A Paucity of Data in Machine Learning: Applications in Single Cell RNA Sequencing and Ranking

by

Umang Varma

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mathematics)
in the University of Michigan
2019

Umang Varma

uvarma@umich.edu

ORCID iD: 0000-0003-1326-7970

ACKNOWLEDGMENTS

I am incredibly grateful for the support that my advisor, Anna C. Gilbert, has shown me over the last two years. Her faith in me as I dove headfirst into a field that was new to me has been invaluable.

I would also like to thank a number of other people who have helped me with the work in this dissertation. I have learned a lot from everyone at the Michigan Center for Single-Cell Genomic Data Analytics, especially from Jul Z. Li and Justin Colacino, who have patiently helped me understand the biological motivations behind single cell genomics research and they have been helpful collaborators for the work in Chapter 2. Lalit Jain has been a patient mentor and important collaborator for all the work in Chapter 3. The inspiration for studying the topics in Chapter 3 came from work I did with Michael Bell, who was a great mentor for the summer that I worked with him.

Audra McMillan helped me navigate my graduate school career from the beginning—her encouragement and advice helped me persist through the toughest parts. Alex Vargo and Amanda Bower have been good friends and allies who have overcome the challenges of this program together with me. Karen Smith has always been available to help me over the past five years and I am thankful for her support. I feel lucky to have worked with Mark Conger on his “Math and the Internet” MMSS course twice during my time at Michigan. He has trusted and supported me and I have learned a lot about my goals by talking with him. My college professors John Fink and Michele Intermont shared their love for math with me at Kalamazoo College and inspired me to pursue a Ph.D. in mathematics.

On a more personal note, I am very fortunate to have had the support of friends and family as I worked towards completing this program. I am especially grateful for my partner Nicole Buccalo and my friends Megan Davis and Stann-Omar Jones for always believing in me. I cherished my time with the incredible community at the Graduate Employees Organization (GEO) during my time at UM. I would also like to thank my parents and sister for deeply valuing my education and learning.

Finally, I would like to thank the University of Michigan Department of Mathematics, the Michigan Institute for Data Science, and the Chan Zuckerberg Initiative. Their funding has supported me through this work.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

ABSTRACT

A driving force behind the development of machine learning techniques is the availability of vast amounts of data that are continuously generated and the enormous potential of using these data. It is, however, not uncommon to have a paucity of data. This paucity can come in different forms: we may have an insufficient amount of data to perform the task at hand, we may have missing entries, or we may be working with aggregate statistics that only capture a fraction of the data we seek to study.

To extract meaningful conclusions from scarce data is challenging and requires creative approaches that account for the scarcity in their assumptions and/or finding other means to make up for insufficient data. It is not always possible to adapt the standard algorithm for a given problem and the challenge often lies in finding the right basis from which to build a viable solution. This thesis takes on two problems within this realm, where there is a paucity of data, and develops techniques to overcome the challenges such paucities can pose.

In single cell RNA-sequencing, entries of a gene expression matrix are counts of the number of molecules observed where only a small fraction of the molecules in the cell have been observed. This is particularly challenging because biologists are often concerned with whether a gene is expressed in a cell at all; however, a zero entry in a gene expression matrix only says that no such molecules were *seen* in the given cell, not that there *were* no such molecules in the given cell. In practice, a vast majority (often over 90%) of entries in gene expression matrices are zero. We focus on the problem of feature selection: we give theoretical guarantees for information-theoretic algorithms and address practical issues involved in their implementation.

The problem of ranking items from pairwise comparisons is also often constrained by a paucity of data—collecting pairwise comparisons made by humans can be expensive—and we propose ways to incorporate other data to overcome a small number of pairwise comparison observations. In addition to giving a better sample complexity bound for the RankCentrality algorithm with simpler proofs that use matrix concentration inequalities, we introduce λ-regularized RankCentrality (theoretical analysis for this regularization depends on our new proofs for RankCentrality), that is capable of giving non-trivial output even when the number of observations is small, and a similarity-based regularization that can use features of the items being ranked to significantly improve performance in the small-sample regime.

CHAPTER 1

Introduction

A driving force behind the development of machine learning techniques is the availability of vast amounts of data that are continuously generated and the enormous potential of using these data. It is, however, not uncommon to have insufficient, missing, or partially observed data. One of the most famous examples of such a problem is the "Netflix Prize" matrix completion problem [BL] [BKV], where users have rated some of the movies they have watched and ratings for about 1% of the user-movie pairs are known. The goal is to predict how users may rate movies that they have not yet watched. In this case, a movie recommendation system has to overcome missing data—where only a fraction of the entries in a matrix have been observed.

Scarcity of data can come in various forms. Unlike in the Netflix problem, where entries in a matrix were missing, single cell RNA-sequencing data suffer from a subtly different problem: entries in the resulting matrix are not "observed" or "not observed" but rather are counts of the number of molecules observed where only a small fraction of the molecules in each cell have been observed. This is particularly challenging because biologists are often concerned with whether a gene is expressed in a cell at all; however, a zero entry in a gene expression matrix only says that no such molecules were *seen* in the given cell, not that there *were* no such molecules in the given cell. In practice, a vast majority (often over 90%) of entries in gene expression matrices are zero. The problem of ranking items from pairwise comparisons also faces hurdles that arise from a paucity of data—getting humans to make pairwise comparisons can be expensive—and we propose ways to incorporate other data to overcome a small number of pairwise comparison observations.

To extract meaningful conclusions from scarce data is challenging and requires creative approaches that account for the scarcity in their assumptions and/or finding other means to make up for insufficient data. It is not always possible to adapt the standard algorithm for the given problem and the challenge often lies in finding the right basis from which to build a viable solution. This thesis takes on two problems within this realm, where there is a paucity of data, and develops techniques to overcome the challenges such paucities can pose.

Single cell RNA-sequencing is a field that has grown tremendously in recent years. It has allowed researchers to study the gene expression levels in cells at the resolution of individual cells, a vast improvement from bulk RNA sequencing which averaged expression levels across many cells (often on the order of 10^3 to 10^7 cells). New techniques allow researchers to process millions of cells and detect RNA molecules from across the transcriptome. This ability to analyze the heterogeneity cells even within a tissue gives biologists a much deeper insight into the workings of complex tissues that may have a large number of cell types, each performing different functions and producing different proteins in order to do so. Researchers anticipate significant breakthroughs in medicine as a result of the studies that biologists are now able to conduct using this technology. There is, however, a caveat that comes with this newfound resolution in sequencing. Due to technical reasons, the "read depth" of these techniques tends to be very small and this is particularly true when we want to be able to sequence larger number of cells. The "read depth" is the fraction of RNA molecules in a cell that we expect to actually detect. In typical experiments, we usually detect between 5% to 15% of the RNA molecules in a cell.

A problem that we focus on in this field is that of marker selection—the process of selecting a small number of genes that can distinguish between cells when only expression levels of those genes are available. In machine learning, this is called "feature selection", choosing a small but informative subset of the available features, not to be confused with "feature extraction" and other forms of dimensionality reduction that create *new* features from existing features.

In this thesis, we develop and analyze supervised and unsupervised methods that are able to be agnostic to underlying distributions of observations. This agnositicity is important especially because there is a lack of consensus on what distributions best model the scarcity of reads and a vast majority of the existing algorithms are generative models that make strong assumptions on the distribution and fit parameters according to those assumptions.

The problem of ranking items from pairwise comparisons is also often constrained by a paucity of data. For example, we may be interested ranking a large number of text documents in terms certain properties (for example, news articles ranked by how worrisome the news might be, social media posts by the perceived age of the author), but the time required to collect sufficiently many pairwise comparisons might be prohibitive.

Because standards for scoring such items are extremely prone to fluctuations—both between the human evaluators and also over time as an evaluator observes the various items. Such scoring systems require strict and detailed rules and rubrics. On the other hand, asking humans which of two items should be ranked higher gives far more consistent results, both between different evaluators and over time.

This problem is distinct from that of sorting because these pairwise comparisons are usually not deterministic and not always consistent. There are various models that can be used to describe the randomness of such comparisons, usually by making assumptions about P_{ij}, the probability that j is preferred to i in comparison of i and j. Some models make very weak assumptions about such a matrix, for example, the low-noise model [RA14] assumes that for all $i, j \in \{1, \ldots, n\} =: [n]$, we have

$$P_{ij} > P_{ji} \implies \sum_{k=1}^{n} P_{kj} > \sum_{k=1}^{n} P_{ki}.$$

Note that this does not require that $P_{kj} > P_{ki}$ for all k. On the other hand, a model that makes stronger assumptions, and one that we will use repeatedly in this thesis, is the Bradley-Terry-Luce (BTL) model that assumes that these probabilities can be modeled by parameters w_i that we call the BTL score of item $i \in [n]$. Under the BTL model,

$$P_{ij} = \frac{w_j}{w_i + w_j},$$

and it is easy to check that this is strictly stronger than the assumptions of the low-noise model.

There are various algorithms in the literature for learning such a ranking from pairwise comparisons, each for a model under which we can give theoretical guarantees. Of these algorithms, we discuss one, RankCentrality, in detail. We focus on RankCentrality because of its ability to be regularized in novel and creative ways that allow us to incorporate other forms of data into the ranking process. To this end, we also address a major issue with RankCentrality that prevents it from giving any meaningful results when an insufficient number of pairwise comparisons have been made. We propose λ-regularized RankCentrality to ensure the underlying empirical Markov chain is irreducible even the sample size is small, which is not true of the original algorithm. We build on top of λ-regularized RankCentrality, which does not take into account any other information, a similarity based regularization for RankCentrality that can use similarity information (for example, from features of the n items being compared) to impute pairwise comparisons which yields significantly better estimators, especially when the training dataset is small.

CHAPTER 2

Information Theoretic Feature Selection for scRNA-seq

2.1 Background

Single cell RNA-sequencing (scRNA-seq) technologies have generated an expansive amount of new biological information, revealing new cellular populations and hierarchical relationships. These methods quantify expression of thousands of genes across tens or hundreds of thousands of individual cells derived from a single cell suspension [Gie+17; Mac+15]. Using unbiased methods for cell clustering, researchers define cell types based on these gene expression patterns [DRS18]. When information derived from these transcriptionally-defined cell clusters is integrated with spatial or protein level data, additional biological insights can be gleaned. A number of technologies complementary to scRNA-seq rely on the selection of a smaller number of marker genes to accurately differentiate cell types within a complex mixture of cells. For example, fluorescence activated cell sorting (FACS) can isolate specific cell populations from a single cell suspension for further in depth biological characterization based on the expression of a distinguishing set of a small number of cell surface markers. Mass cytometry provides data on expression of up to 40 proteins for cells in a single cell suspension, adapting traditional flow cytometry methods by labeling antibodies with heavy metal isotopes rather than fluorophores and using a time-of-flight mass spectrometry readout [Ben+11]. Imaging mass cytometry further allows for in situ assessment of approximately 40 markers in tissue sections, quantifying the spatial distribution of marker expression in the tissue [Gie+14]. Finally, spatial transcriptomic profiling approaches use fluorescence in situ hybridization in tissue sections for imaging based quantification of gene expression [Raj+08; Eng+19]. Optimal usage of each of these techniques requires selection of a small set of marker genes (or features) that can accurately assign an individual cell to a known cell type based on expression.

Feature selection is a challenging combinatorial problem; it is already known that feature selection is NP-hard [AK98]. In the field of scRNA-seq, the most commonly used techniques for marker gene selection data are based on highly variable gene identification [Sat+15; Zhe+17] and differential expression/logistic regression [Ntr+18], the former being unsupervised and the later supervised. Most of these methods are univariate: they consider properties of each feature individually and do not consider any interactions between features. Additionally, these methods make strong assumptions about how gene expression is distributed. They typically measure the mean and standard deviation of features and use these in their determination of variability (or differential expression). Such techniques, especially ones that output probability values, are implicitly or explicitly making assumptions about the underlying distributions (e.g., Gaussian, or Student's t-distribution). Even when the data suggest that these assumptions are reasonable, it is usually only the case for individual features in isolation. Supervised feature selection methods used in scRNA-seq such as differential expression can natively handle binary class labels and need strategies such as one-vs-rest (OvR) or one-vs-one (OvO) to transform the problem back into a binary classification problem. This transformation causes two additional problems. First, by pooling clusters together (e.g., with OvR), we blur out valuable information by artificially suppressing genes that are informative about multiple class labels and boosting genes that are differentially expressed in one class when compared to all other classes. This puts an unnecessary importance on choosing class labels with the correct resolution. Secondly, methods to combining the lists of features identified in each pairwise comparison (either OvO or OvR) tend to be arbitrary and there is no canonical approach to reconciling the many ($\binom{k}{2}$ or $k-1$ in OvO and OvR respectively, where k is the number of class labels) lists of features into one list.

In contrast to differential expression based methods for marker selection, information-theoretic methods address a number of these issues. Rather than considering properties of each marker individually, information-theoretic algorithms consider interactions between features to various degrees. Specifically, the "degree K" algorithm considers interactions between K features. For example, the degree 1 algorithm is a univariate algorithm, much like the differential expression methods, and the degree 2 algorithm considers interactions between pairs of features, etc. Additionally, in contrast to differential expression methods, information-theoretic methods can run as native multiclass methods as well as binary methods.

Our goal is to benchmark the efficacy of differential expression and information-theoretic methods for marker selection in scRNA-seq experiments, quantifying the strengths and weaknesses of a range of objective functions. We then focus on the degree 2 approximation, which corresponds to the conditional infomax feature extraction (CIFE) feature selection algorithm [LT06], and give theoretical guarantees for its performance. We then evaluate the performance of these algorithms on publicly available scRNA-seq datasets and describe optimizations for sparse data that improve the computational efficiency of these algorithms considerably.

2.2 Introduction to Information Theoretic Methods

We seek to find an small subset of genes that is most informative about the class labels. In the language of information theory, we can frame this as the problem finding a small subset S of the variables $\mathcal{X}$ (the set of all genes) such that $I(S; y)$, the mutual information between a set of features S and target labels y, is maximized. This warrants some definitions before we proceed.

2.2.1 Definitions, Notation, and Abuses Thereof

We begin with the definition of mutual information. For a more extensive resource on information theory, we recommend [CT06]. The mutual information between random variables U and V can be thought of as the number of bits of information knowing U tells us about V. For example, if U is a outcome of a fair coin toss (i.e., U takes values in $\{H, T\}$ with equal probability) and V is the opposite of U (i.e., V is T if U is H and vice-versa), then their mutual information is 1 bit. However, if V is a "noisy" opposite of U so that V is the opposite of U with probability 0.99 and is equal to U with probability 0.01 then $I(U; V)$ should be just under 1 (in fact it is approximately 0.92).

Definition. If U and V are discrete random variables, then the mutual information $I(U; V)$ between U and V is

$$\begin{aligned} I(U;V) &= \sum_u \sum_v P_{U,V}(u,v) \log\left(\frac{P_{U,V}(u,v)}{P_U(u)P_V(v)}\right) \\ &= H(U) + H(V) - H(\{U,V\}), \end{aligned}$$

where

- P_X is the probability mass function of X, a random variable, in particular $P_{U,V}(u,v) = P(U = u \text{ and } V = v)$,
- $H(\cdot)$ is the Shannon entropy of a discrete random variable, and
- the above logarithms are with base 2, by convention.

We will only consider the case when U and V are discrete random variables (in practice, such algorithms are used after data has been discretized, although continuous analogues that use density estimators exist, but are computationally harder).

By abuse of notation and in line with existing literature, we will often

- treat a set of random variables as an ensemble of random variables. That is,

$$H(\{X_1, \ldots, X_n\}) = -\sum_{x_1,\ldots,x_n} p(x_1, \ldots, x_n) \log p(x_1, \ldots, x_n),$$

- drop the $\cup$ symbol and write $H(UV)$ to mean $H(U \cup V)$, and
- treat a feature x_i as a singleton $\{x_i\}$, so $H(Ux_i) = H(U \cup \{x_i\})$.

2.2.2 Information-theoretic Feature Selection

We may now return to making the statement of our problem precise. We want to find $S \subset \mathcal{X}$ such that $I(S; y)$ is maximum over subsets of size k or less of $\mathcal{X}$. Here k is a user-prescribed number of features to select. There are $\binom{P}{k}$ such subsets (for fixed k, this is $O(P^k)$). Datasets often have many thousands of features from which we may seek to select on the order of a hundred features, which makes an exhaustive search computations intractable. It is already known that feature selection is NP-hard (see [AK98]), as are sub-modular maximization problems for coverage-type functions (see [Von07]) that our problem can be reduced to by constructions similar to those discussed in Section 2.3. Various greedy algorithms, usually of the form of Algorithm 1, have been studied in the literature. We give a summary of these algorithms in Table 2.1. These algorithms use different objective functions $J : \mathcal{X} \to \mathbb{R}$, where $J(x_i)$ estimates the (relative) value of feature x_i given that $S \subset \mathcal{X}$ has already been chosen.

Algorithm 1 General Greedy Algorithm for Feature Selection with Objective J

```
1: procedure GreedySearch($\mathcal{X}, y, n$)
2:     $S \leftarrow \emptyset$
3:     while $|S| < n$ do
4:         $S \leftarrow S \cup \left\{ \arg\max_{x_i \in \mathcal{X} - S} J(x_i) \right\}$
5:     end while
6:     return $S$
7: end procedure
```

If we were to naïvely maximize $I(S; y)$ via a greedy search, we would run Algorithm 1 with $J_{\text{cmi}}(x_i) = I(S \cup \{x_i\}; y)$. Computing this requires estimating the joint probability distribution of Sx_iy (which has $|S| + 2$ dimensions). This is not only computationally intractable, but also impossible to estimate reasonably in the absence of a very large number of samples. Therefore, the objective function

$$J_{\text{cmi}}(x_i) = I(S \cup \{x_i\}; y) \tag{2.1}$$

is difficult to estimate. We follow the terminology of Brown, et al. [Bro+12] and will refer to this as the conditional mutual information (CMI) objective because a greedy search with the conditional mutual information objective, $J(x_i) = I(x_i; y|S)$, is equivalent to using J_{cmi} from (2.1). This is because $I(x_i; y|S) - I(Sx_i; y)$ is constant with respect to x_i.

In this work, we present a series expansion of $I(S \cup \{x_i\}; y)$ in terms of multivariate mutual information through which we can derive degree K approximations to $I(S \cup \{x_i\}; y)$. This expansion gives us insight into the strengths and weaknesses of various objective functions in the literature. We then focus on the degree 2 approximation, which corresponds to the CIFE feature selection algorithm [LT06], and give new theoretical guarantees for its performance. We then evaluate the performance of these algorithms on publicly available scRNA-seq datasets and describe optimizations for sparse data that improve the computational efficiency of these algorithms considerably.

2.2.3 Previous work

Because of the computational challenges in empirically evaluating J_{cmi}, many approximation algorithms have been proposed. Brown et al. [Bro+12] generalize a vast number algorithms in the literature as

$$J(x_i) = I(x_i; y) - \beta \sum_{j \in S} I(x_i; x_j) + \gamma \sum_{j \in S} I(x_i; x_j|y). \tag{2.2}$$

Specifically, the widely used mRMR algorithm in [PLD05] has $\beta = \frac{1}{|S|}$ and $\gamma = 0$. The MIFS algorithm by [Bat94] has $\gamma = 0$ and leaves β as a (constant) hyperparameter chosen from $[0, 1]$. The CIFE algorithm in [LT06] has $\beta = \gamma = 1$. The JMI algorithm ([YM00]) and IGFS[1] algorithm ([EEA08]) use $\beta = \gamma = \frac{1}{|S|}$. See Table 2.1 for a summary of the various algorithms described.

Brown et al. provide an interpretation of this class of objective functions in terms of three assumptions.

1. Given an unselected feature x_k,
   (a) the selected features are independent: $p(S|x_k) = \prod_{x_j \in S} p(x_j|x_k)$ and
   (b) the selected features are class-conditionally independent: $p(S|yx_k) = \prod_{x_j \in S} p(x_j|yx_k)$.
2. Features are pairwise class-conditionally independent: $p(x_i x_j|y) = p(x_i|y)p(x_j|y)$.
3. Features are pairwise independent: $p(x_i x_j) = p(x_i)p(x_j)$.

[1]Brown et al. incorrectly claim IGFS is equilvalent to CIFE, it is, in fact, equivalent to JMI.

Objective function	**Description**
$J_{\text{cmi}}(x_i) = I(S \cup \{x_i\}; y)$	This is the objective we would ideally like to optimize against, but the cost of computing its empirical value is prohibitive.
$J_{\text{mim}}(x_i) = I(x_i; y)$	The Mutual Information Maximization (MIM) is a univariate feature selection method—it does not consider the interactions between variables and only considers features in isolation.
$J_{\text{mrmr}}(x_i) = I(x_i; y) - \frac{1}{\lvert S\rvert} \sum_{x_j \in S} I(x_i; x_j)$	The minimum Redundancy Maximum Relevance (mRMR) algorithm proposed by [PLD05] uses a diminishing penalty on redundancy and does not distinguish between relevant and irrelevant redundancy (see Example 2).
$J_{\text{mifs}}(x_i) = I(x_i; y) - \beta \sum_{x_j \in S} I(x_i; x_j)$	The Mutual Information Feature Selection (MIFS) algorithm proposed by [PLD05] uses a diminishing penalty on redundancy and does not distinguish between relevant and irrelevant redundancy.
$J_{\text{jmi}}(x_i) = I(x_i; y) - \frac{1}{\lvert S\rvert} \sum_{x_j \in S} I(x_i; x_j; y)$	The Joint Mutual Information (JMI) algorithm proposed by [YM00], like the mRMR algorithm, uses a diminishing penalty on redundancy. However, it only penalizes relevant redundancy which, we explain in Section 2.3, is desirable.
$J_{\text{cife}}(x_i) = I(x_i; y) - \sum_{x_j \in S} I(x_i; x_j; y)$	The (CIFE) algorithm proposed by [LT06] does not diminish its redundancy penalty (which it applies only to relevant redundancy, as desired). Although it can penalize redundancy somewhat aggressively, we will show that it it the degree 2 approximation to J_{cmi} and, under certain conditions, is a submodular set function. We focus on these properties of the CIFE objective to give theoretical guarantees on its performance.

