvalues below 1. However, unlike in CCA, the eigenvalues of C_{mcca} are not symmetric about 1 and so we do not have an elegant closed form relationship. Our intuition leads us to believe that these eigenvalues of C_{mcca} are coupled and represent the same correlation structure. Theorem 10.4.1 states that if we are in a certain sample deficient regime, the largest eigenvalues are deterministic. Similarly, Theorem 10.4.2 states that in a different sample deficient regime, the smallest eigenvalues are deterministic. These sample deficient regimes are different for the largest and smallest eigenvalues; the regime is larger for the smaller eigenvalues. Therefore, due to the hypothesized coupling of the largest and smallest eigenvalues we believe Conjecture 10.4.1 holds. In this setting, like in empirical CCA, we conjecture that the canonical vectors are simply random and we would not like to use them in an algorithm.

10.4.1 Low-Rank Multi-dataset Model

Let $y_1^{(i)} \in \mathbb{C}^{d_1 \times 1}, \dots, y_m^{(i)} \in \mathbb{C}^{d_m \times 1}$ be modeled as

$$y_j^{(i)} = U_j s_j^{(i)} + z_j^{(i)} \tag{10.5}$$

where for $j = 1, \dots, m$, $U_j^H U_j = I_{k_j}$ $z_j^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_{d_j})$. Furthermore, assume that

$$s_j^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, \Theta_j)$$

where $\Theta_j = \mathbf{diag}\left(\left(\theta_j^{(1)}\right)^2, \dots, \left(\theta_j^{(k_j)}\right)^2\right)$. Assume that for all i and j, $z_j^{(i)}$ are mutually independent and independent from all $s_j^{(i)}$. Finally, assume that

$$\mathbb{E}\left[s_j^{(i)} s_\ell^{(i)H}\right] =: K_{j\ell} = \Theta_j^{1/2} P_{j\ell} \Theta_\ell^{1/2}$$

where the entries of $P_{j\ell}$ are between -1 and 1 and represent the correlation between the entries of s_j and s_ℓ. For reasons to be made clear later, for $j, \ell = 1, \dots, m$ define

$$\widetilde{K}_{j\ell} = \left(\Theta_j + I_{d_j}\right)^{-1/2} K_{j\ell} \left(\Theta_\ell + I_{d_\ell}\right)^{-1/2}.$$

Under this model, we define the following covariance matrices

$$\begin{aligned} \mathbb{E}\left[y_j y_j^H\right] &= U_j \Theta_j U_j^H + I_{d_j} =: R_{jj} \\ \mathbb{E}\left[y_j y_\ell^H\right] &= U_j K_{j\ell} U_\ell^H =: R_{j\ell}. \end{aligned} \tag{10.6}$$

With this model we have that our target matrix in MAXVAR is

$$C_{\text{mcca}} = R_D^{-1/2} R R_D^{-1/2}$$

where

$$\begin{aligned} R &= \mathbb{E}\left[\begin{bmatrix} y_1^{(i)} \\ y_2^{(i)} \\ \vdots \\ y_m^{(i)} \end{bmatrix}\begin{bmatrix} y_1^{(i)H} & y_2^{(i)H} & \cdots & y_m^{(i)H} \end{bmatrix}\right] \\ &= \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{m1} & R_{m2} & \cdots & R_{mm} \end{bmatrix} \end{aligned}$$

and

$$R_D = \mathbf{blkdiag}(R_{11}, R_{22}, \ldots, R_{mm}).$$

With these definitions, the block diagonal elements of C_{mcca} are I_{d_j} and the block off-diagonal elements are $R_{jj}^{-1/2} R_{j\ell} R_{\ell\ell}^{-1/2}$. Using our model in (10.5), we have that

$$\begin{aligned} R_{jj}^{-1/2} R_{j\ell} R_{\ell\ell}^{-1/2} &= \left(U_j \Theta_j U_j^H + I_{d_j}\right)^{-1/2} U_j K_{j\ell} U_\ell^H \left(U_\ell \Theta_\ell U_\ell^H + I_{d_\ell}\right)^{-1/2} \\ &= U_j \left(\Theta_j + I_{d_j}\right)^{-1/2} K_{jk} \left(\Theta_\ell + I_{d_\ell}\right)^{-1/2} U_\ell^H \\ &= U_j \widetilde{K}_{j\ell} U_\ell^H. \end{aligned}$$

Therefore, by defining $U = \mathbf{blkdiag}(U_1, \ldots, U_m)$, we have that

$$C_{\text{mcca}} = U \underbrace{\begin{bmatrix} I_{k_1} & \widetilde{K}_{12} & \cdots & \widetilde{K}_{1m} \\ \widetilde{K}_{21} & I_{k_2} & \cdots & \widetilde{K}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{K}_{m1} & \widetilde{K}_{m2} & \cdots & I_{k_m} \end{bmatrix}}_{\widetilde{K}} U^H.$$

Finally, let $U_{\widetilde{K}} K_{\widetilde{K}} U_{\widetilde{K}}^H$ be the eigenvalue decomposition of $\widetilde{K}$ where

$$K_{\widetilde{K}} = \mathbf{diag}(\kappa_1, \ldots, \kappa_k),$$

where $k = \sum_{j=1}^m k_j$. The number of correlated components is thus equal to the number of eigenvalues of C_{mcca} that are greater than one,

$$\# \text{ of correlated components} =: t = \sum_{i=1}^{k} \mathbb{1}_{\{\kappa_i > 1\}}.$$

10.4.2 Informative MCCA

In many applications, however, we do not know the covariance matrices $R_{j\ell}$. Therefore, we cannot determine the number of correlated components by examining the rank of C_{mcca}. Instead, we are given multiple snapshots from (10.5) that we stack columnwise into data matrices

$$Y_j = \left[y_j^{(1)}, \ldots, y_j^{(n)} \right].$$

Using these data matrices, we form estimates of the unknown covariance matrices via

$$\widehat{R}_{j\ell} = \frac{1}{n} Y_j Y_\ell^H.$$

Define the data SVDs as

$$Y_j = \widehat{U}_j \widehat{\Sigma}_j \widehat{V}_j^H,$$

the trimmed matrices as

$$\begin{aligned} \widetilde{U}_j &= \widehat{U}_j(:, 1 : \min(d_j, n)) \\ \widetilde{U}_j &= \widehat{V}_j(:, 1 : \min(d_j, n)), \end{aligned}$$

and the matrices

$$\widetilde{U} = \mathbf{blkdiag}(\widetilde{U}_1, \ldots, \widetilde{U}_m), \quad \widetilde{V} = \left[\widetilde{V}_1, \ldots \widetilde{V}_m \right].$$

Empirical MAXVAR examines the eigen-decomposition of the matrix

$$\widehat{R}_{\text{mcca}} = \widetilde{U} \widetilde{V}^H \widetilde{V} \widetilde{U}.$$

Define the eigenvalues of this matrix as $\widehat{\kappa}_1, \ldots, \widehat{\kappa}_d$,which are the MAXVAR estimates of the correlations present in the multiple datasets. MAXVAR may return up to a maximum of

$$r = \left\lfloor \frac{\sum_{i=1}^m d_i}{2} \right\rfloor$$

correlations. However, based on the model in (10.5), we know that our datasets are low-rank and it is common to return

$$\widehat{r} = \left\lfloor \frac{\sum_{i=1}^m \widehat{k}_i}{2} \right\rfloor$$

where $\widehat{k}_j$ are estimates of the number of underlying signals in each dataset.

In the spirit of the ICCA algorithm and motivated by the low-rank data model in (10.5), we propose the following informative version of MAXVAR. We know that not all right singular vectors are informative and so we trim our data matrices

$$\begin{aligned}\mathring{U}_j &= \widehat{U}_j\left(:, 1:\widehat{k}_j\right)\\ \mathring{V}_j &= \widehat{V}_j\left(:, 1:\widehat{k}_j\right)\end{aligned}$$

and define

$$\mathring{U} = \mathbf{blkdiag}(\mathring{U}_1, \ldots, \mathring{U}_m), \quad \mathring{V} = \left[\mathring{V}_1, \ldots \mathring{V}_m\right].$$

Using these trimmed estimates, we define the informative MAXVAR (IMCCA) matrix as

$$\widehat{R}_{\text{imcca}} = \mathring{U}\mathring{V}^H\mathring{V}\mathring{U}^H.$$

Define the eigenvalues of this IMCCA matrix as $\widetilde{\kappa}_1, \ldots, \widehat{\kappa}_{\widehat{r}}$.

To estimate the number of correlated signals present in our datasets, we examining the eigenvalues of the matrices $\widehat{R}_{\text{mcca}}$ and $\widehat{R}_{\text{imcca}}$. We know from the population model that the number of eigenvalues above 1 represent correlations. Therefore, we can set a threshold to estimate the number of correlations via

$$\begin{aligned}\widehat{t}_{\text{mcca}} &= \sum_{i=1}^{\widehat{r}} \mathbb{1}_{\{\widehat{\kappa}_1 > 1+\tau^{\alpha}_{\text{mcca}}\}}\\ \widehat{t}_{\text{imcca}} &= \sum_{i=1}^{\widehat{r}} \mathbb{1}_{\left\{\widehat{\kappa}_1 > 1+\tau^{\alpha}_{\text{imcca}}\right\}},\end{aligned} \tag{10.7}$$

where the thresholds $\tau^{\alpha}_{\text{mcca}}$ and $\tau^{\alpha}_{\text{immca}}$ are set to achieve a desired false alarm rate α.

Similar to the ICCA Theorems, we make the following two conjectures.

Conjecture 10.4.2. *Let $d_1, \ldots, d_m, n \to \infty$ with $d_j/n \to c_j$. Given the data model in (10.5), the IMCCA estimate of the number of correlated components in (10.7) converges to the actual number of correlated components under the following condition*

$$\widehat{t}_{imcca} \xrightarrow{a.s.} t \quad if\ \forall j = 1, \ldots, m, \min_{i=1,\ldots,k_j} \theta_j^{(i)} > c_j^{1/4}$$

Conjecture 10.4.3. *Consider the missing data setting where our the entries of our data matrices may only be partially observed as*

$$Y_j = \left(U_j\Theta_j^{1/2}V_j^H + Z_j\right) \odot M_j$$

where

$$M_{k\ell}^{j} = \begin{cases} 1 & \textit{with probability } \gamma_j \\ 0 & \textit{with probability } 1-\gamma_j \end{cases}.$$

Let $d_1, \ldots, d_m, n \to \infty$ with $d_j/n \to c_j$. Then the estimated number of correlated components in (10.7) converges to the actual number of correlated components under the following condition

$$\widehat{t}_{imcca} \xrightarrow{a.s.} t \quad \textit{if } \forall j = 1, \ldots, m, \min_{i=1,\ldots,k_j} \theta_j^{(i)} > \frac{c_j^{1/4}}{\sqrt{\gamma_j}}$$

10.5 Controlled Video-Video-Video Experiment

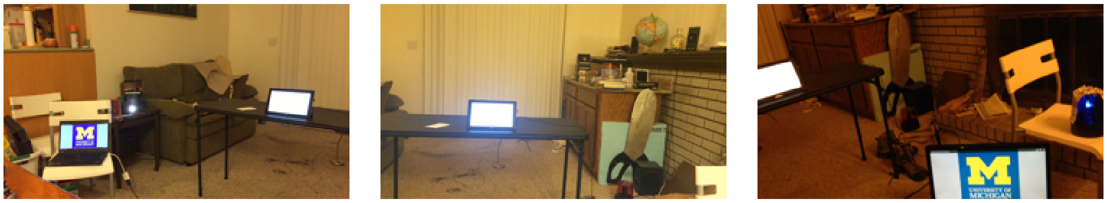
To verify the effectiveness of IMCCA for real world applications, and to showcase the extreme sub-optimality of empirical MCCA, we setup a controlled experiment consisting of four stationary flashing lights and three stationary iPhone cameras. Figure 10.1 shows the left, middle, and right camera views for one frame of the video experiment. Figure 10.2 manually identifies each source in each camera view by drawing a colored box around it. The left camera can see a flashing laptop screen (L1), a flashing phone light (PH1), and a flashing tablet screen (T1). The middle camera has only one source, the flashing tablet screen (T1). The right camera can see the flashing tablet screen (T1), the flashing laptop screen (L1) via an external monitor, and a flashing police light (PL1). We summarize these sources in Table 10.3. Based on our setup, all cameras share the T1 source, while the left and right views share the L1 source. The left and right views also each have an independent source in their view.

To synchronize the cameras we used the RecoLive MultiCam iPhone app [1]. After turning on all light sources, we recorded 20 seconds of video at 30 frames per second. The resolutions of the iPhone's cameras were all 1920 × 1080 pixels.

To post-process the video data, we first converted the video streams to grayscale and then downsampled each spatial dimension by a factor of 8, resulting in a resolution of 240 × 135. We then vectorized each image and stacked the 600 frames into data matrices, all of dimension 32400 × 600. Finally, we subtract the mean from each dataset so that we may run PCA, MCCA, and IMCCA on the zero-mean datasets, Y_{left}, Y_{middle}, and Y_{right}.

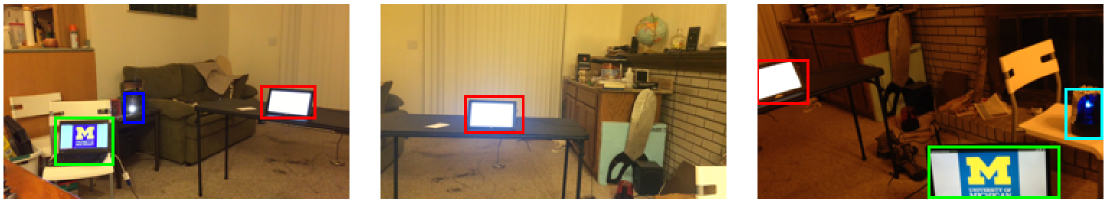
First, we run PCA on the individual datasets Y_{left}, Y_{middle}, and Y_{right} to identify

[1] http://recolive.com/en/



(a) Left Camera (b) Middle Camera (c) Right Camera

Figure 10.1: Left, middle, and right camera views of our four sources for the controlled MCCA flashing light experiment.

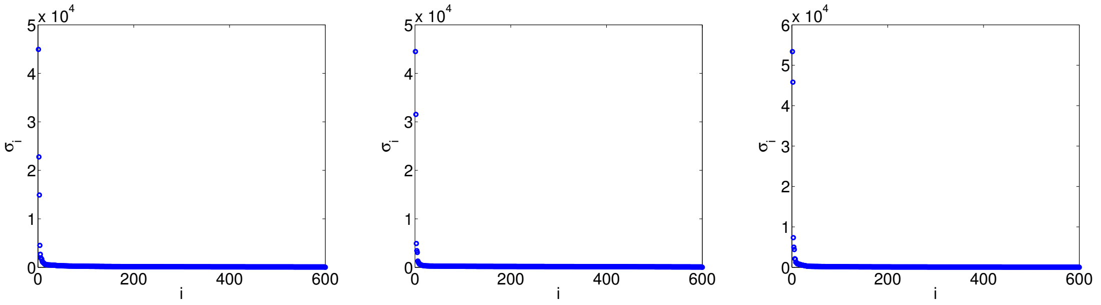


(a) Left Camera (b) Middle Camera (c) Right Camera

Figure 10.2: Manual identification of each source in each camera. All three sources share a common flashing tablet, outlined in red. The left and right camera views share a common flashing laptop screen, outlined in green. The left camera has an independent flashing phone light, outlined in dark blue. The right camera has an independent flashing police light, outlined in cyan.

Camera	Source
Left	Laptop L1
	Phone PH1
	Tablet T1
Middle	Tablet T1
Right	Tablet T1
	Laptop L1
	Police Light PL1

Table 10.3: Visual sources for each camera view. All three cameras share Tablet T1. The left and right cameras share Laptop L1. The left and right cameras each have an independent flashing light source.

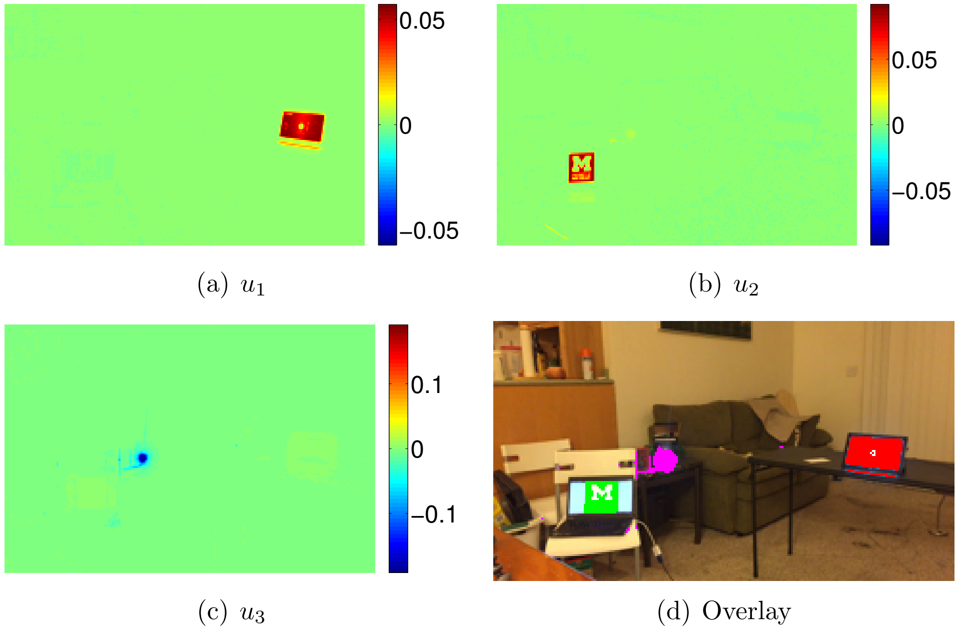


(a) Left Camera (b) Middle Camera (c) Right Camera

Figure 10.3: Singular value spectra of Y_{left}, Y_{middle}, and Y_{right}

the signals residing in each dataset. We know from our setup that the left and right cameras both have three sources. Figure 10.3 plots the singular values of Y_{left}, Y_{middle}, and Y_{right}. Figures 10.4, 10.5 and 10.6 plot the singular vector heatmaps corresponding to the top 3 singular values of Y_{left}, Y_{middle}, and Y_{right}, respectively. Each figure also overlays a thresholded version of the singular vectors onto the raw video. The threshold that we use is $\sqrt{\log(d_i)/d_i}$. From these figures, PCA does a good job at identifying the pixels containing a signal (flashing light).

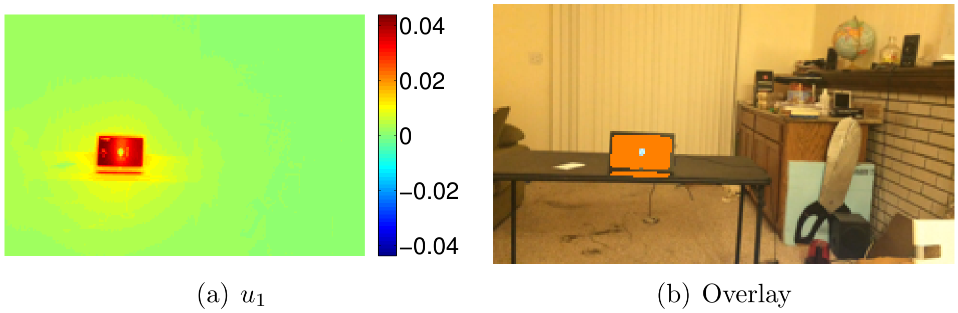
While PCA does a nice job at identifying pixels in each view with a flashing light, it does not provide any information about whether these pixels are correlated across cameras. To accomplish this, we turn to MCCA and IMCCA. In an adaptive setting, we can run these algorithms after every new frame. Specifically, for frame ℓ, we construct the $32400 \times \ell$ submatrices Y_{left}^{ℓ}, Y_{middle}^{ℓ}, and Y_{right}^{ℓ} by taking the matrix of the first ℓ original vectorized frames in each view and then subtracting the mean of the resulting submatrix. We then use these resulting submatrices as inputs to MCCA



(a) u_1 (b) u_2

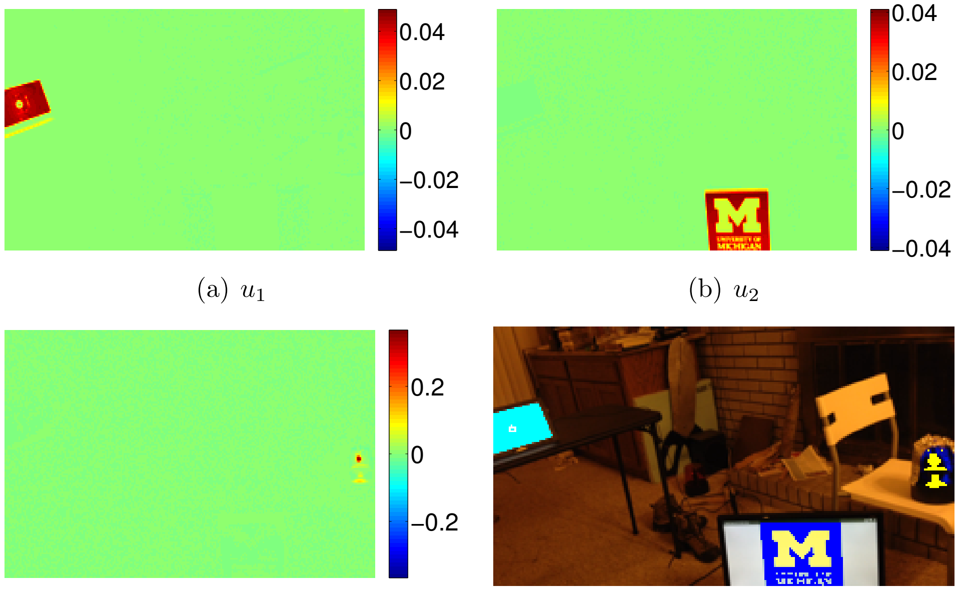
(c) u_3 (d) Overlay

Figure 10.4: (a)-(c) Left singular vectors of Y_{left} corresponding to the top 3 singular values. (d) Thresholded singular vectors from (a)-(c) overlayed onto the original scene. These pixels correspond to the flashing light sources visible in the left camera.



(a) u_1 (b) Overlay

Figure 10.5: (a) Left singular vector of Y_{middle} corresponding to the top singular value. (b) Thresholded singular vector from (a) overlayed onto the original scene. These pixels correspond to the flashing light source visible in the middle camera.



(a) u_1 (b) u_2

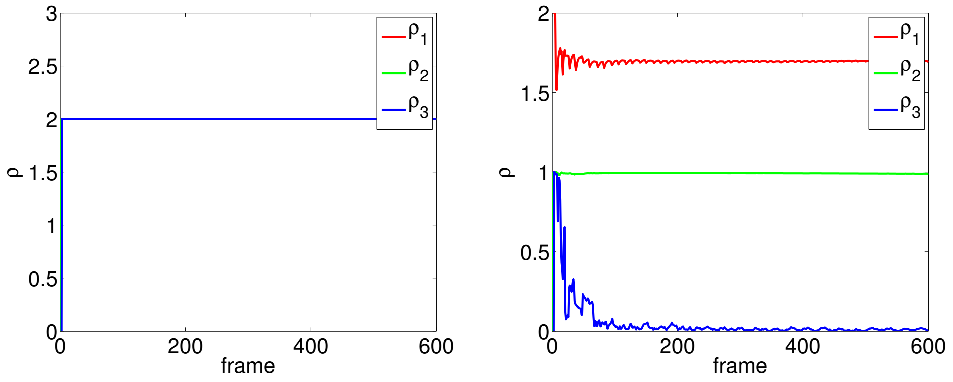
(c) u_3 (d) Overlay

Figure 10.6: (a)-(c) Left singular vectors of Y_{right} corresponding to the top 3 singular values. (d) Thresholded singular vectors from (a)-(c) overlayed onto the original scene. These pixels correspond to the flashing light sources visible in the right camera.

and IMCCA. Using our knowledge of the number of sources in each camera, we set $\widehat{k}_{\text{left}} = 3$, $\widehat{k}_{\text{middle}} = 1$, and $\widehat{k}_{\text{right}} = 3$. Figure 10.7 plots the top 3 correlation coefficients returned by MCCA and IMCCA over the 600 frames of the video. As expected due to our extreme sample deficient regime, MCCA returns coefficients equal to $2 = m - 1$.