Table 2.1: Summary of various objective functions that are used with Algorithm 1.

2.2.4 Our Work

We attempt to address two sets of important questions about the information theoretic methods that we discuss in this chapter:

- What are the differences between the various objective functions? Under what conditions does the CMI objective function (the ideal objective function that we would use if computing power and quantity of data were not restricted) reduce to the approximate objective functions such as MIM, mRMR, JMI, and CIFE? What explains the relative strengths and weaknesses of these objective functions?

- What guarantees can we make for the greedy search algorithm? How far from optimal are the outputs of the greedy search?

We address the first point by expanding the CMI objective function $J(x) = I(S \cup \{x\}, y)$, which considers all possible interactions, in terms of lower-order interactions in Section 2.4. We show that truncating this series after the degree 2 terms gives us the CIFE objective. The reader may also find it helpful to read the example-based Section 2.3 which demonstrates where various objective functions fail.

To address the second set of questions, on the effectiveness of the greedy search algorithm, we demonstrate in Section 2.5 that under certain conditions, mutual information and its degree 2 approximation (i.e., CIFE) are submodular functions and we use existing literature on submodular functions to give theoretical guarantees for the performance of the greedy search algorithm for CIFE.

In Section 2.7, we explain various considerations that the practitioner will find useful. This includes a section our approach to preprocessing datasets for information-theoretic feature selection methods as these methods require the counts to be in discrete bins, as well as a section on trade offs between running information-theoretic methods as mutli-class and binary methods.

Finally, we note that we have implemented fast versions of these algorithms in the PICTUREDROCKS Python package and we explain some of the optimizations that we made in Section 2.8.

2.3 Analysis of Existing Methods

Most of these algorithms are affected by three broad theoretical weaknesses. We analyse these problems in the context of existing algorithms and present simple examples where they arise. Obviously, many assumptions from [Bro+12] that we discussed in Section 2.2.3 are violated in these simple examples. The aim of these examples, however, is to present the theoretical weaknesses of existing algorithms in information-theoretic terms. This will motivate our approach to mutual information based feature selection[2].

2.3.1 Diminishing penalty on redundancy

The mRMR algorithm in [PLD05] is a prominent example of an objective that progressively diminishes the penalty on redundancy. Specifically,

$$J_{\text{mrmr}}(x_i) = I(x_i; y) - \frac{1}{|S|} \sum_{x_j \in S} I(x_i; x_j). \tag{2.3}$$

The JMI algorithm in [YM00] also has a factor of $\frac{1}{|S|}$ on its redundancy term:

$$J_{\text{jmi}}(x_i) = I(x_i; y) - \frac{1}{|S|} \sum_{x_j \in S} \left(I(x_i; x_j) - I(x_i; x_j | y) \right). \tag{2.4}$$

When $|S|$ is large, the redundancy penalty is so small that these algorithms, when asked for a large set of features (but still considerably smaller than the total number of features), will yield similar outputs as MIM. We demonstrate this empirically in Tables 2.2, 2.3, 2.5, and 2.4.

In practice, the $\frac{1}{|S|}$ factor in algorithms such as mRMR and JMI serves two purposes. First, the $\frac{1}{|S|}$ factor accounts for shared redundancy between multiple variables (see examples in Section 2.3.3). Second, it acts to mitigate the accumulating noise in empirical estimations of $I(x_i; x_j)$, because even if x_i and x_j are independent, the empirical estimation of their mutual information might be positive. In either case, the $\frac{1}{|S|}$ factor is a somewhat arbitrary approach that suppresses both signal and noise and, in doing so, impedes our ability to detect true redundancies.

Example 1. Let $b_1, b_2, \ldots$ be random variables chosen from $\{0, 1\}$ independently with uniform probability. Let $x_1 = b_1 b_2 \ldots b_5$ (a binary concatenation of bits), $x_2 = b_6 b_7 \ldots b_{10}$, $x_3 = b_{11}$, and $x_4 = b_1 b_2 b_3$. Suppose $y = b_1 b_2 \ldots b_{11}$. Algorithms such as mRMR and JMI would select x_1 and x_2 first. At this point, $J(x_3) = 1$ and $J(x_4) = 3 - \frac{1}{2} \cdot 3 = 1.5$. But clearly, x_4 contributes no new information, while x_3 does.

2.3.2 Penalizing Irrelevant Redundancy

Algorithms like mRMR and MIFE penalize all redundancy, even if the redundancy does provide any information about y. For example, if x_i and x_j are highly correlated, then it is possible that $I(x_i; x_j) > \max\left(I(x_i; y), I(x_j; y)\right)$. This is demonstrated in the following example.

[2]It should be noted, however, that no algorithm can truly overcome all the weaknesses below for that will require being able to infer a joint probability distribution over a large number of random variables which is meaningless without an enormous number of samples and also computationally intractable.

Example 2. Let $b_1, b_2, \ldots$ be random variables chosen from $\{0,1\}$ independently with uniform probability. Let $x_1 = b_1b_2b_3$, $x_2 = b_1b_2b_4$, $x_3 = b_5$, and $y = b_3b_4$. Without loss of generality, assume the algorithm chooses x_1 first. Now $I(x_2; y) = 1$, but $I(x_1; x_2) = 2$. Such algorithms might choose x_3 instead of x_2, because $I(x_1; x_2)$ is large, but the redundancy between x_1 and x_2 is not relevant to y and should not be penalized.

2.3.3 Ignoring Higher-degree Interactions Between Features

Features can say both more and less about the target labels together than the sum of their parts. All features covered by Brown, et al.'s framework (objectives of the form (2.2)) are unable to detect higher-order synergies and redundancies because they only detect two-way interactions between features.

Example 3 (Synergy). Let $b_1, b_2, \ldots$ be random variables chosen from $\{0,1\}$ independently with uniform probability. Let $x_1 = b_1$ and $x_2 = b_2$. If $y = b_1 \oplus b_2$ (where $\oplus$ denotes the XOR operation on bits), then $I(x_1; y) = I(x_2; y) = 0$, but $I(x_1x_2; y) = 1$. This can be understood in terms of the multivariate mutual information $I(x_1; x_2; y) = -1$ (which we saw in section 2.4)[3]

Example 4 (Shared Redundancy). Let $b_1, b_2, \ldots$ be random variables chosen from $\{0,1\}$ independently with uniform probability. Let $x_1 = b_1b_4$, $x_2 = b_2b_4$, $x_3 = b_3b_4$, $x_4 = b_5$, and $y = b_1b_2b_3b_4$. Without loss of generality, after two iterations, an algorithm of the form (2.2) will likely have $S = \{x_1, x_2\}$. Now, $I(\{x_1, x_2, x_3\}, y) = 4$ and $I(\{x_1, x_2, x_4\}, y) = 3$, but objectives of the form (2.2) penalize x_3 twice for its redundant bit (b_4).

2.4 Series Expansion of Mutual Information

In Section 2.3.3, we give examples of *shared redundancy* and *synergy*. These interactions between features is explained by multivariate mutual information. Intuitively, we want to measure how much information a set of random variables share simultaneously. If the one bit of information that x_3 shares with x_1 is the same as the one bit of information that x_3 shares with x_2, then we want $I(x_1; x_2; x_3) = 1$, and if x_3 shares different bits of information with x_1 and x_2, then $I(x_1; x_2; x_3) = 0$.

This suggests a generalization of mutual information that was studied in [Han75]. We let $I_1(x) = H(x)$ to be the Shannon entropy of x (i.e., the self-information). Define the multivariate mutual information $I_n(x_1; \ldots; x_n)$ recursively as $I_n(x_1; \ldots; x_n) = I_{n-1}(x_1; \ldots; x_{n-1}) - I_{n-1}(x_1; \ldots; x_{n-1} | x_n)$, where $I_{n-1}(x_1; \ldots; x_{n-1} | x_n)$ is the information shared between $x_1, \ldots, x_{n-1}$ given we know the

[3]In cases of extreme synergy, such as the current example (where individual feature provide no information on their own) even the J_{cmi} objective will fail to choose an optimal set of features as a result of the shortsightedness of a greedy approach. In less extreme cases, it is still beneficial for the objective function to recognize such synergy).

value of x_n. If $S = \{x_1, \ldots, x_n\}$, we denote by $I_n(S)$ the multivariate mutual information $I_n(x_1; x_2; \ldots; x_n)$. Note the subtle, but crucial, difference in notation: $I_n(S)$ is the information simultaneously shared by the n variables in S (intuitively, the intersection of information), while $I(S)$ is the self-information of the ensemble of random variables of S (intuitively, the union of the information). The following equality (à la Principle of Inclusion and Exclusion) expresses the multivariate mutual information $I_n(S)$ in terms of lower degree information terms.

Proposition 2.1.

$$I_n(x_1; x_2; \ldots; x_n) = \sum_{X \subseteq \{x_1, \ldots, x_n\}} (-1)^{|X|+1} H(X)$$

Proof. We prove the equality above by induction. For $n = 2$, this follows from the definition of mutual information. For $n > 2$, we have

$$\begin{aligned}
&I_n(x_1; x_2; \ldots; x_n) \\
&= I_n(x_1; \ldots; x_{n-1}) - I(x_1; \ldots; x_{n-1} | x_n) \\
&= \sum_{X \subseteq \{x_1, \ldots, x_{n-1}\}} (-1)^{|X|+1} (H(X) - H(X|x_n)) \\
&= \sum_{X \subseteq \{x_1, \ldots, x_{n-1}\}} (-1)^{|X|+1} \left(H(X) - H(Xx_n) + H(x_n)\right) \\
&= \left(\sum_{X \subseteq \{x_1, \ldots, x_n\}} (-1)^{|X|+1} H(X) \right) + \sum_{i=0}^{n} \binom{n}{i} H(x_n) \\
&= \sum_{X \subseteq \{x_1, \ldots, x_n\}} (-1)^{|X|+1} H(X).
\end{aligned}$$

□

Corollary 2.2.

$$H(A) = \sum_{S \subseteq A} (-1)^{|S|+1} I_{|S|}(S).$$

Proof. This follows from a version of the principle of inclusion and exclusion, see [GGL95, p. 1049 Theorem 12.1]. □

We now give a series expansion of the mutual information $I(\{x_1, \ldots, x_n\}, y)$ that suggests approximations to $I(\{x_1, \ldots, x_n\}, y)$ using lower degree terms (i.e., without needing to estimate the joint probability distribution of a large set of variables).

Lemma 2.3.

$$I(\{x_1,\dots,x_n\},y)=\sum_{k=1}^{n}\sum_{\substack{X\subseteq\{x_1,\dots,x_n\}\\|X|=k}}(-1)^{k+1}I_{k+1}(X;y).$$

Proof.

$$\begin{aligned}
&I(\{x_1,\dots,x_n\},y)\\
&=H(\{x_1,\dots,x_n\})+H(y)-H(\{x_1,\dots,x_n,y\})\\
&=\sum_{X\subseteq\{x_1,\dots,x_n\}}(-1)^{|X|-1}I_{|X|}(X)+I(y)\\
&-\sum_{X\subseteq\{x_1,\dots,x_n\}}(-1)^{|X|-1}I_{|X|}(X)\\
&=\sum_{\emptyset\subsetneq X\subseteq\{x_1,\dots,x_n\}}(-1)^{|X|+1}I_{|X|+1}(X;y)\\
&=\sum_{k=1}^{n}\sum_{\substack{X\subseteq\{x_1,\dots,x_n\}\\|X|=k}}(-1)^{k+1}I_{k+1}(X;y). \qquad (2.5)
\end{aligned}$$

□

2.4.1 Degree K Supervised Feature Selection Algorithm

It is clear that if $X\subseteq Y$, then $I_{|X|}(X)\geq I_{|Y|}(Y)$. In a practical setting it is also reasonable to believe that as k gets larger, the terms in the series in (2.5) converge to 0. As mentioned earlier, the computational cost of the terms in the series above grows exponentially with respect to k. This suggests that a fast "degree K" approximation to the mutual information $I(\{x_1,\dots,x_n\};y)$:

$$I(\{x_1,\dots,x_n\},y)=\sum_{k=1}^{K}\sum_{\substack{X\subseteq\{x_1,\dots,x_n\}\\|X|=k}}(-1)^{k+1}I_{k+1}(X;y).$$

At each iteration of the greedy algorithm, our goal is to compute

$$\arg\max_{x_i\notin S} I(S\cup\{x_i\};y). \qquad (2.6)$$

If we use the above degree K approximation, we only need to consider the terms involving x_i in the series expansion. Therefore, at each step, the degree K mRMR algorithm selects the next feature to add to S as:

$$J_{\text{deg}K}(x_i) = \arg\max_{x_i \notin S} \sum_{k=0}^{K-1} \sum_{\substack{X \subseteq S \\ |X|=k}} (-1)^k I_{k+2}(X; x_i; y).$$

Notice that for $K = 2$, this simplifies to

$$J_{\text{deg2}}(x_i) = \arg\max_{x_i \notin S} I(x_i; y) - \sum_{x_j \in S} I(x_i; x_j; y),$$

the CIFE objective in [LT06]. These functions $J_{\text{deg}K}$ serve as the objective function J in the greedy search algorithm (Algorithm 1).

2.4.2 Degree K Unsupervised Feature Selection Algorithm

In the unsupervised setting, we are trying to solve

$$\arg\max_{S \subseteq \mathcal{X}; |S| \leq n} H(S).$$

This involves approximating n-dimensional joint probability distributions, which necessitates the need for lower-degree approximations.

We use Corollary 2.2 to expand $H(S)$ and write

$$H(S) = \sum_{k=1}^{|S|} \sum_{U \subseteq S; |U|=k} I_{|U|}(U).$$

If we assume that there are no higher-degree interactions, we get a low degree approximation for $H(S)$.

Corollary 2.4. *If $I_{|U|}(U) = 0$ whenever $|U| \geq 3$, then*

$$H(S) = \widetilde{H_2}(S) := \sum_{x_i \in S} H(x_i) - \sum_{x_i, x_j \in S} I(x_i; x_j).$$

□

Using the greedy search (Algorithm 1), at each step, we identify

$$\begin{aligned} J_{\text{unsup deg}K}(x_i) &= \arg\max_{x_i \notin S} \left(\widetilde{H_2}(S \cup \{x_i\}) - \widetilde{H_2}(S) \right) \\ &= \arg\max_{x_i \notin S} \left(H(x_i) - \sum_{x_j \in S} I(x_i; x_j) \right). \end{aligned}$$

2.5 Guarantees for Greedy Search

The greedy search algorithm (Algorithm 1) is a necessary compromise to avoid the computation cost of an exhaustive search over all subsets of size at most n. However, there are some cases where such an approach is too shortsighted, even when we use the ideal objective function $J_{\text{cmi}}(x) = I(S \cup \{x\}; y)$.

Example 5. Let b_1, b_2 be random variables chosen from $\{0, 1\}$ independently with uniform probability. Let b_3 be a random variable (independent of b_1, b_2) such that $P(b_3 = 0) = \frac{1}{2} + \varepsilon$ and $P(b_3 = 1) = \frac{1}{2} - \varepsilon$ where $0 < \varepsilon < \frac{1}{2}$. Finally, let $x_1 = b_1$, $x_2 = b_2$, $x_3 = b_1 \oplus b_2 \oplus b_3$, and $y = b_1 \oplus b_2$. It is clear that $I(x_1; y) = I(x_2; y) = 0$ and $I(x_3; y) = 1 - H(b_3)$. The optimal 2-element subset S of $\{x_1, x_2, x_3\}$ that maximizes $I(S; y)$ is obviously $\{x_1, x_2\}$, but a greedy approach will always choose x_3 at the first step.

In this section, we describe conditions under which we can give performance guarantees for the greedy search algorithm. These results rely heavily on results for the maximization of submodular functions. A function f defined on sets is submodular if it satisfies the diminishing returns property: $f(T + \{x\}) - f(T) \leq f(S + \{x\}) - f(S)$ whenever $S \subseteq T$. The seminal paper on this topic by [NWF78] proves the following result.

Theorem 2.5 ([NWF78]). *Consider the maximization problem*

$$\arg\max_{S \subseteq \mathcal{X}, |S| \leq k} f(S).$$

If f is a non-decreasing submodular function, the greedy algorithm yields a $(1 - 1/e)$-approximation. In other words, if S_o is an optimal set and S_g is the set found via a greedy search, then

$$\frac{f(S_o) - f(S_g)}{f(S_o) - f(\emptyset)} \leq \frac{1}{e}.$$

2.5.1 Submodularity of Entropy and its Degree 2 Approximation

Proposition 2.6. *The entropy function $H : 2^{\mathcal{X}} \to \mathbb{R}$ is submodular.*

Proof. We want to show $H(T \cup \{x\}) - H(T) \leq H(S \cup \{x\}) - H(S)$ whenever $S \subseteq T$. Let $T' = T - S$. We have $I(T'; x|S) \geq 0$ because (conditional) mutual information is always non-negative. This gives us

$$\begin{aligned}
&H(x|S) - H(x|ST') \geq 0 \\
&H(S \cup \{x\}) - H(S) \geq H(S \cup T' \cup \{x\}) - H(S \cup T') \\
&H(S \cup \{x\}) - H(S) \geq H(T \cup \{x\}) - H(T)
\end{aligned}$$

□

Proposition 2.7. *The degree 2 approximation $\widetilde{H_2} : 2^{\mathcal{X}} \to \mathbb{R}$ of H, given by $\widetilde{H_2}(S) = \sum_{x_i \in S} H(x_i) - \sum_{x_i, x_j \in S} I(x_i; x_j)$ is submodular.*

Proof. Note that $\widetilde{H_2}(S \cup \{x\}) - \widetilde{H_2}(S) = H(x) - \sum_{x_i \in S} I(x; x_i)$. Recall that $I(\cdot;\cdot)$ is non-negative. Clearly, $\widetilde{H_2}(T \cup \{x\}) - \widetilde{H_2}(T) \leq \widetilde{H_2}(S \cup \{x\}) - \widetilde{H_2}(S)$ whenever $S \subseteq T$. □

2.5.2 Submodularity of Mutual Information and its Degree 2 Approximation

In general, mutual information is not submodular. As we saw in Example 5, the "value" of a feature is not necessarily diminishing. In that particular example we had $I(x_2; y) - I(\emptyset; y) = 0$, but $I(\{x_1, x_2\}; y) - I(x_1; y) = 1$; crudely put, a greedy algorithm only appreciates the value of x_2 when it already knows x_1. This specific phenomenon is a direct consequence of synergy and we explain this relation rigorously in this section.

Proposition 2.8. *If for all $x \notin T \supseteq S$, we have $I(T - S; x; y|S) \geq 0$, then the set function $f : 2^{\mathcal{X}} \to \mathbb{R}$ given by $f(S) = I(S; y)$ is submodular.*

Proof. As before, let $T' = T - S$. We have $I(T'; x; y|S) = I(T'; x; y) - I(S; T'; x; y)$. We can now rewrite $I(T - S; x; y|S) \geq 0$ as $I(S; T'; x; y) - I(T'; x; y) \leq 0$. Adding many terms to each side of the inequality, we get

$$\begin{aligned}
I(S; y) + I(T'; y) + I(x; y) - I(S; x; y)& \\
-I(T'; x; y) - I(S; T'; y) + I(S; T'; x; y)& \\
-I(S; y) - I(T'; y) + I(S; T'; y)& \\
\leq\ I(S; y) + I(x; y) - I(S; x; y) - I(S; y)& \qquad (2.7)
\end{aligned}$$

Therefore,

$$I(T \cup \{x\}; y) - I(T; y) \leq I(S \cup \{x\}; y) - I(S; y) \qquad (2.8)$$

□

The analogous result for the degree approximation $\widetilde{I_2}$ of I given by $\widetilde{I_2}(S;y) = \sum_{x_i \in S} I(x_i;y) - \sum_{x_i,x_j \in S} I(x_i;x_j;y)$ (which is the function that the CIFE algorithm maximizes) is a little easier to describe, since we ignore higher order interactions. We only need to assume that there is no two-way synergy.

Proposition 2.9. *If for all x_i, x_j, we have $I(x_i;x_j;y) \geq 0$, then the set function $f : 2^{\mathcal{X}} \to \mathbb{R}$ given by $f(S) = \sum_{x_i \in S} I(x_i;y) - \sum_{x_i,x_j \in S} I(x_i;x_j;y)$ is submodular.*

Proof. Note that $\widetilde{I_2}(S \cup \{x\};y) - \widetilde{I_2}(S;y) = I(x;y) - \sum_{x_i \in S} I(x;x_i;y)$. Because we have assumed $I(x_i;x_j;y) \geq 0$ for all x_i, x_j, we have $f(T \cup \{x\}) - f(T) \leq f(S \cup \{x\}) - f(S)$ whenever $S \subseteq T$. □

2.5.3 Guarantees for Feature Selection

It is well known that $I(\cdot;y)$ and $H(\cdot)$ are non-decreasing set functions. We immediately have the following corollaries.

Corollary 2.10. *The greedy search algorithm to maximize $H(S)$ over $S \subseteq \mathcal{X}$ with $|S| \leq k$ for some user-specified k yields a $(1 - 1/e)$-approximation (i.e., $H(S_{greedy}) \geq (1 - 1/e)H(S_{optimal})$.*

Proof. Follows by Theorem 2.5 and Proposition 2.6. □

Corollary 2.11. *If for all $x \notin T \supseteq S$, we have $I(T - S;x;y|S) \geq 0$, then the greedy search algorithm to maximize $I(S;y)$ over $S \subseteq \mathcal{X}$ with $|S| \leq k$ for some user-specified k yields a $(1 - 1/e)$-approximation (i.e., $I(S_{greedy};y) \geq (1 - 1/e)I(S_{optimal};y)$.*

Proof. Follows by Theorem 2.5 and Proposition 2.8. □

Unfortunately, our approximations $\widetilde{I_2}(\cdot;y)$ and $\widetilde{H_2}$ of $I(\cdot;y)$ and H functions are not non-decreasing, even though $I(\cdot;y)$ and H themselves are. This leaves us with a slightly weaker result for optimizing $\widetilde{I_2}$ and $\widetilde{H_2}$. The following is a generalization of Theorem 2.5.