Figures 10.8 and 10.9 overlay the thresholded canonical vectors corresponding to the correlations in Figure 10.7 onto the original scene for MCCA and IMCCA, respectively. Unsurprisingly, the MCCA canonical vectors appear extremely random and noisy while the IMCCA canonical vectors correctly identify the two sources of correlation in our video. Additionally, IMCCA identifies that once source of correlation appears in all three camera views (red pixels) and that one source of correlation appears in only two camera views (green pixels).

To overlay the thresholded canonical correlations on the original scene, we use a different threshold than $\sqrt{\log(n)/n}$. The main reason for this is that our eigenvector returned by MCCA is unit norm and contains information for all three canonical vectors. Therefore, the energy may be dispersed across all views. Consider the following examples of (possible) canonical vectors returned by IMCCA for our situation



(a) MCCA (b) IMCCA

Figure 10.7: Top 3 correlations returned by MCCA and IMCCA.

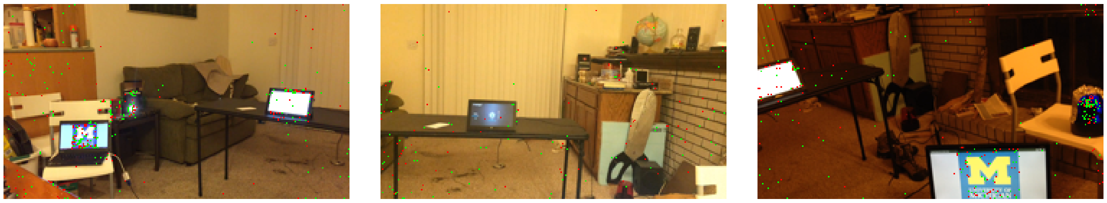
of $k = 7$ signals,

$$u_1 = \left[1/\sqrt{3}, 0, 0, 1/\sqrt{3}, 1/\sqrt{3}, 0, 0\right]^T$$
$$u_2 = \left[0, 1/\sqrt{2}, 0, 0, 0, 1/\sqrt{2}, 0\right]^T$$
$$u_3 = [0, 0, 1, 0, 0, 0, 0,]^T .$$

In these examples, the first three components could correspond to the left camera signals, the fourth component could correspond to the middle camera, and the last three components could correspond to the right camera. Examining u_1, we see that this vector structure tells us that there is a correlation between all three cameras. Examining u_2, this vector structure tells us that there is a correlation between only the left and right cameras. Finally, u_3 shows that there is no correlation between the cameras. In this noise-free setting, it is easy to see that components with zero weight are not correlated. However, in the noisy settings, these non-correlated components will be small but non-zero. Therefore, we propose the following thresholding technique

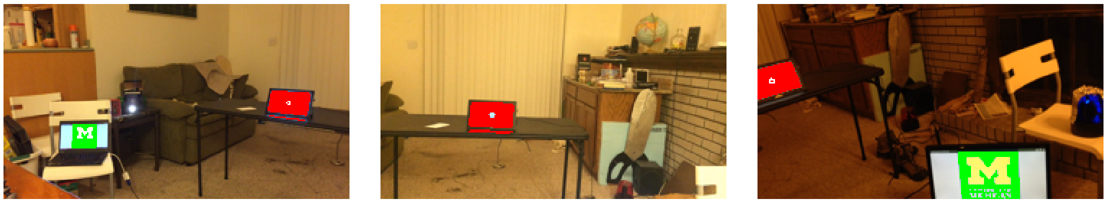
1. $\widetilde{u}_i = \sqrt{m} u_i$
2. Extract the components for each dataset from $\widetilde{u}_i$, resulting in $\widetilde{u}_i^{(j)}$ for $j = 1, \ldots, m$.
3. If $\|\widetilde{u}_i^{(j)}\|_2 > 1$, then $\widetilde{u}_i^{(j)} = \widetilde{u}_i^{(j)} / \|\widetilde{u}_i^{(j)}\|_2$

The resulting $\widetilde{u}_i^j$ now have at most norm 1, but uncorrelated components still remain small. Therefore, using $\widetilde{u}_i^j$ to weight the principal component vectors, we may use our normal threshold of $\sqrt{\log(d_i)/d_i}$. The above steps are a heuristic to determine the number of views that a large correlation represents. The worst case is when



(a) Left Camera (b) Middle Camera (c) Right Camera

Figure 10.8: Top 2 thresholded MCCA canonical vectors overlayed onto the original scene. The red pixels are the pixels corresponding to the largest correlation and the green pixels correspond to the pixels with the second largest correlation. Since we are in the sample deficient regime, MCCA returns random pixels as the canonical vectors are random.



(a) Left Camera (b) Middle Camera (c) Right Camera

Figure 10.9: Top 2 thresholded IMCCA canonical vectors overlayed onto the original scene. The red pixels correspond to the largest correlation and the green pixels correspond to the second largest correlation. Clearly, the red pixels identify the shared flashing tablet light in all 3 views and the green pixels identify the shared flashing laptop in the left and right views.

all m datasets are correlated and the energy in u_i is distributed evenly across the m components. The first step accounts for this by scaling all components by $\sqrt{m}$. In our toy example, the subvectors of u_1 each have unit norm for each dataset. However, in cases where only a subset of the datasets are correlated, as in u_2, this scaling overcompensates and so we use step 2 to make all subvectors at most unit norm. However as we don't normalize all subvectors, those with small norm will stay small, correctly indicating their dataset is not correlated.

CHAPTER XI

Afterword

In the first part of this dissertation, we considered the classical problem of matched subspace detection. Using insights from random matrix theory about the accuracy of subspaces in the low-sample, high dimensional regime, we showcased the suboptimality of the standard plug-in detector and derived a new detector that can avoid some of the performance loss of the plug-in detector. It is amazing that random matrix theory reveals new surprises is classically solved problems. We hope this application will continue to accelerate this trend and that others will similarly reconsider other classical signal processing applications to find such surprises in the low-sample high-dimensionality regime.

In the second part of this dissertation, we explored correlation detection in multimodal datasets. Motivated by the suboptimality of canonical correlation analysis in the sample deficient regime, we considered informative CCA (ICCA). Using insights from random matrix theory, ICCA first trims data SVDs to contain only informative singular vectors. This allows ICCA to robustly detect correlations in the sample deficient regime. We provided a statistical significance test for the ICCA correlation estimates and derived a consistency bound for it. We then considered the accuracy of the canonical vector returned by ICCA and used insights from random matrix to derive improved estimates of the canonical vectors. Finally, we extended these ideas to algorithms that detect correlations in more than two datasets and proposed an informative version, IMCCA, that is able to robustly detect correlations for multiple datasets in the sample deficient regime. We verified these informative correlation algorithms on new low-rank real-world datasets that we created.

The correlation algorithms considered herein are all linear. The work presented in this thesis unifies and completes much of the theory on linear correlation detection in the sample deficient regime. However, if there are nonlinear correlations present between the datasets, ICCA and IMCCA are the wrong algorithms to use. An important area of future research is to extend these insights from random matrix theory to the kernel versions of CCA (KCCA). While KCCA has been used in practice, the theoretical limits of it are generally unknown. An important first step is to develop a universal data model that encodes non-linear correlations. Ideally, similar to the work present in this thesis, we would like to see a fundamental limit dependent on the system dimensionality, number of samples, data SNR, and choice of kernel parameters. Similarly to CCA, one can expect KCCA to behave poorly in this sample deficient regime and so an informative version of KCCA seems within reach.

Finally, we hope that the work on MCCA presented in the final chapter will serve as a springboard for future research in the area. We showcased that reliable detection of correlations between more than two datasets is possible in the sample deficient regime. However, further investigation into the theoretical properties of such algorithms is necessary. In the thesis we touched on the close relationship between the algorithms MAXVAR and MINVAR. Further exploration of this relationship is very important as it may reveal structure in the problem that we can exploit. Similar to the work presented for ICCA, improving the estimates of MCCA canonical vectors seems within reach.

Today's technological landscape offers the ability to collect as much data as possible. It is our job as machine learning and statistical signal processing specialists to theoretically fuse such a wide variety of data. We hope that the work presented in this thesis serves as a starting point for such a discussion.

APPENDICES

APPENDIX A

Proof of Theorem 2.5.1

We restate the theorem for exposition:

> Assume the same hypothesis as in Proposition 2.5.1. Let $\widehat{k} = k_{\text{eff}} = k$. For $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, and $i \neq j$, as $n, m \to \infty$ with $n/m \to c$, then $\langle u_j, \widehat{u}_i \rangle \xrightarrow{\text{a.s.}} 0$.

Proof. Let $U_{n,k}$ be a $n \times k$ real or complex matrix with orthonormal columns, u_i for $1 \leq i \leq k$. Let $\Sigma = \mathbf{diag}\left(\sigma_1^2, \ldots, \sigma_k^2\right)$ such that $\sigma_1^2 > \sigma_2^2 > \cdots > \sigma_k^2 > 0$ for $k \geq 1$. Define $P_n = U_{n,k} \Sigma U_{n,k}^H$ so that P_n is rank-k. Let Z_n be a $n \times m$ real or complex matrix with independent $\mathcal{CN}\left(0, 1\right)$ entries. Let $X_n = \frac{1}{m} Z_n Z_n^H$, which is a random Wishart matrix, have eigenvalues $\lambda_1(X_n) \geq \cdots \geq \lambda_n(X_n)$. Let $\widehat{X}_n = X_n \left(I_n + P_n\right)$. X_n and P_n are independent by assumption. Define the empirical eigenvalue distribution as $\mu_{X_n} = \frac{1}{n} \sum_{j=1}^n \delta_{\lambda_j(X_n)}$. We assume that as $n \to \infty$, $\mu_{X_n} \xrightarrow{\text{a.s.}} \mu_X$.

For $i = 1, \ldots, \widehat{k} = k$, let $\widehat{v}_i$ be an arbitrary unit eigenvector of $\widehat{X}_n$. By the eigenvalue master equation, $\widehat{X}_n \widehat{v}_i = \widehat{\lambda}_i \widehat{v}_i$, it follows that

$$U_{n,k}^H \left(\widehat{\lambda}_i I_n - X_n\right)^{-1} X_n U_{n,k} \Sigma U_{n,k}^H \widehat{v}_i \quad = U_{n,k}^H \widehat{v}_i. \tag{A.1}$$

Let $X_n = V_n \Lambda_n V_n^H$ be the eigenvalue decomposition of X_n such that

$$\Lambda_n = \mathbf{diag}(\lambda_1(X_n), \ldots, \lambda_n(X_n))$$

and $\lambda_1(X_n) \geq \cdots \geq \lambda_n(X_n)$. Using this decomposition and defining $W_{n,k} = V^H U_{n,k}$, (A.1) simplifies to

$$W_{n,k}^H \left(\widehat{\lambda}_i I_n - \Lambda_n\right)^{-1} \Lambda_n W_{n,k} \Sigma U_{n,k}^H \widehat{v}_i \quad = U_{n,k}^H \widehat{v}_i. \tag{A.2}$$

Define the columns of $W_{n,k}$ to be $w_j^{(n)} = [w_{1,j}^{(n)}, \ldots, w_{n,j}^{(n)}]^T$ for $j = 1, \ldots, k$. These columns are orthonormal and isotropically random. We can rewrite (A.2) as

$$\left[T_{\mu_{r,j}^{(n)}}\left(\widehat{\lambda}_i\right)\right]_{r,j=1}^{k} \Sigma U_{n,k}^H \widehat{v}_i = U_{n,k}^H \widehat{v}_i \tag{A.3}$$

where for $r = 1, \ldots, k$, $j = 1, \ldots, k$, $\mu_{r,j}^{(n)} = \sum_{\ell=1}^{n} \overline{w_{\ell,r}^{(n)}} w_{\ell,j}^{(n)} \delta_{\lambda_\ell(X_n)}$ is a complex measure and $T_{\mu_{r,j}^{(n)}}$ is the T-transform defined by $T_\mu(z) = \int \frac{t}{z-t} d\mu(t) \quad \text{for } z \notin \text{supp } \mu$. We may rewrite (A.3) as

$$\left(I_k - \left[\sigma_j^2 T_{\mu_{r,j}^{(n)}}\left(\widehat{\lambda}_i\right)\right]_{r,j=1}^{k}\right) U_{n,k}^H \widehat{v}_i = 0.$$

Therefore, $U_{n,k}^H \widehat{v}_i$ must be in the kernel of $M_n\left(\widehat{\lambda}_i\right) = I_k - \left[\sigma_j^2 T_{\mu_{r,j}^{(n)}}\left(\widehat{\lambda}_i\right)\right]_{r,j=1}^{k}$. By Proposition 9.3 of [85]

$$\mu_{r,j}^{(n)} \xrightarrow{\text{a.s.}} \begin{cases} \mu_X & \text{for } i = j \\ \delta_0 & \text{o.w.} \end{cases}$$

where μ_X is the limiting eigenvalue distribution of X_n. Therefore,

$$M_n\left(\widehat{\lambda}_i\right) \xrightarrow{\text{a.s.}} \mathbf{diag}\left(1 - \sigma_1^2 T_{\mu_X}\left(\widehat{\lambda}_i\right), \ldots, 1 - \sigma_k^2 T_{\mu_X}\left(\widehat{\lambda}_i\right)\right).$$

As $k_{\text{eff}} = k$, for $i = 1, \ldots, k$, $\sigma_i^2 > 1/T_{\mu_X}(b^+)$, where b is the supremum of the support of μ_X. As $\widehat{\lambda}_i$ is the eigenvalue corresponding to the eigenvector $\widehat{v}_i$, by Theorem 2.6 of [85] $\widehat{\lambda}_i \xrightarrow{\text{a.s.}} T_{\mu_X}^{-1}(1/\sigma_i^2)$. Therefore,

$$M_n\left(\widehat{\lambda}_i\right) \xrightarrow{\text{a.s.}} \mathbf{diag}\left(1 - \frac{\sigma_1^2}{\sigma_i^2}, \ldots, 1 - \frac{\sigma_{i-1}^2}{\sigma_i^2}, 0, 1 - \frac{\sigma_{i+1}^2}{\sigma_i^2}, \ldots, 1 - \frac{\sigma_k^2}{\sigma_i^2}\right) \tag{A.4}$$

Recall that $U_{n,k}^H \widehat{v}_i$ must be in the kernel of $M_n\left(\widehat{\lambda}_i\right)$. Therefore, any limit point of $U_{n,k}^H \widehat{v}_i$ is in the kernel of the matrix on the right hand side of (A.4). Therefore, for $i \neq j$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, we must have that $\left(1 - \frac{\sigma_j^2}{\sigma_i^2}\right) \langle u_j, \widehat{v}_i \rangle = 0$. As $\sigma_i^2 \neq \sigma_j^2$, for this condition to be satisfied we must have that for $j \neq i$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, $\langle u_j, \widehat{v}_i \rangle \xrightarrow{\text{a.s.}} 0$.

Recall that our observed vectors $y_i \in \mathbb{C}^{n \times 1}$ have covariance matrix $U_{n,k} \Sigma U_{n,k}^H + I_n = P_n + I_n$. Therefore, our observation matrix, Y_n which is a $n \times m$ matrix, may be written $Y_n = (P_n + I_n)^{1/2} Z_n$. The sample covariance matrix, $S_n = \frac{1}{m} Y_n Y_n^H$, may be written $S_n = (I_n + P_n)^{1/2} X_n (I_n + P_n)^{1/2}$. By similarity transform, if $\widehat{v}_i$ is a unit-norm eigenvector of $\widehat{X}_n$ then $\widehat{s}_i = (I_n + P_n)^{1/2} \widehat{v}_i$ is an eigenvector of S_n. If

$\widehat{u}_i = \widehat{s}_i/\|\widehat{s}_i\|$ is a unit-norm eigenvector of S_n, it follows that

$$\langle u_j, \widehat{u}_i\rangle = \frac{\sqrt{\sigma_i^2+1}\langle u_j, \widehat{v}_i\rangle}{\sqrt{\sigma_i^2|\langle u_j, \widehat{v}_i\rangle|^2+1}}$$

As $\langle u_j, \widehat{v}_i\rangle \xrightarrow{\text{a.s.}} 0$ for all $i \neq j$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, it follows that $\langle u_j, \widehat{u}_i\rangle \xrightarrow{\text{a.s.}} 0$ for all $i \neq j$ $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$. □

Claim 5.1: We conjecture that this result holds for the general case of $i \neq j$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, not just when $\widehat{k} = k_{\text{eff}} = k$. Consider the case when $k = 1$. For $i > 2$, if $\widehat{\lambda}_i$ is an eigenvalue of $\widehat{X}_n = X_n(I_n + \sigma^2 uu^H)$, then it satisfies $\det(\widehat{\lambda}_i I_n - X_n(I_n + \sigma^2 uu^H)) = \det(\widehat{\lambda}_i I_n - X_n)\det(I_n - (\widehat{\lambda}_i I_n - X_n)^{-1} X_n \sigma^2 uu^H) = 0$. Therefore, if $\widehat{\lambda}_i$ is not an eigenvalue of X_n, the corresponding unit norm eigenvector $\widehat{v}_i$ is in the kernel of $I_n - (\widehat{\lambda}_i I_n - X_n)^{-1} X_n \sigma^2 uu^H$. Therefore

$$|\langle \widehat{v}_i, u\rangle|^2 = \frac{1}{\sigma^4 u^H X_n \left(\widehat{\lambda}_i I_n - X_n\right)^{-2} X_n u}.$$

Recall that Weyl's interlacing lemma for eigenvalues gives $\lambda_i(X_n) \leq \widehat{\lambda}_i \leq \lambda_{i-1}(X_n)$. Letting $X_n = V_n \Lambda_n V_n^H$ and $w = V_n^H u$, we see the importance of the asymptotic spacing of eigenvalues of X_n in

$$\begin{aligned} u^H X_n(\widehat{\lambda}_i I_n - X_n)^{-2} X_n u \quad &= \sum_{\ell=1}^{n} \frac{|w_\ell|^2 \lambda_\ell^2(X_n)}{\left(\widehat{\lambda}_i - \lambda_\ell\right)^2} \\ &\geq \frac{\min_j \lambda_j^2(X_n) \min_j |w_j|^2}{\max_j |\lambda_{j-1} - \lambda_j|^2} \end{aligned}$$

In [159] it is shown that $\min_j \lambda_j^2(X_n) = \lambda_n^2(X_n) \xrightarrow{\text{a.s.}} (1-\sqrt{c})^2$. The typical spacing between eigenvalues is $O(1/n)$ while the typical magnitude of w_j^2 is $O(1/n)$ [160]. Therefore, the right hand side of the above inequality will typically be $O(n)$ and we get the desired result of $|\langle \widehat{v}_i, u\rangle|^2 \xrightarrow{\text{a.s.}} 0$. More generally, it is the behavior of the largest eigenvalue gap and the smallest element of w_i that drives this convergence. Thus, so long as the eigenvector whose elements are w_i are delocalized (i.e. having elements of $O(1/\sqrt{n})$) and the smallest gap between k successive eigenvalues is at least as large as $O(1/(n^{(0.5+\epsilon)})$, the right hand side of the inequality will be unbounded with n. The claim follows after applying a similarity transform as in the proof of Theorem 5.1.

APPENDIX B

Theoretical and Empirical MCCA Derivations

We use the following notation. Let Y_i for $i = 1, \ldots, m$ denote the $d_i \times n$ data matrix, with each column an observation from the ith dataset. Let $Y = [Y_1^H \ldots Y_m^H]^H \in \mathbb{C}^{d \times n}$ be the entire observation matrix made by stacking Y_i on top of each other. Let $U_i \Sigma_i V_i^H$ be the individual data SVDs of Y_i. Let $\widehat{R} = \frac{1}{n} Y^H Y$ be the sample covariance matrix. Defining $U = \mathbf{blkdiag}(U_1, \ldots, U_m) \in \mathbb{C}^{d \times dm}$, $\Sigma = \mathbf{blkdiag}(\Sigma_1, \ldots, \Sigma_m) \in \mathbb{C}^{d \times nm}$, $V = [V_1, \ldots, V_m] \in \mathbb{C}^{n \times nm}$, we can write $\widehat{R} = U \Sigma V^H V \Sigma U^H$. Similarly, define $\widehat{R}_D = \frac{1}{n} \mathbf{blkdiag}(Y_i^T Y_i) = U \Sigma \Sigma^H U^H$. Recall that $x = [x_1^H \ldots x_m^H]^H$ is the vector of canonical vectors.

B.1 Problem 1a

B.1.1 Theory

Our optimization problem is

$$\begin{array}{ll} \underset{x_1, \ldots, x_m}{\operatorname{argmax}} & \displaystyle\sum_{i=1}^{m} \sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ \text{s.t.} & x_i^H x_i = 1, \quad i = 1, \ldots, m. \end{array} \tag{B.1}$$

The Lagrangian for this problem is

$$L(x, \underline{\lambda}) \quad = x^H R x - \sum_{i=1}^{m} \lambda_i \left(x_i^H x_i - 1 \right).$$

Define $\Lambda = \mathbf{blkdiag}(\lambda_1 I_{d_1}, \ldots, \lambda_m I_{d_m})$ to be the matrix with the Lagrange multipliers on the diagonal. The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2\Lambda_D x.$$

Setting the derivative equal to the zero vector results in the following non-normal generalized eigensystem.

$$R\widetilde{x} = \Lambda_D \widetilde{x},$$

where $\widetilde{x}$ is a unit norm vector that may be decomposed as $\widetilde{x} = [\widetilde{x}_1^H, \ldots, \widetilde{x}_m^H]^H$ with $\widetilde{x}_i \in \mathbb{C}^{d_i}$ Therefore, the canonical vectors are

$$x = \begin{bmatrix} \|\widetilde{x}_1\|^{-1} I_{d_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \|\widetilde{x}_m\|^{-1} I_{d_m} \end{bmatrix} \widetilde{x}.$$

To obtain the canonical correlation, we substitute the canonical vectors into the objective function.

B.1.2 Empirical

As shown in the previous section, the solution to (B.1) is a non-normal eigenvalue system. To solve this problem, we use the manopt software package to solve cost functions on manifolds. The manifold for this problem is the product of m sphere manifolds constraining the canonical vectors x_i to lie on the $\mathbb{C}^{d_i}$ unit sphere. We use the SUMCORR cost function and its gradient

$$\frac{\partial}{\partial x} = 2Rx$$

in the manopt solution.

B.2 Problem 1b

B.2.1 Theory

Our optimization problem is

$$\begin{aligned} &\underset{x_1,\ldots,x_m}{\text{argmax}} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ &\text{s.t.} \qquad\quad x^H x = 1. \end{aligned}$$

The Lagrangian for this problem is

$$L(x,\lambda) \quad = x^H R x + \lambda(1 - x^H x).$$

The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2x.$$

Setting the derivative equal to the zero vector results in the following eigensystem.

$$Rx = \lambda x \tag{B.2}$$

From this relationship, if we substitute this solution into the objective function, we obtain

$$\rho = x^H R x = x^H(\lambda x) = \lambda. \tag{B.3}$$

B.2.2 Empirical

We plug in $\widehat{R}$ into (B.2) for R and solve the eigenvalue decomposition. The eigenvector x is the canonical vector and the eigenvalue λ is the canonical correlation. This problem is typically ill-posed as the maximum solution is typically found by setting only one x_i to be nonzero corresponding to the R_{ii} with the largest variance. We advise to not use this formulation of MCCA.