Theorem 2.12 ([NWF78]). *Consider the maximization problem*

$$\arg\max_{S \subseteq \mathcal{X}, |S| \leq k} f(S).$$

If f is a submodular function and $f(S \cup \{x\}) - f(S) \geq -\theta$ for all $S \subseteq \mathcal{X}$ and $x \in \mathcal{X}$, we have

$$\frac{f(S_o) - f(S_g)}{f(S_o) - f(\emptyset) + k\theta} \leq \frac{1}{e},$$

where S_o is an optimal set and S_g is the set found via a greedy search.

Corollary 2.13. *If $\widetilde{H_2}(S \cup \{x\}) - \widetilde{H_2}(S) \geq -\theta$ for all $S \subseteq \mathcal{X}$ and $x \in \mathcal{X}$, then the greedy solution $S_g \subseteq \mathcal{X}$ to $\arg\max_{S \subseteq \mathcal{X}, |S| \leq k} \widetilde{H_2}(S)$ relates to the optimal solution S_o as follows:*

$$\widetilde{H_2}(S_o) - \widetilde{H_2}(S_g) \leq \frac{\widetilde{H_2}(S_o) + k\theta}{e}$$

Corollary 2.14. *If $I(x_i; x_j; y) \geq 0$ for all $x_i, x_j \in \mathcal{X}$ and $\widetilde{I_2}(S \cup \{x\}; y) - \widetilde{I_2}(S; y) \geq -\theta$ for all $S \subseteq \mathcal{X}$ and $x \in \mathcal{X}$, then the greedy solution $S_g \subseteq \mathcal{X}$ (i.e., the output of the CIFE algorithm) to $\arg\max_{S \subseteq \mathcal{X}, |S| \leq k} \widetilde{I_2}(S; y)$ relates to the optimal solution S_o as follows:*

$$\widetilde{I_2}(S_o; y) - \widetilde{I_2}(S_g; y) \leq \frac{\widetilde{I_2}(S_o; y) + k\theta}{e}$$

In practical terms, we can interpret corollary 2.14 as follows: if we know that shared redundancy between any pair of features outweigh synergies and that our degree 2 approximation to mutual information is monotone, then the greedy algorithm will output an $1 - \frac{1}{e}(\approx 0.632)$ approximation (i.e., the score of the set returned by a greedy search will be at least 63.2% of the score of the optimal set). If we know that the degree 2 approximation is not monotone but can bound how much it might decrease by at any step, we can still recover a (weaker) guarantee.

2.6 Empirical Results

To assess the performance of mutual information methods in practice, we performed numerous experiments with publicly available single cell RNA-seq datasets. For each of the datasets listed below, we treat each cell as an observation and each gene as a feature.

- Paul dataset from [Pau+15] with 2730 mouse bone marrow cells and 3451 genes. Bone marrow cells form a continuous trajectory rather than distinct clusters because many of the cells are in the process of differentiating. The authors of the study assign class labels to cells based on expression of known marker genes.

- Green dataset [Gre+18] with 34633 cells from adult mouse testis and 37241 genes. The cells in dataset form a continuous trajectory, as should be expected for spermatogenesis.

- Zheng dataset from [Zhe+17] with 68579 human blood cells and 32738 genes. This dataset consists of human blood cells that were clustered into distinct cell types. Zheng, et al̃then used these clusters to isolate more cells based on these types to validate their computational results.

- Zeisel dataset [Zei+15] with 3005 mouse neuron cells and 19972 genes. Because neuron cells lie in discrete clusters in gene space, we expect clustering and classification tasks to be easier for this dataset. Further, Zeisel, et al. have carefully assigned labels to clusters and validated their cluster labels as corresponding to known biology.

Recall that our goal is to find a small set of genes that explain the given target labels (e.g., cluster labels or biologically verified cell type labels). To this end, we performed 5-fold cross validation by splitting the datasets into 5 folds, and then, for each fold, we selected markers on the four remaining folds, projected the data onto the selected features (i.e., only considered the columns of the matrix that correspond to the selected genes), trained a classifier on the four folds, and predicted class labels on the current fold. The feature selection methods that perform well are those that select discriminative features that, together, help predict the class of an unseen observation. In other words, feature selection methods with the lowest error rates choose genes that explain the class labels best.

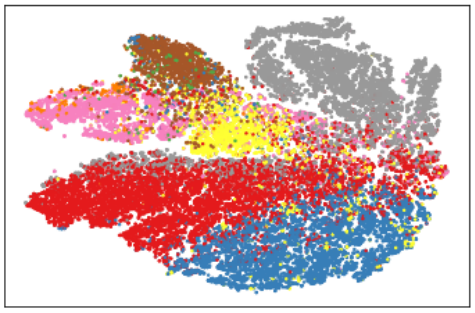
We tested three mutual information based feature selection algorithms:

- the CIFE algorithm (which is equivalent to the degree 2 approximation presented in Section 2.4.1),
- the MIM algorithm (which is equivalent to the degree 1 approximation), and
- the JMI algorithm (which dampens the redundancy penalty in CIFE by a factor of $\frac{1}{|S|}$, where S is the set of previously selected features).

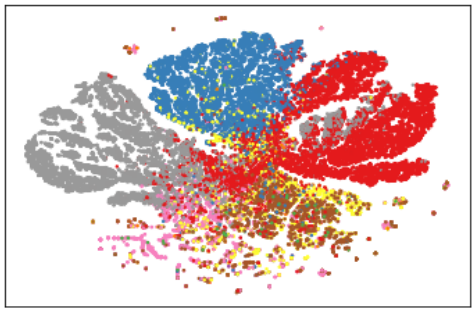
For each dataset, we ran these algorithms in both a multi-class and binary fashion (see Section 2.7.2). For all information-theoretic algorithms, we began by selecting the 5000 most relevant features (as found by the univariate MIM) before running bivariate algorithms (the run times in Figure 2.6 includes the MIM run time). For binary versions, this univariate filter was run for each class label separately.

We tested our algorithms with two different classifiers. Note that even for the datasets where class labels were originally found using an (unsupervised) classifier (rather than biologically), and we use supervised classifiers because we want to assess how informative the selected features are about the class labels.

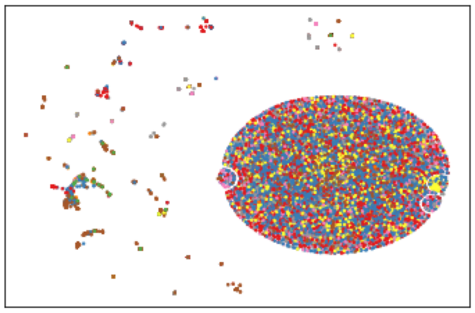
- a Nearest Centroids Classifier that computes the centroid (unweighted mean of vectors) for each class label and to each unseen observation assigns the class label with the nearest centroid. We ran our experiments with this classifier to evaluate performance with a transparent classifier with minimal bells and whistles that can be easily implemented from scratch. Plots for classification errors with Nearest Centroids are available in Section 2.6.3.



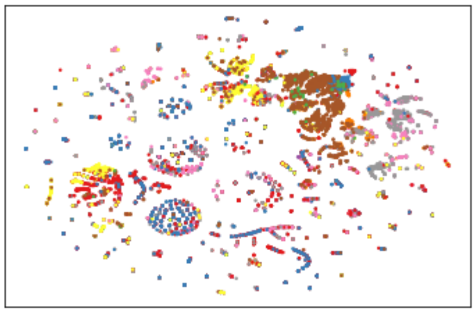
(a) t-SNE Plot for Green Dataset using 10 features selected by CIFE (multiclass)



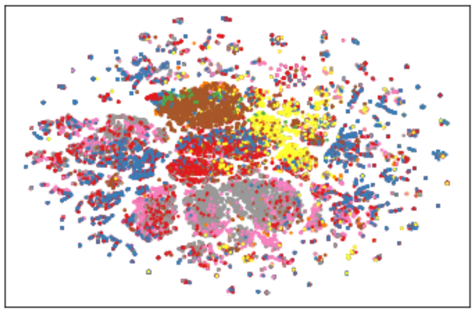
(b) t-SNE Plot for Green Dataset using 10 features selected by JMI (multiclass)



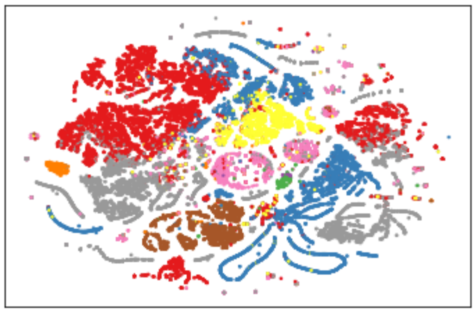
(c) t-SNE Plot for Green Dataset using 11 features selected by Logistic Regression



(d) t-SNE Plot for Green Dataset using 11 features selected by t-test



(e) t-SNE Plot for Green Dataset using 11 features selected by Wilcoxon Rank Sum



(f) t-SNE Plot for Green Dataset using 11 features selected by CIFE (binary)

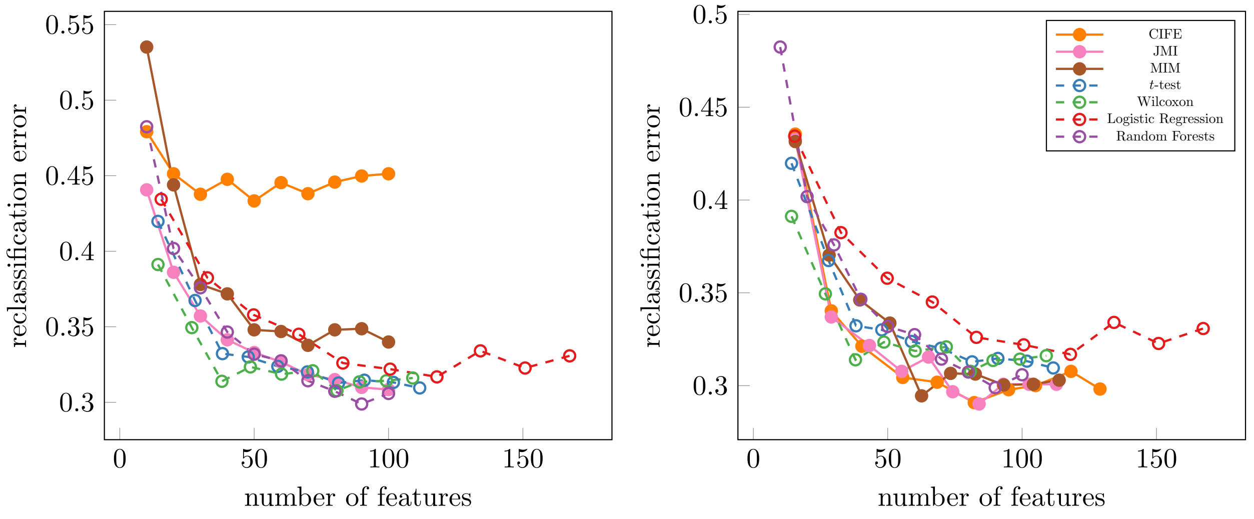


Figure 2.2: Classification Errors For Paul Dataset (left: multiclass MI methods, right: binary MI methods)

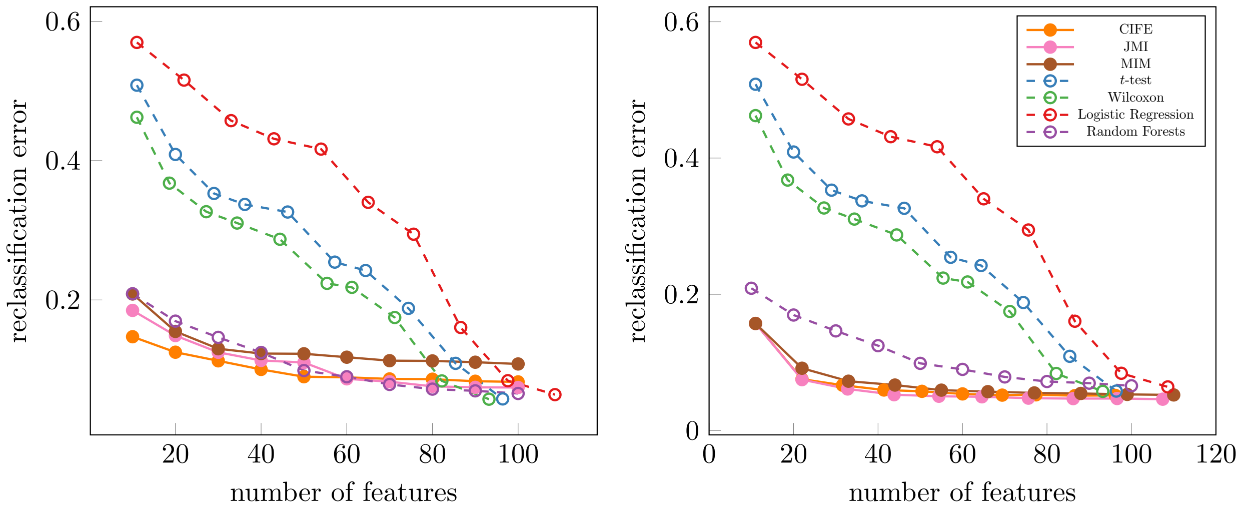


Figure 2.3: Classification Errors For Green Dataset (left: multiclass MI methods, right: binary MI methods)

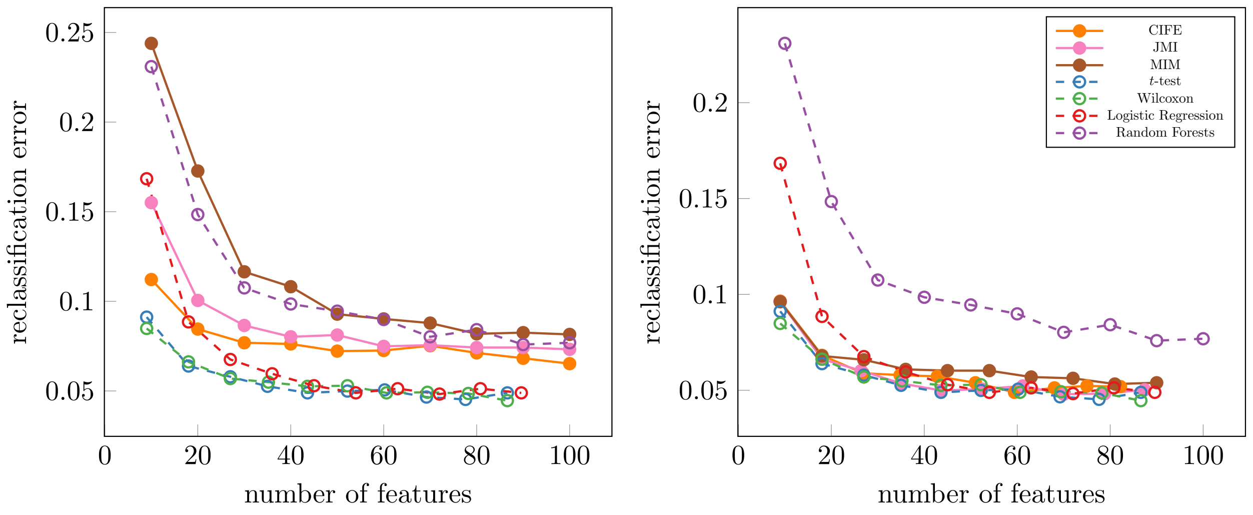


Figure 2.4: Classification Errors For Zeisel Dataset (left: multiclass MI methods, right: binary MI methods)

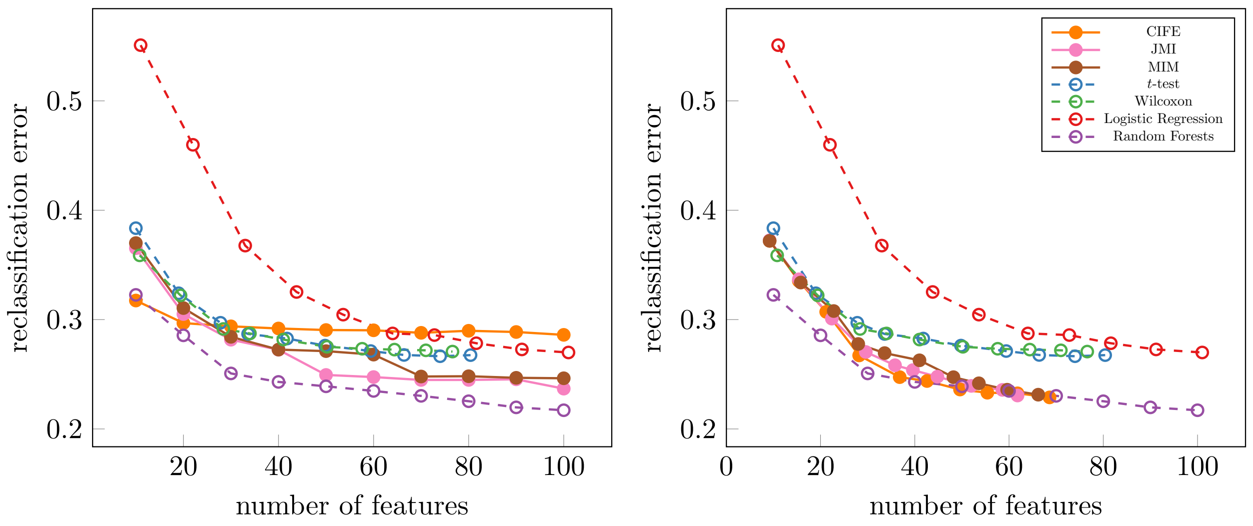


Figure 2.5: Classification Errors For Zheng Dataset (left: multiclass MI methods, right: binary MI methods)

Table 2.2: Number of Genes in Common between algorithms (Paul Dataset)

	LogReg	t-Test	Wilcoxon	RForest	JMI	MIM	CIFE	JMI(B)	MIM(B)	CIFE(B)
LogReg	100	40	36	24	18	18	16	36	31	28
t-Test	40	101	83	52	47	41	19	68	65	39
Wilcoxon	36	83	101	55	53	44	18	66	63	43
RForest	24	52	55	100	69	62	11	54	55	50
JMI	18	47	53	69	100	84	6	50	55	46
MIM	18	41	44	62	84	100	4	45	52	40
CIFE	16	19	18	11	6	4	100	9	7	12
JMI(B)	36	68	66	54	50	45	9	104	87	41
MIM(B)	31	65	63	55	55	52	7	87	104	40
CIFE(B)	28	39	43	50	46	40	12	41	40	101

- a Random Forests Classifier that uses an ensemble of decision trees. We use the Scikit Learn ([Ped+11]) implementation, which has numerous tweaks to improve accuracy and reduce overfitting. Although it is a more opaque classifier, the Random Forests Classifier does not give each feature equal weight and will use new features only when they are helpful. This aligns more closely with our motivation of finding an efficient set of features that are collective informative about the class labels rather than features that, individually, distinguish classes from one another.

We also tested against three differential expression algorithms (t-test with overestimated variation, Wilcoxon Rank Sum, and Logistic Regression) for single cell RNA sequencing (using the implementations in the widely used SCANPY package[4], see [WAT18]). Since differential expression algorithms take a One-vs-Rest approach, we combined the lists into a single list by taking the first k genes (features) for each class label and removing duplicates.

Finally, since we are using a Random Forests classifier, we also evaluated Random Forests' "feature importance" as a feature selection method. We should note that using Random Forests for feature selection and comparing against other methods using Random Forests is not a fair comparison, but we include it nonetheless for sake of comparison.

Figures 2.1a, 2.1b, 2.1c, 2.1d, 2.1e, 2.1f are t-SNE plots (the projection of the dataset onto the top two principal components) for the Green dataset [Gre+18] generated using only a few genes. The colors of each point represents its class label (cell type).

The error rates in 5-fold cross validation with random forests are summarized in Figures 2.2, 2.3, 2.5, and 2.4 (the corresponding figures for the nearest centroids classifier are Figures 2.8, 2.9, 2.10, and 2.11 in Section 2.6.3).

In addition to error rates, which suggest how well the algorithms select features that explain

[4]We used the latest developmental version (commit 80e635d2b78) available at the time as this revision fixed a bug that prevented us from using scanpy during K-fold cross validation.

Table 2.3: Number of Genes in Common between algorithms (Green Dataset)

	LogReg	t-Test	Wilcoxon	RForest	JMI	MIM	CIFE	JMI(B)	MIM(B)	CIFE(B)
LogReg	108	38	32	8	9	7	3	49	45	20
t-Test	38	106	89	28	24	15	6	55	47	27
Wilcoxon	32	89	103	27	28	17	11	49	41	27
RForest	8	28	27	100	45	34	3	25	19	6
JMI	9	24	28	45	100	73	8	21	16	9
MIM	7	15	17	34	73	100	3	16	12	9
CIFE	3	6	11	3	8	3	100	4	1	18
JMI(B)	49	55	49	25	21	16	4	108	79	27
MIM(B)	45	47	41	19	16	12	1	79	100	23
CIFE(B)	20	27	27	6	9	9	18	27	23	109

Table 2.4: Number of Genes in Common between algorithms (Zeisel Dataset)

	LogReg	t-Test	Wilcoxon	RForest	JMI	MIM	CIFE	JMI(B)	MIM(B)	CIFE(B)
LogReg	100	48	43	27	29	21	4	48	50	23
t-Test	48	100	87	39	48	34	5	65	71	26
Wilcoxon	43	87	100	38	50	33	7	64	68	26
RForest	27	39	38	100	57	42	1	42	36	30
JMI	29	48	50	57	100	71	3	42	41	36
MIM	21	34	33	42	71	100	2	25	27	31
CIFE	4	5	7	1	3	2	100	4	3	2
JMI(B)	48	65	64	42	42	25	4	115	79	30
MIM(B)	50	71	68	36	41	27	3	79	100	20
CIFE(B)	23	26	26	30	36	31	2	30	20	106

Table 2.5: Number of Genes in Common between algorithms (Zheng Dataset)

	LogReg	t-Test	Wilcoxon	RForest	JMI	MIM	CIFE	JMI(B)	MIM(B)	CIFE(B)
LogReg	102	31	29	19	19	18	2	33	31	19
t-Test	31	110	94	52	60	59	14	61	64	34
Wilcoxon	29	94	108	53	62	62	15	57	60	36
RForest	19	52	53	100	79	77	20	53	51	51
JMI	19	60	62	79	100	94	16	58	58	50
MIM	18	59	62	77	94	100	14	55	56	49
CIFE	2	14	15	20	16	14	100	14	9	28
JMI(B)	33	61	57	53	58	55	14	101	81	45
MIM(B)	31	64	60	51	58	56	9	81	101	37
CIFE(B)	19	34	36	51	50	49	28	45	37	100

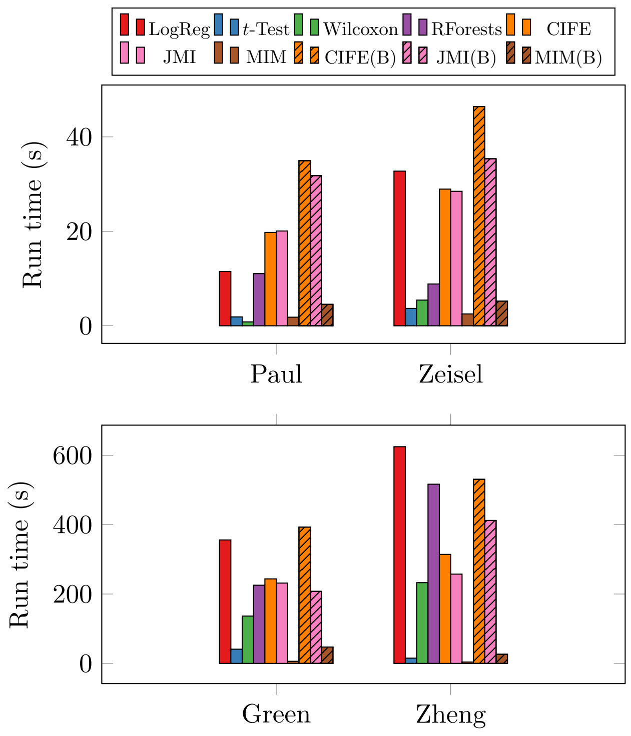


Figure 2.6: Runtimes for various algorithms

class labels, we are also interested in knowing the extent to which these algorithm select the same features. We summarize these intersections in Tables 2.2, 2.3, 2.4, and 2.5. Because binary methods merge lists of features, we cannot ensure that all feature sets are of equal size. Therefore, not all sets are exactly of size 100 features, but they are the smallest possible sets of size ≥ 100.

In order to compare the run times for the various algorithms, we timed individual runs of each algorithm on each dataset (without any cross-validation). The recorded times for bivariate methods (the multiclass and binary variants of both CIFE and JMI) include the run time for the initial univariate selection of 5000 features with MIM.

2.6.1 Discussion

On the datasets we have evaluated, the mutual information based methods seem to consistently outperform differential expression methods when run as binary methods. However, when run as multiclass methods, mutual information based methods perform better than differential expression methods only on some datasets, e.g., the Green dataset (Figure 2.3) and the Zheng dataset (Figure 2.5), while differential expression methods far outperformed multiclass methods in other datasets, e.g., the Paul dataset (Figure 2.2). It is worth noting that it is only on the Zheng dataset that selecting features based on Random Forests' feature importances is significantly better than other methods. We would also add that Figures 2.1a, 2.1b, 2.1c, 2.1d, and 2.1e demonstrate the strength of multiclass information theoretic methods on the Green dataset, but as we see by comparing Figure 2.3 to Figure 2.2, we don't expect this to be the case on all datasets.

Tables 2.2, 2.3, 2.5, and 2.4 suggest that many binary methods choose many of the same features. The notable exceptions are logistic regression and the binary CIFE method, both of which have moderate overlaps with other methods, but still select numerous features that other methods do not. The multiclass CIFE algorithm has, by far, the most distinct feature sets. This proves to be both a source of strength and weakness as the classification errors associated to these feature sets are sometimes very poor (e.g., in the Paul dataset, Figure 2.2, and, to a much milder extent, the Zheng dataset, Figure 2.5) but sometimes very competitive (e.g., in the Green dataset, Figure 2.3, and in the Zeisel dataset, Figure 2.4). This suggests that in some cases, CIFE can be used to discover novel markers genes for cell types and/or cell states in scRNA-seq.

The run times for information-theoretic algorithms, as demonstrated by Figure 2.6, are comparable to their differential expression counterparts. Specifically, the univariate MIM is almost always the fastest and the bivariate CIFE and JMI have similar run times to logistic regression.

Finally, we consider the assumptions we made to give our theoretical guarantees in Section 2.5 and examine how close these datasets are to satisfying those assumptions. One of the most significant assumption we made (e.g., in Corollary 2.14) is that $I(x_i; x_j; y) \geq 0$ for all $x_i, x_j \in \mathcal{X}$. In Figure 2.7, we see that although the hypothesis don't hold perfectly on any dataset, they hold best on the Green dataset (where CIFE seems to perform best, see Figure 2.3) and worst on the Paul dataset (where CIFE seems to perform the worst, see Figure 2.2).

2.6.2 Experimental Setup

Our benchmarking process relied heavily on SCANPY, a widely used package for single cell analysis in Python, and PICTUREDROCKS, a software package that we developed. PICTUREDROCKS is a Python package that can perform information-theoretic feature selection and also contains tools to benchmark feature selection methods via k-fold cross-validation. PICTUREDROCKS includes a interactive user interface for feature selection that can allows researchers to incorporate domain knowledge in the process of feature selection. All the code to reproduce our benchmarking study can also be found on GitHub.

There are numerous possible pitfalls when evaluating the ability of a feature selection method to select informative features. We refer the reader to the excellent section on "The Wrong and Right Way to do Cross-validation" in [HTF13, Section 7.10.2]. In particular, we highlight the following:

- Cross-validation is effective at estimating the generalization error rather than the test error, which is susceptible to artificially being suppressed due to overfitting.

- It is extremely important to ensure that at no point does any data "leak" between folds. For

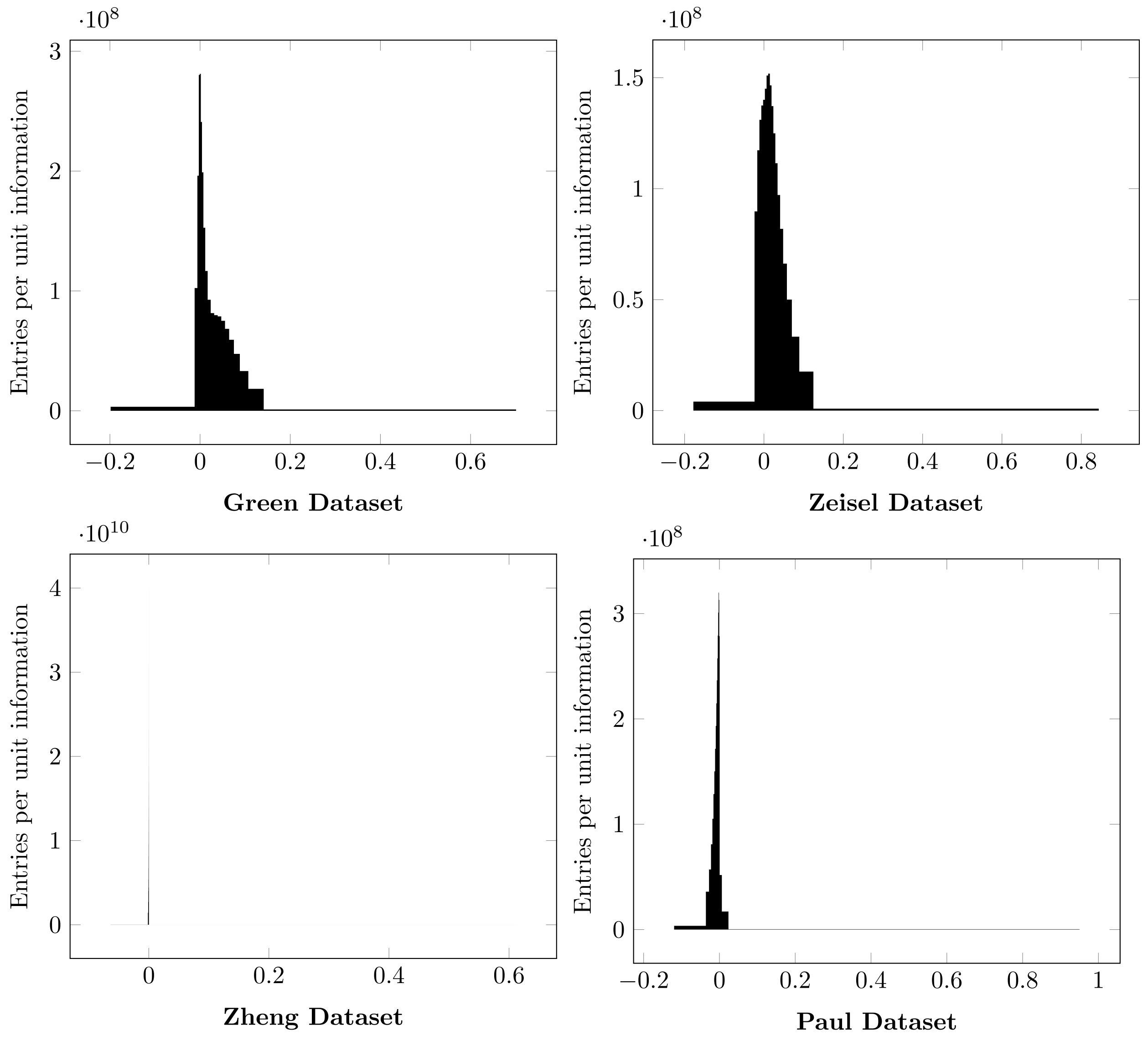


Figure 2.7: Histogram of $I(x_i; x_j; y)$ values for various datasets.

example, as explained in [HTF13], it is not fair to evaluate a feature selection method by first selecting features on the entire dataset and then report the cross-validation error. Even though the classifier used to predict labels for each fold was not trained on that fold, the feature selection algorithm had access to data from all folds.

In order to avoid such pitfalls, we performed our benchmarking in the following manner:

1. Split the dataset into 5 folds
2. For each fold $k = 1, \dots, 5$:
   (a) use all samples not in fold k to select n features; repeat for each feature selection algorithm and store results separately
   Remark 1: For binary feature selection methods (such as t-test, Wilcoxon Rank Sum, Logistic Regression, and when any of the information-theoretic methods are forcibly run as binary methods), this means running the algorithm for each class label and combining the top $\lceil n/c \rceil$ features from each class label, where c is the number of classes. Remove duplicates. Keep track of the number of features selected, since this may not be exactly n.
   Remark 2: the bivariate methods (CIFE and JMI) can slow down considerably in situations when we are selecting many features and there are many candidate features (specifically there is a $O(np)$ term in the time complexity of these methods). To avoid these prohibitively slow run times, we use a univariate method (MIM) to select 5000 features and then use the bivariate method to select among those 5000 features
   (b) train a Random Forests classifier and a Nearest Centroids classifier on all samples not in fold k with only the n (or possibly a few more) features selected; repeat for each feature selection algorithm
   (c) use both classifiers to predict labels for samples in fold k and store the predictions of each classifier separately; repeat for each feature selection algorithm
3. for each feature selection algorithm and classifier combination:
   (a) compute the error rate of the predictions as $\#(\text{wrongly classified})/\#(\text{samples})$.
   (b) for binary methods where the number of features may not be exactly n, record the number of features used as the average number of features used for each fold

For t-test, Wilcoxon Rank Sum, and Logistic Regression feature selection methods we used the standard implementation in SCANPY. As part of the feature selection subroutines, we perform basic preprocessing (as this happens on a copy of the data, it does not affect the data used for classification): `normalize per cell` followed by `log1p`. These methods scale the counts matrix so that each cell's counts sum to the same number (this is usually chosen to the median count prior to scaling) and then apply the $x \mapsto \log(1+x)$ transformation element-wise. For the information-theoretic methods, we use the recursive quantile transformation (with 5 bins) before running the methods, see Section 2.7.1.

At the classification step, we used the Random Forests classifier from the Scikit-learn package[Ped+11] with 100 estimators. As a preprocessing step, we scaled the data so that the sum of counts for each cells was 1000 (this is arbitrary but had to be chosen in order to standardize the preprocessing between training and testing) and then applied the $x \mapsto \log(1+x)$ transformation. For the Nearest Centroids classifier, we applied the $x \mapsto \log(1+x)$ transformation, projected onto the top 30 principal components of the training data (as euclidean distances become less meaningful in high dimensions), and the mapped each training point to the class with the nearest centroid in the training data.

When generating Figure 2.6 and Tables 2.2, 2.3, 2.4, 2.5, we performed the preprocessing for feature selection algorithms listed above and ran the algorithms on the entire dataset rather than use cross validation as these figures are not meant to estimate generalizability. Figures 2.1a, 2.1b, 2.1c, 2.1d, 2.1e, and 2.1f were also generated with these feature sets using the t-SNE implementation in Scikit-Learn [Ped+11].

2.6.3 Classification Errors with Nearest Centroids

In this section, we present classification error rate figures when we use nearest centroids, instead of random forests, to classify points after feature selection. In general, the error rates are higher when we classify with nearest centroids rather than random forests. Moreover, this different is more noticeable with information-theoretic algorithms (particularly CIFE) as random forests can treat features as new information (and therefore benefit from non-redundant features, even when they have weaker signal) while nearest centroids weighs every feature equally (which leads to worse performance when noisy features are added). See Figures 2.8, 2.9, 2.10, and 2.11.

2.7 Practical Implementation Issues

There are numerous issues that arise in the implementation of the information-theoretic algorithms in this chapter. These issues warrant further discussion as they have a significant impact on the output of these algorithms. We present our approach to addressing these issues in this section. Optimizations that improve runtime but do not affect the output of the algorithm are presented in Section 2.8.

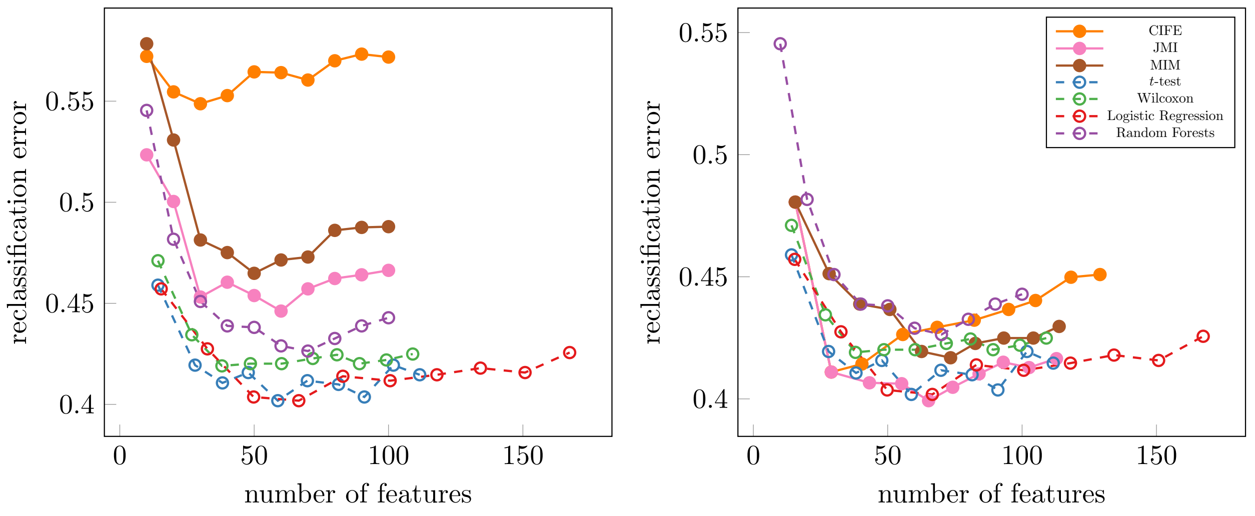


Figure 2.8: Classification Errors with Nearest Centroid Classifier For Paul Dataset (left: multiclass MI methods, right: binary MI methods)

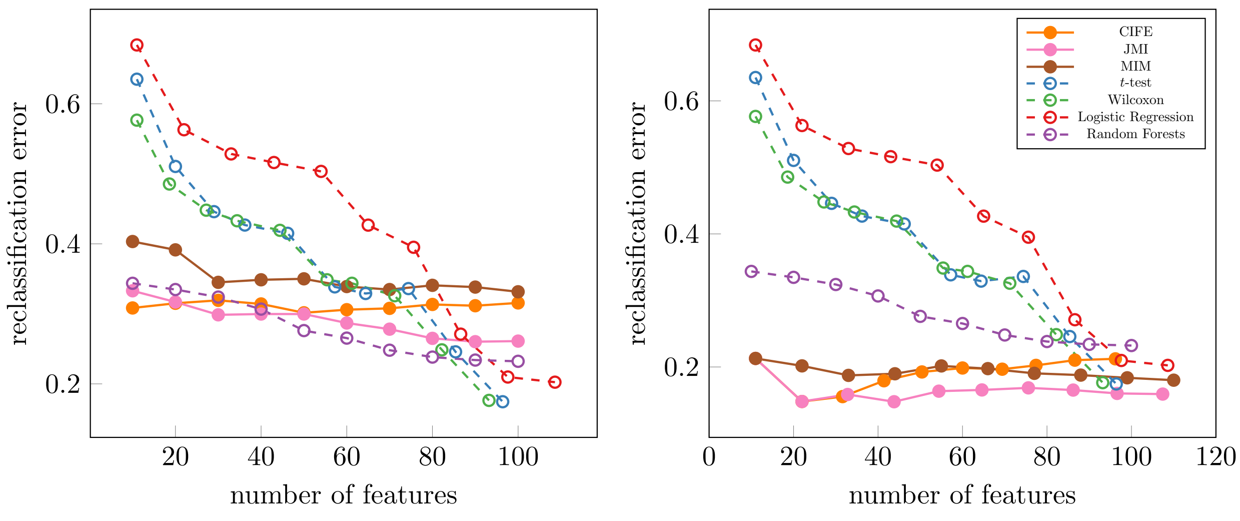


Figure 2.9: Classification Errors with Nearest Centroid Classifier For Green Dataset (left: multiclass MI methods, right: binary MI methods)

2.7.1 Binning Technique

All our discussion thus far has assumed that the x_i's are discrete random variables. When working with continuous data, it is common to discretize the data prior to running information theoretic methods. Although the datasets we have used are inherently discrete (with entries in $\mathbb{Z}_{\geq 0}$), there is a need to limit the number of values that entries can take. To understand why this is so important, suppose that the x_i's take integer values from 0 to $k-1$. In order to empirically compute $I(x_i; x_j; y)$, we need to approximate $P(x_i, x_j, y)$. Without any additional assumptions on the joint probability distribution, we are left to approximate this distribution by counting occurrences of each

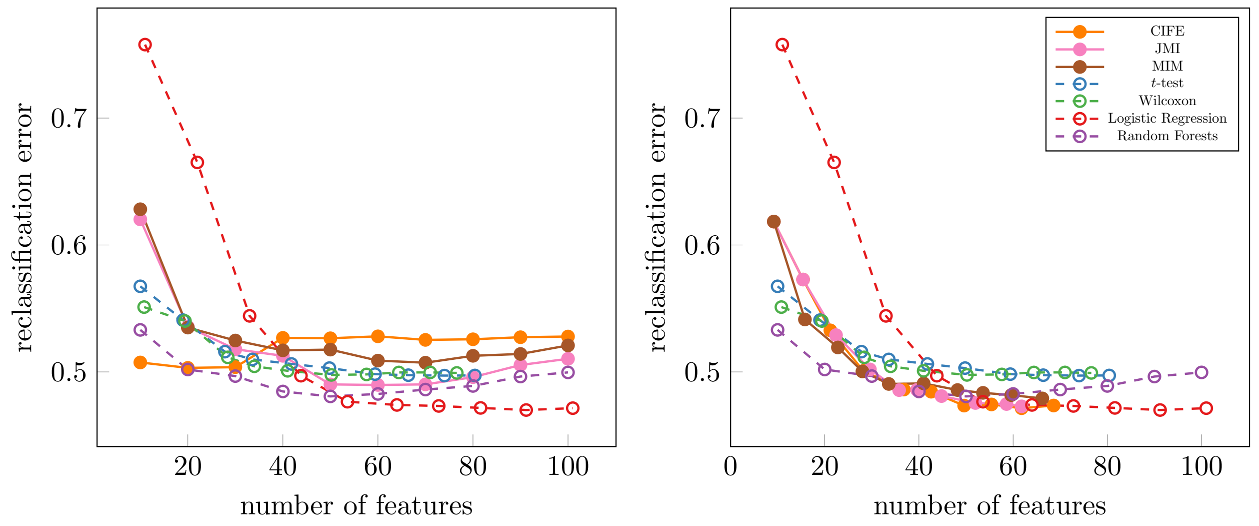


Figure 2.10: Classification Errors with Nearest Centroid Classifier For Zheng Dataset (left: multi-class MI methods, right: binary MI methods)

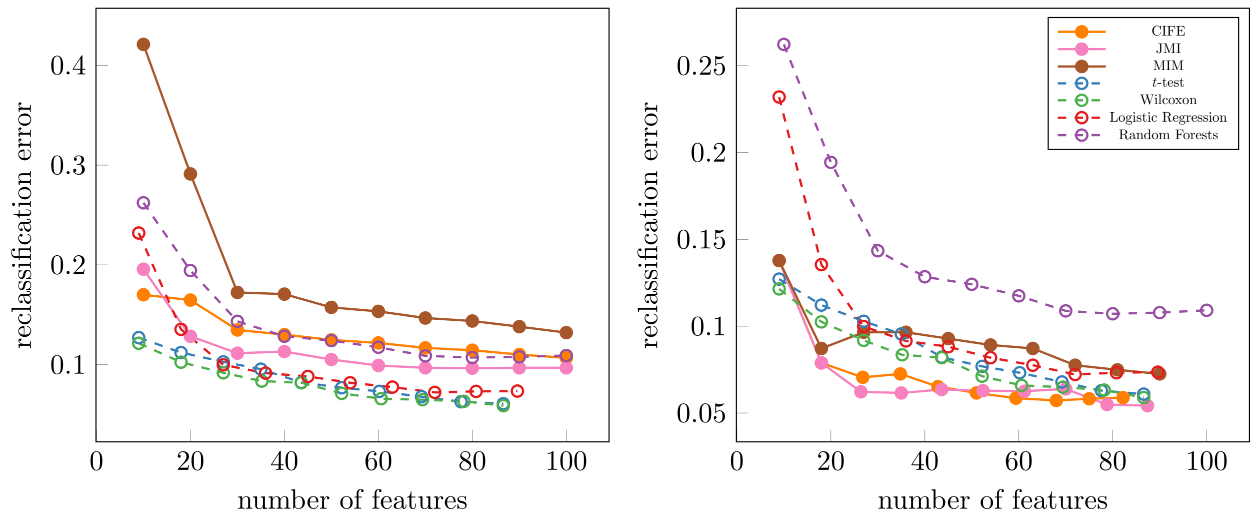


Figure 2.11: Classification Errors with Nearest Centroid Classifier For Zeisel Dataset (left: multi-class MI methods, right: binary MI methods)

possible combination, of which there are k^3. If our dataset has n observations, we need $n \gg k^3$ to have a reasonable approximation of $I(x_i; x_j; y)$. It is also important not to collapse too many values together as that would result in very poor resolution stifling our ability to observe significant differences. We recommend targeting $k = 5$ discrete values as a reasonable default.