B.3 Problem 1c

B.3.1 Theory

Our optimization problem is

$$\begin{aligned} &\underset{x_1,\ldots,x_m}{\text{argmax}} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ &\text{s.t.} \qquad\quad x_i^H R_{ii} x_i = 1. \end{aligned}$$

The Lagrangian for this problem is

$$L(x, \underline{\lambda}) \quad = x^H R x - \sum_{i=1}^{m} \lambda_i \left(x_i^H R_{ii} x_i \right).$$

Define $\Lambda_D \in \mathbb{C}^{d \times d} = \mathbf{blkdiag}(\lambda_1 I_{d_1}, \ldots, \lambda_m I_{d_m})$. The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2\Lambda_D R_D x.$$

Setting the derivative equal to the zero vector results in the non-normal generalized eigensystem.

$$R\widetilde{x} \quad = \Lambda_D R_D \widetilde{x}.$$

To obtain the canonical vectors, we make the transformation

$$x_i = \frac{R_{ii}^{-1/2}\widetilde{x}_i}{\|\widetilde{x}_i\|}.$$

B.3.2 Empirical

Making the transformation

$$x_i = R_{ii}^{-1/2}\widetilde{x}_i,$$

our optimization problem becomes

$$\begin{aligned} &\underset{\widetilde{x}}{\text{argmax}} \quad \widetilde{x}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{x} \\ &\text{s.t.} \qquad\quad \widetilde{x}_i^H \widetilde{x}_i = 1. \end{aligned}$$

As shown in the previous section, the solution to this problem is a non-normal eigenvalue system. To solve the above problem, we use the manopt software package to solve cost functions on manifolds. The manifold for this problem is the product of

m sphere manifolds constraining each canonical vector to lie on the $\mathbb{C}^{d_i}$ unit sphere. We use the SUMCORR cost function and the derivative

$$\frac{\partial}{\partial \widetilde{x}} = 2R_D^{-1/2} R R_D^{-1/2} \widetilde{x}.$$

We substitute the empirical sample covariances

$$\widehat{R} = \frac{1}{n} Y Y^T, \;\; \widehat{R}_D = \frac{1}{n}\,\mathbf{blkdiag}(Y_i Y_i^T)$$

for the unknown R and R_D in the cost and gradient functions. To obtain the canonical vectors x_i, we make the transformation

$$x_i = R_{ii}^{-1/2} \widetilde{x}_i.$$

Using our notation for R and R_D from the data SVDs, we have

$$R_D^{-1/2} R R_D^{-1/2} = U V^H V U^H$$

and

$$x = U \Sigma^{-1} U^H \widetilde{x}.$$

The canonical correlation is

$$\widehat{\rho} = x^H R_D^{-1/2} R R_D^{-1/2} x = \widetilde{x}^H U V^H V U^H \widetilde{x}.$$

B.4 Problem 1d

B.4.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x_1,\ldots,x_m}{\text{argmax}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ \text{s.t.} \quad & x^H R_D x = 1. \end{aligned}$$

The Lagrangian for this problem is

$$L(x, \lambda) = x^H R x + \lambda(1 - x^H R_D x).$$

The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2\lambda R_D x.$$

Setting the derivative equal to the zero vector results in the following generalized eigensystem.

$$R_D^{-1} R x = \lambda x.$$

Let $\widetilde{x} = R_D^{1/2} x$ so that the eigensystem becomes

$$R_D^{-1/2} R R_D^{-1/2} \widetilde{x} = \lambda \widetilde{x}$$

where $\|\widetilde{x}\|_2 = 1$. The canonical vectors are $x = R_D^{-1/2}\widetilde{x}$ and the canonical correlation is $\rho = \lambda$.

B.4.2 Empirical

Our empirical eigen-problem is $\widehat{R}_D^{-1/2} \widehat{R} \widehat{R}_D^{-1/2} \widetilde{x} = \lambda \widetilde{x}$. Using data SVDs,

$$\widehat{R}_D^{-1/2} \widehat{R} \widehat{R}_D^{-1/2} = U V^H V U^H.$$

Let $Q \Lambda Q^H$ be the eigenvalue decomposition of $U V^H V U^H$. To obtain canonical vectors consistent with the the constraint function, we make the transformation

$$x = U \Sigma^{-1} U^H Q.$$

Substituting this expression into the objective function, we obtain

$$\widehat{\rho} = \lambda.$$

B.5 Problem 2a

B.5.1 Theory

Our optimization problem is

$$\begin{array}{ll} \underset{x}{\operatorname{argmax}} & \displaystyle\sum_{i=1}^{m} \sum_{j=1}^{m} (x_i^H R_{ij} x_j)^2 = \|X^H R X\|_F^2 \\ \text{s.t.} & x_i^H x_i = 1, i = 1, \ldots, m. \end{array}$$

To calculate the gradient of the cost function, we use the double summation version of the cost function. We have that

$$\begin{aligned} \frac{\partial}{\partial x_i} &= 4\sum_{j=1}^{m}(x_i^H R_{ij} x_j) R_{ij} x_j \\ &= R_{i,:} X (X^H R X)_{:,i} \end{aligned} \tag{B.4}$$

where

$$R_{i,:} = [R_{i,1}, \ldots, R_{i,m}], \quad (X^H R X)_{:,i} = [x_1^H R_{1,i} x_i, \ldots, x_m^H R_{m,i} x_i]^H.$$

Thus

$$\frac{\partial}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_m} \end{bmatrix}.$$

If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \Lambda_D x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x and $\Lambda_D = \mathbf{diag}(\lambda_1 I_{d1}, \ldots, \lambda_m I_{dm})$, this is a highly non regular eigenvalue problem.

B.5.2 Empirical

To solve the problem above, we use the manopt software package to solve cost functions on manifolds. Each of our canonical vectors are constrained on the d_i unit sphere. We use the SSQCORR cost function and the derivative in (B.4). We substitute the empirical sample covariance

$$\widehat{R} = \frac{1}{n} Y Y^T$$

for the unknown R in the cost and gradient functions.

B.6 Problem 2b

B.6.1 Theory

Our optimization problem is

$$\begin{aligned} &\underset{x}{\text{argmax}} \quad \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^H R_{ij} x_j)^2 \\ &\text{s.t.} \qquad\quad x^H x = 1 \end{aligned}$$

The derivative of our cost function is the same as in (B.4). If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \lambda x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x, this is a highly non regular eigenvalue problem.

B.6.2 Empirical

To solve the problem above, we use the manopt software package to solve cost functions on manifolds. Our manifold is simpler as we only have one constraint that $x^H x = 1$. We use the SSQCORR cost function and the derivative in (B.4). We substitute the empirical sample covariance

$$\widehat{R} = \frac{1}{n} Y Y^T$$

for the unknown R in the cost and gradient functions. However, the solution to this problem will typically set the only one x_i to be non-zero corresponding to the R_{ii} with the largest eigenvalue. We advise not to use this formulation of MCCA.

B.7 Problem 2c

B.7.1 Theory

Our optimization problem is

$$\begin{aligned} &\underset{x}{\text{argmax}} \quad \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^H R_{ij} x_j)^2 \\ &\text{s.t.} \qquad\quad x_i^H R_{ii} x_i = 1. \end{aligned}$$

The derivative of our cost function is the same as in (B.4). If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \Lambda_D R_D x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x and and $\Lambda_D = \mathbf{diag}(\lambda_1 I_{d1}, \ldots, \lambda_m I_{dm})$, this is a highly non regular eigenvalue problem.

B.7.2 Empirical

We first make the transformation

$$x_i = R_{ii}^{-1/2} \widetilde{x}_i.$$

Our optimization problem becomes

$$\begin{array}{ll} \underset{\widetilde{x}}{\operatorname{argmax}} & \|\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{X}\|_F^2 \\ \text{s.t.} & \widetilde{x}_i^H \widetilde{x}_i = 1, i = 1, \ldots, m \end{array}$$

This is the same type of optimization problem as Problem 2a if we replace R with $R_D^{-1/2} R R_D^{-1/2}$.

To solve this problem above, we use the manopt software package to solve cost functions on manifolds. Our manifold consists of m constraints, $\widetilde{x}_i^H \widetilde{x}_i = 1$, that is m vectors constrained on the d_i unit sphere. We use the SSQCORR cost function and the derivative in (B.4). We substitute the empirical sample covariances

$$\widehat{R} = \frac{1}{n} Y Y^T, \ \ \widehat{R}_D = \frac{1}{n} \mathbf{blkdiag}(Y_i Y_i^T)$$

for the unknown R and R_D in the cost and gradient functions.

To obtain the canonical vectors x_i, we make the transformation

$$x_i = R_{ii}^{-1/2} \widetilde{x}_i.$$

Using our notation for R and R_D from the data SVDs, we have

$$R_D^{-1/2} R R_D^{-1/2} = U V^H V U^H$$

and

$$x = U \Sigma^{-1} U^H \widetilde{x}.$$

B.8 Problem 2d

B.8.1 Theory

Our optimization problem is

$$\begin{array}{ll} \underset{x}{\text{argmax}} & \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^H R_{ij} x_j)^2 \\ \text{s.t.} & x^H R_D x = m. \end{array}$$

The derivative of our cost function is the same as in (B.4). If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \lambda R_D x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x and, this is a highly non regular eigenvalue problem.

B.8.2 Empirical

We first make the transformation

$$x_i = R_{ii}^{1/2}\widetilde{x}_i.$$

Our optimization problem becomes

$$\begin{array}{ll} \underset{\widetilde{x}}{\text{argmax}} & \|\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2}\widetilde{X}\|_F^2 \\ \text{s.t.} & \widetilde{x}^H\widetilde{x} = 1 \end{array}$$

This is the same type of optimization problem as Problem 2b if we replace R with $R_D^{-1/2} R R_D^{-1/2}$.

To solve this problem above, we use the manopt software package to solve cost functions on manifolds. Our manifold consists of only one constraint, $\widetilde{x}^H\widetilde{x} = 1$, which is a vector constrained on the $\mathbb{R}^d$ unit sphere. We use the SSQCORR cost function and the derivative in (B.4). We substitute the empirical sample covariances

$$\widehat{R} = \frac{1}{n}YY^T, \ \ \widehat{R}_D = \frac{1}{n}\,\mathbf{blkdiag}(Y_iY_i^T)$$

for the unknown R and R_D in the cost and gradient functions.

To obtain the canonical vectors x_i, we make the transformation

$$x = R_d^{-1/2}\widetilde{x}$$

Using our notation for R and R_D from the data SVDs, we have

$$\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} = UV^H V U^H$$

and

$$x = U\Sigma^{-1}U^H \widetilde{x}.$$

B.9 Problem 3a

B.9.1 Theory

Our optimization problem is

$$\begin{array}{ll} \underset{x}{\text{argmax}} & \lambda_1 \\ \text{s.t.} & x_i^H x_i = 1, i = 1, \ldots, m \\ & \Phi(x) a = \lambda_1 a \\ & a^H a = 1. \end{array}$$

We may write $\Phi(x) = X^H R X$. Using this fact and third constraint of this optimization, the second constraint may be written as $a^H X^H R X a = \lambda_1$. Define $\widetilde{a} = Xa$. As a consequence of the first constraint function,

$$\|\widetilde{a}\|^2 = a^H X^H X a = a^H a = 1.$$

Our modified optimization problem is

$$\begin{array}{ll} \underset{\tilde{a}}{\text{argmax}} & \lambda_1 \\ \text{s.t} & \tilde{a}^H R \tilde{a} = \lambda_1 \\ & \widetilde{a}^H \widetilde{a} = 1. \end{array}$$

Therefore, $\widetilde{a}$ is the unit norm eigenvector corresponding to the largest eigenvalue of R. To solve for the canonical coefficients, we have $\widetilde{a} = Xa$ which implies $x_i = \frac{\widetilde{a}_i}{a_i}$. As a_i is a scalar, and x_i is required to have unit norm, we have that $x_i = \frac{\widetilde{a}_i}{\|\widetilde{a}_i\|}$. This implies $x = \Lambda_{\widetilde{a}}^{-1} \widetilde{a}$ where $\Lambda_{\widetilde{a}} \in \mathbb{C}^{d \times d} = \textbf{blkdiag}(\|\widetilde{a}_i\| I_{d_i})$. The canonical correlation is simply $\rho = \lambda_1$.

B.9.2 Empirical

Our empirical eigen-system is $\widehat{R}\widetilde{a} = \lambda_1 \widetilde{a}$ where $\widehat{R} = \frac{1}{n} YY^H$ is the sample covariance matrix. Let $Q\Lambda Q^H$ be the eigenvalue decomposition of $\widehat{R}$. Let q be the leftmost column of Q and decomposed as $q^H = [q_1^H, \ldots, q_m^H]$ with $q_i \in \mathbb{C}^{d_i}$. Then

$$\begin{aligned} \widehat{\rho} &= \lambda_1 \\ \widehat{x} &= \Lambda_{\widetilde{q}}^{-1} q \end{aligned}$$

where $\Lambda_{\widetilde{q}} \in \mathbb{C}^{d \times d} = \textbf{blkdiag}(\|\widetilde{q}_i\| I_{d_i})$.

B.10 Problem 3b

B.10.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \lambda \\ \text{s.t.} \qquad & x^H x = 1 \\ & \Phi(x) a = \lambda a \\ & a^H a = 1. \end{aligned}$$

We may write $\Phi(x) = X^H R X$. Using this fact and third constraint of this optimization, the second constraint may be written as $a^H X^H R X a = \lambda$.

Let $R = U\Sigma V^H V \Sigma^H U^H$ be a decomposition of R using the block SVDs of the individual covariance matrices R_{ii}. Let $\widetilde{a} = Xa$. We wish to maximize $\lambda = \widetilde{a}^H R \widetilde{a}$, with $\|\widetilde{a}\| = 1$. This is equivalent to

$$\begin{aligned} \underset{\widetilde{a}}{\text{argmax}} \quad & \|R^{1/2} \widetilde{a}\|_2 \\ \text{s.t.} \qquad & \|\widetilde{a}\| = 1. \end{aligned}$$

Now

$$\|R^{1/2} \widetilde{a}\|_2 \quad = \|P \Sigma U^H \widetilde{a}\|_2$$

where $P \in \mathbb{C}^{d \times d}$ is composed of sub-matrices $P_{ij} \in \mathbb{C}^{d_i \times d_j} = \text{corr}(y_i, y_j)$. Note that $P_{ii} = I_{d_i}$. The entries of P are all between -1 and 1. Now since U is an orthonormal matrix and the largest entries in P have norm 1, to maximize this norm, $\widetilde{a}$ should be the column of U corresponding to the largest value in Σ. Since U is block diagonal,

$\widetilde{a} = [0^H \dots 0^H u_{i1}^H 0^H]^H$ where u_{i1} is the leftmost left singular vector of R_{ii} where i is the dataset with the largest singular value. Therefore, $\rho = \widetilde{a}^H U \Sigma P P^T \Sigma^H U^H \widetilde{a} = \sigma_{i1}^2 P_{ii} = \sigma_{i1}^2$ as $P_{ii} = 1$. Therefore, the canonical vectors are

$$x_i = \begin{cases} u_{i1} & \text{dataset } i \text{ has largest singular value} \\ 0 & \text{otherwise} \end{cases}$$

This is obviously undesirable as all but one canonical vector is 0. We advise to not use this formulation of MCCA.

B.10.2 Empirical

In the empirical setting, we substitute $\widehat{R}$ as the sample covariance estimate. Recall that $\widehat{R} = U \Sigma V^H V \Sigma^H U^H$. Letting $\widetilde{a} = Xa$, our optimization problem is

$$\begin{aligned} \underset{\widetilde{a}}{\text{argmax}} \quad & \|R^{1/2}\widetilde{a}\|_2 \\ \text{s.t.} \quad & \|\widetilde{a}\| = 1 \end{aligned}$$

We can rewrite this as

$$\|R^{1/2}\widetilde{a}\|_2 \quad = \|V \Sigma U^H \widetilde{a}\|_2.$$

Now since U is an orthogonal matrix and the columns of V are unit norm, to maximize this norm, $\widetilde{a}$ should be the column of U corresponding to the largest value in Σ. Since U is block diagonal, $\widetilde{a} = [0^H \dots 0^H u_{i1}^H 0^H]^H$ where u_{i1} is the leftmost left singular vector of R_{ii} where i is the dataset whose sample covariance matrix has the largest singular value. The value of $\widehat{\rho}$ is the value of the largest singular value squared. This formulation of MCCA results in canonical vectors that are 0 for all but one dataset. This obviously is very undesirable and we advise to not use this formulation for MCCA.

B.11 Problem 3c

B.11.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \lambda \\ \text{s.t.} \quad & x_i^H R_{ii} x_i = 1 \ , 1 \leq i \leq m \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

We may write $\Phi(x) = X^H R X$. Using this fact and third constraint of this optimization, the second constraint may be written as $a^H X^H R X a = \lambda$. If we assume that R_D is positive definite (which requires it to be full rank), we can rewrite this as $a^H X^H R_D^{1/2} R_D^{-1/2} R R_D^{-1/2} R_D^{1/2} X a = \lambda$. Let $\widetilde{a} = R_D^{1/2} X a$. Now by the first and third constraints

$$\|\tilde{a}\|^2 = a^H X^H R_D X a = a^H I_m a = a^H a = 1.$$

Our modified optimization problem is

$$\begin{aligned} \underset{\widetilde{a}}{\text{argmax}} \quad & \lambda \\ \text{s.t} \quad & \widetilde{a}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{a} = \lambda \\ & \widetilde{a}^H \widetilde{a} = 1. \end{aligned}$$

Therefore, $\widetilde{a}$ is the eigenvector corresponding to the largest eigenvalue of $R_D^{-1/2} R R_D^{-1/2}$. To solve for our original canonical coefficients, recall that $\widetilde{a} = R_D^{1/2} X a$. As R_D and X are block diagonal, we have $\widetilde{a}_i = R_{ii}^{1/2} x_i a_i$, implying $x_i = \frac{1}{a_i} R_{ii}^{-1/2} \widetilde{a}_i$. By the first constraint,

$$x_i^H R_{ii} x_i = \frac{\widetilde{a}_i^H \widetilde{a}_i}{a_i^2} = 1.$$

Letting $a_i = \|\widetilde{a}_i\|$ satisfies this constraint. Therfore, the canonical vector is

$$x_i = \frac{R_{ii}^{-1/2} \widetilde{a}_i}{\|\widetilde{a}_i\|}.$$

Thus

$$x = \Lambda_{\widetilde{a}}^{-1} R_D^{-1/2} \widetilde{a}$$

where $\Lambda_{\widetilde{a}} \in \mathbb{C}^{d \times d} = \textbf{blkdiag}(\|\widetilde{a}_i\| I_{d_i})$.

B.11.2 Empirical

Our empirical eigen-system is $\widehat{R}_D^{-1/2}\widehat{R}\widehat{R}_D^{-1/2}\widetilde{a} = \widehat{\rho}\widetilde{a}$. Using the SVD notation for our empirical data matrices, we have that

$$\begin{aligned}
\widehat{R}_D^{-1/2}\widehat{R}\widehat{R}_D^{-1/2} &= \left(U\Sigma\Sigma^H U^H\right)^{-1/2}\left(U\Sigma V^H V\Sigma^H U^H\right)\left(U\Sigma\Sigma^H U^H\right)^{-1/2} \\
&= U(\Sigma\Sigma^H)^{-1/2}U^H U\Sigma V^H V\Sigma^H U^H U(\Sigma\Sigma^H)^{-1/2}U^H \\
&= U(\Sigma\Sigma^H)^{-1/2}\Sigma V^H V\Sigma^H(\Sigma\Sigma^H)^{-1/2}U^H \\
&= U\widetilde{V}^H\widetilde{V}U^H
\end{aligned}$$

where $\tilde{V} \in \mathbb{C}^{n\times d} = [V_1(:, 1:d_1), \ldots, V_m(:, 1:d_m)]$. Defining $\widehat{C} = \widetilde{V}^H\widetilde{V}$ and its eigenvalue decomposition $\widehat{C} = \widehat{F}\widehat{K}\widehat{F}^H$, then we have that the MCCA empirical solution is

$$\begin{aligned}
\widehat{\rho} &= \widehat{k}_1 \\
\widehat{x} &= U\widetilde{\Sigma}^{-1}\Lambda^{-1}_{\widehat{f}_1}\widehat{f}_1
\end{aligned}$$

where $\widetilde{\Sigma} = \textbf{blkdiag}\left(\Sigma_1(1:d_1, 1:d_1), \ldots, \Sigma_m(1:d_m, 1:d_m)\right)$.

B.12 Problem 3d

B.12.1 Theory

We proceed very similarly as above. Our optimization problem is

$$\begin{aligned}
\underset{x}{\text{argmax}} \quad & \lambda \\
\text{s.t.} \quad & xR_Dx = 1 \\
& \Phi(x)a = \lambda a \\
& a^H a = 1.
\end{aligned}$$

Substituting $\widetilde{x} = R_D^{1/2}x$ into the above problem yields

$$\begin{aligned}
\underset{x}{\text{argmax}} \quad & \lambda \\
\text{s.t.} \quad & \widetilde{x}^H\widetilde{x} = 1 \\
& \widetilde{X}^H R_D^{-1/2} R R_d^{-1/2}\widetilde{X}a = \lambda a \\
& a^H a = 1.
\end{aligned}$$

This is now the same problem as 3b except we replace R with $R_D^{-1/2} R R_D^{-1/2}$. Using the SVD notation as in 3b, we have that $R_D^{-1/2} R R_D^{-1/2} = UP^T PU^H$. Recall that the diagonals of P are 1 and that every entry of P has a norm of no greater than 1. We can clearly see that this problem does not have a unique solution. We can set any $x_i = u_i/\sigma_i$ where u_i is any left singular vector or R_{ii} corresponding to the singular value σ_i. We then set all other $x_i = 0$. Choosing canonical vectors in this fashion results in $\rho = 1$. This solution is non-unique and clearly undesirable. Therefore, the canonical vectors are

$$x_i = \begin{cases} u_i/\sigma_i & \text{for one dataset} \\ 0 & \text{for all others} \end{cases}.$$

We advise to not use this formulation of MCCA.

B.12.2 Empirical

The solutions to this problem are not unique. Take the data SVD of one dataset Y_i and set $x_i = u_i/\sigma_i$ and all others equal to 0.

B.13 Problem 4a

The optimization problem is

$$\begin{array}{ll} \underset{x}{\operatorname{argmin}} & \lambda \\ \text{s.t.} & x_i^H x_i = 1, i = 1, \dots, m \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{array}$$

Here we proceed exactly as in problem 3a except that we choose the eigenvector corresponding to the smallest, (potentially zero) eigenvalue.

B.14 Problem 4b

B.14.1 Theory

The optimization problem is

$$\begin{aligned} \underset{x}{\operatorname{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x^H x = 1 \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Choosing the canonical vectors the same way as in 3b makes $\Phi(x)$ singular. Therefore we can achieve an eigenvalue of 0. This is optimal as $\Phi(x)$ is positive semi-definite. This solution is not unique and undesirable.

B.14.2 Empirical

The solutions to this problem are not unique. Take the data SVD of one dataset Y_i and set $x_i = u_i/\sigma_i$ and all others equal to 0 for any dataset and any singular vector/value pair.

B.15 Problem 4c

The optimization problem is

$$\begin{aligned} \underset{x}{\operatorname{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x_i^H R_{ii} x_i = 1, i = 1, \ldots, m \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Here we proceed exactly as in problem 3c except that we choose the eigenvector corresponding to the smallest, nonzero eigenvalue.

B.16 Problem 4d

B.16.1 Theory

The optimization problem is

$$\begin{aligned} \underset{x}{\operatorname{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x^H R_D x = 1 \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Choosing the canonical vectors the same way as in 3d makes $\Phi(x)$ singular. Therefore we can achieve an eigenvalue of 0. This is optimal as $\Phi(x)$ is positive semi-definite. This solution is not unique and undesirable.

B.16.2 Empirical

The solutions to this problem are not unique. Take the data SVD of one dataset Y_i and set $x_i = u_i/\sigma_i$ and all others equal to 0 for any dataset and any singular vector/value pair.

B.17 Problems 5a-d Theory

The GENVAR problem does not offer a closed form solution. To solve these problems we use the manopt software package. The cost function is

$$|X^H R X| \tag{B.5}$$

where $X = \mathbf{blkdiag}(x_1, \ldots, x_m)$. The gradient with respect to the matrix X is

$$\frac{\partial}{\partial X} = 2|X^H R X| R X (X^H R X)^{-1}.$$

Let $\mathbf{1}_{d_i} \in \mathbb{C}^{d_i}$ be the vector of all ones. Let $A = \mathbf{blkdiag}(\mathbf{1}_{d_1}, \ldots, \mathbf{1}_{d_m})$. Then the gradient with respect to the vector x can be extracted via

$$\frac{\partial}{\partial x} = 2|X^H R X| R X (X^H R X)^{-1} \odot A \tag{B.6}$$

where $\odot$ represents element-wise multiplication. Choosing the appropriate cost function manifolds completes the solution using manopt as we see below in the empirical versions.

B.18 Problem 5a Empirical

The canonical vectors are each constrained on the $\mathbb{C}^{d_i}$ unit sphere. The manifold for the problem is the product of m of these sphere manifolds. We use the sample covariance matrix $\widehat{R}$ for the unknown R in (B.5) and (B.6).

B.19 Problem 5b Empirical

The canonical vectors are each constrained on the $\mathbb{C}^{d}$ unit sphere. The manifold for the problem is therefore one sphere manifolds. We use the sample covariance matrix $\widehat{R}$ for the unknown R in (B.5) and (B.6).