The process of choosing bins has a surprisingly large impact on the performance of information-theoretic algorithms. For single cell RNA-sequencing, we propose a recursive quantile transformation that adjusts for the sparsity of the data. To motivate the need for our modification to the usual quantile transform, suppose we would like $k = 5$ bins for each feature. Because scRNA-seq datasets have a large number of zero entries (in some cases $> 90\%$ of entries are zero), the $\frac{1}{5}$-th, $\frac{2}{5}$-th, $\frac{3}{5}$, and $\frac{4}{5}$-th quantiles could all be 0. This leaves us with only two meaningful bins. To avoid this problem, we recommend a recursive strategy described in Algorithm 2 for selecting right endpoints for k bins. On our experiments, we perform this transformation on each feature individually to account for the different distributions of values across features.

Algorithm 2 Recursive-quantile transform

```
1: procedure QuantBins($C$, $k$)
2:     if $k \leq 1$ then
3:         return $(\max C, )$
4:     end if
5:     $q \leftarrow \frac{1}{k}$-th quantile of $C$
6:     $C' \leftarrow \{c \in C \mid c > q\}$
7:     return concatenate$((q, ), \text{QuantBins}(C', k-1))$
8: end procedure
```

2.7.2 Multiclass vs Binary Feature Selection Methods

The mutual information based feature selection methods are approximations of

$$\arg\max_{|S| \leq n} I(S; y).$$

Unlike many of the differential expression methods, the mutual information based algorithms allow for y to take more than two values (i.e., they are *multiclass* rather than *binary* methods). This, in theory, allows us to select features at once, rather than having to select them for every class label (one-vs-rest) or every pair of class labels (one-vs-one).

However, as we mentioned in Section 2.6, it is also possible to run these algorithms as binary methods by letting $\bar{y}_t = \mathbb{1}_{y=t}$ and treating $\bar{y}_t$ as the class labels. This gives us the opportunity both to compare binary differential expression methods to binary mutual information based methods and to evaluate strengths and weaknesses of multiclass methods vis-a-vis binary methods.

We saw that for almost all datasets (the only exception being the Zheng dataset for a small number of markers), turning the mutual information based methods into binary methods led to a significant improvement in performance. We describe some possible ways in which these differences in performance can be explained. Note that unlike in Section 2.3, this is not an exhaustive list of shortcomings. Instead, these examples demonstrate ways in which these approaches differ.

Sometimes features can be informative about multiple target labels. Unless the feature selection algorithm used is inherently multi-class, we cannot detect features that may have a weak signal about any particular target label but a strong signal that helps discern between sets of target labels.

Example 6. Let b_1, b_2 be random variables chosen from $\{0, 1\}$ independently with uniform probability. Let $y = b_1 b_2$ and let $x_i = \mathbb{1}_{i=(b_1 b_2)_{(b)}}$ for $i \in \{0, 1, \ldots, 3\}$. Here the subscript (b) denotes that we are looking at the binary representation of the number. For example,

$$x_2 = \begin{cases} 1 & \text{if } b_1 = 1, b_2 = 0 \\ 0 & \text{otherwise} \end{cases}.$$

Finally, let $x_4 = b_1$ and $x_5 = b_2$.

It is easy to see now that $I(x_4; y) = 1 = I(x_5; y)$. However, for $i \leq 3$ we have $I(x_i; y) = \frac{1}{4}\log(4) + \frac{3}{4}\log(\frac{4}{3}) \approx 0.811$. That is to say, multiclass methods will correctly identify x_4 and x_5 as being very informative about y.

On the other hand, if our class labels were binary, for $i \leq 3$ we have $I(x_i; \bar{y}_i) = \frac{1}{4}\log 4 + \frac{3}{4}\log\frac{4}{3} \approx 0.811$. However, $I(x_4; \bar{y}_i) = H(x_4) + H(\bar{y}_i) - H(x_4 \bar{y}_i) = 1 + \left(\frac{1}{4}\log 4 + \frac{3}{4}\log\frac{4}{3}\right) - \left(\frac{1}{2}\log(2) + \frac{1}{2}\log(4)\right) \approx 0.311$. A binary method would therefore choose $x_0, \ldots, x_3$ before x_4 and x_5.

It must be noted that if we run a binary method on the example above, we will eventually choose $\{x_0, x_1, x_2, x_3\}$, which, together, are as informative as $\{x_4, x_5\}$. The following modification demonstrates a situation in which this is not true.

Example 7. Let $c_1, c_2, \ldots$ be independent Bernoulli trials with probability p for some constant $p \in [0, \frac{1}{2}]$ (these will introduce noise in x_0, x_1, x_2, x_3). In other words, $\mathbb{P}(c_i = 1) = p$ and $\mathbb{P}(c_i = 0) = 1 - p$. Now follow the previous example, except let $x_i = c_i \oplus \mathbb{1}_{i=(b_1 b_2)_{(b)}}$ for $i \in \{0, 1, \ldots, 3\}$. Recall $\oplus$ is the XOR operator: $a \oplus b = a$ if $b = 0$ and $a \oplus b = \neg a$ if $b = 1$.

It is clear that for small values of p, a binary method will still choose $\{x_0, x_1, x_2, x_3\}$ instead of $\{x_4, x_5\}$, even though y is a deterministic function of x_4 and x_5 but not of $x_0, \ldots, x_3$. With a sufficiently large value of p (i.e., where there is enough noise in $x_0, \ldots, x_3$), even a binary method will choose x_4 and x_5

Recall that objectives like CIFE and JMI only consider low-degree interactions between features. Therefore, it is easy to under/over-estimate redundancies. By measuring relevance ($I(x_i; y)$ term) and redundancies ($I(x_i; x_j; y)$ terms) only in the context of a specific class label, binary methods are less vulnerable to accumulating errors in approximations.

We demonstrate this idea with two example. In the first example, we consider a univariate feature selection method (e.g., MIM) that does not penalize any redundancy (i.e., it underestimates redundancy) and outline a case where the binary version outperforms the multiclass version. In the second example, we demonstrate a situation in which a bivariate feature selection method (e.g., CIFE) overpenalizes redundancy in the multiclass version and the binary version yields a better selection of features.

Example 8. Let b_1, b_2, b_3 be random variables chosen from $\{0, 1\}$ independently with uniform probability. Let $y = b_1b_2b_3$ and let $t_i = \mathbb{1}_{i=(b_1b_2b_3)_{(b)}}$ for $i \in \{0, 1, \ldots, 7\}$. We will use the t_i's to define features. Let $x_i = t_0t_1t_2t_i$ for $i \in \{3, \ldots, 7\}$. Let $x_8 = t_4t_5$ and $x_9 = t_6t_7$. A univariate multiclass method (e.g., MIM) will associate a higher score to $x_3, \ldots, x_7$ because they are individually more informative about y than x_8 and x_9. However, a binary method (when considering the class label $y = 100$ or $y = 101$) will assign the same scores to x_8, x_4, and x_5. Similarly, it would assign the same scores to x_9, x_6, and x_7 when considering $y = 110$ and $y = 111$). To construct an example where the binary method will definitely pick x_8 and x_9 while the multiclass method won't, we can add noise (as in Example 7) to features $x_3, \ldots, x_7$.

In the previous example, we saw how binary methods, in searching for information specific to individual class labels, are able to avoid traps of redundant information. We now demonstrate an example in which binary methods avoid overpenalizing redundancy.

Example 9. Let b_1, b_2 be random variables chosen from $\{0, 1\}$ independently with uniform probability. Let c_2, c_3 be independent Bernoulli trials with probability $p \in [0, \frac{1}{2}]$. Let $y = b_1b_2$ and let $t_i = \mathbb{1}_{i=(b_1b_2)_{(b)}}$ for $i \in \{0, 1, \ldots, 3\}$. We will use the t_i's to define features. Let $x_1 = t_0t_1t_2$, $x_2 = t_0t_1t_3$, $x_3 = t_2 \oplus c_2$, and $x_4 = t_3 \oplus c_3$.

If we use a bivariate multiclass method (e.g., CIFE), without loss of generality we select x_1 at the first iteration (analogous argument if we selected x_2 first). Because $I(x_2; x_1; y)$ is significant, the penalty on x_2 (with CIFE, say) would be large enough that we would pick x_4 instead of x_2, which is more informative. The binary version of CIFE would not have this problem.

We posit that the Paul dataset, with its high cluster resolution (numerous clusters that cannot be separated easily) resembles Example 9 and believe that this (at least partially) explains the stark contrast in the reclassification accuracy between the multiclass and binary algorithms (especially CIFE).

2.8 Efficient Algorithms

2.8.1 Computing Entropy for Sparse Matrices

In this section, we outline a fast computation of entropy for sparse matrices. As a result of Proposition 2.1, this algorithm can be used to efficiently compute various information-theoretic objective functions for feature selection (especially low-dimension approximations, such as CIFE and JMI). In this section, we will use zero-indexing rather than one-indexing (i.e., the first element of an array A is $A[0]$).

Compressed Sparse Column (CSC) is a sparse matrix format that is optimized for column operations (e.g., slicing by columns, iterating over non-zero entries in a given column, etc.). It is defined as follows. For an $n \times m$ matrix M, the CSC format stores three arrays:

- X, the values of the non-zero entries stored in column major order.
- R, the row index of the non-zero entries in X (i.e., X and R have the same length).
- C, an array of length $m+1$ where $C[j+1]-C[j]$ is the number of non-zero entries in column j and $C[0]=0$.

In other words, for any s such that $C[j] \leq s < C[j+1]$, we have $X[s]$ is the value of a non-zero entry in row $R[s]$ and column j.

Assume all values in X come from $\{0, 1, \cdots, k-1\}$ (i.e., there are k distinct values in X). To efficiently compute the (empirical) entropy of an individual column j of an $n \times m$ matrix M, we follow Algorithm 3.

Algorithm 3 Entropy of a Single Column

```
1:  procedure ENTROPY($M = (X, R, C), j$)
2:      counts ← array (length $k$)
3:      counts[0] ← $n$
4:      for $C[j] \leq s < C[j+1]$ do
5:          counts[0] ← counts[0] - 1
6:          counts[X[s]] ← counts[X[s]] + 1
7:      end for
8:      $H \leftarrow 0$
9:      for nonzero $c$ in counts do
10:         $H \leftarrow H + (\frac{c}{n}) \log_2(\frac{c}{n})$
11:     end for
12:     return $H$
13: end procedure
```

We often need to compute the entropy of two or more columns. A trivial way to overcome this problem is to combine multiple columns into one column. One way to combine r columns is to write each entry as an r-tuple. This combined column will retain the sparse structure as the non-zero entries will correspond to rows in which at least one entry was non-zero.[5]

However, the process of computing the tuple representation of multiple columns can sometimes be repetitive. For example, to compute

$$\arg\max_{x_i} I(x_i; x_{j_1}; x_{j_2}; \cdots; x_{j_r}),$$

we need to compute $H(x_i; x_{j_1}; x_{j_2}; \cdots; x_{j_r})$ for all $x_i \in \mathcal{X}$. In such cases, it make sense to precompute the tuple representation of $x_{j_1}, \ldots, x_{j_r}$.

[5]In practice, is can be faster to represent $(a_0, a_1, \cdots, a_{r-1})$ as $a_0 + a_1 k + \cdots + a_{r-1} k^{r-1}$ where $k = \max_i(a_i) + 1$, but we will avoid this conversion for the sake of readability. See our source code for such an implementation.

However, in most cases, the column of class labels y tends not to be a sparse column. Combining y into the tuple representation means that most entries in the column become non-zero. This voids the benefits of exploiting the sparse structure of M. We can address these concerns by precomputing some information and vectorizing the entropy computation as in Algorithm 4. The ENTROPYWRT ("entropy with respect to") takes a list of columns $(x_{j_1}, x_{j_2}, \cdots, x_{j_r}, y)$ and returns the vector $[H(x_i x_{j_1} \cdots x_{j_r} y)]_{i=0}^{m-1}$. Pseudocode for ENTROPYWRT when the y column is not in the list of columns can be written with a trivial modification to Algorithm 4. Here we assume y has entries in $0, 1, \ldots, n_{\text{labels}} - 1$.

Algorithm 4 Entropy of Multiple Columns (all but one fixed)

```
1: procedure ENTROPYWRT($M = (X, R, C)$, $j_1, j_2, \ldots, j_r$, $y$ )
2:     H ← zero-filled array (length $m$)
3:     P ← sparse tuple representation of $[M_{ij}]_{j \in \{j_1, \ldots, j_r\}}$
4:     ycounts ← array (length $n_{\text{labels}}$)
5:     for yval in y do
6:         increment ycounts[yval]
7:     end for
8:     for $0 \leq j < m$ do
9:         counts ← Map: $(r+2)$-tuples → $\mathbb{Z}_{\geq 0}$
10:        for $0 \leq$ yval $< n_{\text{labels}}$ do
11:            counts[(yval, 0, ..., 0)] ← ycounts[yval]
12:        end for
13:        curind ← 0
14:        coljindices ← where ($j \neq 0$)
15:        colPindices ← where ($P \neq 0$)
16:        while exists $x \in$ colPindices ∪ coljindices, $x >$ curind do
17:            curind ← min { $x : x \in$ colPindices ∪ coljindices, $x >$ curind }
18:            decrement counts[(y[curind], 0, ..., 0)]
19:            increment counts[(y[curind], P[curind], M[curind, $j$])]
20:        end while
21:        for $c$ in counts, $c > 0$ do
22:            $H[j] \leftarrow H[j] + (\frac{c}{n}) \log_2(\frac{c}{n})$
23:        end for
24:    end for
25:    return $H$
26: end procedure
```

The time complexity of ENTROPYWRT is $O(n + mp + q + mk^{r+1})$ where p is the number of rows with non-zero entries in columns $j_1, j_2, \ldots, j_r$, and q is the number of non-zero entries in M. Typically, ENTROPYWRT will be called numerous times, in which case ycounts should be precomputed once, rather than on each call to ENTROPYWRT. In this case, the time complexity is $O(mp + q + mk^{r+1})$. Also note that when we use ENTROPYWRT, we will typically fix k

beforehand (depending of discretization strategy) to be about 5. For CIFE and JMI algorithms, we have $r = 2$ because we will only ever compute the joint entropy of three columns or less. In these cases, the time complexity is $O(mp + q)$.

2.8.2 Efficient Optimization of CIFE and JMI

The CIFE and JMI algorithms compute $\arg\max_{x_i \notin S} J(x_i)$ at each iteration. If done naively, computing $J(x_i)$ can take longer as S grows because $J_{\text{cife}}(x_i) = I(x_i; y) - \sum_{x_j \in S} I(x_i; x_j; y)$ and $J_{\text{jmi}}(x_i) = I(x_i; y) - \frac{1}{|S|} \sum_{x_j \in S} I(x_i; x_j; y)$ have $|S| + 1$ terms. The computational complexity to select t features is $O(mt^2 f)$ where f is the time to compute $I(x_i; x_j; y)$ and is a function of n, k, and sparsity parameters of columns in X. A minor modification to vectorize this process can lower the complexity to $O(mtf)$, as shown in Algorithm 5.

Algorithm 5 Vectorization of CIFE/JMI

```
 1: procedure FEATURESELECTION(X, t, method)
 2:     $S \leftarrow \emptyset$
 3:     basescore $\leftarrow [I(x_i; y)]_{i=0}^{m-1}$
 4:     penalty $\leftarrow$ zero-filled vector (length $m$)
 5:     score $\leftarrow$ basescore
 6:     for $0 \le \ell < t$ do
 7:         s $\leftarrow \arg\max_{i \notin S}$ score[i]
 8:         penalty $\leftarrow$ penalty + $[I(x_s; x_j; y)]_{j=0}^{m-1}$
 9:         if method = "CIFE" then
10:             score $\leftarrow$ basescore $-$ penalty
11:         else if method = "JMI" then
12:             score $\leftarrow$ basescore $-\frac{1}{|S|}$ penalty
13:         end if
14:     end for
15: end procedure
```

Note that $I(x_s; x_j; y) = I(x_j; y) + H(x_s) - H(x_s, x_j) - H(x_s, y) + H(x_s, x_j, y)$. Since $I(x_j; y)$ was computed as `basescore`, and $H(x_s) - H(x_s; y)$ is constant with respect to j, the only two calls to ENTROPYWRT are for $H(x_s, x_j)$ and $H(x_s, x_j, y)$.

Under certain conditions, this process can be made even faster with a strategy adapted from the fast algorithm for CMIM in [Fle04]. In the aforementioned paper, the authors propose a objective function that can be rewritten as $J_{\text{cmim}}(x_i) = I(x_i; y) - \max_{x_j \in S} I(x_i; x_j; y)$. They improve the efficiency of the naive algorithm by updating $J_{\text{cmim}}(x_i)$ (i.e., checking if the penalty term has increased since more features were selected into S) only $J(x_i) > J(x_j)$ for all $j < i$. This prevents some unnecessary evaluations of $I(x_i; x_j; y)$. This trick hinges on the monotonicity of $\max_{x_j \in S} I(x_i; x_j; y)$ with respect to S. In our case $\sum x_j \in SI(x_i; x_j; y)$ is monotone if $I(x_i; x_j; y) \ge 0$. If we do make this assumption (as we did in Proposition 2.9), we could make a similar optimization. In fact, we believe such an optimization would be best implemented using a priority queue.

2.9 Problems with mRMR

A false theorem in [PLD05, §2.3] states that maximizing the mRMR objective in a greedy search is "equivalent" to maximizing the CMI objective in a greedy search. We believe that the most reasonable interpretation of "equivalent" is that given the same input, the two algorithms will return the same output. In this case, their claim (and the proof thereof) is incorrect. We have already seen in Examples 1 and 2 cases where the mRMR objective makes different choices than the CMI objective. To understand why these examples are counterexamples to the claim in [PLD05], we walk through their proof.

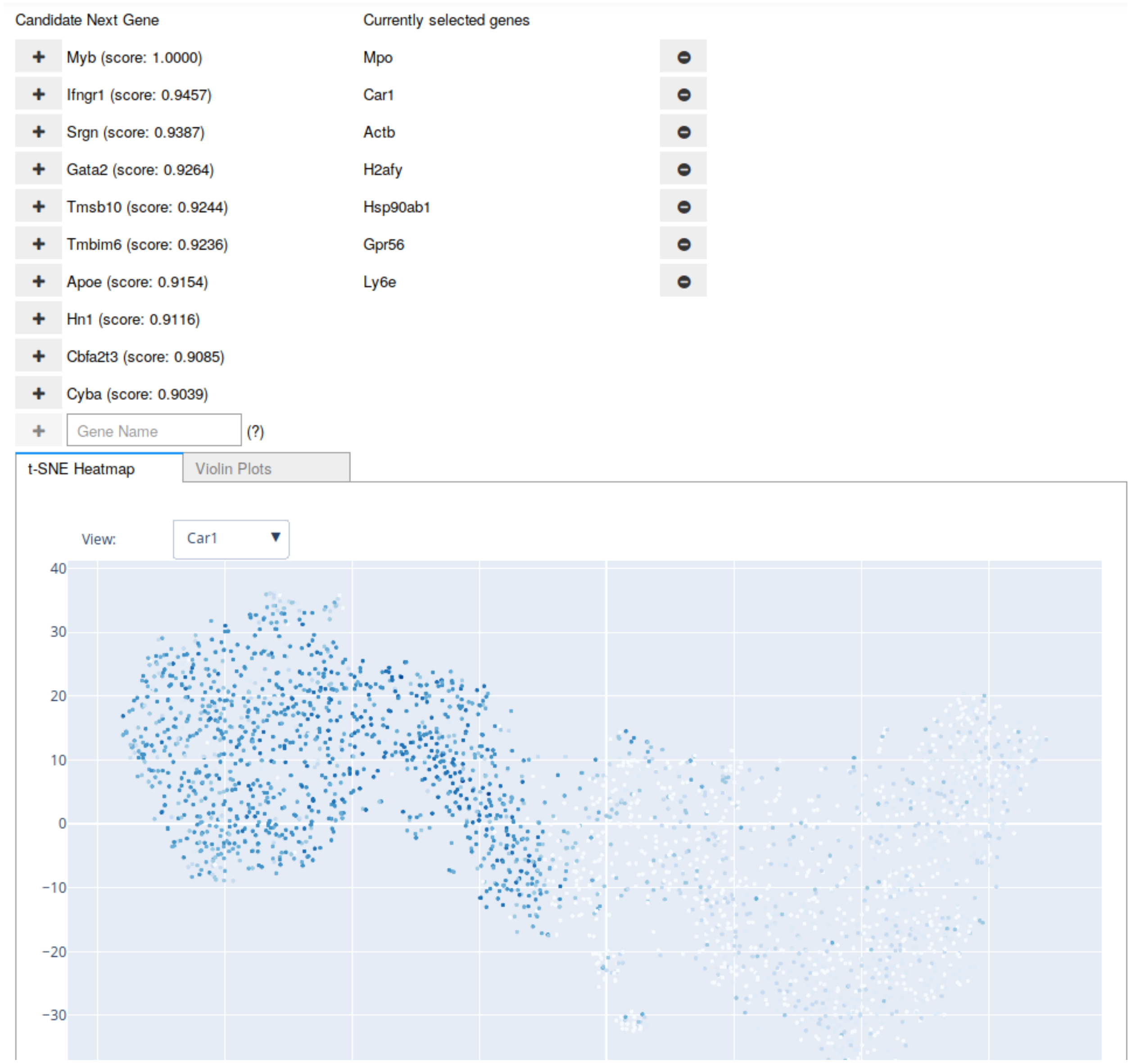
For notational convenience, we let $\mathcal{X}$ be the set of all features, $S = S_{m-1}$ be the set of $m-1$ features selected after $m-1$ steps, $D(x_i) = I(x_i; y)$ be the *relevance* of feature x_i and $R(x_i) = \frac{1}{|S|}\sum_{x_j \notin S} I(x_i; x_j)$ be the *redundancy* of feature x_i. They definite a quantity $J : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ and show that $I(S_{m-1} \cup \{x_i\}; y) = J(S_{m-1}, x_i, y) - J(S_{m-1}, x_i)$. Fixing S_{m-1}, they then show that the first term, $J(S_{m-1}, x_i, y)$, is maximized when $D(x_i) = I(x_i; y)$ is maximized, and the second term, $J(S_{m-1}, x_i)$ is minimized when $R(x_i) = \frac{1}{|S|}\sum_{x_j \notin S} I(x_i; x_j)$ is minimized. They then falsely conclude that

$$\arg\max_{x_i \notin S_{m-1}} I(S_{m-1} \cup \{x_i\}; y) = \arg\max_{x_i \notin S_{m-1}} D(x_i) - R(x_i). \tag{2.9}$$

2.10 PicturedRocks: Information-Theoretic Feature Selection Software for Single Cell RNA-Sequencing

As part of the work in this chapter of the dissertation, I developed PicturedRocks, a Python package with efficient implementations of the algorithms in this chapter. PicturedRocks allows users to import data using the widely-used ScanPy package [WAT18] and perform various information-theoretic feature selection algorithms. In addition to implementing CIFE, JMI, and MIM, it also supports the degree 1 and degree 2 approximations to unsupervised feature selection as described in Section 2.4.2. PicturedRocks can compute empirical entropy over columns of sparse matrix natively, which significantly reduces the runtime and memory usage.

PicturedRocks also features an interactive marker selection user interface that allows biologists to use domain knowledge to select markers in a data-informed (rather than data-driven) manner. The user interface (see Figure 2.12) presents the user with candidate genes that they can add to their current selection of genes, updating the candidate genes after every change to the current selection of genes. A visualization of various statistics about candidate genes are available at each step. This



Candidate Next Gene	Currently selected genes
Myb (score: 1.0000)	Mpo
Ifngr1 (score: 0.9457)	Car1
Srgn (score: 0.9387)	Actb
Gata2 (score: 0.9264)	H2afy
Tmsb10 (score: 0.9244)	Hsp90ab1
Tmbim6 (score: 0.9236)	Gpr56
Apoe (score: 0.9154)	Ly6e
Hn1 (score: 0.9116)	
Cbfa2t3 (score: 0.9085)	
Cyba (score: 0.9039)	

Figure 2.12: A screenshot of the PicturedRocks Interactive Marker Selection tool on the Paul dataset.

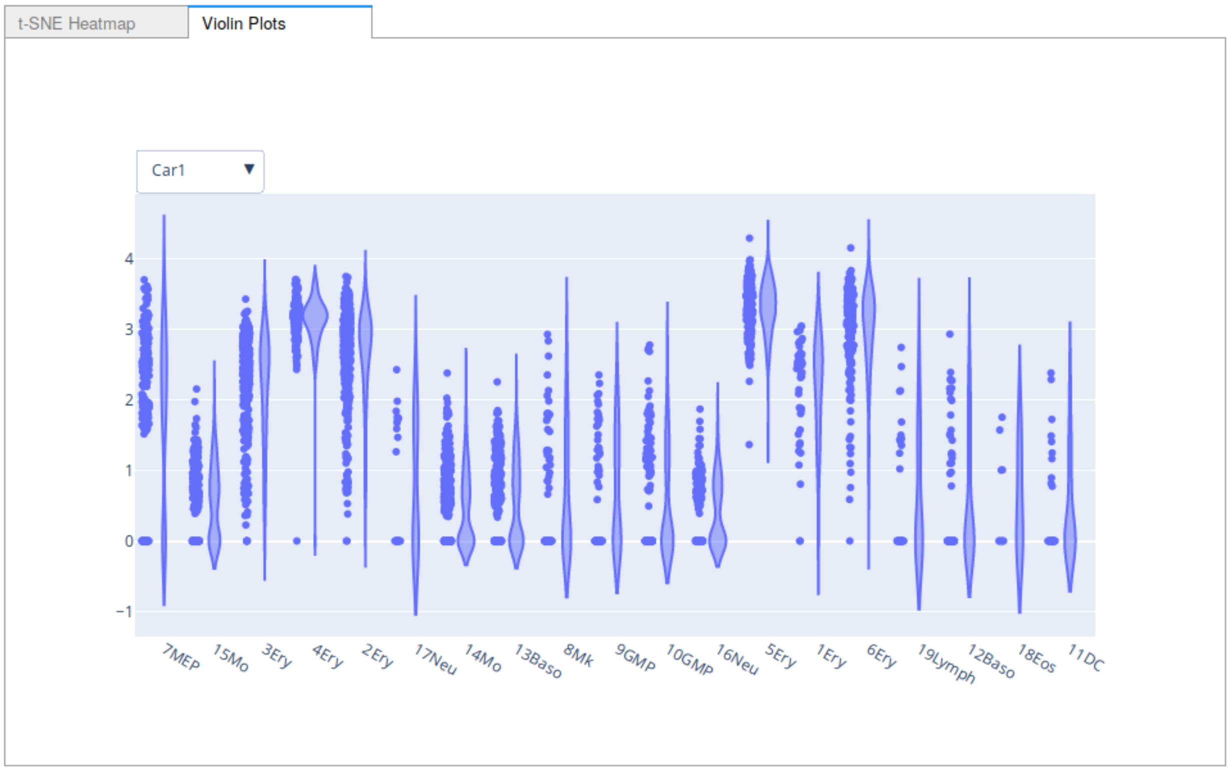


Figure 2.13: A screenshot of the “Violin Plots” tab of the PicturedRocks Interactive Marker Selection tool on the Paul dataset.

allows biologists not only to understand the algorithm more intuitively but to overrule the decisions at various steps (for example, to select a biologically verified marker gene over a correlated gene) but continue to use the results of the algorithms. The interactive tool is modular and users may add their own visualization tools to the interface (see for example the violin plot in Figure 2.13).

To help perform many of the benchmarking experiments in this chapter, PicturedRocks has a variety of tools for evaluating feature selection methods using cross-validation. These include tools that store and recover folds for k-fold cross-validation, select markers for each fold, and train a classifier and classify points using the markers selected. In addition to automating this process, these tools can serialize output at various stages of the process which allows users to parallelize parts of this experiment through multiple jobs on an high performance computing (HPC) cluster.

Software packages such as PicturedRocks written for niche audiences often suffer from reproducibility problems especially because of dependency and compatibility issues. I have taken the following steps to minimize the extent to which these issues affect PicturedRocks.

- To allow researchers anywhere to reproduce the results in this chapter, I have made a “Docker” image[6] with PicturedRocks installed. It includes code that downloads the datasets in Section 2.6, performs all preprocessing steps, and runs the various analyses in Section 2.6. I have also made the code for these analyses directly available[7].

- To ensure that every dependency is explicitly stated, that PicturedRocks remains compatible with its dependencies as they change, and that core features of PicturedRocks are regularly tested, I have used a continuous integration tool[8] that regularly builds PicturedRocks, install dependencies, and runs unit tests in a minimal environment.

- PicturedRocks is extensively documented[9] with instructions for installing, an API reference, and a tutorial on using PicturedRocks for feature selection.

2.11 Conclusion

The opportunity presented by single cell RNA-sequencing also poses challenges in its analysis. While traditional differential expression reveal meaningful information about the data, we believe that a multiplicity of feature selection techniques allows researchers to discover novel marker genes. Specifically, we believe that information-theoretic feature selection algorithms can detect marker genes that may otherwise be obscured by interactions with other genes or distributional assumptions of statistical tests.

[6]https://hub.docker.com/r/umangv/picturedrocksbenchmarks
[7]https://github.com/umangv/picturedrocksbenchmarks/
[8]https://travis-ci.com/umangv/picturedrocks
[9]https://picturedrocks.readthedocs.io/en/latest/

We do not expect (in fact, we believe it is unrealistic to hope) that a single marker selection technique is appropriate for all purposes and datasets. The algorithms in this chapter (especially CIFE) have proven to select sets of genes that are very distinct from those selected by traditional differential expression methods. In most cases the reclassification accuracy when restricting datasets to the chosen genes was comparable between information-theoretic algorithms and traditional differential expression algorithms and in some cases information-theoretic algorithms performed considerably better.

The reader will likely have noted that there are numerous possible combinations of configurations and versions of these algorithms to choose from. We attempt to summarize these options in the form of suggestions below.

- MIM is a univariate algorithm. Because it does not consider interactions between genes, it yields somewhat similar results to traditional differential expression algorithms, especially when run as a binary method rather than a multiclass method. As a multiclass method, it is particularly susceptible to selecting features with a strong signal without any consideration of how informative those features are together or what class labels they are informative about. Therefore, the output of MIM is more meaningful and easier to interpret when it is run as a binary method.

- CIFE is a bivariate algorithm that aggressively penalizes redundancy. When used as a multiclass method, it is particularly strong at identifying a small set of informative genes (in our examples, CIFE often far outperformed other algorithms when selecting 10 to 30 genes).

  CIFE's ability to consider interactions allows it to pick genes that may not be as informative in isolation but are informative in the context of other selected genes. However, when selecting a large number of genes, the interactions (particularly redundancies) overwhelm the algorithm and it eventually picks very uninformative features. This is especially true when individual genes have information about numerous class labels (for example, when there are many class labels in the dataset; see Example 9).

  If we need a larger set of genes or want marker genes for specific class labels, we recommend using CIFE as a binary feature selection method. Running the algorithm as a binary method ensures that the number of features chosen per class label remains relatively low.

- JMI is a bivariate algorithm that acts as an intermediate between CIFE and MIM. As the algorithm selects more features, it dampens the impact of interactions resulting, eventually, in a convergence in behavior between JMI and MIM. As a consequence of this damping, it is unlikely to unearth informative features hidden behind interactions as CIFE can nor does its output reflect the strength of the signal of features in isolation. The damping in JMI is a heuristic that is not justified theoretically and should be used cautiously. However, empirical results show that its performance is consistently decent and avoids the occasional extreme poor performance of CIFE.

In all of these cases, we strongly recommend using the recursive quantile transform we describe in Section 2.7.1 to discretize datasets.

Used in conjunction with traditional differential expression methods, the information-theoretic algorithms in this chapter have the potential to discover novel marker genes that reveal information that otherwise be lost on differential expression algorithms that prioritize signal strength of genes over new information. Future directions include more sophisticated analysis and benchmarking of unsupervised feature selection methods for single cell RNA-sequencing.

CHAPTER 3

Ranking from Pairwise Comparisons

3.1 Introduction

Rank aggregation is the problem of determining an ordering or preference scores for n items based on various kinds of comparison data. Learning such rankings has applications in sport ranking [ES08], web search [Dwo+01], and social sciences [KS40], [Thu45]. Often these comparisons are pairwise comparisons: for example, in a league of n sports teams that play total of m games and in the k-th game, which is between i_k and j_k, one of the two is judged the winner. Alternatively, we may wish to understand user preferences and display a pair of items on a website and record which item the user click on. When gathering user feedback on n items, it is helpful to present comparisons pairwise as opposed to asking for scores on individual items as standards' for assigning scores vary from user to user and can shift over time, especially with exposure to more items. In both of these examples, we should not expect our observations (pairwise comparisons) to be self-consistent (for example item i may beat item j only a fraction of the time, or in some cases, the probability that i beats j and that j beats ℓ are both greater than $\frac{1}{2}$ but the probability that i beats ℓ may not be), but we still believe that asking to rank these items is a meaningful question. There are numerous frameworks in the literature that account for this lack of consistency. For a survey on some of the most prominent frameworks, we refer the reader to [RA14].

A variant of this problem also falls under the category of active learning (for example, the extreme case when the outcome of each observation is completely deterministic and i beats j in an observation if and only if i should be ranked higher than j can be handled by traditional sorting algorithms where the algorithm can choose which pairs are compared). While it is conceivable that active learning approaches [JN11] to this problem could perform better than non-adaptive algorithms that do not influence the choice of pairs that are compared, there is little evidence that active learning algorithms can significantly outperform non-adaptive algorithms in practice [Kat+18].

The contributions of this work fall under three broad categories.

1. We analyze the RankCentrality algorithm [NOS16] under a natural sampling scheme considered by [RA14] and improve their sample complexity upper bounds from $O(n^5 \log n)$ to

$O(n \log n)$ using proofs that are significantly simpler that those in [NOS16] and [RA14].

2. We introduce a λ-regularized RankCentrality algorithm that, unlike unregularized RankCentrality, is capable of returning non-trivial output even when the number of observations is very small. Our theoretical analysis of this algorithm relies on our proofs of convergence for RankCentrality.

3. We propose similarity-based regularization for RankCentrality that can leverage features and similarity information to impute pairwise comparisons and vastly improve the predicted ranking when the number of pairwise comparisons is small. We give empirical evidence that similarity-based regularization can outperform SVM and Siamese Networks for certain kinds of datasets.

One framework of particular interest to us is the Bradley-Terry-Luce (BTL) model, which assumes that each item i has a positive score w_i and the probability that j is preferred to i ("j beats i") in a comparison is

$$P_{ij} := P(i \prec j) = \frac{w_j}{w_i + w_j} \tag{3.1}$$

(here $\prec$ isn't comparing i and j as integers, rather refers to the event that j is preferred to i in an individual observation). In this chapter, we propose algorithms to approximate w (and hence a linear ordering on the n items) from a set of pairwise comparisons. We analyze an unbiased estimator, RankCentrality (modified slightly from the literature), and present a novel approach to regularizing RankCentrality that addresses a major problem practitioners will encounter when the number of pairwise comparisons is small relative to the number of items. In addition to giving a bias-variance trade-off for the latter algorithm (which we call λ-regularized RankCentrality), we also demonstrate the power of our regularization technique that allows for creative ways to regularize RankCentrality when other information (such as features for the n items being compared) is available.

There are multiple methods to learn w given pairwise comparisons $(i_1, j_1, y_1), \ldots, (i_m, j_m, y_m)$, where the 3-tuple (i_k, j_k, y_k) denotes that the k-th comparison was between items i_k and j_k, and $y_k = 0$ denotes that i_k was preferred in this observation, whereas $y_k = 1$ denotes that j_k was preferred. Because w_i's are all positive, we often write $v_i = \ln(w_i)$ and

$$P(i \prec j) = \frac{e^{v_j}}{e^{v_i} + e^{v_j}} = \frac{1}{1 + e^{v_i - v_j}}.$$

An obvious approach, if we assume the BTL model, is to maximize (log-)likelihood. Specifically, solve

$$\arg\max_{v \in \mathbb{R}^n} \sum_{i=1}^{k} \log\left(\frac{1}{1 + e^{y'_k (v_{j_k} - v_{i_k})}}\right) = \arg\max_{v \in \mathbb{R}^n} \sum_{i=1}^{k} \log \operatorname{expit}\left(y'_k v^T (e_{j_k} - e_{i_k})\right). \tag{3.2}$$

Here $y'_k := 2y_k - 1$ is used to ensure that y'_k takes values in $\{1, -1\}$. We have now reduced this to logistic regression where each observation (i_k, j_k, y_k) is represented as $e_{i_k} - e_{j_k}$ (here $\{e_i\}_{i=1}^n$ is the elementary basis on $\mathbb{R}^n$). The target label in this case is y_k (but can be $1 - y_k$ if we represent the input instead as $e_{j_k} - e_{i_k}$).

Another method, RankCentrality, originally proposed by Negahban, et al. in [NOS16], learns the ranking as the stationary distribution of a Markov chain. Let $\hat{P}_{ij}$ be the proportion of comparisons between i and j in which j was preferred to i (if there are no comparisons between i and j, we let $\hat{P}_{ij} = 0$). We call $\hat{P}$ the *empirical comparison matrix* and distinguish it from P, the *pairwise preference matrix* (defined in (3.1) above). Now construct a directed weighted graph G with vertices $\{1, \ldots, n\}$. If i and j were compared (equivalently, if $\hat{P}_{ij} + \hat{P}_{ji} = 1$) then add an edge from i to j with weight $\hat{P}_{ij}$ and an edge from j to i with weight $\hat{P}_{ji}$. Let $d_{\max}$ be the maximum out-degree of a node on G. Finally, we definite the time-invariant transition matrix for the Markov chain as

$$\hat{Q}_{ij}^{\text{NOS}} = \begin{cases} \frac{1}{d_{\max}} \hat{P}_{ij} & \text{if } i \neq j \\ 1 - \sum_{k \neq i} \frac{1}{d_{\max}} \hat{P}_{ik} & \text{if } i = j. \end{cases} \tag{3.3}$$

(The second case ensures row-stochasticity of $\hat{Q}^{\text{NOS}}$). We will use $\hat{Q}^{\text{NOS}}$ to denote Negahban, Oh, and Shah's construction of the transition matrix in order to disambiguate between the different constructions of the transition matrix that we will present in this chapter. The RankCentrality estimator is $\hat{\pi}$, the unique stationary distribution of this Markov chain, provided this exists. In [NOS16], Negahban, et. al. prove that as the number of pairwise comparisons increase, the vector $\hat{\pi}$ approaches the vector of BTL scores w (appropriately scaled).

Algorithm 6 RankCentrality algorithm

```
1: procedure RankCentrality($n, S = \{(i_k, j_k, y_k)_k\}$)
2:     compute $\hat{Q}$ as in (3.5) (for [NOS16] compute $\hat{Q}^{\text{NOS}}$ as in (3.3), for [RA14] compute $\hat{Q}^{\text{RA}}$
       as in (3.4))
3:     if $\hat{Q}$ defines an ergodic (irreducible and aperiodic) Markov chain then
4:         return leading left eigenvector of $\hat{Q}$ (i.e., unique stationary distribution of $\hat{Q}$)
5:     else
6:         return uniform distribution $[1/n]_{i=1}^n$
7:     end if
8: end procedure
```

After the original paper on RankCentrality by Negahban, et. al. [NOS16], Rajkumar and Agarwal [RA14] proved convergence of a modified version of this algorithm under a different sampling scheme. In the following paragraphs, we explain the differences in how they observe pairwise comparisons and in their algorithms. We will also highlight where both their approaches fall short and present various improvements to the RankCentrality algorithm.

Negahban, et. al. [NOS16] assume that pairwise comparisons are observed by constructing an Erdős–Rényi graph on n vertices and for each edge (pair) observing k comparisons. This involves fixing a constant $p \in [0,1]$, the probability that an edge (pair) is included in the graph independent of all other edges. We believe that the sampling scheme proposed by Rajkumar and Agarwal [RA14] is more practical: they assume that each pair (i,j), where $i<j$, is observed with probability μ_{ij}, independent of previous observations. Here $\sum_{1\le i<j\le n}\mu_{ij}=1$ and each $\mu_{ij}>0$. Observe that in Rajkumar and Agarwal's sampling scheme each observation is made independent of the other observations, which is not true of Negahban, et. al.'s scheme. This observation will be important in our analysis later in the chapter.

Since Rajkumar and Agarwal do not explicitly construct an underlying graph as part of their sampling scheme, their algorithm varies slightly. They construct Q^{RA} in terms of P and $\hat{Q}^{\mathrm{RA}}$ by computing the empirical comparison matrix $\hat{P}$:

$$\hat{Q}^{\mathrm{RA}}_{ij} = \begin{cases} \frac{\hat{P}_{ij}}{n} & \text{if } i \neq j \\ 1-\sum_{\ell\neq i}\frac{\hat{P}_{i\ell}}{n} & \text{if } i=j. \end{cases}, \qquad \text{and} \qquad Q^{\mathrm{RA}}_{ij} = \begin{cases} \frac{P_{ij}}{n} & \text{if } i \neq j \\ 1-\sum_{\ell\neq i}\frac{P_{i\ell}}{n} & \text{if } i=j. \end{cases}. \tag{3.4}$$

As the reader might expect, the estimator for Rajkumar and Agarwal's algorithm is limiting distribution of $\hat{Q}^{\mathrm{RA}}$. However, Rajkumar and Agarwal add an important caveat that their algorithm outputs $\frac{1}{n}\mathbf{1}=[\frac{1}{n}]_{i=1}^{n}$ if $\hat{Q}^{\mathrm{RA}}$ is not an ergodic markov chain (recall that an ergodic markov chain is such that every state is aperiodic and positive recurrent, which together guarantee a unique stationary distribution).

For completeness, we state the results by both Negahban, et. al. [NOS16] and Rajkumar and Agarwal [RA14] in the notation that we use in this chapter. First, we state Negahban, et. al.'s theorem for random Erdős–Rényi graphs. We state this result instead of their result on general graphs because that requires that a graph be constructed under certain constraints and states the result in terms of the spectral gap of the graph Laplacian, which is harder to interpret and compare to other results both in this chapter and in [RA14].