B.20 Problem 5c Empirical

The constraints for this problem are $x_i^H R_{ii} x_i = 1$ for $i = 1, \ldots, m$. Here we make the transformation

$$\widetilde{x} = R_{ii}^{1/2} x$$

which results in the constraints $\widetilde{x}_i^H \widetilde{x}_i = 1$ for $i = 1, \ldots, m$. The cost function becomes

$$|X^H R X| = |\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{X}|.$$

We see that this is the same type of problem as 5a with $\widetilde{x}$ replacing x and $R_D^{-1/2} R R_D^{-1/2}$ replacing R. We make this substitution and use the sample covariance matrices $\widehat{R}$ and $\widehat{R}_D$ in (B.5) and (B.6). The manifold for this problem is the product of m $\mathbb{C}^{d_i}$ sphere manifolds.

B.21 Problem 5d Empirical

The single constraint for this problem is $x^H R_D x = 1$. Here we make the transformation

$$\widetilde{x} = R_{ii}^{1/2} x$$

which results in the constraint $\widetilde{x}^H \widetilde{x} = 1$ for $i = 1, \ldots, m$. The cost function becomes

$$|X^H R X| = |\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{X}|.$$

We see that this is the same type of problem as 5a with $\widetilde{x}$ replacing x and $R_D^{-1/2} R R_D^{-1/2}$ replacing R. We make this substitution and use the sample covariance matrices $\widehat{R}$

and $\widehat{R}_D$ in (B.5) and (B.6). The manifold for this problem is one $\mathbb{C}^d$ sphere manifold.

APPENDIX C

Derivations of Empirical CCA Canonical Correlations and Accuracy of Canonical Vectors

In this appendix we first derive the almost sure limit of the top empirical CCA correlation estimates using the low-rank signal-plus-noise data model presented in Chapter 4. This is a recent result by [2] but we present a similar derivation using our data model and our random matrix theory notation. We then consider the accuracy of the corresponding canonical vectors returned by empirical CCA. We derive a non-closed form expression for this vector accuracy and discuss the approximations we make to compute the needed terms. Finding a closed form expression for the canonical vector accuracy remains future work.

C.1 Model

We repeat the data model from Chapter 4 for simplicity. Let $x_i \in \mathbb{C}^{p\times 1}$ and $y_i \in \mathbb{C}^{q\times 1}$ be modeled as

$$\begin{aligned} x_i &= U_x s_{x,i} + z_{x,i} \\ y_i &= U_y s_{y,i} + z_{y,i}, \end{aligned} \tag{C.1}$$

where $U_x^H U_x = I_{k_x}$, $U_y^H U_y = I_{k_y}$, $z_{x,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_p)$ and $z_{y,i} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_q)$. Furthermore, assume that

$$\begin{aligned} s_{x,i} &\sim \mathcal{CN}(0, \Theta_x) \\ s_{y,i} &\sim \mathcal{CN}(0, \Theta_y), \end{aligned}$$

where $\Theta_x = \mathbf{diag}\left(\left(\theta_1^{(x)}\right)^2, \ldots, \left(\theta_{k_x}^{(x)}\right)^2\right)$ and $\Theta_y = \mathbf{diag}\left(\left(\theta_1^{(y)}\right)^2, \ldots, \left(\theta_{k_y}^{(y)}\right)^2\right)$. Assume that $z_{x,i}$ and $z_{y,i}$ are mutually independent and independent from both $s_{x,i}$

and $s_{y,i}$. Finally, assume that

$$\mathbb{E}\left[s_{x,i}s_{y,i}^H\right] =: K_{xy} = \Theta_x^{1/2} P_{xy} \Theta_y^{1/2}$$

where the entries of P_{xy} are $-1 \leq \rho_{kj} \leq 1$ and represent the correlation between $s_{x,i}^{(k)}$ and $s_{y,i}^{(j)}$. For reasons to be made clear later, define

$$\widetilde{K}_{xy} = \left(\Theta_x + I_{k_x}\right)^{-1/2} K_{xy} \left(\Theta_y + I_{k_y}\right)^{-1/2}$$

and define the singular values of $\widetilde{K}_{xy}$ as $\kappa_1, \ldots, \kappa_{\min(k_x,k_y)}$. Under this model, we define the following covariance matrices

$$\begin{aligned} \mathbb{E}\left[x_i x_i^H\right] &= U_x \Theta_x U_x^H + I_p =: R_{xx} \\ \mathbb{E}\left[y_i y_i^H\right] &= U_y \Theta_y U_y^H + I_q =: R_{yy} \\ \mathbb{E}\left[x_i y_i^H\right] &= U_x K_{xy} U_y^H =: R_{xy}. \end{aligned} \tag{C.2}$$

We define the rank of R_{xy} to be k.

C.2 Almost Sure Limit of CCA Eigenvalues

Bao et. al [2] solve for the canonical correlation estimates in the following setting

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} I_p & R \\ R^H & I_q \end{bmatrix}^{1/2} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \tag{C.3}$$

where W_1 has independent columns that are $\mathcal{N}(0, I_p)$ and W_2 has independent columns that are $\mathcal{N}(0, I_q)$; W_1 is independent of W_2 and $R = \mathbf{diag}(\sqrt{r_1}, \ldots, \sqrt{r_k}, 0, \ldots 0)$. In this setup, $[X^H \;\; Y^H]^H$ has covariance matrix

$$\begin{bmatrix} I_p & R \\ R^H & I_q \end{bmatrix}$$

and we may view (C.3) as a sample from this covariance matrix. Here, we first show that the above data model in (C.1) can be transformed via invertible transformations to achieve the form of (C.3). Using our own notation, we then provide an alternative but equivalent derivation to Bao et al. for the almost sure limit of the correlation estimates of empirical CCA. In this setting, we assume that $n > p + q$ as below this limit, we know via simple geometric arguments that $\widehat{\rho}_{\text{cca}} = 1$ deterministically.

We now show that our data model in (C.1) may be formulated as (C.3). In our model,

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix}^{1/2} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$$

where W_1 and W_2 are the same as above. This views the data matrices as a sample from a covariance matrix. Define

$$\begin{bmatrix} \widetilde{X} \\ \widetilde{Y} \end{bmatrix} = \begin{bmatrix} R_{xx} & 0 \\ 0 & R_{yy} \end{bmatrix}^{-1/2} \begin{bmatrix} X \\ Y \end{bmatrix}.$$

In this setup, $[X^H \;\; Y^H]^H$ has covariance matrix

$$\begin{bmatrix} I_p & M \\ M^H & I_q \end{bmatrix}$$

where

$$M = R_{xx}^{-1/2} R_{xy} R_{yy}^{-1/2}.$$

With the definitions of the population covariance matrices for our data model,

$$M = U_x \left(\Theta_x + I_{k_x}\right)^{1/2} \Theta_x^{1/2} P_{xy} \Theta_y^{1/2} \left(\Theta_y + I_{k_y}\right)^{-1/2} U_y^H.$$

Define $F_M D_M G_M^H$ to be the SVD of M, noting that D_M has at most k nonzero singular values. Then make the transformation

$$\begin{aligned} \begin{bmatrix} \widetilde{\widetilde{X}} \\ \widetilde{\widetilde{Y}} \end{bmatrix} &= \begin{bmatrix} F_M^H & 0 \\ 0 & G_M^H \end{bmatrix}^{1/2} \begin{bmatrix} \widetilde{X} \\ \widetilde{Y} \end{bmatrix} = \begin{bmatrix} F_M^H R_{xx}^{-1/2} & 0 \\ 0 & G_M^H R_{yy}^{-1/2} \end{bmatrix}^{1/2} \begin{bmatrix} X \\ Y \end{bmatrix} \\ &= \begin{bmatrix} I_p & D_M \\ D_M^H & I_q \end{bmatrix}^{1/2} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}. \end{aligned}$$

Therefore, from this transformation, we are in the same setting as Bao et al. with, for $i = 1, \ldots, k$,

$$d_i = \sqrt{r_i}.$$

We note that d_i are the singular values of $\widetilde{K}_{xy}$, which we defined as $\kappa_1, \ldots, \kappa_k$. If we are in the special case where $P_{xy} = \mathbf{diag}(\rho_1, \ldots, \rho_k)$, then

$$d_i = \frac{\theta_i^{(x)} \theta_i^{(y)} \rho_i}{\sqrt{\left(\theta_i^{(x)}\right)^2 + 1} \sqrt{\left(\theta_i^{(y)}\right)^2 + 1}}.$$

Next we proceed with an analogous proof of Bao et al. As noted in their paper, transforming X and Y to $\widetilde{\widetilde{X}}$ and $\widetilde{\widetilde{Y}}$ preserves the canonical correlation estimates because the transformation matrix is non-singular. Our target matrix in CCA is

$$C_{\text{cca}} = R_{xx}^{-1} R_{xy} R_{yy}^{-1} R_{xy}^{H}. \tag{C.4}$$

Clearly, making invertible transformations of X and Y preserves the eigenvalues of (C.4). We proceed assuming X and Y are the transformed versions $\widetilde{\widetilde{X}}$ and $\widetilde{\widetilde{Y}}$ to ease notation.

First for $i = 1, \ldots, k$, define

$$\alpha_i = \frac{\sqrt{1+d_i} + \sqrt{1-d_i}}{2}, \quad \beta_i = \frac{\sqrt{1+d_i} - \sqrt{1-d_i}}{2}$$

and the matrices

$$P_1 = \begin{bmatrix} \mathbf{diag}(\alpha_1, \ldots, \alpha_k) & 0 \\ 0 & I_{p-k} \end{bmatrix}, P_2 = \begin{bmatrix} \mathbf{diag}(\alpha_1, \ldots, \alpha_k) & 0 \\ 0 & I_{q-k} \end{bmatrix},$$

$$P_3 = \begin{bmatrix} \mathbf{diag}(\beta_1, \ldots, \beta_k) & 0 \\ 0 & 0 \end{bmatrix}.$$

With these definitions, we observe that

$$\begin{bmatrix} I_p & D_M \\ D_M^H & I_q \end{bmatrix}^{1/2} = \begin{bmatrix} P_1 & P_3 \\ P_3^H & P_2 \end{bmatrix}.$$

Defining the transformation (again re-using notation for simplicity)

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} P_1^{-1} & 0 \\ 0 & P_2^{-1} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix},$$

we have that

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} I_p & P \\ P & I_q \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix},$$

where

$$P = \begin{bmatrix} \mathbf{diag}(\tau_1, \ldots, \tau_k) & 0 \\ 0 & 0 \end{bmatrix},$$

with $\tau_i = \beta_i/\alpha_i$. Next, define

$$Q = \begin{bmatrix} \mathbf{diag}(2\tau_1/\left(1+\tau_1^2\right), \ldots, 2\tau_k/\left(1+\tau_k^2\right)) & 0 \\ 0 & 0 \end{bmatrix},$$

and

$$W = \left(I - QP^H\right) W_1 + (P-Q)W_2,$$

so that by construction W and Y are independent and

$$X = W + QY.$$

The covariance matrices for Y and W are

$$\begin{aligned} R_W = \mathbb{E}\left[\frac{1}{n}WW^H\right] &= \begin{bmatrix} \mathbf{diag}(1+\frac{\tau_1^4-3\tau_1^2}{1+\tau_1^2}, \ldots, 1+\frac{\tau_k^4-3\tau_k^2}{1+\tau_k^2}) & 0 \\ 0 & I_{p-k} \end{bmatrix} \\ R_Y = \mathbb{E}\left[\frac{1}{n}YY^H\right] &= \begin{bmatrix} \mathbf{diag}(1+\tau_1, \ldots, 1+\tau_k) & 0 \\ 0 & I_{q-k} \end{bmatrix}. \end{aligned}$$

Finally making the transformation

$$\begin{bmatrix} \widetilde{X} \\ \widetilde{Y} \end{bmatrix} = \begin{bmatrix} R_W^{-1/2} & 0 \\ 0 & R_Y^{-1/2} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix},$$

yields

1. $\widetilde{Y}$ has independent columns that are $\mathcal{N}(0, I_q)$
2. $\widetilde{W} = R_W^{-1/2}W$ has independent columns that are $\mathcal{N}(0, I_p)$.
3. $\widetilde{X}$ has independent columns that are $\mathcal{N}(0, R_W^{-1/2}QR_YQ^HR_W^{-1/2})$
4. $\widetilde{W}$ and $\widetilde{Y}$ are independent.

Denoting

$$T = R_W^{-1/2}QR_Y^{1/2} = \begin{bmatrix} \mathbf{diag}(t_1, \ldots, t_k) & 0 \\ 0 & 0 \end{bmatrix},$$

with $t_i = 2\tau_i/(1-\tau_i^2)$, we finally arrive at the setting (again dropping the tildes)

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} W + TY \\ Y \end{bmatrix}. \tag{C.5}$$

This is wonderful because we now have a perturbation model for X and Y. Defining the sample covariance matrices of our matrices as

$$\begin{aligned} S_{xx} &= \frac{1}{n} X X^H \\ S_{yy} &= \frac{1}{n} Y Y^H \\ S_{xy} &= \frac{1}{n} X Y^H \\ S_{yx} &= \frac{1}{n} Y X^H \\ S_{ww} &= \frac{1}{n} W W^H \\ S_{wy} &= \frac{1}{n} W Y^H \\ S_{yw} &= \frac{1}{n} Y W^H, \end{aligned}$$

we have the relationship. that

$$\begin{aligned} S_{xx} &= S_{ww} + T S_{yw} + S_{wy} T^H + T^H S_{yy} T^H \\ S_{xy} &= S_{wy} + T S_{yy} \\ S_{yx} &= S_{yw} + S_{yy} T^H. \end{aligned}$$

Therefore the CCA matrix in (C.4) becomes a low rank matrix plus a product of independent noise matrices. We show this beginning with the characteristic equation for (C.4).

$$\begin{aligned} 0 \quad &= \det\left(S_{xx}^{-1} S_{xy} S_{yy}^{-1} S_{yx} - \lambda I\right) \\ &= \det\left(S_{xy} S_{yy}^{-1} S_{yx} - \lambda S_{xx}\right) \\ &= \det\left(\left(S_{wy} + T S_{yy}\right) S_{yy}^{-1} \left(S_{yw} + S_{yy} T^H\right) - \lambda \left(S_{ww} + T S_{yw} + S_{wy} T^H + T^H S_{yy} T^H\right)\right) \\ &= \det\left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} + (1-\lambda) \underbrace{\left(T S_{yw} + S_{wy} T^H + T S_{yy} T^H\right)}_{\Delta} \right) \end{aligned}$$

This give a nice low rank perturbation of a random matrix product. Therefore, if λ is not an eigenvalue of $S_{ww}^{-1} S_{wy} S_{yy}^{-1} S_{yw}$. we have

$$0 = \det\Big(I_p + (1-\lambda)\left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww}\right)^{-1} \Delta\Big).$$

Examining T, we see that the rank of T is at most k as therefore Δ will be low rank.

The structure of Δ is ugly but necessary for the remainder of the proof and we slightly alter the notation of Bao et al. We have that

$$\Delta = UV^H$$

where

$$\begin{aligned} U &= [A_1, \dots, A_k, F_1, \dots, F_k] \\ V &= [B_1, \dots, B_k, C_1, \dots, C_k] \end{aligned}$$

where

$$\begin{aligned} A_i &= [\chi_{ii} e_i, t_i e_i, t_i u_i] \\ B_i &= [e_i, u_i, e_i] \\ C_i &= \left[\underbrace{e_i, \dots, e_i}_{k-1} \right] \\ F_1 &= [\chi_{12} e_1, \dots, \chi_{1k} e_k] \\ F_i &= [\chi_{i1} e1, \dots \chi_{i,i-1} e_{i-1}, \chi_{i,i+1} e_{i+1}, \dots, \chi_{ik} e_k] \end{aligned}$$

where e_i is i-th elementary vector, u_i is the i-th column of S_{wy} and $\chi_{ij} = t_i t_j S_{yy}(i,j)$. With these definitions, and using the identity that $\det(I + AB) = \det(I + BA)$, we have that

$$0 = \det(I_{k^2+2k} + (1-\lambda) V^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} U).$$

In this form, we have our standard characteristic equation of a low rank perturbation of a random matrix. In this case Δ is our perturbation and $S_{ww}^{-1} S_{wy} S_{yy}^{-1} S_{yw}$ is the random matrix. This highlights the importance of the previous transformations needed to write $X = W + TY$. The random matrix product is now of independent components and we can therefore compute statistics on its eigenvalues.

First define

$$M(\lambda) = I_{k^2+2k} + (1-\lambda) V^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} U \tag{C.6}$$

and based on the structure of U and V, we have that

$$M(\lambda) = I_{k^2+2k} + (1-\lambda)\, \mathbf{blkdiag}(G_1(\lambda), \dots, G_k(\lambda), 0, \dots, 0),$$

where

$$G_i(\lambda) = \begin{bmatrix} t_i^2 f(\lambda) & t_i f(\lambda) & 0 \\ 0 & 0 & t_i h(\lambda) \\ t_i^2 f(\lambda) & t_i f(\lambda) & 0 \end{bmatrix}$$

with

$$\begin{aligned} f(\lambda) &= e_i^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} e_i^H \\ h(\lambda) &= u_i^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} u_i^H. \end{aligned}$$

We get such a nice structure for $M(\lambda)$ in (C.6) due to many cancellations of terms such as

$$\begin{aligned} & e_i^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} e_j^H \\ & u_i^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} u_j^H \\ & e_i^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} u_j^H. \end{aligned}$$

Therefore, to solve our characteristic equation, we may look at the sub-blocks of M of the form $I_3 + (1-\lambda)G_i(\lambda)$. The determinant of this 3×3 matrix is

$$1 + (1-\lambda) t_i^2 f(\lambda) - (1-\lambda)^2 t_i^2 f(\lambda) h(\lambda). \tag{C.7}$$

To complete the proof, we must find closed form expressions for $f(\lambda)$ and $h(\lambda)$ and substitute them into (C.7) to solve for λ.

First, define the projection matrix $P_Y = Y^H \left(Y Y^H \right)^{-1} Y$ and the matrices $E = \frac{1}{n} W P_Y W^H$ and $H = \frac{1}{n} W (I - P_Y) W^H$. Therefore, with these definitions

$$\left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} = (E - \lambda(E + H))^{-1} = ((1-\lambda)E - \lambda H)^{-1}.$$

These definitions are wonderful because E and H are independent by construction and

$$\begin{aligned} E &\sim \text{Wishart}_p(I_p, q) \\ H &\sim \text{Wishart}_p(I_p, n-q). \end{aligned}$$

This formulation is sufficient for $f(\lambda)$ but not $h(\lambda)$. Examining $h(\lambda)$, we have that

$$h(\lambda) = u_1^H \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} u_1 = e_1^H S_{yw} \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} S_{wy} e_1.$$

Next, we attempt to get a similar expression for this additional matrix product. Defining

$$\Phi(\lambda) = S_{yw} \left(S_{wy} S_{yy}^{-1} S_{yw} - \lambda S_{ww} \right)^{-1} S_{wy}$$

and applying the Woodbury matrix identity, we have

$$\begin{aligned}\Phi(\lambda) &= S_{yw}S_{ww}^{-1/2}\left(S_{ww}^{-1/2}S_{wy}S_{yy}^{-1}S_{yw}S_{ww}^{-1/2}-\lambda I_p\right)^{-1}S_{ww}^{-1/2}S_{wy}\\ &= S_{yw}S_{ww}^{-1/2}\left[-\frac{1}{\lambda}-\frac{1}{\lambda^2}S_{ww}^{-1/2}S_{wy}\left(S_{yy}-\frac{1}{\lambda}S_{yw}S_{ww}^{-1}S_{wy}\right)^{-1}S_{wy}S_{ww}^{-1/2}\right]S_{ww}^{-1/2}S_{wy}.\end{aligned}$$

Next define $A = S_{yw}S_{ww}^{-1}S_{wy} = \frac{1}{n}YP_wY^H$ and $B = \frac{1}{n}Y(I-P_w)Y^H$, similar to above. Recall that with these definitions,

$$\begin{aligned}A &\sim \text{Wishart}_q(I_q, p)\\ B &\sim \text{Wishart}_q(I_q, n-p).\end{aligned}$$

With these definitions, we have

$$\begin{aligned}S_{yw}\left(S_{wy}S_{yy}^{-1}S_{yw}-\lambda S_{ww}\right)^{-1}S_{wy} &= -\frac{1}{\lambda}A+\frac{1}{\lambda}A\left(A-\lambda(A+B)\right)^{-1}A\\ &= -\frac{1}{\lambda}A+\frac{1}{\lambda}A\left((1-\lambda)A-\lambda B\right)^{-1}A\end{aligned}$$

After another application of the Woodbury matrix identity, we have that

$$\begin{aligned}\Phi(\lambda) &= -\frac{1}{\lambda}A+\frac{1}{\lambda}A\left[\frac{1}{1-\lambda}A^{-1}-\frac{1}{(1-\lambda)^2}A^{-1}\left(-\frac{1}{\lambda}B^{-1}+\frac{1}{1-\lambda}A^{-1}\right)^{-1}A^{-1}\right]A\\ &= \frac{1}{1-\lambda}A-\frac{1}{\lambda(1-\lambda)^2}\left(-\frac{1}{\lambda}B^{-1}+\frac{1}{1-\lambda}A^{-1}\right)^{-1}\\ &= \frac{1}{1-\lambda}A+\frac{1}{1-\lambda}\left((1-\lambda)B^{-1}-\lambda A^{-1}\right)^{-1}.\end{aligned}$$

While this may look ugly, it is quite useful.

Recall that the Stieltjes transform of a spectral distribution of a matrix X with eigenvalues $\gamma_1, \ldots, \gamma_n$ is

$$\begin{aligned}m_{\mu_X}(z) &= \int \frac{1}{\gamma-z}d_{\mu_X}\\ &= \frac{1}{n}\,\mathbf{tr}((X-zI)^{-1}) \text{ in the finite case} \qquad\qquad (C.8)\\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\gamma_i-z}\end{aligned}$$

In addition, recall the R-transform of a spectral distribution is

$$R_{\mu_X}(z) = K_{\mu_X}(z) - \frac{1}{z}, \tag{C.9}$$

where K_{μ_X} is the Blue function of the spectral distribution with the property

$$-m_{\mu_X}(K_{\mu_X}(z)) = K_{\mu_X}(-m_{\mu_X}(z)) = z. \tag{C.10}$$

Specifically, we have the relationship that

$$\begin{aligned} R_{\mu_X}(-m_{\mu_X}(z)) \quad &= K_{\mu_X}(-m_{\mu_X}(z)) + \frac{1}{m_{\mu_X}(z)} \\ &= z + \frac{1}{m_{\mu_X}(z)}. \end{aligned}$$

Therefore we can recover the Stieltjes transform from the R-transform. A very nice property of the R-transform is free additive convolution. Mainly, if we have matrices X_1 and X_2,

$$R_{X_1+X_2}(z) = R_{X_1}(z) + R_{X_2}(z). \tag{C.11}$$

With these definitions in mind, we first note that $m_{\mu_X}(0) = \frac{1}{n}\,\mathbf{tr}(X^{-1})$. This is extremely helpful for our problem! Define the matrices

$$\begin{aligned} J &= (1-\lambda)E + \lambda H \\ \widetilde{J} &= (1-\lambda)B^{-1} + \lambda A^{-1}. \end{aligned}$$

To solve for $f(\lambda)$ and $h(\lambda)$, we need so solve for the Stieltjes transforms of J and $\widetilde{J}$. To accomplish this, we first will solve for the Stieltjes transforms of the component matrices, find the associated R-transform, use (C.11) to find the R-transform of the sum, and then transform back to Stieltjes transforms. Recall that E, H, A, B are all Wishart random matrices. The basic Stieltjes transform for a Wishart random matrix X with parameter c is

$$m_{\mu_X}(z) = \frac{(1-c) - z + \sqrt{(z-1-c)^2 - 4c}}{2cz}.$$

We define $\widetilde{E} = (1-\lambda)E$, $\widetilde{H} = -\lambda H$, $\widetilde{B} = (1-\lambda)B^{-1}$, and $\widetilde{A} = -\lambda A^{-1}$. The Stieltjes transforms of the limiting spectral densities of these matrices are (through change of variables and some calculation)

$$m_{\mu_{\widetilde{E}}}(z) = \frac{(1-\lambda)(c_y - c_x) - z + \sqrt{(z-(1-\lambda)(c_x+c_y))^2 - 4(1-\lambda)^2 c_x c_y}}{2(1-\lambda)c_x z}$$
$$m_{\mu_{\widetilde{H}}}(z) = \frac{\lambda(1-c_y-c_x) + z - \sqrt{(z+\lambda(1+c_x-c_y))^2 - 4c_x(\lambda)^2(1-c_y)}}{2\lambda c_x z}$$
$$m_{\mu_{\widetilde{A}}}(z) = -z^{-1} - \frac{c_x - c_y + \lambda z^{-1} - \sqrt{(z^{-1}\lambda + c_x + c_y)^2 - 4c_x c_y}}{2c_y z}$$
$$m_{\mu_{\widetilde{B}}}(z) = -z^{-1} - \frac{1 - c_x - c_y - (1-\lambda)z^{-1} + \sqrt{(z^{-1}(1-\lambda) - (1-c_x+c_y))^2 - 4(1-c_x)c_y}}{2c_y z}$$

Using (C.9) and (C.10), we have that the R-transforms of these expressions are

$$\begin{aligned}
R_{\mu_{\widetilde{E}}}(w) &= \frac{(1-\lambda)c_y}{1-(1-\lambda)c_x w} \\
R_{\mu_{\widetilde{H}}}(w) &= -\frac{\lambda(1-c_y)}{1+\lambda c_x w} \\
R_{\mu_{\widetilde{A}}}(w) &= \frac{c_x - c_y - \sqrt{(c_x-c_y)^2 + 4\lambda c_y w}}{2c_y w} \\
R_{\mu_{\widetilde{B}}}(w) &= \frac{1 - c_x - c_y - \sqrt{(1-c_x-c_y)^2 + 4(\lambda-1)c_y w}}{2c_y w}
\end{aligned}$$

By observation we have that

$$f(\lambda) \to m_{\mu_J}(0).$$

We know by definition of J and using (C.11), (C.9), and (C.10) that

$$\begin{aligned}
R_{\mu_J}(w) &= R_{\mu_{\widetilde{E}}}(w) + R_{\mu_{\widetilde{H}}}(w) \\
R_{\mu_J}(-m_{\mu_J}(0)) &= R_{\mu_{\widetilde{E}}}(-m_{\mu_J}(0)) + R_{\mu_{\widetilde{H}}}(-m_{\mu_J}(0)) \\
K_{\mu_J}(-m_{\mu_J}(0)) + \frac{1}{m_{\mu_J}(0)} &= \frac{(1-\lambda)c_y}{1+(1-\lambda)c_x m_{\mu_J}(0)} - \frac{\lambda(1-c_y)}{1-\lambda c_x m_{\mu_J}(0)} \\
\frac{1}{m_{\mu_J}(0)} &= \frac{(1-\lambda)c_y}{1+(1-\lambda)c_x m_{\mu_J}(0)} - \frac{\lambda(1-c_y)}{1-\lambda c_x m_{\mu_J}(0)}
\end{aligned}$$

Solving the above for $m_{\mu_J}(0)$ yields an expression for $f(\lambda)$

$$f(\lambda) = \frac{-(c_y - c_x + 2\lambda c_x - \lambda) - \sqrt{\lambda^2 + (4c_x c_y - 2c_x - 2c_y)\lambda + (c_x - c_y)^2}}{2\lambda(1-\lambda)(c_x^2 - c_x)}. \quad \text{(C.12)}$$

We proceed similarly to get an expression for $h(\lambda)$. However, we note that

$$\begin{aligned} h(\lambda) \quad &= \frac{1}{1-\lambda} e_i^H A e_i + \frac{1}{1-\lambda} e_1 \left((1-\lambda)B^{-1} + \lambda A^{-1}\right)^{-2} e_1 \\ &\to c_x + \frac{1}{1-\lambda} m_{\mu_{\widetilde{J}}}(0) \end{aligned}$$

as $e_i^H A e_i$ converges to the expected value of the limiting spectral density of A. As A is Wishart with parameter c_x, we know that this expectation is simply c_x. Therefore, we are left to solve for $m_{\mu_{\widetilde{J}}}(0)$. We solve this again via the R-transform

$$\begin{aligned} R_{\mu_{\widetilde{J}}}(w) \quad &= R_{\mu_{\widetilde{A}}}(w) + R_{\mu_{\widetilde{B}}}(w) \\ R_{\mu_{\widetilde{J}}}(-m_{\mu_{\widetilde{J}}}(0)) \quad &= R_{\mu_{\widetilde{A}}}(-m_{\mu_{\widetilde{J}}}(0)) + R_{\mu_{\widetilde{B}}}(-m_{\mu_{\widetilde{J}}}(0)) \\ K_{\mu_{\widetilde{J}}}(-m_{\mu_{\widetilde{J}}}(0)) + \frac{1}{m_{\mu_{\widetilde{J}}}(0)} \quad &= -\frac{c_x - c_y - \sqrt{(c_x-c_y)^2 - 4\lambda c_y m_{\mu_{\widetilde{J}}}(0)}}{2c_y m_{\mu_{\widetilde{J}}}(0)} \\ &\quad - \frac{1 - c_x - c_y - \sqrt{(1-c_x-c_y)^2 - 4(\lambda-1)c_y m_{\mu_{\widetilde{J}}}(0)}}{2c_y m_{\mu_{\widetilde{J}}}(0)} \\ \frac{1}{m_{\mu_{\widetilde{J}}}(0)} \quad &= -\frac{c_x - c_y - \sqrt{(c_x-c_y)^2 - 4\lambda c_y m_{\mu_{\widetilde{J}}}(0)}}{2c_y m_{\mu_{\widetilde{J}}}(0)} \\ &\quad - \frac{1 - c_x - c_y - \sqrt{(1-c_x-c_y)^2 - 4(\lambda-1)c_y m_{\mu_{\widetilde{J}}}(0)}}{2c_y m_{\mu_{\widetilde{J}}}(0)}. \end{aligned}$$

Solving the above for $m_{\mu_{\widetilde{J}}}(0)$ yields

$$m_{\mu_{\widetilde{J}}}(0) = \frac{c_x + c_y - 2c_x c_y - \lambda + \sqrt{\lambda^2 + (4c_x c_y - 2c_x - 2c_y)\lambda + (c_x - c_y)^2}}{2c_y}$$

Therefore, we have

$$h(\lambda) = \frac{c_x}{1-\lambda} + \frac{c_x + c_y - 2c_x c_y - \lambda + \sqrt{\lambda^2 + (4c_x c_y - 2c_x - 2c_y)\lambda + (c_x - c_y)^2}}{2c_y(1-\lambda)}. \tag{C.13}$$

To conclude, we must substitute (C.13) and (C.12) into (C.7) and solve for λ. This is largely an algebra problem and we point the reader to Bao et al. if interested. After some calculation we arrive at the final result

$$\widehat{\rho}_{\text{cca}}^{(i)} \xrightarrow{\text{a.s.}} \begin{cases} \sqrt{d_i^2\left(1 - c_x + \frac{c_x}{d_i^2}\right)\left(1 - c_y + \frac{c_y}{d_i^2}\right)} & d_i^2 \geq r_c \\ \sqrt{d_r} & d_i^2 < r_c \end{cases}$$

where

$$\begin{aligned} r_c &= \frac{c_x c_y + \sqrt{c_y c_y (1 - c_x)(1 - c_y)}}{(1 - c_x)(1 - c_y) + \sqrt{c_x c_y (1 - c_x)(1 - c_y)}} \\ d_r &= c_x + c_y - 2c_x c_y + 2\sqrt{c_x c_y (1 - c_x)(1 - c_y)}. \end{aligned}$$

C.3 Canonical Vectors

We now solve for the accuracy of the canonical vectors in empirical CCA. We consider, without loss of generality, the accuracy of the canonical vector of only one dataset. Let X and Y be drawn from (C.1). With the same definition of the target matrix C_{cca}, the estimated canonical vectors, $\widehat{w}_x^{(i)}$ solves the generalized eigenvalue problem $C\widehat{w}_x^{(i)} = \widehat{\rho}_{\text{cca}}^2 \widehat{w}_x^{(i)}$. Recall from Chapter 5 that the unit-norm population canonical vector that we are trying to estimate is

$$w_x^{(i)} = \frac{R_{xx}^{-1/2} U_x U_{\widetilde{K}}(:, i)}{\sqrt{U_{\widetilde{K}}(:, i)^H U_x^H R_{xx}^{-1} U_x U_{\widetilde{K}}(:, i)}}$$

where $U_{\widetilde{K}}$ are the left singular vectors of $\widetilde{K}_{xy}$. In this section, we want to find a closed form expression for $|\langle w_x^{(i)}, \widehat{w}_x^{(i)} \rangle|^2$. We first introduce the change of variables similar to the correlation computation

$$\begin{aligned} \widetilde{X} &= F_M^H R_{xx}^{-1/2} X \\ \widetilde{Y} &= G_M^H R_{yy}^{-1/2} Y \end{aligned}$$

where $F_M D_M G_M^H$ is the SVD of M, defined above. Then with this transformation, we have

$$C_{\text{cca}} = \left(R_{xx}^{1/2} F_M\right) \underbrace{\left(\widetilde{X}\widetilde{X}^H\right)^{-1} \widetilde{X}\widetilde{Y}^H \left(\widetilde{Y}\widetilde{Y}^H\right)^{-1} \widetilde{Y}\widetilde{X}^H}_{\widetilde{C}} \left(F_M^H R_{xx}^{1/2}\right).$$

Then, if $\widetilde{u}^{(i)}$ is a unit-norm eigenvector of $\widetilde{C}$, via the similarity transform,

$$\widehat{w}_x^{(i)} = \frac{\left(F_M^H R_{xx}^{1/2}\right)^{-1} \widetilde{u}^{(i)}}{\sqrt{\widetilde{u}^{(i)H} \left(F_M^H R_{xx}^{1/2}\right)^{-H} \left(F_M^H R_{xx}^{1/2}\right)^{-1} \widetilde{u}^{(i)}}}.$$

Therefore

$$\begin{aligned} |\langle w_x^{(i)}, \widehat{w}_x^{(i)}\rangle|^2 &= \frac{\left(U_{\widetilde{K}}(:,i)^H U_x^H R_{xx}^{-1/2} \left(F_M^H R_{xx}^{1/2}\right)^{-1} \widetilde{u}^{(i)}\right)^2}{\left(U_{\widetilde{K}}(:,i)^H U_x^H R_{xx}^{-1} U_x U_{\widetilde{K}}(:,i)\right) \left(\widetilde{u}^{(i)H} \left(F_M^H R_{xx}^{1/2}\right)^{-T} \left(F_M^H R_{xx}^{1/2}\right)^{-1} \widetilde{u}^{(i)}\right)} \\ &= \frac{\left(U_{\widetilde{K}}(:,i)^H U_x^H R_{xx}^{-1} F_M \widetilde{u}^{(i)}\right)^2}{\left(U_{\widetilde{K}}(:,i)^H U_x^H R_{xx}^{-1} U_x U_{\widetilde{K}}(:,i)\right) \left(\widetilde{u}^{(i)H} F_M^H R_{xx}^{-1} F_M \widetilde{u}^{(i)}\right)}. \end{aligned} \tag{C.14}$$

Now due to the structure of $\widetilde{K}_{xy}$ and M, we have that

$$F_M = \left[U_x U_{\widetilde{K}} \quad \left(U_x U_{\widetilde{K}}\right)^{\perp}\right],$$

which allows us to rewrite

$$|\langle w_x^{(i)}, \widehat{w}_x^{(i)}\rangle|^2 = \frac{\left(e_i^H F_M^H R_{xx}^{-1} F_M \widetilde{u}^{(i)}\right)^2}{\left(e_i^H F_M^H R_{xx}^{-1} F_M e_i\right) \left(\widetilde{u}^{(i)H} F_M^H R_{xx}^{-1} F_M \widetilde{u}^{(i)}\right)}.$$

Also note that we can write our unit norm eigenvector as

$$\widetilde{u}^{(i)} = \sum_{j=1}^{p} (\widetilde{u}^{(i)H} e_j) e_j.$$

We note that by the Theorem 4.6.1, for $j = 1, \ldots, k_x, j \neq i$, $(\widetilde{u}^{(i)H} e_j) \xrightarrow{\text{a.s.}} 0$. Examining the structure of the population covariance matrix of our data model, the final $p - k_x$ eigenvalues of R_{xx} are 1 so we have that the last term in the denominator above is

$$\widetilde{u}^{(i)H} F_M^H R_{xx}^{-1} F_M \widetilde{u}^{(i)} \xrightarrow{\text{a.s.}} \left(e_i^H F_M^H R_{xx}^{-1} F_M e_i\right) \left(\widetilde{u}^{(i)H} e_i\right)^2 + \left[1 - \left(\widetilde{u}^{(i)H} e_i\right)^2\right].$$

By a similar argument, the term in the numerator is

$$e_i^H F_M^H R_{xx}^{-1} F_M \widetilde{u}^{(i)} \xrightarrow{\text{a.s.}} e_i^H F_M^H R_{xx}^{-1} F_M e_i \left(\widetilde{u}^{(i)H} e_i\right).$$

Therefore, (C.14) becomes

$$|\langle w_x^{(i)}, \widehat{w}_x^{(i)} \rangle|^2 \xrightarrow{\text{a.s.}} \frac{\gamma_i \left(\widetilde{u}^{(i)H} e_i\right)^2}{(\gamma_i - 1) \left(\widetilde{u}^{(i)H} e_i\right)^2 + 1}, \tag{C.15}$$

where

$$\gamma_i = e_i^H F_M^H R_{xx}^{-1} F_M e_i.$$

Thus, it suffices to find the accuracy of $\widetilde{u}^{(i)}$ with respect to e_i to solve (C.15).

We proceed by first noting that in the above CCA correlation derivation, we first transformed $\widetilde{X}$ and $\widetilde{Y}$ by a series of invertible linear transformations to get the final matrix perturbation form. While this does not affect the eigenvalues of the target matrix, it does affect the eigenvectors and so we correct for that here via similarity transformations. These transformations may be done in one step and since the transformation matrices are all diagonal, it makes the story a little easier. Specifically we have

$$\widetilde{\widetilde{X}} = M_x \widetilde{X}$$
$$\widetilde{\widetilde{Y}} = M_y \widetilde{Y}$$

where

$$M_x = \begin{bmatrix} \mathbf{diag}(m_{x1}, \ldots, m_{xk_x}) & 0 \\ 0 & I_{p-k_x} \end{bmatrix}, M_y = \begin{bmatrix} \mathbf{diag}(m_{y1}, \ldots, m_{yk_y}) & 0 \\ 0 & I_{q-k_y} \end{bmatrix}$$

with

$$m_{xi} = \frac{1}{\alpha_i \sqrt{1 + \sqrt{\frac{\tau_i^4 - 3\tau_i^2}{1+\tau_i^2}}}}, \quad m_{yi} = \frac{1}{\alpha_i \sqrt{1 + \tau_i^2}},$$

where these parameter were defined in the correlation derivation. After these transformations, our target matrix is

$$\widetilde{C} = M_x^{-1} \widetilde{\widetilde{C}} M_x.$$

Again via a similarity transform, we see that if $\widetilde{\widetilde{u}}$ is an eigenvector of $\widetilde{\widetilde{C}}$, then $\widetilde{u} =$

$\frac{M_x\widetilde{\widetilde{u}}}{\sqrt{\widetilde{\widetilde{u}}^H M_x^2 \widetilde{\widetilde{u}}}}$. Then via a similar computation, we have

$$\left|\langle e_i, \widetilde{u}^{(i)}\rangle\right|^2 = \frac{m_{xi}^2 \left(e_i^H \widetilde{\widetilde{u}}^{(i)}\right)^2}{(m_{xi}^2-1)\left(e_i^H\widetilde{\widetilde{u}}^{(i)}\right)^2+1}.$$

Therefore, solving for the eigenvector accuracy of $\widetilde{\widetilde{u}}$ we can recover the accuracy of our canonical vector via

$$\left|\langle w_x^{(i)}, \widehat{w}_x^{(i)}\rangle\right|^2 = \frac{\gamma_i m_{xi}^2 \left|\langle e_i, \widetilde{\widetilde{u}}^{(i)}\rangle\right|^2}{\left|\langle e_i,\widetilde{\widetilde{u}}^{(i)}\rangle\right|^2 \left(m_{xi}^2\left(\gamma_i-1\right)+m_{xi}^2-2\right)+1}. \tag{C.16}$$

As γ_i and m_{xi} are parameters of our problems, what is left is to determine the eigenvector accuracy of $\widetilde{\widetilde{u}}^{(i)}$. We proceed using the same matrix perturbation model as in the eigenvalue derivation. Our master equation is

$$\begin{aligned} \widetilde{\widetilde{C}}\widetilde{\widetilde{u}} &= \lambda\widetilde{\widetilde{u}} \\ S_{xy}S_{yy}^{-1}S_{yx}\widetilde{\widetilde{u}} &= \lambda S_{xx}\widetilde{\widetilde{u}} \end{aligned}$$

We made these specific transformations to achieve the setting of (C.5). Using similar derivations as the eigenvalue setting, we arrive at the low-rank matrix we desire, which just so happens to have a component of e_i.

$$\left(S_{wy}S_{yy}^{-1}S_{yw} - \lambda S_{ww} + (1-\lambda)UV^H\right)\widetilde{\widetilde{u}}^{(i)} \quad = 0$$

The low rank matrix UV^H will contain terms of the form $e_iu_i^H$, $u_ie_i^H$, and $e_ie_i^H$. As shown in Theorem 2.7 c) in [116], the energy of $\widetilde{\widetilde{u}}^{(i)}$ lying in orthogonal components of $e_{j\neq i}$ will be zero and hence for notational simplicity, we ignore them going forward. Proceeding, we have

$$\begin{aligned} \left(S_{wy}S_{yy}^{-1}S_{yw} - \lambda S_{ww} + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H) + (1-\lambda)t_i^2 S_{yy}(i,i)e_ie_i^H\right)\widetilde{\widetilde{u}}^{(i)} &= 0 \\ \left(S_{wy}S_{yy}^{-1}S_{yw} - \lambda S_{ww} + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H)\right)\widetilde{\widetilde{u}}^{(i)} &= (\lambda-1)t_i^2 S_{yy}(i,i)e_ie_i^H\widetilde{\widetilde{u}}^{(i)}. \end{aligned}$$

Noticing that the right hand side is a scaled version of the vector e_1, we have that

$$\widetilde{\widetilde{u}}^{(i)} = \left(S_{wy}S_{yy}^{-1}S_{yw} - \lambda S_{ww} + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H)\right)^{-1} e_i,$$

but this vector may not be unit norm, and we divide by it's norm. Define $\Psi(\lambda) = S_{wy}S_{yy}^{-1}S_{yw} - \lambda S_{ww}$. Then

$$\left|\langle e_i, \widetilde{\widetilde{u}}^{(i)}\rangle\right|^2 = \frac{\left(e_i^H\left(\Psi(\lambda) + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H)\right)^{-1} e_i\right)^2}{e_i^H\left(\Psi(\lambda) + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H)\right)^{-2} e_i}. \tag{C.17}$$

We notice that the matrix inverse is a rank-2 addition to $\Phi(\lambda)$ and so can be simplified with the matrix inversion lemma. Also, define $\Phi(\lambda) = A^{-1}$ and $f(\lambda) = e_i^H\Phi(\lambda)e_1$, $h(\lambda) = u_i^H\Phi(\lambda)u_i$ with $u_i = S_{wy}(:,i)$, $q(\lambda) = e_i^H\Phi(\lambda)^2e_i$ and $s(\lambda) = u_i^H\Phi(\lambda)^2u_i$. From the correlation derivation above, we recall that $e_1^H\Phi(\lambda)u_1 = 0$. With this notation and the Woodubry inversion lemma, we have that the numerator is

$$e_i^H\left(\Psi(\lambda) + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H)\right)^{-1} e_i = \frac{f(\lambda)}{1-(1-\lambda)^2t_i^2f(\lambda)h(\lambda)} \tag{C.18}$$

and that the denominator

$$\begin{aligned}
&e_i^H\left(\Psi + (1-\lambda)t_i(e_iu_i^H + u_ie_i^H)+\right)^{-2} e_i = q(\lambda)+ \\
&\quad \frac{2f(\lambda)h(\lambda)q(\lambda)}{\frac{1}{(1-\lambda)^2t_i^2} - f(\lambda)h(\lambda)} + \left(\frac{1}{(1-\lambda)^2t_i^2} - f(\lambda)h(\lambda)\right)^2\left(f(\lambda)^2h(\lambda)^2q(\lambda) + \frac{f(\lambda)^2s(\lambda)}{(1-\lambda)^2t_i^2}\right)
\end{aligned} \tag{C.19}$$

We derived expressions for $f(\lambda)$ and $h(\lambda)$ in the correlation derivation, specifically, (C.12) and (C.13). Therefore, we need expressions for $q(\lambda)$ and $s(\lambda)$. Let's begin with $q(\lambda)$.

Similar to our correlation derivation, define the projection matrix

$$P_Y = Y^H(YY^H)^{-1}Y,$$

and matrices $B_1 = WPW^H$ and $B_2 = W(I-P)W^H$. Then $\Psi = (1-\lambda)B_1 - \lambda B_2$. As discussed before, B_1 and B_2 are independent Wishart matrices. Then

$$q(\lambda) = e_1^H\left((1-\lambda)B_1 - \lambda B_2\right)^{-2} e_1.$$

Let $(1-\lambda)B_1 - \lambda B_2$ have limiting eigenvalue distribution $\sigma(w)$. Then

$$q(\lambda) \xrightarrow{\text{a.s.}} \int \frac{1}{x^2} d_{\mu_\Psi}(x).$$

Let $m_{\mu_\Psi}(z)$ be the Stieltjes transform of Ψ. Then we see that

$$m_{\mu_\Psi}(z) = \int \frac{1}{x-z} d_{\mu_\Psi}(x), \; m'_{\mu_\Psi}(z) = -\int \frac{1}{(x-z)^2} d_{\mu_\Psi}(x)$$

Therefore,

$$q(\lambda) \to -m'_{\mu_\Psi}(0).$$

To compute this, we use the R-transform trick that we employed in the correlation derivation. First define $R_{b_1}(w)$ and $R_{b_2}(w)$ as the R-transforms of $(1-\lambda)B_1$ and $-\lambda B_2$. From Bao et al. and above, we have

$$\begin{aligned} R_{b_1}(w) &= \frac{(1-\lambda)c_y}{1-(1-\lambda)c_x w} \\ R_{b_2}(w) &= \frac{-\lambda(1-c_y)}{1+\lambda c_x w} \end{aligned}$$

By (C.11), $R_\Psi = R_{b_1} + R_{b_2}$. Substituting $w = -m_{\mu_\Psi}(z)$ and using the relationship (C.10), we have

$$\begin{aligned} R_\Psi(-m_{\mu_\Psi}(z)) \quad &= R_{b_1}(-m_{\mu_\Psi}(z)) + R_{b_2}(-m_{\mu_\Psi}(z)) \\ z + \frac{1}{m_{\mu_\Psi}(z)} \quad &= R_{b_1}(-m_{\mu_\Psi}(z)) + R_{b_2}(-m_{\mu_\Psi}(z)). \end{aligned} \tag{C.20}$$

Plugging in the expressions for the individual R-transforms into (C.20), and doing some algebra yields the following equality

$$\lambda(1-\lambda)(c_x^2 - c_x)m_{\mu_\Psi}^2(z) + (cy - cx + 2\lambda c_x - \lambda)m_{\mu_\Psi}(z) - 1 = \\ z m_{\mu_\Psi}(z)\left(1 + c_x m_{\mu_\Psi}(z) - 2\lambda c_x m_{\mu_\Psi}(z) - \lambda(1-\lambda)c_x^2 m_{\mu_\Psi}^2(z)\right).$$

Taking the derivative of both sides with respect to z, setting $z = 0$ and solving for $m'_{\mu_\Psi}(0)$ yields

$$q(\lambda) = m'_{\mu_\Psi}(0) = \frac{m_{\mu_\Psi}(0) + c_x m_{\mu_\Psi}^2(0) - 2\lambda c_x m_{\mu_\Psi}^2(0) - \lambda(1-\lambda)c_x^2 m_{\mu_\Psi}^3(0)}{\lambda(1-\lambda)(c_x^2 - c_x)2m_{\mu_\Psi}(0) + c_y - c_x + 2\lambda c_1 - \lambda}.$$

We know from the correlation derivation that $m_{\mu_\Psi}(0) = f(\lambda)$, which completes the derivation for $q(\lambda)$.