Theorem (Negahban, et. al. [NOS16], Theorem 2)**.** *Given n objects, let the comparison graph $G=([n],E)$ be generated by selecting each pair (i,j) to be in E with probability d/n independently of everything else. Each such chosen pair of objects is compared k times with the outcomes of comparisons produced as per a BTL model with parameters $w_1,\ldots,w_n$. Then, if $d\ge 10C^2\log n$ and $k\,d\ge 128C^2b^5\log n$ (here $b=\max_{i,j}w_i/w_j$), the following bound on the error rate holds with* *probability at least* $1 - 10n^{-C/8}$*:*

$$\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \leq 8Cb^{5/2}\sqrt{\frac{\log n}{k\,d}}.$$

In order to make a reasonable comparison between this result and the following, we note that the number of pairwise comparisons made in the above sampling scheme is $O(nkd)$. While noting that the sampling scheme above deviates significantly from the one proposed by Rajkumar and Agarwal (which we also use in our analysis), we can informally conclude that the sample complexity for the above version of the RankCentrality algorithm is $O(n \log n)$.

Now the result in Rajkumar and Agarwal [RA14].

Theorem (Rajkumar and Agarwal [RA14], Theorem 7)**.** *Let* $\mu_{\min} := \min_{ij} \mu_{ij}$*. Assume* $\mu_{\min} > 0$*. Here* μ_{ij} *is the probability that the pair* (i, j) *is chosen for a given comparison. Let* $P_{ij} = w_j/(w_i + w_j)$ *as per the BTL model. Let* Q^{RA} *and* $\hat{Q}^{\mathrm{RA}}$ *be defined as in* (3.4) *and let* π *and* $\hat{\pi}$ *be their unique stationary distributions (if* $\hat{Q}^{\mathrm{RA}}$ *is not ergodic, let* $\hat{\pi} = \frac{1}{n}\mathbf{1}$*. Let* $P_{\min} = \min_{i \neq j} P_{ij}$*,* $\pi_{\max} = \max_i \pi_i$*, and* $\pi_{\min} = \min_i \pi_i$*. Let* $0 < \varepsilon \leq 1$ *and* $\delta \in (0, 1]$*. If*

$$m \geq \max\left(\frac{1024n}{\varepsilon^2 P_{\min}^2 \mu_{\min}^2}\left(\frac{\pi_{\max}}{\pi_{\min}}\right)^3 \ln\left(\frac{16n^2}{\delta}\right), 3\left(\frac{12}{\mu_{\min}^2} + 3\right)\ln\left(\frac{12}{\mu_{\min}^2} + 3\right)\right)$$

then with probability at least $1-\delta$ *(over the random draw of* m *samples from which* $\hat{P}$ *is constructed), the score vector* $\hat{\pi}$ *produced by the rank centrality algorithm satisfies*

$$\|\hat{\pi} - \pi\|_2 \leq \varepsilon.$$

The sample complexity here scales as $O(n^5 \log n)$ since $\mu_{\min}^{-1} \geq \binom{n}{2}$, with equality achieved only when μ is uniform. Intuition suggests that a much stronger bound is possible, which is indeed the case. We modify the RankCentrality algorithm yet again and demonstrate that under Rajkumar and Agarwal's sampling scheme, we are able to give a $O(n \log n)$ sample complexity bound. It is also worth noting that Rajkumar and Agarwal give their bound as $\|\pi - \hat{\pi}\| \leq \varepsilon$ instead of $\|\pi - \hat{\pi}\|/\|\pi\| < \varepsilon$. This makes their result weaker because if π is close to uniform, $\|\pi\| \approx n^{-1/2}$.

Algorithm 7 λ-regularized RankCentrality algorithm

```
1: procedure RankCentrality($n, S = \{(i_k, j_k, y_k)_k\}, \lambda$)
2:     compute $\hat{Q}$ as in (3.5)
3:     set $D_\lambda \leftarrow (1-\lambda)I + \frac{\lambda}{n}\mathbf{1}\mathbf{1}^T$
4:     return leading left eigenvector of $\hat{Q}D_\lambda$
5: end procedure
```

A crucial point to note is that in both of these cases, it is assumed that the empirical estimate $\hat{Q}$ of the markov transition matrix is ergodic. When the number of comparisons m is small, this is usually not the case. In the same way that linear classifiers (such as logistic regression and linear SVM) need regularization when the dimension of the feature space exceeds the number of samples, our λ-regularized RankCentrality algorithm (Section 3.3) addresses this concern. This is a primary motivation for the work in this chapter.

Our work in this chapter begins by modifying RankCentrality and giving a stronger sample complexity bound.

Theorem (Corollary 3.9). *Fix $\delta \in (0,1)$ and $\varepsilon \in (0,1)$. If*

$$m \geq 32 b^3 \varepsilon^{-2} n^{-1} \mu_{\min}^{-1} \log \frac{2n}{\delta}$$

and the empirical Markov chain $\hat{Q}$ constructed as in (3.5) *is ergodic, then with probability at least $1-\delta$, we have*

$$\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \leq \varepsilon.$$

We then introduce the λ-regularized RankCentrality algorithm (Algorithm 7), which constructs a Markov chain that is guaranteed to be ergodic, and give the following sample complexity bound (note that $\varepsilon > 2\lambda\gamma^{-1}$, which reflects that this is a biased estimator).

Theorem (Corollary 3.12). *Recall $\gamma = \frac{n\mu_{\min}}{2(1+\sqrt{2})b^{3/2}}$. Let $\lambda \in (0, \frac{\gamma}{2})$. Choose $\delta \in (0,1)$ and $\varepsilon \in (2\lambda\gamma^{-1}, 1)$. If*

$$m > \frac{52(1-\lambda)b^3}{n\mu_{\min}\left(\varepsilon - 2\lambda\gamma^{-1}\right)^2}$$

then with probability at least $1-\delta$, we have

$$\frac{\|\tilde{\hat{\pi}} - \pi\|}{\|\pi\|} \leq \varepsilon.$$

Finally, in Section 3.5, we give examples of other creative ways to regularize RankCentrality and give an empirical analysis of these methods.

3.2 Convergence of RankCentrality: Unregularized and Regularized

We begin this section by reformulating RankCentrality to make it easier for analysis. We present significantly simpler proofs for the convergence of the RankCentrality algorithm. We loosely follow notation from Rajkumar and Agarwal [RA14]. Our "RankCentrality" algorithm differs from both Negahban, Oh, and Shah [NOS16] as well as from Rajkumar and Agarwal [RA14].

Symbol	Definition
$\|\cdot\|$	unless stated otherwise, vector norms are ℓ_2 norms, and matrix norms are operator (spectral) norms
γ	$\frac{n\mu_{\min}}{2(1+\sqrt{2})b^{3/2}}$
π	limiting distribution of Q
$\hat{\pi}$	limiting distribution of $\hat{Q}$
λ	regularization constant, see D_λ
$\lambda_{\max}(R)$	second largest eigenvalue of matrix R (because the largest eigenvalue of an irreducible Markov chain is always 1)
μ_{ij}	probability that pair (i,j) is observed
$\mathbf{1}$	vector of all one entries, usually in $\mathbb{R}^n$
b	$\max_{i,j} \frac{\pi_i}{\pi_j}$
k	number of comparisons per pair in sampling scheme in [NOS16]
n	number of items being compared
m	number of comparisons total
P	pairwise preference matrix
$\hat{P}$	empirical comparison matrix
Q	true markov chain (requires knowing P)
$\hat{Q}$	empirical markov chain
$\hat{Q}^{\text{NOS}}$	empirical markov chain as defined in [NOS16]
$\hat{Q}^{\text{RA}}$	empirical markov chain as defined in [RA14]
D_λ	$(1-\lambda)I + \frac{\lambda}{n}\mathbf{1}\mathbf{1}^T$

Table 3.1: Notation used in Chapter 3

Negahban, et al. [NOS16] assume pairs are sampled *without* replacement and each pair is compared d times. We prefer the approach in [RA14] by Rajkumar and Agarwal: each pair is sampled *with* replacement and compared once (comparing with replacement allows for each pair to be compared different number of times). Our arguments should extend to the case where each pair is observed a multiple of d times, i.e., pairs are chosen with replacement but each time a pair is chosen, we observe it d times, but we focus on the case $d = 1$. Let $S = ((i_1, j_1, y_1), \dots, (i_m, j_m, y_m)) \in (\mathcal{X} \times \{0,1\})^m$ be the training sample, where i_k, j_k denotes the items compared on the k-th observation and $y_k \in \{0,1\}$ denotes the outcome of the comparison (0 implies i_k won, 1 implies j_k won).

To construct the Markov transition matrix $\hat{Q}$, let C_{ij} be the *number* (not fraction, as in P_{ij}) of comparisons between i and j that j won, then,

$$\hat{Q}_{ij} = \begin{cases} \mu_{\min}\mu_{ij}^{-1}\frac{C_{ij}}{m} & \text{if } i \neq j \\ 1 - \mu_{\min}\sum_{\ell \neq i}\mu_{ij}^{-1}\frac{C_{i\ell}}{m} & \text{if } i = j \end{cases}. \tag{3.5}$$

We want to view $\hat{Q}$ as a sum of simpler matrices and, to that end, make the following definitions. Let

$$Q^{(ij)} = e_i e_j^T - e_i e_i^T \tag{3.6}$$

and let

$$Q_k = \begin{cases} \mu_{ij}^{-1} Q^{(j_k i_k)} & \text{if } y_k = 0 \\ \mu_{ij}^{-1} Q^{(i_k j_k)} & \text{if } y_k = 1 \end{cases}. \tag{3.7}$$

We see now that

$$\hat{Q} = I + \frac{\mu_{\min}}{m}\sum_{k=1}^{m} Q_k, \tag{3.8}$$

and for the remainder of our analysis shall consider (3.8) as the definition of $\hat{Q}$.

We pause, briefly, to give intuition into the motivation for our choice of $\hat{Q}$ and to highlight the differences between $\hat{Q}$ and the analogous $\hat{Q}^{\mathrm{NOS}}$ and $\hat{Q}^{\mathrm{RA}}$. First, we observe that the choice of denominator in the off-diagonal entries does not affect the output of algorithm. Indeed, we can see by inspection that for any row-stochastic matrix R the leading left eigenvectors of R and $\frac{1}{s}R + (1 - \frac{1}{s})I$ are equal. That $\hat{Q}^{\mathrm{NOS}}$ has a denominator of $d_{\max}$, the maximum degree on the underlying graph, $\hat{Q}^{\mathrm{RA}}$ has a denominator of n, the number of items, and $\hat{Q}$ has a denominator of m, the number of comparisons, does not affect the leading left eigenvalue returned. We then engineer $\hat{Q}$ so that it is row-stochastic and the entries are sums of independent random variables. We use μ_{ij}^{-1} to scale each entry so that the expected value of the ij-entry depends on P_{ij} but not μ_{ij}. However, we need to ensure that the entries in $\hat{Q}$ are never negative, so we multiply these entries by

$\mu_{\min}$. Having ensured that each summand's row-sums don't exceed 1, we need to ensure that that the same is true of $\hat{Q}$ and do so by dividing by m.

The advantage of defining $\hat{Q}$ in this way, as opposed to $\hat{Q}^{\mathrm{RA}}$ in (3.4), is that we can write $\hat{Q}$ as a sum of independent random variables, which allows for stronger central-limit type theorems.

Recall $P_{ij} = \mathbb{P}(i \prec j)$. Set

$$Q = \mu_{\min} \sum_{i \neq j} P_{ij} Q^{(ij)}.$$

More concretely,

$$Q_{ij} = \begin{cases} \mu_{\min} P_{ij} & \text{if } i \neq j \\ 1 - \sum_{\ell \neq i} \mu_{\min} P_{i\ell} & \text{if } i = j \end{cases} \tag{3.9}$$

and observe that $\mathbb{E}\,(\hat{Q}) = Q$. It is also worth noting that Q satisfies the detailed balance equation and is time-reversible: for all $i \neq j$, we have

$$\pi_i Q_{ij} = \frac{\mu_{\min} \pi_i \pi_j}{\pi_i + \pi_j} = \pi_j Q_{ji}.$$

Note that our definition satisfies $Q = \mathbb{E}\,(\hat{Q})$ regardless of the number of pairwise comparisons. To contrast this with [RA14], there we had that $\mathbb{E}\,(\hat{Q}^{\mathrm{RA}})$ is a function of m, the number of comparisons observed, and it is only true that $\lim_{m \to \infty} \mathbb{E}\,(\hat{Q}^{\mathrm{RA}}) = Q^{\mathrm{RA}}$. That $\mathbb{E}\,(\hat{Q}) = Q$ is the first of two main reasons that we are able to give stronger sample complexity bounds. The second is that $\hat{Q}$ can be written as a sum of m *independent* random matrices. This enables us to use matrix concentration inequalities (such as those in [Tro12]) for $\|Q - \hat{Q}\|$, unlike Rajkumar and Agarwal, who had to prove element-wise convergence and apply a union bound over all elements.

We begin our analysis of the RankCentrality algorithm by giving a bound on the spectral gap of the transition matrix Q constructed from pairwise preferences.

Proposition 3.1. *The spectral gap $1 - \lambda_{\max}$ of Q is at least $\frac{n\mu_{\min}}{2b}$, where $b = \max_{i,j} \frac{w_i}{w_j}$.*

Proof. We will use the following lemma from Negahban, et al [NOS16, Lemma 6].

Lemma 3.2 (Comparison Inequality for Spectral Gaps[NOS16]). *Let Q, π and R, τ be reversible Markov chains on a finite set $[n]$ representing random walks on a graph $G = ([n], E)$, i.e. $R(i,j) = 0$ and $Q(i,j) = 0$ if $(i,j) \notin E$. For $\alpha \equiv \min_{(i,j) \in E}\{\pi_i Q_{ij} / \tau_i R_{ij}\}$ and $\beta \equiv \max_i\{\pi_i/\tau_i\}$,*

$$\frac{1 - \lambda_{\max}(Q)}{1 - \lambda_{\max}(R)} \geq \frac{\alpha}{\beta}$$

Let $R = \frac{1}{n}\mathbf{1}\mathbf{1}^T = [\frac{1}{n}]_{ij}$ and $\tau = \frac{1}{n}\mathbf{1} = [\frac{1}{n}]_i$ and observe that these define a reversible Markov chain. Since R has rank 1, we have $\lambda_{\max}(R) = 0$, which gives us that $1 - \lambda_{\max}(Q) \geq \frac{\alpha}{\beta}$. Now we bound α and β.

We have

$$\alpha = \min_{i,j} \frac{\pi_i Q_{ij}}{\tau_i R_{ij}} = \min_{ij} \frac{\frac{w_i}{\sum_l w_l} \frac{w_j}{w_i + w_j} \mu_{\min}}{\frac{1}{n}\frac{1}{n}}$$
$$= \min_{i,j} \frac{n^2 \mu_{\min} w_i w_j}{(w_i + w_j) \sum_l w_l} \geq \frac{n^2 \mu_{\min} \min_i w_i}{2 \sum_l w_l} = \frac{n^2 \mu_{\min} \min_i \pi_i}{2}.$$

We also see $\beta = \max_i \frac{\pi_i}{\tau_i} = n \max_i \pi_i$. Thus, $\frac{\alpha}{\beta} \geq \frac{n\mu_{\min}}{2b}$. □

It is easy to see that this bound is close to optimal. Since the diagonal entries are each at least $1 - n\mu_{\min}$, we know $(n\mu_{\min})^{-1}(Q - (1 - n\mu_{\min})I)$ is non-negative and row stochastic. By the Perron-Frobenius Theorem, the eigenvalues of $(n\mu_{\min})^{-1}(Q - (1 - n\mu_{\min})I)$ lie in $[-1, 1]$ and the eigenvalues of Q must lie in $[1 - 2n\mu_{\min}, 1]$. The difference between 1 and the smallest possible eigenvalue of Q is only a factor of $4b$ larger than our bound on the spectral gap.

Proposition 3.3 (Effect of perturbing Q)**.** *Let Q be the true transition matrix as defined in* (3.9)*. For any ergodic Markov chain on $[n]$ with row-stochastic transition matrix $\tilde{Q}$ and stationary distribution $\tilde{\pi}$, if $\|Q - \tilde{Q}\| < \frac{n\mu_{\min}}{2b^{3/2}}$, we have*

$$\frac{\|\tilde{\pi} - \pi\|}{\|\pi\|} \leq \frac{2\|\Delta\| b^{3/2}}{n\mu_{\min} - 2\|\Delta\| b^{3/2}},$$

where $\Delta = \tilde{Q} - Q$.

Proof. We begin by citing a lemma by Negahban, et al. [NOS16, Lemma 2].

Lemma 3.4. *For any Markov chain $\tilde{Q} = Q + \Delta$ with a reversible Markov chain Q, let p_t be the distribution of the Markov chain $\tilde{Q}$ when started with initial distribution p_0. Then,*

$$\frac{\|p_t - \pi\|}{\|\pi\|} \leq \rho^t \frac{\|p_0 - \pi\|}{\|\pi\|} \sqrt{\frac{\pi_{\max}}{\pi_{\min}}} + \frac{1}{1 - \rho} \|\Delta\|_2 \sqrt{\frac{\pi_{\max}}{\pi_{\min}}} .$$

where π is the stationary distribution of Q and $\rho = \lambda_{\max}(Q) + \|\Delta\|_2 \sqrt{\pi_{\max}/\pi_{\min}}$.

As before, let $b = \max_{i,j} \frac{\pi_i}{\pi_j}$. Consider the limit as $t \to \infty$:

- when $0 \leq \rho < 1$ we have $\rho^t \to 0$, and
- when the Markov chain $\tilde{Q}$ is irreducible we have $p_t \to \tilde{\pi}$.

In this case,

$$\frac{\|\tilde{\pi} - \pi\|}{\|\pi\|} \leq \frac{1}{1-\rho}\|\Delta\|_2\sqrt{b}.$$

Recall that $1-\lambda_{\max}(Q) > \frac{n\mu_{\min}}{2b}$ by Proposition 3.1. Now we have that $\rho < 1$ when $\|\Delta\| < \frac{n\mu_{\min}}{2b^{3/2}}$ because when this is the case, we have $\|\Delta\|\sqrt{b} < \frac{n\mu_{\min}}{2b}$ and hence $\rho \leq 1 - \frac{n\mu_{\min}}{2b} + \|\Delta\|\sqrt{b} < 1$. Assuming $\|\Delta\| < \frac{n\mu_{\min}}{2b^{3/2}}$, we have

$$\frac{\|\tilde{\pi} - \pi\|}{\|\pi\|} \leq \frac{\|\Delta\|\sqrt{b}}{\frac{n\mu_{\min}}{2b} - \|\Delta\|\sqrt{b}} = \frac{2\|\Delta\|b^{3/2}}{n\mu_{\min} - 2\|\Delta\|b^{3/2}}.$$

□

For transition matrices Q and $\hat{Q}$ we define the centered transition matrix Q' and $\hat{Q}'$ by subtracting I. That is, $Q' = Q - I$ and $\hat{Q}' = \hat{Q} - I$. These centered matrices Q' and $\hat{Q}'$, as well as Q_k and $Q^{(ij)}$ defined previously, have non-negative entries everywhere except on the diagonal (where they are non-positive) and their rows sum to zero. These centered matrices significantly simplify the algebra in the following computations.

Lemma 3.5. *The difference* $Z_k := \frac{\mu_{\min}Q_k - Q'}{m}$ *is bounded in norm:* $\|Z_k\| < \frac{3}{m}$.

Proof. To bound $\|Q_k\|$, recall that Q_k is of the form $\mu_{ij}^{-1}Q^{(ij)} = \mu_{ij}^{-1}(e_ie_j^T - e_ie_i)$. Observe that $Q^{(ij)}Q^{(ij)T} = 2e_ie_i^T$. Therefore, $\|Q_k\| \leq \mu_{\min}\mu_{ij}^{-1}\sqrt{2} \leq \sqrt{2}$. By convexity of norms, $\|Q'\| = \|\mathbb{E}\,\mu_{\min}Q_k\| \leq \mathbb{E}\,\|\mu_{\min}Q_k\| \leq \sqrt{2}$. Using the triangle inequality we get $\|\mu_{\min}Q_k - Q'\| \leq 2\sqrt{2} < 3$. □

Lemma 3.6. *Let* $Z_k = \frac{\mu_{\min}Q_k - Q'}{m}$*, as before. We can bound the variance term as:*

$$\sigma^2 := \max\left\{\left\|\sum_{k=1}^{m}\mathbb{E}\,Z_kZ_k^*\right\|, \left\|\sum_{k=1}^{m}\mathbb{E}\,Z_k^*Z_k\right\|\right\} \leq \frac{3(n-1)\mu_{\min}}{m}.$$

Proof. To bound $\|\mathbb{E}\,Z_kZ_k^*\|$, we see

$$\begin{aligned}\mathbb{E}\,Z_kZ_k^* &= \frac{1}{m^2}\mathbb{E}\,\left(\mu_{\min}^2Q_kQ_k^T - \mu_{\min}Q_kQ'^T - Q'\mu_{\min}Q_k^T + Q'Q'^T\right)\\ &= \frac{1}{m^2}\mathbb{E}\,\left(\mu_{\min}^2Q_kQ_k^T - Q'Q'^T\right).\end{aligned}$$

We can compute these explicitly.

Begin by considering the $\mu_{\min}^2Q_kQ_k^T$ term. We know $Q^{(ij)}Q^{(ij)T} = 2e_ie_i^T$. By simple algebra, we get $\mathbb{E}\,Q_kQ_k^T = \sum_i\sum_{j\neq i}2\mu_{ij}^{-1}P_{ji}e_ie_i^T$. Therefore, $\|\mathbb{E}\,\mu_{\min}^2Q_kQ_k^T\| \leq \mu_{\min}^2\max_i\sum_{j\neq i}2\mu_{ij}^{-1}P_{ji} \leq 2(n-1)\mu_{\min}$.