A closed form expression for $s(\lambda)$ remains an open problem. However, substituting empirically realizations of $\frac{1}{p}\,\mathbf{tr}(e_i^H \Phi(\lambda)^2 e_i)$ into (C.19) combined with the other closed form expressions to complete (C.19) and (C.18) result in a good approximation of (C.17). This good approximation then can be used to solve for the canonical vector accuracy in (C.16)

APPENDIX D

Significance Test for Canonical Correlations

D.1 Problem Setup

Let X be a zero-mean $p \times n$ matrix and let Y be a zero-mean $q \times n$ matrix. Define

$$C_{\text{cca}} = \left(XX^H\right)^{-1} XY^H \left(YY^H\right)^{-1} YX^H. \tag{D.1}$$

The largest eigenvalue of C_{cca} is the largest canonical correlation between X and Y returned by CCA. Note that C_{cca} is a $p \times p$ matrix.

Define $\widetilde{V}_x$ and $\widetilde{V}_y$ as the $n \times k_x$ and $n \times k_y$ matrices of the right singular vectors corresponding to the largest k_x and k_y singular vectors of X and Y, respectively. Similarly, define

$$C_{\text{icca}} = \mathring{V}_x^H \mathring{V}_y \mathring{V}_y^H \mathring{V}_x \tag{D.2}$$

The largest eigenvalue of C_{icca} is the largest canonical correlation between X and Y returned by ICCA. Note that C_{icca} is a $k_x \times k_y$ matrix. In this framework, k_x and k_y represent the number of informative signals individually present in X and Y.

Here, we would like to determine a statistical test to determine when the the canonical correlations returned by ICCA are statistically different from noise, which indicates the presence of correlated signals between X and Y. In our null model, we assume that X and Y are independent and that the entries of X and Y are independent $\mathcal{N}(0, 1)$ or $\mathcal{CN}(0, 1)$. We derive the distribution of the top eigenvalue of C_{icca} in this null model. This distribution then allows us to set a threshold to achieve a desired significance level when detecting the presence of correlated signals with ICCA (and consequently with empirical CCA).

D.2 Distribution of Largest eigenvalue of ICCA

The distribution of the largest canonical correlation in CCA was previously derived in [115]. We provide a similar derivation using our notation for completeness. We then use this to provide a significance test for CCA. We begin with a classical result for the largest eigenvalue of a double Wishart model that will give the distribution of our top canonical correlations in the null model for both empirical CCA and ICCA.

Proposition D.2.1. *[Johnstone 2008] Let $A \sim W_p(I, m)$ and $B \sim W_p(I, n)$ where $W_p(\Sigma, n)$ denotes a Wishart matrix formed by the product of XX^T where X is a $p \times n$ matrix with i.i.d. $\mathcal{N}_p(0, \Sigma)$ columns. Assume that $m \geq p$ and that A and B are independent. Denote the largest eigenvalue of $(A+B)^{-1}B$ as $\theta_1(p, m, n)$. Then*

$$\frac{\log\left(\frac{\theta_1}{1-\theta_1}\right) - \mu_p(p, m, n)}{\sigma_p(p, m, n)} \overset{\mathcal{D}}{\Rightarrow} F_1 \tag{D.3}$$

where F_1 is the Tracy-Widom Distribution and

$$\begin{aligned} \mu_p(p, m, n) &= 2 \log \tan\left(\frac{\varphi + \gamma}{2}\right) \\ \sigma_p^3(p, m, n) &= \frac{16}{(m+n-1)^2 \sin^2(\varphi+\gamma) \sin\varphi \sin\gamma} \end{aligned} \tag{D.4}$$

and

$$\begin{aligned} \sin^2\left(\frac{\gamma}{2}\right) &= \frac{\min(p, n) - 1/2}{m+n-1} \\ \sin^2\left(\frac{\varphi}{2}\right) &= \frac{\max(p, n) - 1/2}{m+n-1} \end{aligned} \tag{D.5}$$

Proof. See [115] for the result. □

As stated and proved by Johnstone [115], empirical CCA falls into this double Wishart model. We state the result as a proposition but provide a proof using our notation. We note that both empirical CCA and ICCA fall into this model and the only difference between the two is the dimension of the problem. An important consequence of this result is that we may use the result in Proposition D.2.1 to find the distribution of the largest canonical correlations in empirical CCA and ICCA.

Proposition D.2.2. *Let X be a $p \times n$ matrix with $\mathcal{N}(0, 1)$ entries and let Y be an independent $q \times n$ matrix with $\mathcal{N}(0, 1)$ entries. Assume that $p \leq q$ and that $n > p+q$. Let λ_1 be the largest eigenvalue of C_{cca}. Then*

$$\lambda_1 \sim \theta_1(p, n-q, q). \tag{D.6}$$

Proof. Johnstone shows the result for empirical CCA in [115]. □

This proposition will allow us to determine whether the correlations returned by empirical CCA are statistically different from the correlations returned when the data matrices are uncorrelated. We provide an analogous result for ICCA.

Theorem D.2.1. *Let X be a $p \times n$ matrix with $\mathcal{N}(0,1)$ entries and let Y be an independent $q \times n$ matrix with $\mathcal{N}(0,1)$ entries. Assume that $p \leq q$ and that $n > p+q$. Let $0 < k_x \leq p$ and $0 < k_y \leq q$ be two integers. Let λ_1 be the largest eigenvalue of C_{icca}. Then*

$$\lambda_1 \sim \theta_1(k_x, n-k_y, k_y). \tag{D.7}$$

Proof. Recalling C_{icca}, define $\widetilde{X} = \widetilde{U}_x^T X$ and $\widetilde{Y} = \widetilde{U}_y^T Y$, where $\widetilde{U}_x$ and $\widetilde{U}_y$ are the left singular vectors corresponding to the largest k_x and k_y singular values of X and Y, respectively. Define $P = \widetilde{Y}^T \left(\widetilde{Y}\widetilde{Y}^T\right)^{-1} \widetilde{Y}$, which is a $n \times n$ projection matrix of rank k_y. Let $P^\perp = I - P$ be the orthogonal complement of P. Note that $P^\perp$ is also a projection matrix of dimension $n - k_y$ and that P and $P^\perp$ are independent by construction. Define $B = \widetilde{X} P \widetilde{X}^T$ and $A = \widetilde{X} P^\perp \widetilde{X}^T$. Then C_{icca} may be written

$$C_{\text{icca}} = (A+B)^{-1}B.$$

Using these definitions of A and B, it is clear that $A \sim W_{k_x}(I, k_y)$ and $B \sim W_{k_x}(I, n-k_y)$. Applying Johnstone's Theorem gives the desired result. □

This theorem allows us to similarly determine whether the correlations returned by ICCA are statistically different from the ICCA correlations returned when the data matrices are uncorrelated. As Theorem D.2.1 is a new result, we summarize the necessary parameters needed from Proposition D.2.1 in Tables D.1 and D.2. These propositions and theorems allow us to complete the statistical tests in (4.14). Recall that the thresholds needed for these tests, first given in (4.15), are

$$\begin{aligned} \tau_{\text{cca}}^{\alpha} &= F_{\text{cca}}^{-1}(1-\alpha) \\ \tau_{\text{icca}}^{\alpha} &= F_{\text{icca}}^{-1}(1-\alpha). \end{aligned} \tag{D.8}$$

In Chapter 4, we approximated these thresholds as

$$\begin{aligned} \tau_{\text{cca}}^{\alpha} &\approx \sigma_{n,p,q} \mathsf{TW}_{\mathbb{C}}^{-1}(1-\alpha) + \mu_{n,p,q}, \\ \tau_{\text{icca}}^{\alpha} &\approx \sigma_{n,\widehat{k}_x,\widehat{k}_y} \mathsf{TW}_{\mathbb{C}}^{-1}(1-\alpha) + \mu_{n,\widehat{k}_x,\widehat{k}_y}. \end{aligned}$$

	$\mu_{\{\cdot\}}\left(k_x, k_y, n\right)$	$\sigma_{\{\cdot\}}\left(k_x, k_y, n\right)$
$x_i, y_i \in \mathbb{R}$	$2\log\tan\left(\frac{\varphi+\gamma}{2}\right)$	$\left(\frac{16}{(n-1)^2}\frac{1}{\sin^2(\varphi+\gamma)\sin(\varphi)\sin(\gamma)}\right)^{1/3}$
$x_i, y_i \in \mathbb{C}$	$\frac{\frac{\mu_N}{\tau_N}+\frac{\mu_{N-1}}{\tau_{N-1}}}{\frac{1}{\tau_N}+\frac{1}{\tau_{N-1}}}$	$\frac{2}{\frac{1}{\tau_N}+\frac{1}{\tau_{N-1}}}$

Table D.**1:** Parameters for distributions of ICCA correlation coefficients. See Table D.2 for related parameters necessary for computation.

	Related parameters
$x_i, y_i \in \mathbb{R}$	$\gamma = 2\sin^{-1}\left(\sqrt{\frac{\min(k_x,k_y)-1/2}{n-1}}\right)$ $\varphi = 2\sin^{-1}\left(\sqrt{\frac{\max(k_x,k_y)-1/2}{n-1}}\right)$
$x_i, y_i \in \mathbb{C}$	$N = \min(k_x, k_y)$ $\alpha = n - k_x - k_y$ $\beta = \lvert k_x - k_y\rvert$ $\gamma_N = 2\sin^{-1}\left(\sqrt{\frac{N+1/2}{2N+\alpha+\beta+1}}\right)$ $\varphi_N = 2\sin^{-1}\left(\sqrt{\frac{N+\beta+1/2}{2N+\alpha+\beta+1}}\right)$ $\mu_N = 2\log\tan\frac{\varphi_N+\gamma_N}{2}$ $\tau_N = \left(\frac{16}{(2N+\alpha+\beta+1)^2}\frac{1}{\sin^2(\gamma_N+\varphi_N)\sin(\varphi_N)\sin(\gamma_N)}\right)^{1/3}$

Table D.**2:** Related parameters for distributions of ICCA correlation coefficients presented in Table D.1.

These approximations are a direct consequence of the propositions and theorem above. We note that the logit transform is used by Johnstone and one would similar want to use these in practice. We provide the inverse CDF values of the Tracy-Widom distribution for a number of significance levels in Table D.3. A practitioner would select a value best fit for the specific application.

D.3 Empirical Results

Figures D.1 D.3 explore the accuracy of Theorem D.2.2 where our matrices are real valued and imaginary, respectively. Each figure plots the theoretical cumulative distribution function (c.d.f.) in a red dashed line and empirically generated cdf in a blue line. Figures D.2 and D.4 plot the corresponding probability density function (p.d.f.) for real and complex data, respectively. For these figures, we plot the the-

α	$1-\alpha$	$\mathsf{TW}_{\mathbb{R}}^{-1}(1-\alpha)$	$\mathsf{TW}_{\mathbb{C}}^{-1}(1-\alpha)$
0.990000	0.010000	-3.895432673064243	-3.724445946400548
0.950000	0.050000	-3.180379976937733	-3.194166732158107
0.900000	0.100000	-2.782427905695298	-2.901350938475908
0.700000	0.300000	-1.910379746199262	-2.266182039849163
0.500000	0.500000	-1.268574616581076	-1.804912408936580
0.300000	0.700000	-0.592287191016136	-1.324859556060199
0.100000	0.900000	0.450143289058243	-0.596851297117349
0.050000	0.950000	0.979316053469545	-0.232474469763996
0.010000	0.990000	2.023449281380126	0.477636047390792
0.001000	0.999000	3.272196059001973	1.314419480086017
0.000100	0.999900	4.359420343910324	2.034691754570250
0.000010	0.999990	5.344295940484186	2.682207321677930
0.000001	0.999999	6.256354429605480	3.278588282048362

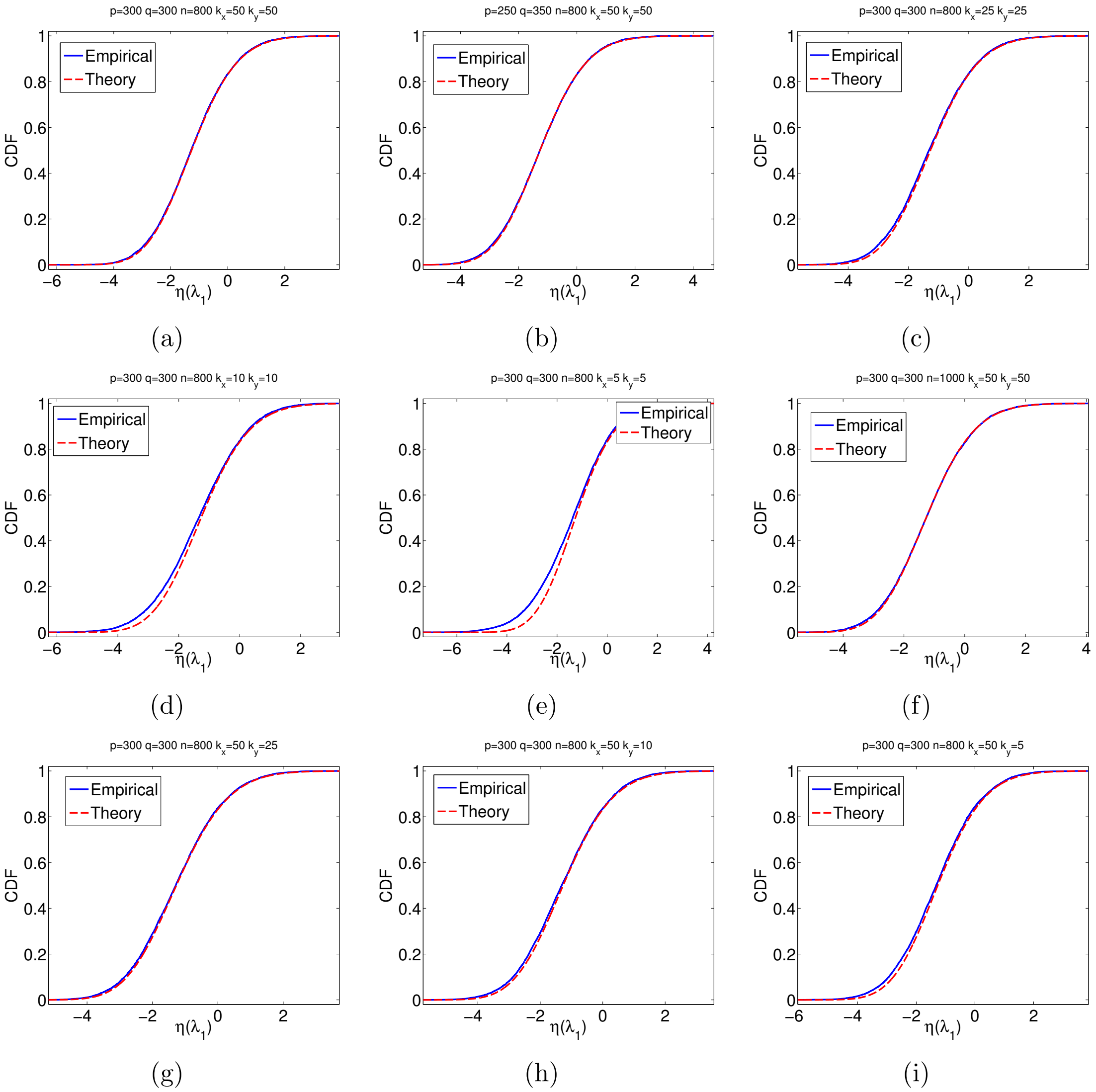
Table D.3: Percentiles of the Tracy-Widom real and complex distribution.

oretical pdf in a red line and plot the empirical pdf as a histogram. All theoretical predictions are Tracy Widom distributions that use the scaling and mean parameters given in the theorem. For the empirical results, we employ the inverse logit transform of 10000 realizations of the largest eigenvalue of (D.2) generated from random X and Y.

Each figures sweeps over various values of k_x and k_y. For large k_x and k_y, the approximation is very good (Figures 4.2(a), 4.2(c)). We lose some accuracy as these values decrease (Figures 4.2(d), 4.2(e)). Interestingly, the approximation is better on the upper tail than the lower tail, which is good for our application since we will use the c.d.f. on the upper tail.

Finally, we make a quick note on estimating k_x and k_y. Following [83], we use a Algorithm 2 to determine the number of signals present in each dataset. As Nadakuditi et al. showed that the largest eigenvalues of the sample covariance matrices in Gaussian noise-only setting follow the Tracy-Widom distribution and derived a statistical test to determine the rank of a data matrix. We use these tests to find estimates $\widehat{k}_x$ and $\widehat{k}_y$ from the top eigenvalues of the individual sample covariance matrices $\widehat{R}_{xx}$ and $\widehat{R}_{yy}$.

The parameters in Tables D.1 and D.2 employ a correction term of 0.5 that is shown to increase the convergence of the approximation. Figures D.5, D.6, D.7,and D.8 plot the convergence as a function of n for a fixed false alarm rate of 0.01 and 0.05 for real, imaginary data. These convergence plots are shown for the PCA statistical test [83] and ICCA statistical test presented here.

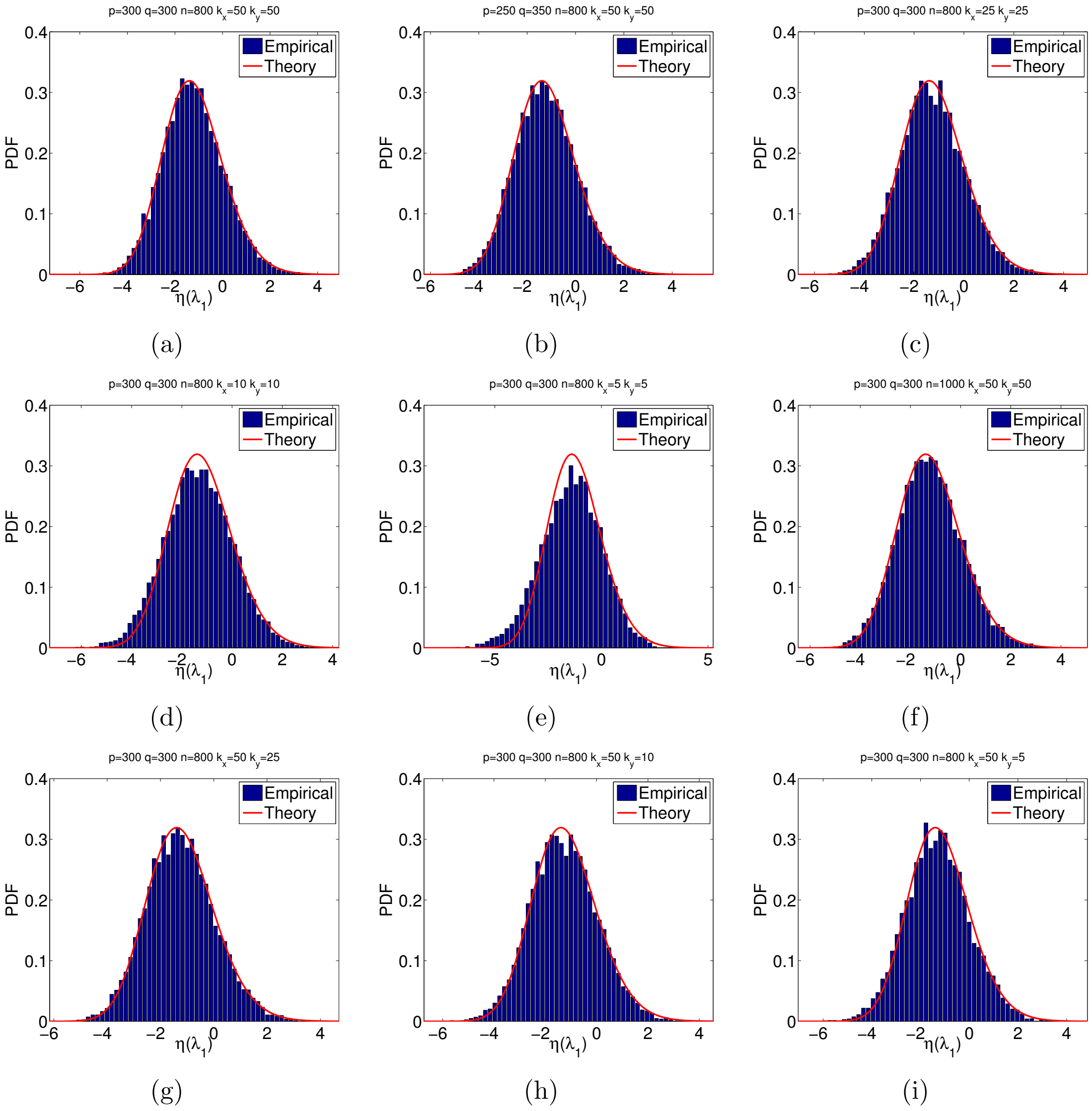


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure D.1: Empirical and theoretically predicted cumulative distribution functions (cdf) for ICCA under various parameters k_x, k_y and n for real valued X and Y.

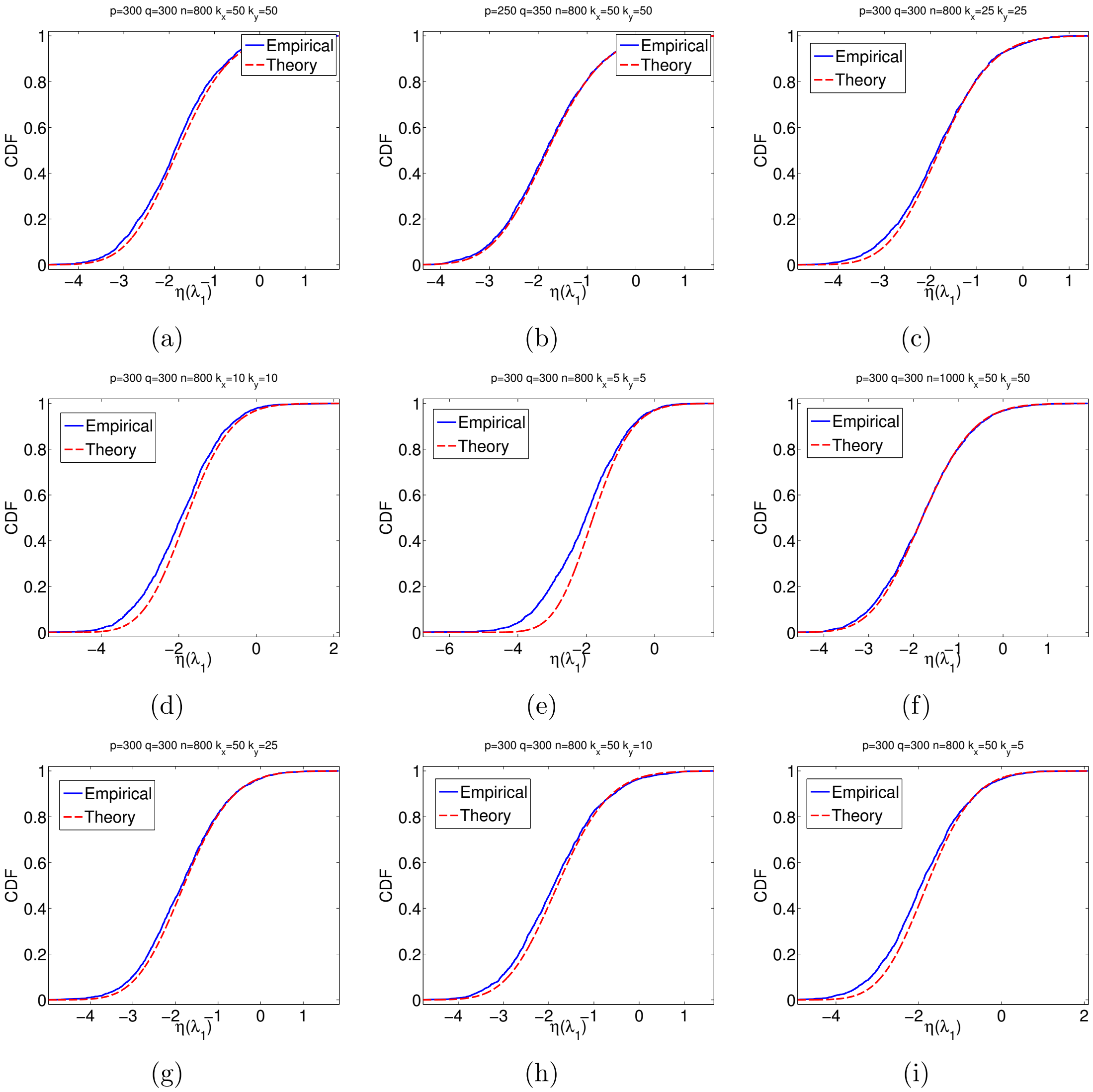


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure D.2: Empirical and theoretically predicted probability density functions (pdf) for ICCA under various parameters k_x, k_y and n for real valued X and Y.

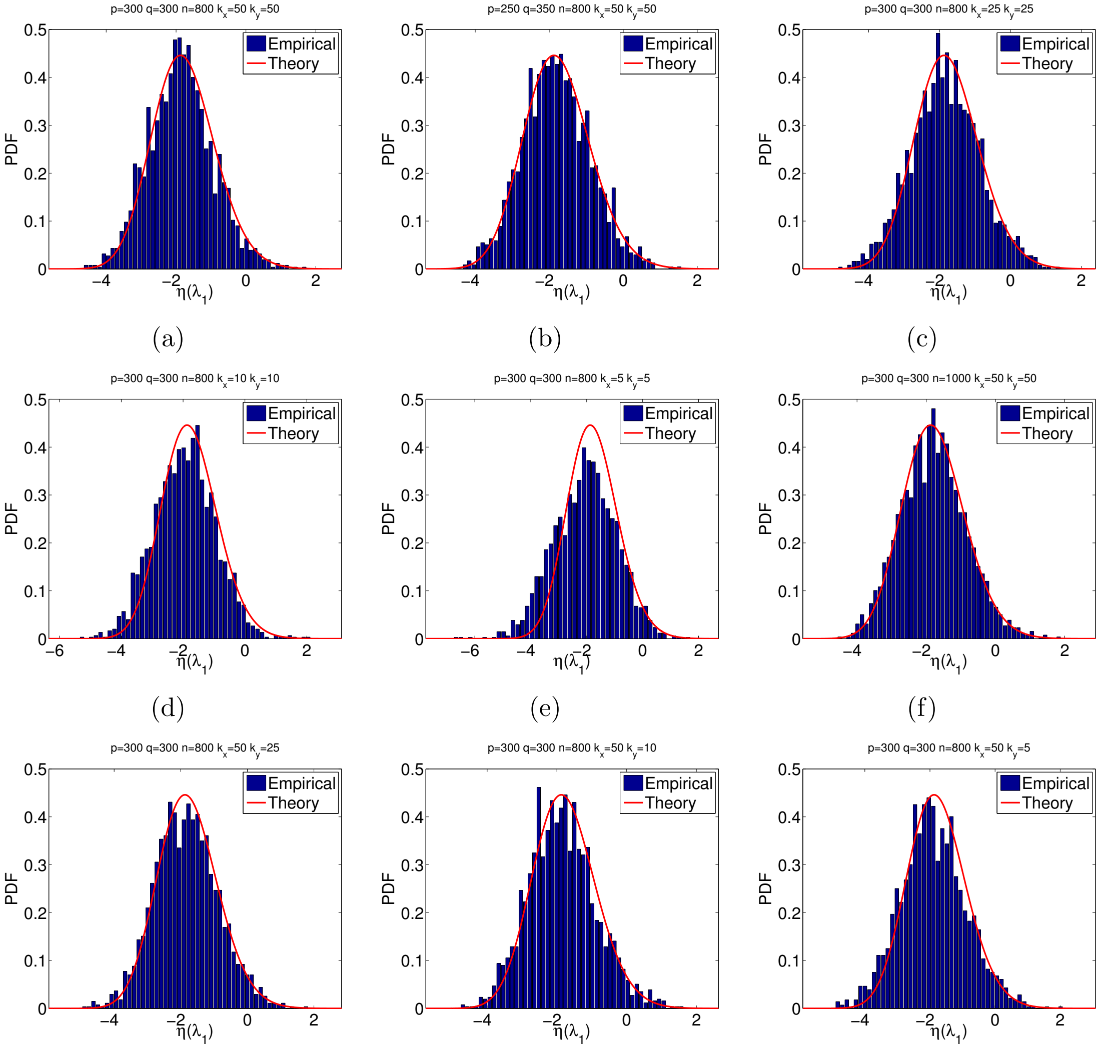


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure D.3: Empirical and theoretically predicted cumulative distribution functions (cdf) for ICCA under various parameters k_x, k_y and n for complex valued X and Y.

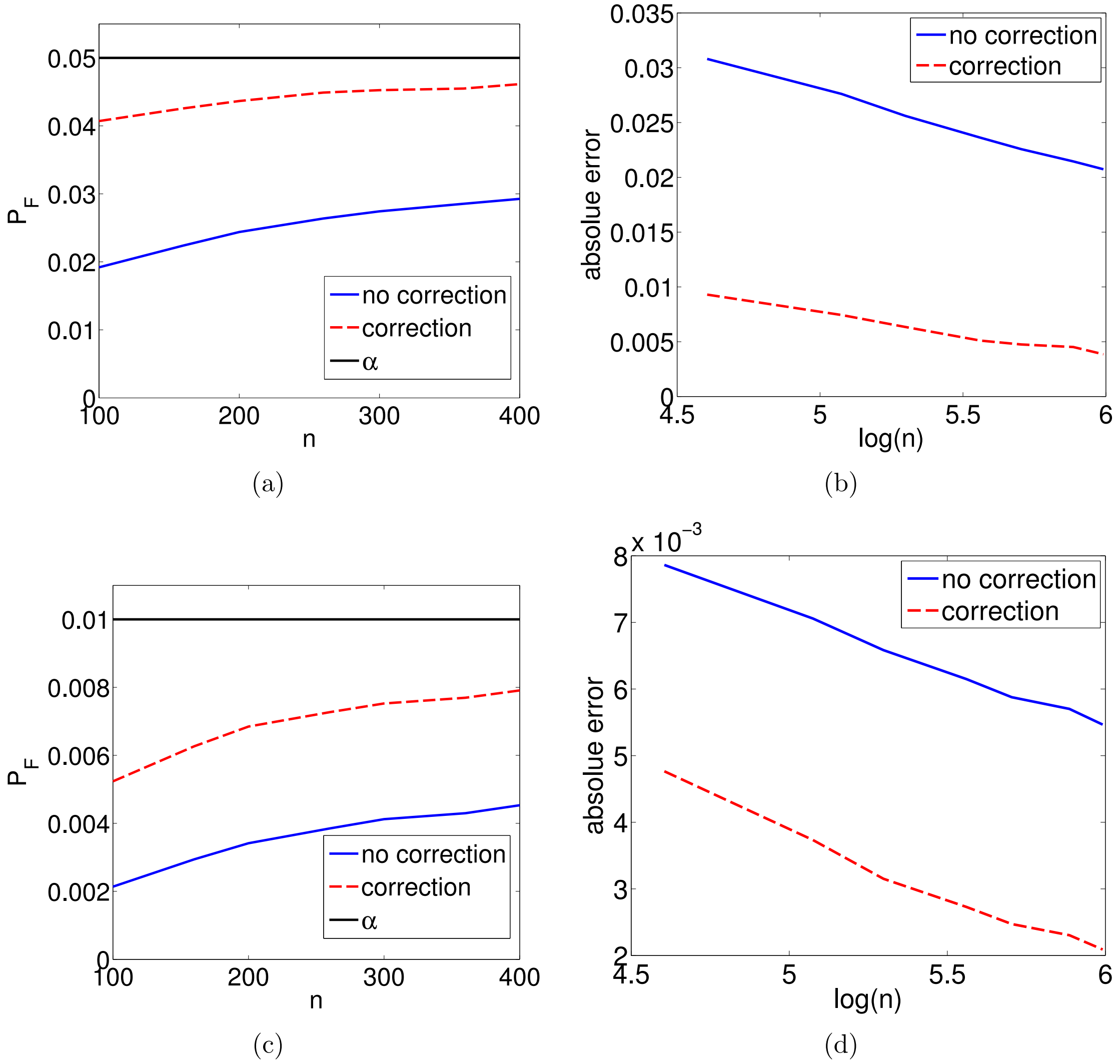


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure D.4: Empirical and theoretically predicted probability density functions (pdf) for ICCA under various parameters k_x, k_y and n for complex valued X and Y.



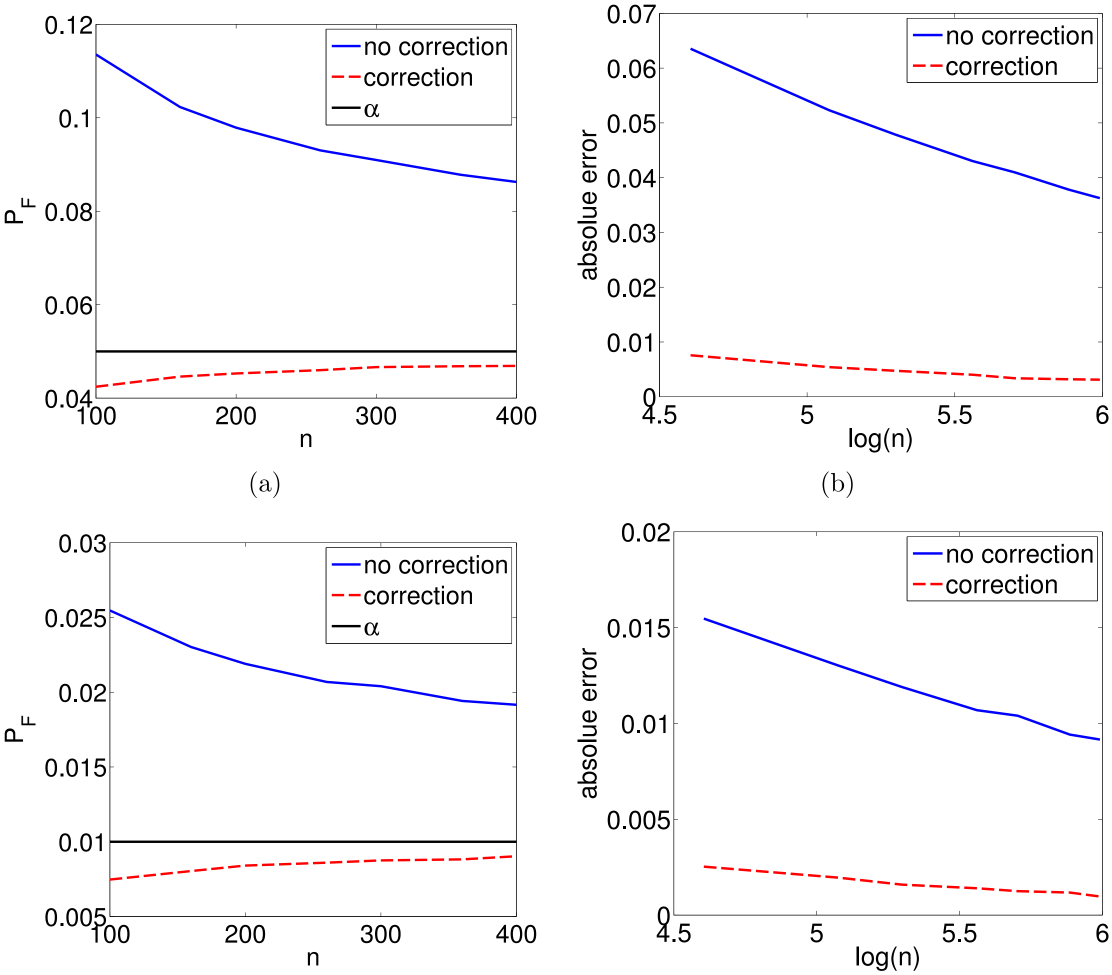
(a)

(b)

(c)

(d)

Figure D.5: Convergence plots for the false alarm rate of the proposed ICCA test statistic for real data. The false alarm rate is plotted as a function of n for fixed k_x/n, k_y/n. The black line shows the desired false alarm rate. The absolute error is also plotted. We show plots for $\alpha = 0.05$ and $\alpha = 0.01$. We show convergence plots when using the test statistic with and without the correction term.



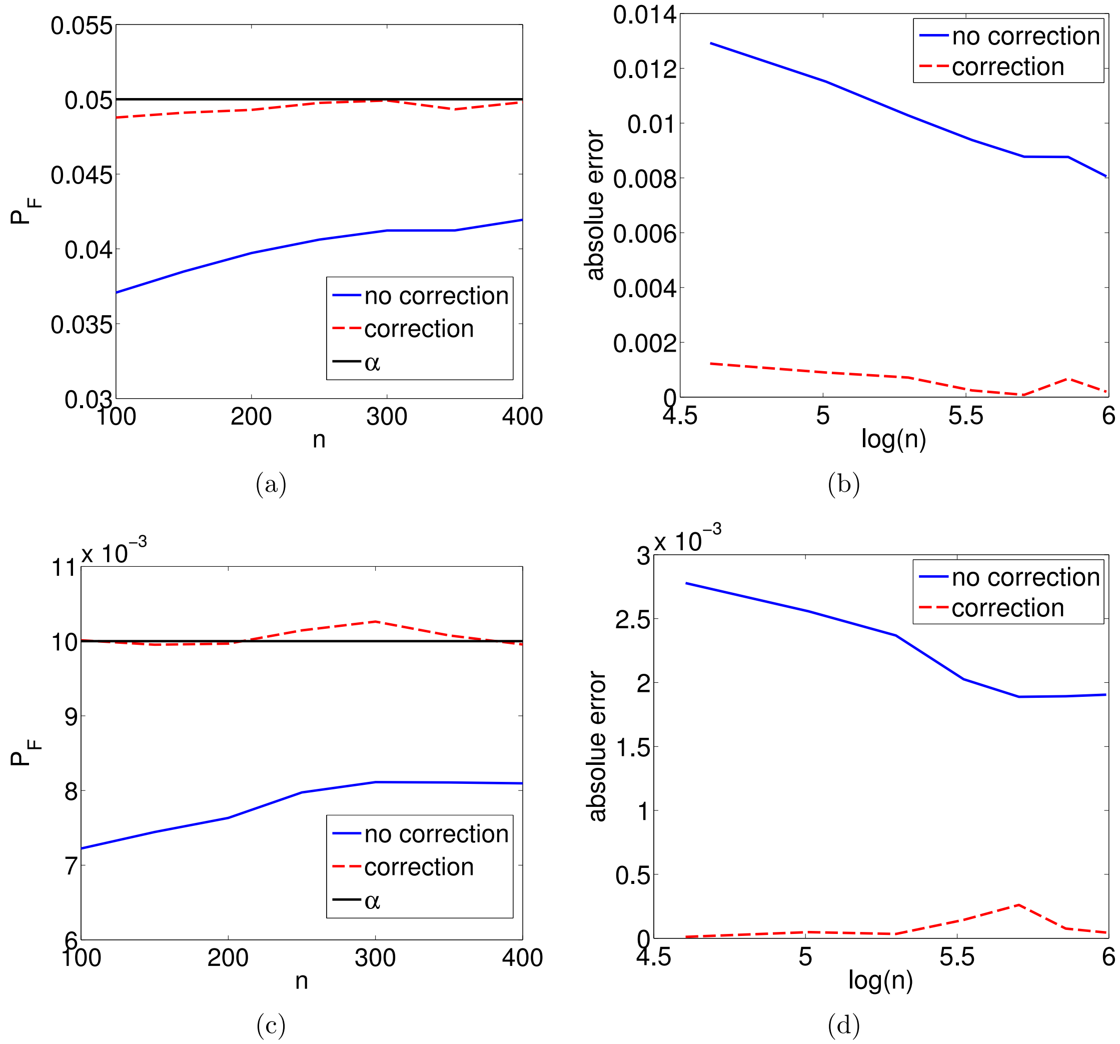
(a)

(b)

(c)

(d)

Figure D.6: Convergence plots for the false alarm rate of the proposed ICCA test statistic for complex data. The false alarm rate is plotted as a function of n for fixed k_x/n, k_y/n. The black line shows the desired false alarm rate. The absolute error is also plotted. We show plots for $\alpha = 0.05$ and $\alpha = 0.01$. We show convergence plots when using the test statistic with and without the correction term.



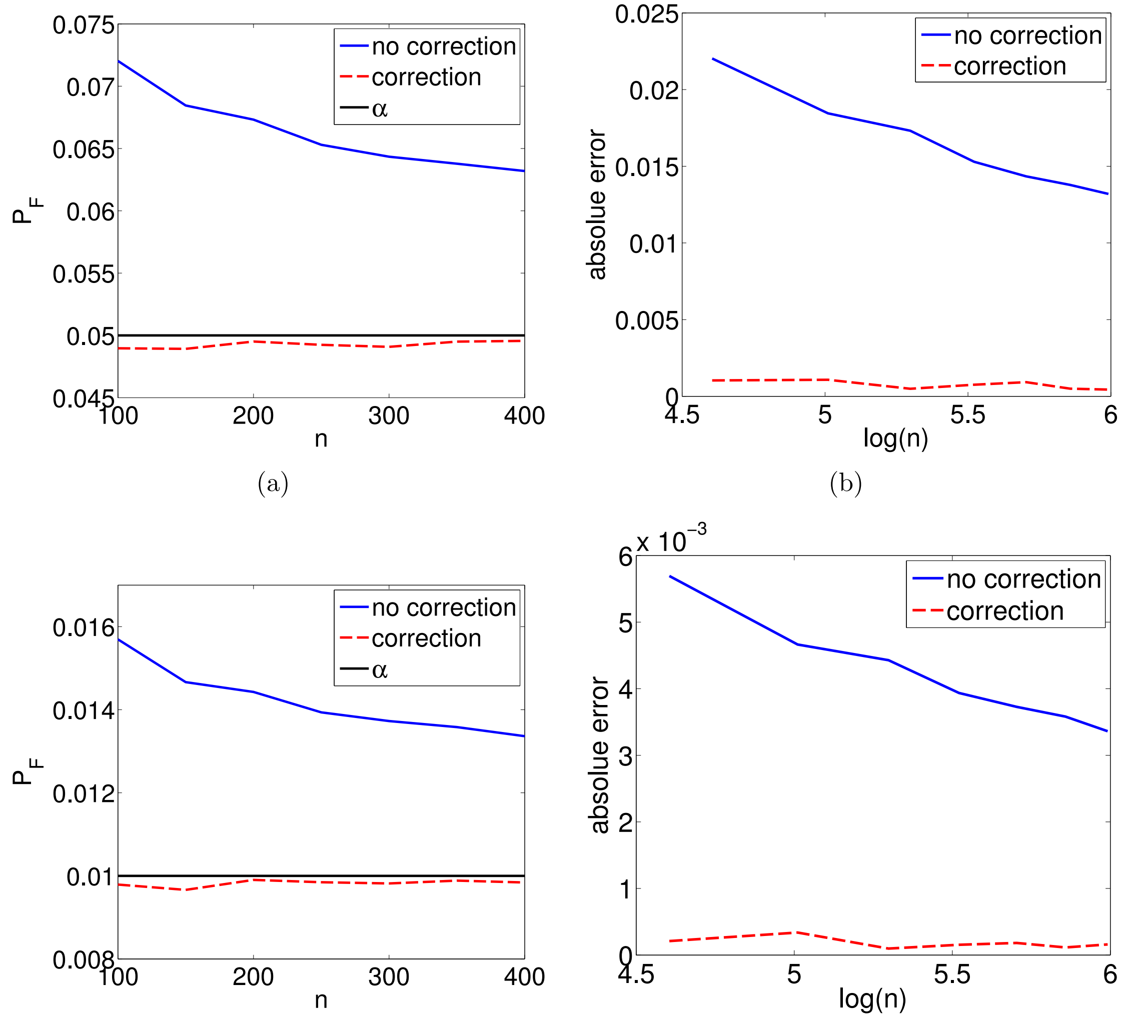
(a)

(b)

(c)

(d)

Figure D.7: Convergence plots for the false alarm rate of the PCA test statistic for real data. The false alarm rate is plotted as a function of n for fixed p/n. The black line shows the desired false alarm rate. The absolute error is also plotted. We show plots for $\alpha = 0.05$ and $\alpha = 0.01$. We show convergence plots when using the test statistic with and without the correction term.



(a)

(b)

(c)

(d)

Figure D.8: Convergence plots for the false alarm rate of the PCA test statistic for complex data. The false alarm rate is plotted as a function of n for fixed p/n. The black line shows the desired false alarm rate. The absolute error is also plotted. We show plots for $\alpha = 0.05$ and $\alpha = 0.01$. We show convergence plots when using the test statistic with and without the correction term.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[2] Z. Bao, J. Hu, G. Pan, and W. Zhou, “Canonical correlation coefficients of high-dimensional normal vectors: finite rank case,” *arXiv preprint arXiv:1407.7194*, 2014.

[3] N. R. Rao and A. Edelman, “The polynomial method for random matrices,” *Foundations of Computational Mathematics*, vol. 8, no. 6, pp. 649–702, 2008.

[4] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[5] B. Gunderson and R. Muirhead, “On estimating the dimensionality in canonical correlation analysis,” *Journal of Multivariate Analysis*, vol. 62, no. 1, pp. 121–136, 1997.

[6] A. Pezeshki, L. Scharf, M. Azimi-Sadjadi, and M. Lundberg, “Empirical canonical correlation analysis in subspaces,” in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 1. IEEE, 2004, pp. 994–997.

[7] H. Ge, I. Kirsteins, and X. Wang, “Does canonical correlation analysis provide reliable information on data correlation in array processing?” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on.* IEEE, 2009, pp. 2113–2116.

[8] R. Nadakuditi, “Fundamental finite-sample limit of canonical correlation analysis based detection of correlated high-dimensional signals in white noise,” in *Statistical Signal Processing Workshop (SSP), 2011 IEEE.* IEEE, 2011, pp. 397–400.

[9] H. D. Vinod, “Canonical ridge and econometrics of joint production,” *Journal of Econometrics*, vol. 4, no. 2, pp. 147–166, 1976.

[10] A. Thum, S. Mönchgesang, L. Westphal, T. Lübken, S. Rosahl, S. Neumann, and S. Posch, “Supervised penalized canonical correlation analysis,” *arXiv preprint arXiv:1405.1534*, 2014.

[11] R. Cruz-Cano and M.-L. T. Lee, "Fast regularized canonical correlation analysis," *Computational Statistics & Data Analysis*, vol. 70, pp. 88–100, 2014.

[12] A. Tenenhaus and M. Tenenhaus, "Regularized vgeneralized canonical correlation analysis for multiblock or multigroup data analysis," *European Journal of Operational Research*, vol. 238, no. 2, pp. 391–403, 2014.

[13] S. Akaho, "A kernel method for canonical correlation analysis," *arXiv preprint cs/0609071*, 2006.

[14] M. Welling, "Kernel canonical correlation analysis," 2005.

[15] S. Waaijenborg and A. H. Zwinderman, "Correlating multiple snps and multiple disease phenotypes: Penalized nonlinear canonical correlation analysis," *Bioinformatics*, p. btp491, 2009.

[16] A. Mandal and A. Cichocki, "Non-linear canonical correlation analysis using alpha-beta divergence," *Entropy*, vol. 15, no. 7, pp. 2788–2804, 2013.

[17] B. Chang, U. Kruger, R. Kustra, and J. Zhang, "Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 316–324.

[18] D. R. Hardoon and J. Shawe-Taylor, "Sparse canonical correlation analysis," *Machine Learning*, vol. 83, no. 3, pp. 331–353, 2011.

[19] J. Yan, H. Zhang, L. Du, E. Wernert, A. J. Saykin, and L. Shen, "Accelerating sparse canonical correlation analysis for large brain imaging genetics data," in *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment.* ACM, 2014, p. 4.

[20] J. Sun and S. Keates, "Canonical correlation analysis on data with censoring and error information," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 12, pp. 1909–1919, 2013.

[21] H. Shin and S. Lee, "Canonical correlation analysis for irregularly and sparsely observed functional data," *Journal of Multivariate Analysis*, vol. 134, pp. 1–18, 2015.

[22] L. Tao, H. Ip, Y. Wang, and X. Shu, "Exploring shared subspace and joint sparsity for canonical correlation analysis," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management.* ACM, 2014, pp. 1887–1890.

[23] C. Gao, Z. Ma, and H. H. Zhou, "An efficient and optimal method for sparse canonical correlation analysis," *arXiv preprint arXiv:1409.8565*, 2014.

[24] Z. Zhang, M. Zhao, and T. W. Chow, “Binary-and multi-class group sparse canonical correlation analysis for feature extraction and classification,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 10, pp. 2192–2205, 2013.

[25] D. M. Witten, R. Tibshirani, and T. Hastie, “A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis,” *Biostatistics*, p. kxp008, 2009.

[26] A. Klami, S. Virtanen, and S. Kaski, “Bayesian canonical correlation analysis,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 965–1003, 2013.

[27] D. Chu, L.-Z. Liao, M. K. Ng, and X. Zhang, “Sparse canonical correlation analysis: new formulation and algorithm,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 12, pp. 3050–3065, 2013.

[28] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[29] P. Dhillon, D. P. Foster, and L. H. Ungar, “Multi-view learning of word embeddings via cca,” in *Advances in Neural Information Processing Systems*, 2011, pp. 199–207.

[30] T. Melzer, M. Reiter, and H. Bischof, “Nonlinear feature extraction using generalized canonical correlation analysis,” *Artificial Neural NetworksICANN 2001*, pp. 353–360, 2001.

[31] D. Zhai, Y. Zhang, D.-Y. Yeung, H. Chang, X. Chen, and W. Gao, “Instance-specific canonical correlation analysis,” *Neurocomputing*, 2015.

[32] G. Lisanti, I. Masi, and A. Del Bimbo, “Matching people across camera views using kernel canonical correlation analysis,” in *Proceedings of the International Conference on Distributed Smart Cameras.* ACM, 2014, p. 10.

[33] U. Ahsan and I. Essa, “Clustering social event images using kernel canonical correlation analysis,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on.* IEEE, 2014, pp. 814–819.

[34] D. R. Hardoon, C. Saunders, S. Szedmak, and J. Shawe-Taylor, “A correlation approach for automatic image annotation,” in *Advanced Data Mining and Applications.* Springer, 2006, pp. 681–692.

[35] K. Chaudhuri, S. Kakade, K. Livescu, and K. Sridharan, “Multi-view clustering via canonical correlation analysis,” in *Proceedings of the 26th annual international conference on machine learning.* ACM, 2009, pp. 129–136.

[36] F. Deleus and M. Van Hulle, "Functional connectivity analysis of fmri data based on regularized multiset canonical correlation analysis," *Journal of Neuroscience Methods*, 2011.

[37] M. R. Arbabshirani, M. Nakhkash, and H. Soltanian-Zadeh, "Comparison of canonical correlation analysis and ica techniques for fmri," in *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on.* IEEE, 2010, pp. 1–5.

[38] M. U. Khalid and A.-K. Seghouane, "Improving functional connectivity detection in fmri by combining sparse dictionary learning and canonical correlation analysis," in *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on.* IEEE, 2013, pp. 286–289.

[39] P. Guccione, L. Mascolo, G. Nico, P. Taurisano, G. Blasi, L. Fazio, and A. Bertolino, "Functional brain networks and schizophrenia analysis with fmri by multiset canonical correlation analysis," in *Advances in Biomedical Engineering (ICABME), 2013 2nd International Conference on.* IEEE, 2013, pp. 207–210.

[40] N. Correa, T. Adali, Y. Li, and V. Calhoun, "Canonical correlation analysis for data fusion and group inferences," *Signal Processing Magazine, IEEE*, vol. 27, no. 4, pp. 39–50, 2010.

[41] D. Lin, J. Zhang, J. Li, V. Calhoun, and Y.-P. Wang, "Identifying genetic connections with brain functions in schizophrenia using group sparse canonical correlation analysis," in *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on.* IEEE, 2013, pp. 278–281.

[42] J. A. Seoane, C. Campbell, I. N. Day, J. P. Casas, and T. R. Gaunt, "Canonical correlation analysis for gene-based pleiotropy discovery," *PLoS computational biology*, vol. 10, no. 10, p. e1003876, 2014.

[43] D. Lin, J. Zhang, J. Li, V. D. Calhoun, H.-W. Deng, and Y.-P. Wang, "Group sparse canonical correlation analysis for genomic data integration," *BMC bioinformatics*, vol. 14, no. 1, p. 245, 2013.

[44] Y. Zhang, G. Zhou, J. Jin, M. Wang, X. Wang, and A. Cichocki, "L1-regularized multiway canonical correlation analysis for ssvep-based bci," 2013.

[45] M. Nakanishi, Y. Wang, Y.-T. Wang, Y. Mitsukura, and T.-P. Jung, "Enhancing unsupervised canonical correlation analysis-based frequency detection of ssveps by incorporating background eeg," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE.* IEEE, 2014, pp. 3053–3056.

[46] Y. Zhang, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Frequency recognition in ssvep-based bci using multiset canonical correlation analysis," *International journal of neural systems*, vol. 24, no. 04, 2014.

[47] M. Spuler, A. Walter, W. Rosenstiel, and M. Bogdan, “Spatial filtering based on canonical correlation analysis for classification of evoked or event-related potentials in eeg data,” 2013.

[48] C. Campi, L. Parkkonen, R. Hari, and A. Hyvärinen, “Non-linear canonical correlation for joint analysis of meg signals from two subjects,” *Frontiers in neuroscience*, vol. 7, 2013.

[49] X. Chen, C. He, and H. Peng, “Removal of muscle artifacts from single-channel eeg based on ensemble empirical mode decomposition and multiset canonical correlation analysis,” *Journal of Applied Mathematics*, vol. 2014, 2014.

[50] J. Kuzilek, V. Kremen, and L. Lhotska, “Comparison of jade and canonical correlation analysis for ecg de-noising,” in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE.* IEEE, 2014, pp. 3857–3860.

[51] J. Vıa, I. Santamarıa, and J. Pérez, “Canonical correlation analysis (cca) algorithms for multiple data sets: Application to blind simo equalization,” in *Proc. of European Signal Processing Conference (EUSIPCO)*, 2005.

[52] A. Pezeshki, L. Scharf, J. Thomas, and B. Van Veen, “Canonical coordinates are the right coordinates for low-rank gauss–gauss detection and estimation,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 12, pp. 4817–4820, 2006.

[53] L. L. Scharf and J. K. Thomas, “Wiener filters in canonical coordinates for transform coding, filtering, and quantizing,” *Signal Processing, IEEE Transactions on*, vol. 46, no. 3, pp. 647–654, 1998.

[54] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, “Joint blind source separation by multiset canonical correlation analysis,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 10, pp. 3918–3929, 2009.

[55] A. Nielsen, “Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data,” *Image Processing, IEEE Transactions on*, vol. 11, no. 3, pp. 293–305, 2002.

[56] L. Scharf and C. Mullis, “Canonical coordinates and the geometry of inference, rate, and capacity,” *Signal Processing, IEEE Transactions on*, vol. 48, no. 3, pp. 824–831, 2000.

[57] J. Manco-Vásquez, S. V. Vaerenbergh, J. Vía, and I. Santamaría, “Kernel canonical correlation analysis for robust cooperative spectrum sensing in cognitive radio networks,” *Transactions on Emerging Telecommunications Technologies*, 2014.

[58] K. Todros and A. Hero, “Measure transformed canonical correlation analysis with application to financial data,” in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th.* IEEE, 2012, pp. 361–364.

[59] D. Torres, D. Turnbull, B. Sriperumbudur, L. Barrington, and G. Lanckriet, "Finding musically meaningful words by sparse cca," in *Neural Information Processing Systems (NIPS) Workshop on Music, the Brain and Cognition*, 2007.

[60] D. S. Wilks, "Probabilistic canonical correlation analysis forecasts, with application to tropical pacific sea-surface temperatures," *International Journal of Climatology*, vol. 34, no. 5, pp. 1405–1413, 2014.

[61] A. J. Prera, K. M. Grimsrud, J. A. Thacher, D. W. McCollum, and R. P. Berrens, "Using canonical correlation analysis to identify environmental attitude groups: Considerations for national forest planning in the southwestern us," *Environmental management*, vol. 54, no. 4, pp. 756–767, 2014.

[62] J. L. Steward, Z. Haddad, S. Hristova-Veleva, and T. Vukicevic, "Assimilating scatterometer observations of tropical cyclones into an ensemble kalman filter system with a robust observation operator based on canonical-correlation analysis," in *SPIE Asia Pacific Remote Sensing.* International Society for Optics and Photonics, 2014, pp. 926 507–926 507.

[63] J. LI, M. ZHANG, Z.-r. YUAN, J.-y. LI, Z.-k. ZHOU, J.-n. HE, X. GAO, J.-b. CHEN, and S.-z. XU, "Canonical correlation analysis of bovine growth traits, meat quality traits and carcass traits," *China Animal Husbandry & Veterinary Medicine*, vol. 6, p. 025, 2010.

[64] J. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[65] A. Nielsen, "Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data," *Science*, vol. 21, no. 4, pp. 555–567, 1994.

[66] L. L. Scharf and C. Demeure, *Statistical signal processing: detection, estimation, and time series analysis.* Addison-Wesley Publishing Company, 1991, vol. 1.

[67] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning.* Springer Series in Statistics, 2001, vol. 1.

[68] O. Besson, L. L. Scharf, and F. Vincent, "Matched direction detectors and estimators for array processing with subspace steering vector uncertainties," *Signal Processing, IEEE Transactions on*, vol. 53, no. 12, pp. 4453–4463, 2005.

[69] O. Besson and L. L. Scharf, "Cfar matched direction detector," *Signal Processing, IEEE Transactions on*, vol. 54, no. 7, pp. 2840–2844, 2006.

[70] F. Bandiera, A. De Maio, A. S. Greco, and G. Ricci, "Adaptive radar detection of distributed targets in homogeneous and partially homogeneous noise plus subspace interference," *Signal Processing, IEEE Transactions on*, vol. 55, no. 4, pp. 1223–1237, 2007.

[71] F. Bandiera, O. Besson, D. Orlando, G. Ricci, and L. L. Scharf, "Glrt-based direction detectors in homogeneous noise and subspace interference," *Signal Processing, IEEE Transactions on*, vol. 55, no. 6, pp. 2386–2394, 2007.

[72] E. Maris, "A resampling method for estimating the signal subspace of spatio-temporal eeg/meg data," *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 8, pp. 935–949, 2003.

[73] A. C. K. Soong and Z. J. Koles, "Principal-component localization of the sources of the background eeg," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 1, pp. 59–67, 1995.

[74] L. L. Scharf and B. Friedlander, "Matched subspace detectors," *Signal Processing, IEEE Trans. on*, vol. 42, no. 8, pp. 2146–2157, 1994.

[75] F. Vincent, O. Besson, and C. Richard, "Matched subspace detection with hypothesis dependent noise power," *Signal Processing, IEEE Transactions on*, vol. 56, no. 11, pp. 5713–5718, 2008.

[76] T. McWhorter and L. L. Scharf, *Matched subspace detectors for stochastic signals.* Defense Technical Information Center, 2003.

[77] Y. Jin and B. Friedlander, "A cfar adaptive subspace detector for second-order gaussian signals," *Signal Processing, IEEE Trans. on*, vol. 53, no. 3, pp. 871–884, 2005.

[78] L. Elden, "Matrix methods in data mining and pattern recognition," 2007.

[79] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, [Online; accessed 10-October-2012].

[80] B. Thai and G. Healey, "Invariant subpixel material detection in hyperspectral imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 40, no. 3, pp. 599–608, 2002.

[81] G. Healey and D. Slater, "Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 6, pp. 2706–2717, 1999.

[82] H. Kwon and N. M. Nasrabadi, "Kernel matched subspace detectors for hyperspectral target detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 2, pp. 178–194, 2006.

[83] R. Nadakuditi and A. Edelman, "Sample eigenvalue based detection of high-dimensional signals in white noise using relatively few samples," *Signal Processing, IEEE Trans. on*, vol. 56, no. 7, pp. 2625–2638, 2008.

[84] D. Paul, “Asymptotics of sample eigenstructure for a large dimensional spiked covariance model,” *Statistica Sinica*, vol. 17, no. 4, p. 1617, 2007.

[85] F. Benaych-Georges and R. Nadakuditi, “The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices,” *Adv. in Math.*, 2011.

[86] F. Benaych-Georges and R. R. Nadakuditi, “The singular values and vectors of low rank perturbations of large rectangular random matrices,” *Arxiv preprint arXiv:1103.2221*, 2011.

[87] N. Asendorf and R. R. Nadakuditi, “Improving and characterizing the performance of stochastic matched subspace detectors when using noisy estimated subspaces,” in *Proceedings of the Asilomar Conference of Signals and Systems*, November 2011.

[88] M. Zhu and A. Ghodsi, “Automatic dimensionality selection from the scree plot via the use of profile likelihood,” *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 918–930, 2006.

[89] R. Nadakuditi and J. Silverstein, “Fundamental limit of sample generalized eigenvalue based detection of signals in noise using relatively few signal-bearing and noise-only samples,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 3, pp. 468–480, 2010.

[90] I. M. Johnstone, “On the distribution of the largest eigenvalue in principal components analysis,” *The Ann. of stat.*, vol. 29, no. 2, pp. 295–327, 2001.

[91] N. El Karoui, “Tracy–widom limit for the largest eigenvalue of a large class of complex sample covariance matrices,” *The Ann. of Prob.*, vol. 35, no. 2, pp. 663–714, 2007.

[92] R. J. Muirhead, *Aspects of multivariate statistical theory.* Wiley Online Library, 1982, vol. 42.

[93] H. L. Van Trees, *Detection, estimation, and modulation theory: Detection, estimation, and linear modulation theory.* Wiley, 1968.

[94] H. Cox, “Resolving power and sensitivity to mismatch of optimum array processors,” *J. Acoust. Soc. Amer*, vol. 54, no. 3, pp. 771–785, 1973.

[95] A. T. A. Wood, J. G. Booth, and R. W. Butler, “Saddlepoint approximations to the cdf of some statistics with nonnormal limit distributions,” *Journal of the American Statistical Association*, pp. 680–686, 1993.

[96] G. Cui, A. DeMaio, and M. Piezzo, “Performance prediction of the incoherent radar detector for correlated generalized swerling-chi fluctuating targets,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 1, pp. 356–368, 2013.

[97] J. Arribas, C. Fernandez-Prades, and P. Closas, "Antenna array based gnss signal acquisition for interference mitigation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 1, pp. 223–243, 2013.

[98] A. Gorji, R. Tharmarasa, and T. Kirubarajan, "Widely separated mimo versus multistatic radars for target localization and tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 4, pp. 2179–2194, 2013.

[99] S. Zhou and H. Liu, "Space-partition-based target detection for distributed mimo radar," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 4, pp. 2717–2729, 2013.

[100] E. A. Santiago and M. Saquib, "Noise subspace-based iterative technique for direction finding," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 4, pp. 2281–2295, 2013.

[101] N. Hu, Z. Ye, X. Xu, and M. Bao, "Doa estimation for sparse array via sparse signal reconstruction," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 2, pp. 760–773, 2013.

[102] B. Liao and S.-C. Chan, "Direction-of-arrival estimation in subarrays-based linear sparse arrays with gain/phase uncertainties," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 4, pp. 2268–2280, 2013.

[103] Y. Chen, Y. Nijsure, C. Yuen, Y. H. Chew, Z. Ding, and S. Boussakta, "Adaptive distributed mimo radar waveform optimization based on mutual information," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 2, pp. 1374–1385, 2013.

[104] Y. Kwon, R. M. Narayanan, and M. Rangaswamy, "Multi-target detection using total correlation for noise radar systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 2, pp. 1251–1262, 2013.

[105] S. Sirianunpiboon, D. Howard, and D. Cochran, "Multiple-channel detection of signals having known rank," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2013.

[106] G. Vazquez-Vilar, D. Ramírez, R. López-Valcarce, J. Vía, and I. Santamaría, "Spatial rank estimation in cognitive radio networks with uncalibrated multiple antennas," in *Proceedings of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management.* ACM, 2011, p. 35.

[107] N. Asendorf and R. Nadakuditi, "The performance of a matched subspace detector that uses subspaces estimated from finite, noisy, training data," *Signal Processing, IEEE Trans. on*, vol. 61, no. 8, pp. 1972–1985, 2013.

[108] L. Balzano, B. Recht, and R. Nowak, "High-dimensional matched subspace detection when data are missing," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on.* IEEE, 2010, pp. 1638–1642.

[109] J. He, M. O. Ahmad, and M. Swamy, "Near-field localization of partially polarized sources with a cross-dipole array," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 2, pp. 857–870, 2013.

[110] J. J. Fuchs, "A robust matched detector," *Signal Processing, IEEE Transactions on*, vol. 55, no. 11, pp. 5133–5142, 2007.

[111] S. Yu, B. De Moor, and Y. Moreau, "Learning with heterogenous data sets by weighted multiple kernel canonical correlation analysis," in *Machine Learning for Signal Processing, 2007 IEEE Workshop on.* IEEE, 2007, pp. 81–86.

[112] W. Härdle and L. Simar, *Applied multivariate statistical analysis.* Springer Science & Business Media, 2007.

[113] M. S. Bartlett, "A note on the multiplying factors for various χ 2 approximations," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 296–298, 1954.

[114] A. Constantine and R. J. Muirhead, "Asymptotic expansions for distributions of latent roots in multivariate analysis," *Journal of Multivariate Analysis*, vol. 6, no. 3, pp. 369–391, 1976.

[115] I. M. Johnstone, "Multivariate analysis and jacobi ensembles: Largest eigenvalue, tracy–widom limits and rates of convergence," *Annals of statistics*, vol. 36, no. 6, p. 2638, 2008.

[116] F. Benaych-Georges and R. R. Nadakuditi, "The singular values and vectors of low rank perturbations of large rectangular random matrices," *Journal of Multivariate Analysis*, vol. 111, pp. 120–135, 2012.

[117] R. Nadakuditi, "Optshrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage," 2014.

[118] R. Latała, "Some estimates of norms of random matrices," *Proceedings of the American Mathematical Society*, vol. 133, no. 5, pp. 1273–1282, 2005.

[119] I. Wilms and C. Croux, "Sparse canonical correlation analysis from a predictive point of view," *FEB Research Report KBI_1320*, 2013.

[120] A. Singanamalli, H. Wang, G. Lee, N. Shih, M. Rosen, S. Master, J. Tomaszewski, M. Feldman, and A. Madabhushi, "Supervised multi-view canonical correlation analysis: fused multimodal prediction of disease diagnosis and prognosis," in *SPIE Medical Imaging.* International Society for Optics and Photonics, 2014, pp. 903 805–903 805.

[121] D. Lin, V. D. Calhoun, and Y.-P. Wang, "Correspondence between fmri and snp data by group sparse canonical correlation analysis," *Medical image analysis*, vol. 18, no. 6, pp. 891–902, 2014.

[122] S. F. dos Santos and H. S. Brandi, “A canonical correlation analysis of the relationship between sustainability and competitiveness,” *Clean Technologies and Environmental Policy*, pp. 1–12, 2014.

[123] C. L. Vilsaint, S. M. Aiyer, M. N. Wilson, D. S. Shaw, and T. J. Dishion, “The ecology of early childhood risk: a canonical correlation analysis of childrens adjustment, family, and community context in a high-risk sample,” *The journal of primary prevention*, vol. 34, no. 4, pp. 261–277, 2013.

[124] F. Travis and Y. Lagrosen, “Creativity and brain-functioning in product development engineers: A canonical correlation analysis,” *Creativity Research Journal*, vol. 26, no. 2, pp. 239–243, 2014.

[125] J. Chen, H. Gu, and W. Su, “A new method for joint dod and doa estimation in bistatic mimo radar,” *Signal Processing*, vol. 90, no. 2, pp. 714–718, 2010.

[126] J.-F. Gu, P. Wei, and H.-M. Tai, “2-d direction-of-arrival estimation of coherent signals using cross-correlation matrix,” *Signal Processing*, vol. 88, no. 1, pp. 75–85, 2008.

[127] J.-F. Gu and P. Wei, “Joint svd of two cross-correlation matrices to achieve automatic pairing in 2-d angle estimation problems,” *Antennas and Wireless Propagation Letters, IEEE*, vol. 6, pp. 553–556, 2007.

[128] S. Kikuchi, H. Tsuji, and A. Sano, “Pair-matching method for estimating 2-d angle of arrival with a cross-correlation matrix,” *Antennas and Wireless Propagation Letters, IEEE*, vol. 5, no. 1, pp. 35–40, 2006.

[129] K. I. Diamantaras and S.-Y. Kung, “Cross-correlation neural network models,” *Signal Processing, IEEE Transactions on*, vol. 42, no. 11, pp. 3218–3223, 1994.

[130] K. J. Worsley, J.-I. Chen, J. Lerch, and A. C. Evans, “Comparing functional connectivity via thresholding correlations and singular value decomposition,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, pp. 913–920, 2005.

[131] R. R. Nadakuditi, “Applied Stochastic Eigen-Analysis,” Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[132] M.-A. Belabbas and P. J. Wolfe, “Fast low-rank approximation for covariance matrices,” in *Computational Advances in Multi-Sensor Adaptive Processing, 2007. CAMPSAP 2007. 2nd IEEE International Workshop on.* IEEE, 2007, pp. 293–296.

[133] M. Gu and S. C. Eisenstat, “Efficient algorithms for computing a strong rank-revealing qr factorization,” *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.

[134] M. Rudelson and R. Vershynin, "Sampling from large matrices: An approach through geometric functional analysis," *Journal of the ACM (JACM)*, vol. 54, no. 4, p. 21, 2007.

[135] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation."

[136] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1100–1124, 2009.

[137] N. Halko, P.-G. Martinsson, Y. Shkolnisky, and M. Tygert, "An algorithm for the principal component analysis of large data sets," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2580–2594, 2011.

[138] D. Achlioptas and F. Mcsherry, "Fast computation of low-rank matrix approximations," *Journal of the ACM (JACM)*, vol. 54, no. 2, p. 9, 2007.

[139] S. Arora, E. Hazan, and S. Kale, "A fast random sampling algorithm for sparsifying matrices," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques.* Springer, 2006, pp. 272–279.

[140] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices," *Proceedings of the National Academy of Sciences*, vol. 104, no. 51, pp. 20 167–20 172, 2007.

[141] P. Ramachandra and M. Sartipi, "Compressive sensing based imaging via belief propagation," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on.* IEEE, 2011, pp. 254–256.

[142] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[143] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.

[144] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[145] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1349–1380, 2000.

[146] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, “Evaluating bag-of-visual-words representations in scene classification,” in *Proceedings of the international workshop on Workshop on multimedia information retrieval.* ACM, 2007, pp. 197–206.

[147] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[148] J. E. Solem, *Programming Computer Vision with Python.* O’Reilly Media, 2012.

[149] C. W. Leong, R. Mihalcea, and S. Hassan, “Text mining for automatic image tagging,” in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters.* Association for Computational Linguistics, 2010, pp. 647–655.

[150] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, “Collecting image annotations using amazon’s mechanical turk,” in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk.* Association for Computational Linguistics, 2010, pp. 139–147.

[151] Y. Feng and M. Lapata, “Automatic image annotation using auxiliary text information.” in *ACL*, 2008, pp. 272–280.

[152] H. Schmid, “Probabilistic part-of-speech tagging using decision trees,” in *Proceedings of the international conference on new methods in language processing*, vol. 12. Citeseer, 1994, pp. 44–49.

[153] B. Vinograde, “Canonical positive definite matrices under internal linear transformations,” *Proceedings of the American Mathematical Society*, vol. 1, no. 2, pp. 159–161, 1950.

[154] R. G. Steel, “Minimum generalized variance for a set of linear functions,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 456–460, 1951.

[155] P. Horst, “Relations among m sets of measures,” *Psychometrika*, vol. 26, no. 2, pp. 129–149, 1961.

[156] ——, “Generalized canonical correlations and their applications to experimental data,” *Journal of Clinical Psychology*, vol. 17, no. 4, pp. 331–347, 1961.

[157] F. Bach and M. Jordan, “Kernel independent component analysis,” *The Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2003.

[158] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “Manopt, a matlab toolbox for optimization on manifolds,” *arXiv preprint arXiv:1308.5200*, 2013.

[159] J. Silverstein, “The smallest eigenvalue of a large dimensional wishart matrix,” *The Annals of Probability*, vol. 13, no. 4, pp. 1364–1368, 1985.

[160] A. Barvinok, “Measure concentration lecture notes,” *See http://www. math. lsa. umich. edu/barvinok/total710. pdf*, 2005.