Computing $Q'Q'^T$ is more tedious.

$$\begin{aligned}Q'Q'^T &= \left(\sum_{i\neq j}\mu_{\min}P_{ij}(e_ie_j^T - e_ie_i^T)\right)\left(\sum_{u\neq v}\mu_{\min}P_{uv}(e_ve_u^T - e_ue_u)\right)\\ &= \sum_{i\neq j, u\neq v}\mu_{\min}^2 P_{ij}P_{uv}(e_ie_j^Te_ve_u^T - e_ie_j^Te_ue_u - e_ie_i^Te_ve_u^T + e_ie_i^Te_ue_u^T).\end{aligned}$$

By ignoring zero terms (notice that the first of four summands is non-zero only when $j = v$, the second when $j = u$, etc.) and re-indexing, we get

$$Q'Q'^T = \mu_{\min}^2\left(\sum_{i\neq\ell\neq j}P_{i\ell}P_{j\ell}e_ie_j^T - \sum_{i\neq j\neq\ell}P_{ij}P_{j\ell}e_ie_j^T - \sum_{j\neq i\neq\ell}P_{i\ell}P_{ji}e_ie_j^T + \sum_{u\neq i\neq v}P_{iu}P_{iv}e_ie_i^T\right),$$

where statements such as $i \neq \ell \neq j$ mean $i \neq \ell$ and $j \neq \ell$ (but i may be equal to j). This is a symmetric matrix, so its singular values are its eigenvalues. We can now invoke the Gershgorin circle theorem, a consequence of which is that $\|M\| < \max_i \sum_j |M_{ij}|$ for symmetric matrices. Therefore, $\|Q'Q'^T\| \leq 4n^2\mu_{\min}^2$. Finally, the triangle inequality gives $\|\mathbb{E}\, Z_kZ_k^*\| \leq \frac{1}{m^2}\left(2(n-1)\mu_{\min} + 4n^2\mu_{\min}^2\right) \leq \frac{3(n-1)\mu_{\min}}{m^2}$ for all $n \geq 4$ (we used the fact that $\mu_{\min} \leq \binom{n}{2}^{-1}$).

We now turn to $Z_k^*Z_k$. Similar to the calculations above, simple algebra gets us

$$\mathbb{E}\, Q_k^TQ_k = \sum_i\sum_{j\neq i}\mu_{ij}^{-1}(P_{ij} + P_{ji})(e_ie_i^T - e_ie_j^T).$$

As before, this is a symmetric matrix and we can use the Gershgorin circle theorem to give a bound on the largest singular value of $\mathbb{E}\,\mu_{\min}^2Q_k^TQ_k$:

$$\|\mathbb{E}\,\mu_{\min}^2Q_k^TQ_k\| \leq \max_i\sum_{j\neq i}\mu_{\min}^2(2\mu_{ij}^{-1}) \leq 2(n-1)\mu_{\min}.$$

As before computing Q'^TQ' is more tedious but gives

$$\begin{aligned}Q'^TQ' &= \mu_{\min}^2\sum_{i\neq j}\sum_{u\neq v}P_{ij}P_{uv}(e_je_i^T - e_ie_i^T)(e_ue_v^T - e_ue_u^T)\\ &= \mu_{\min}^2\sum_{i\neq j}\sum_{v\neq i}P_{ij}P_{iv}(e_je_v^T - e_je_i^T - e_ie_v^T + e_ie_i^T)\\ &= \mu_{\min}^2\left(\sum_{i\neq j}\left(\sum_{\ell\neq i;\ell\neq j}P_{\ell i}P_{\ell j} - P_{ji}P_{j\ell} - P_{i\ell}P_{ij}\right)e_ie_j^T\right.\\ &\qquad\left. + \sum_i\left(\sum_{u\neq i, v\neq i}P_{iu}P_{iv} + \sum_{\ell\neq i}P_{\ell i}P_{\ell i}\right)e_ie_i^T\right)\end{aligned}$$

Again, we can invoke the Gershgorin circle theorem and see that $\|Q'Q'^T\| \le 4n^2\mu_{\min}^2$. As before, the triangle inequality gives $\|\mathbb{E}\ Z_k^* Z_k\| \le \frac{1}{m^2}\left(2(n-1)\mu_{\min} + 4n^2\mu_{\min}^2\right) \le \frac{3(n-1)\mu_{\min}}{m^2}$ for all $n \ge 4$.

Finally, note that Z_k are not only independent but also identically distributed and hence

$$\max\left\{\left\|\mathbb{E}\ \sum_k Z_k^* Z_k\right\|, \left\|\mathbb{E}\ \sum_k Z_k Z_k^*\right\|\right\} = m\max\left\{\|\mathbb{E}\ Z_k^* Z_k\|, \|\mathbb{E}\ Z_k Z_k^*\|\right\} \le \frac{3(n-1)\mu_{\min}}{m}.$$

□

We will soon need to use the Matrix Bernstein Inequality from [Tro12, Theorem 1.6] and state it here as a lemma.

Lemma 3.7 (Matrix Bernstein [Tro12]). *Consider a finite sequence $\{\mathbf{Z}_k\}$ of independent, random matrices with dimensions $d_1 \times d_2$. Assume that each random matrix satisfies*

$$\mathbb{E}\ \mathbf{Z}_k = \mathbf{0} \quad \textit{and} \quad \|\mathbf{Z}_k\| \le R \quad \textit{almost surely}.$$

Define

$$\sigma^2 := \max\left\{\left\|\sum\nolimits_k \mathbb{E}\left(\mathbf{Z}_k\mathbf{Z}_k^*\right)\right\|, \ \left\|\sum\nolimits_k \mathbb{E}\left(\mathbf{Z}_k^*\mathbf{Z}_k\right)\right\|\right\}.$$

Then, for all $t \ge 0$,

$$\mathbb{P}\left(\left\|\sum\nolimits_k \mathbf{Z}_k\right\| \ge t\right) \le (d_1 + d_2)\cdot\exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right).$$

Finally, we put this all together.

Theorem 3.8 (Convergence of Unregularized RankCentrality). *Let $\hat{Q}$ be constructed as in* (3.5). *If $\hat{Q}$ is ergodic and $\hat{\pi}$ is the stationary distribution of $\hat{Q}$, then we have (where probability is taken over the m comparisons made under the BTL model and each pair is equally likely to get picked)*

$$\mathbb{P}\left(\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \le \varepsilon\right) > 1 - 2n\exp\left(\frac{-\varepsilon^2 m}{5nb^3(1+\varepsilon)^2 + nb^{3/2}\varepsilon(1+\varepsilon)}\right).$$

Proof. Assuming $\|\Delta\| < \frac{1}{nb^{3/2}}$, by Proposition 3.3 we have

$$\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \le \frac{2\|\Delta\|b^{3/2}}{n\mu_{\min} - 2\|\Delta\|b^{3/2}}.$$

This means we want

$$\frac{2\|\Delta\|b^{3/2}}{n\mu_{\min} - 2\|\Delta\|b^{3/2}} < \varepsilon,$$

which happens when $\|\Delta\| \leq \frac{\varepsilon n \mu_{\min}}{2b^{3/2}(1+\varepsilon)}$. Note that this is stronger than $\|\Delta\| < \frac{n\mu_{\min}}{2b^{3/2}}$, so our previous assumption will hold.

Finally, we let $t = \frac{\varepsilon n \mu_{\min}}{2b^{3/2}(1+\varepsilon)}$ and use Lemma 3.7 to get

$$\mathbb{P}\left(\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \geq \varepsilon\right) \leq \mathbb{P}\left(\|\hat{Q} - Q\| \geq t\right) \leq -2n \exp\left(\frac{-t^2}{\sigma^2 + Rt/3}\right),$$

where we have $\sigma^2 \leq \frac{3n\mu_{\min}}{m}$ by Lemma 3.6 and $R < \frac{3}{m}$ by Lemma 3.5. Therefore, we get

$$\begin{aligned}\mathbb{P}\left(\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \geq \varepsilon\right) &\leq 2n \exp\left(\frac{-\left(\frac{\varepsilon n \mu_{\min}}{2b^{3/2}(1+\varepsilon)}\right)^2}{\frac{3n\mu_{\min}}{m} + \frac{\varepsilon n \mu_{\min}}{2mb^{3/2}(1+\varepsilon)}}\right) \\ &\leq 2n \exp\left(\frac{-\mu_{\min}\varepsilon^2 nm}{6b^3(1+\varepsilon)^2 + 2b^{3/2}\varepsilon(1+\varepsilon)}\right) \leq 2n \exp\left(\frac{-\mu_{\min}\varepsilon^2 nm}{8b^3(1+\varepsilon)^2}\right).\end{aligned}$$

□

Corollary 3.9. *Fix* $\delta \in (0,1)$ *and* $\varepsilon \in (0,1)$. *If*

$$m \geq 32b^3\varepsilon^{-2}n^{-1}\mu_{\min}^{-1} \log \frac{2n}{\delta}$$

and the empirical Markov chain $\hat{Q}$ *constructed as in* (3.5) *is ergodic, then with probability at least* $1-\delta$*, we have*

$$\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \leq \varepsilon.$$

Proof. We need

$$\mathbb{P}\left(\frac{\|\hat{\pi} - \pi\|}{\|\pi\|} \geq \varepsilon\right) \leq 2n \exp\left(\frac{-\mu_{\min}\varepsilon^2 nm}{8b^3(1+\varepsilon)^2}\right) < \delta.$$

By re-writing in terms of m, we see that the second inequality is true when

$$m > 8b^3(1+\varepsilon)^2 n^{-1}\mu_{\min}^{-1}\varepsilon^{-2} \log \frac{2n}{\delta}.$$

The desired inequality now follows immediately from $\varepsilon < 1$ (we make this assumption for simplicity; the statement of the theorem is not very strong when $\varepsilon > 1$). □

When μ is uniform, the above theorem requires $m > 16b^3\varepsilon^{-2}n \log(\frac{2n}{\delta})$. We have given an $O\left(\varepsilon^{-2}n \log \frac{n}{\delta}\right)$ upper bound on the sample complexity. This is a much better bound than Rajkumar and Agarwal give in [RA14]. Their $O(\varepsilon^{-2}\mu_{\min}^{-2}n \log(\frac{n}{\delta}))$ scales as $O(\varepsilon^{-2}n^5 \log(\frac{n}{\delta}))$ when μ is uniform, and worse otherwise.

Our results, however, require that we know μ beforehand whereas Rajkumar and Agarwal's version of RankCentrality does not assume μ is known beforehand. Future work on our algorithm could study the impact of using an approximate μ (e.g., by sampling from μ before running RankCentrality) on the output of RankCentrality.

The b^3 term may seem counterintuitive to the reader as it should be harder to rank n items when b, the dynamic range of their BTL scores, is small. However, we are using the ℓ_2 norm to evaluate performance, and, under the ℓ_2 norm, both π and $\hat{\pi}$ are close to $\frac{1}{n}\mathbf{1} = [\frac{1}{n}]_{i=1}^n$ when b is small.

3.3 λ-Regularized RankCentrality

There is a crucial problem with the RankCentrality algorithm that prevents it from being useful when only a small number of pairwise comparisons are available. In such a scenario, the empirical markov chain $\hat{Q}$ is not ergodic, which means we cannot guarantee a unique stationary distribution (the output of the RankCentrality algorithm). We propose the following approach to regularizing the RankCentrality algorithm that also ensures that the resulting markov chain is irreducible. We call this method λ-Regularized RankCentrality.

Let $D_\lambda := (1-\lambda)I + \frac{\lambda}{n}\mathbf{1}\mathbf{1}^T$. The Regularized RankCentrality algorithm computes $\hat{Q}$ as before, but returns the leading left eigenvector of $\hat{Q}D_\lambda$ instead of the leading left eigenvector of $\hat{Q}$. Note that $\hat{Q}D_\lambda = (1-\lambda)\hat{Q} + \frac{\lambda}{n}\mathbf{1}\mathbf{1}^T$. This is the sum of a non-negative matrix and a positive matrix, giving us a positive matrix, which must be ergodic. The intuition here is that we are allowing, with probability λ, a transition to any state $i \in [n]$ uniformly at random, thus the Markov chain is irreducible and aperiodic.

The rest of this subsection is devoted to an analysis of the bias-variance trade-off of λ-Regularized RankCentrality. We will compare

- $\hat{\tilde{\pi}}$, the leading left eigenvector of $\hat{Q}D_\lambda$, i.e., the output of λ-regularized RankCentrality, and
- $\tilde{\pi}$, the leading left eigenvector of QD_λ, i.e., the expected output of λ-regularized RankCentrality as $m \to \infty$,
- π, the leading left eigenvector of Q, and the expected output of RankCentrality as $m \to \infty$.

For notational convenience, let $\gamma := \frac{n\mu_{\min}}{2(1+\sqrt{2})b^{3/2}}$. Note that γ is not constant—in fact it is $\Theta(\frac{1}{n})$ when μ is uniform and $O(\frac{1}{n})$ in general. We will be careful not to use γ anywhere that it might obfuscate sample complexity.

Proposition 3.10 (Regularized RankCentrality Bias). *Fix $\lambda \in (0,\gamma)$. The asymptotic ($m \to \infty$) expectation of the output of the λ-Regularized RankCentrality algorithm is $\tilde{\pi}$ and the bias*

$\|\pi - \tilde{\pi}\|/\|\pi\|$ can be bounded as

$$\frac{\|\pi - \tilde{\pi}\|}{\|\pi\|} \leq \frac{\lambda}{\gamma - \lambda}$$

Proof. Let $\tilde{Q} = QD_\lambda$. We now have $Q - \tilde{Q} = \lambda(\frac{1}{n}\mathbf{1}\mathbf{1}^T - Q)$ and $\|Q - \tilde{Q}\| \leq \lambda(1 + \sqrt{2})$. Now we apply Proposition 3.3 to see that

$$\frac{\|\pi - \tilde{\pi}\|}{\|\pi\|} \leq \frac{2(1+\sqrt{2})\lambda b^{3/2}}{n\mu_{\min} - 2(1+\sqrt{2})\lambda b^{3/2}} = \frac{\lambda}{\gamma - \lambda}.$$

$\square$

Theorem 3.11 (Regularized RankCentrality). *Fix $\lambda \in (0, \frac{\gamma}{2})$ and choose $\varepsilon \in (2\lambda\gamma^{-1}, 1)$. We construct $\hat{Q}$ as before and let $\tilde{\hat{\pi}}$ be the stationary distribution (leading left eigenvector) of $\hat{Q}D_\lambda$ (i.e., the output of λ-regularized RankCentrality). We have*

$$\mathbb{P}\left(\frac{\|\tilde{\hat{\pi}} - \pi\|}{\|\pi\|} < \varepsilon\right)$$
$$> 1 - 2n\exp\left(\frac{-(n\mu_{\min}\varepsilon - 4(1+\sqrt{2})b^{3/2}\lambda)^2 m}{48(1-\lambda)^2 n\mu_{\min}b^3 + 4b^{3/2}(1-\lambda)(n\mu_{\min}\varepsilon - 4b^{3/2}(1+\sqrt{2})\lambda)}\right)$$

Proof. As we noted in the proof of Theorem 3.8, to guarantee $\|\pi - \tilde{\hat{\pi}}\|/\|\pi\| \leq \varepsilon$, we need $\|Q - \hat{Q}D_\lambda\| \leq \frac{\varepsilon n\mu_{\min}}{2(1+\varepsilon)b^{3/2}}$. Using the triangle inequality, we have $\|Q - \hat{Q}D_\lambda\| \leq \|Q - QD_\lambda\| + \|QD_\lambda + \hat{Q}D_\lambda\|$. We showed in Proposition 3.10 that $\|Q - QD_\lambda\| \leq \lambda(1+\sqrt{2})$. So we need

$$\begin{aligned}\|QD_\lambda - \hat{Q}D_\lambda\| &\leq \frac{\varepsilon n\mu_{\min}}{2(1+\varepsilon)b^{3/2}} - \lambda(1+\sqrt{2}) \leq \frac{\varepsilon n\mu_{\min}}{4b^{3/2}} - \lambda(1+\sqrt{2}) \\ &= \frac{(1+\sqrt{2})}{2}\varepsilon\gamma \; - \lambda(1+\sqrt{2}) = \frac{(1+\sqrt{2})}{2}(\varepsilon\gamma - 2\lambda)\end{aligned}$$

Note that this quantity is positive when $\varepsilon \in (2\lambda\gamma^{-1}, 1)$ (which is precisely the requirement in the hypothesis above). We have required that $\varepsilon < 1$ for ease of computation. Note that the theorem is not very useful in practice otherwise. We now require that

$$\|QD_\lambda - \hat{Q}D_\lambda\| \leq \frac{\varepsilon n\mu_{\min}}{4b^{3/2}} - \lambda(1+\sqrt{2}).$$

We can now invoke Lemma 3.7 with $Z_k = \frac{1}{m}(Q'D_\lambda - \mu_{\min}Q_kD_\lambda) = \frac{1}{m}(1-\lambda)(Q' - \mu_{\min}Q_k)$. By our previous calculations in Lemmas 3.5 and 3.6, we have the variance term $\sigma^2 \leq (1-\lambda)^2\frac{3n\mu_{\min}}{m}$

and the norm term $R \leq (1-\lambda)\frac{3}{m}$. The resulting inequality is

$$\mathbb{P}\left(\|QD_\lambda - \hat{Q}D_\lambda\| \geq \frac{n\mu_{\min}\varepsilon}{4b^{3/2}} - (1+\sqrt{2})\lambda\right) \\ \leq 2n\exp\left(\frac{-\left(\frac{n\mu_{\min}\varepsilon}{4b^{3/2}} - (1+\sqrt{2})\lambda\right)^2}{(1-\lambda)^2\frac{3n\mu_{\min}}{m} + (1-\lambda)\frac{1}{m}\left(\frac{n\mu_{\min}\varepsilon}{4b^{3/2}} - (1+\sqrt{2})\lambda\right)}\right),$$

which simplifies to the desired inequality. □

Corollary 3.12. *Recall $\gamma = \frac{n\mu_{\min}}{2(1+\sqrt{2})b^{3/2}}$. Let $\lambda \in (0, \frac{\gamma}{2})$. Choose $\delta \in (0,1)$ and $\varepsilon \in (2\lambda\gamma^{-1}, 1)$. If*

$$m > \frac{52(1-\lambda)b^3}{n\mu_{\min}\left(\varepsilon - 2\lambda\gamma^{-1}\right)^2}$$

then with probability at least $1-\delta$, we have

$$\frac{\|\tilde{\hat{\pi}} - \pi\|}{\|\pi\|} \leq \varepsilon.$$

Proof. As before, we need

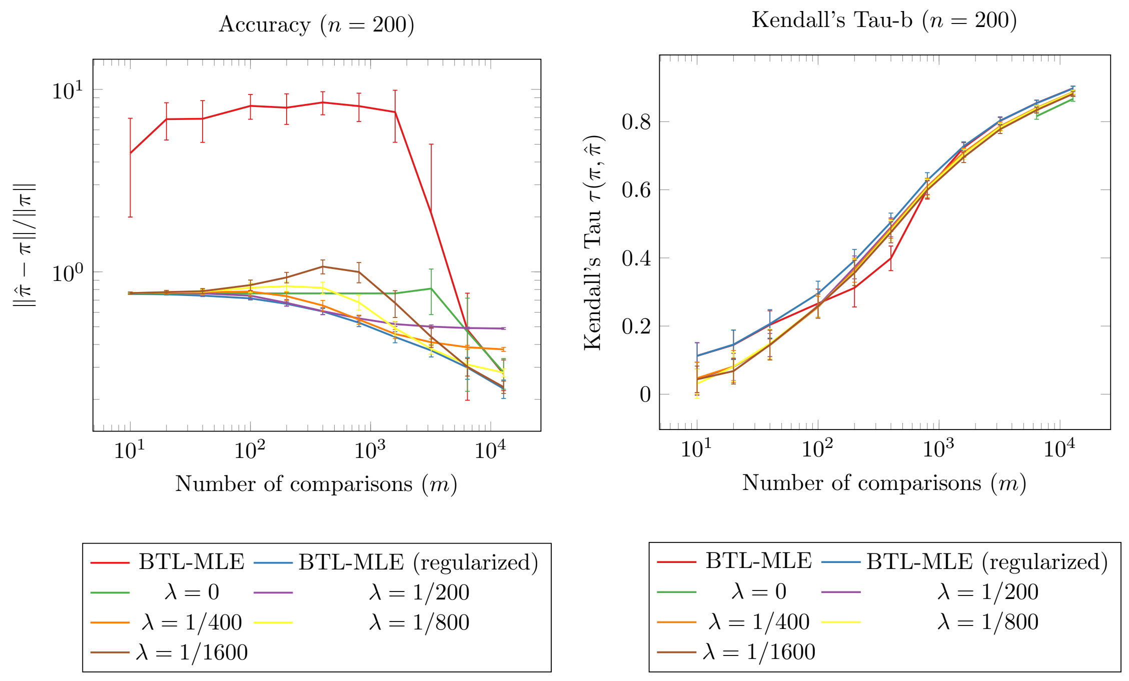
$$\mathbb{P}\left(\frac{\|\tilde{\hat{\pi}} - \pi\|}{\|\pi\|} > \varepsilon\right) \\ \leq 2n\exp\left(\frac{-(n\mu_{\min}\varepsilon - 4(1+\sqrt{2})b^{3/2}\lambda)^2 m}{48(1-\lambda)^2 n\mu_{\min}b^3 + 4b^{3/2}(1-\lambda)(n\mu_{\min}\varepsilon - 4b^{3/2}(1+\sqrt{2})\lambda)}\right) \\ < \delta$$

Rewriting in terms of m, we see that the second inequality is true when

$$m > \frac{48(1-\lambda)^2 n\mu_{\min}b^3 + 4b^{3/2}(1-\lambda)(n\mu_{\min}\varepsilon - 4b^{3/2}(1+\sqrt{2})\lambda)}{-(n\mu_{\min}\varepsilon - 4(1+\sqrt{2})b^{3/2}\lambda)^2}\log\frac{2n}{\delta}$$

The desired inequality now follows by replacing various terms in the above inequality with upper bounds for them (e.g., $(1-\lambda)^2 < 1-\lambda$, $b^{3/2} < b^3$, and $\varepsilon < 1$). □

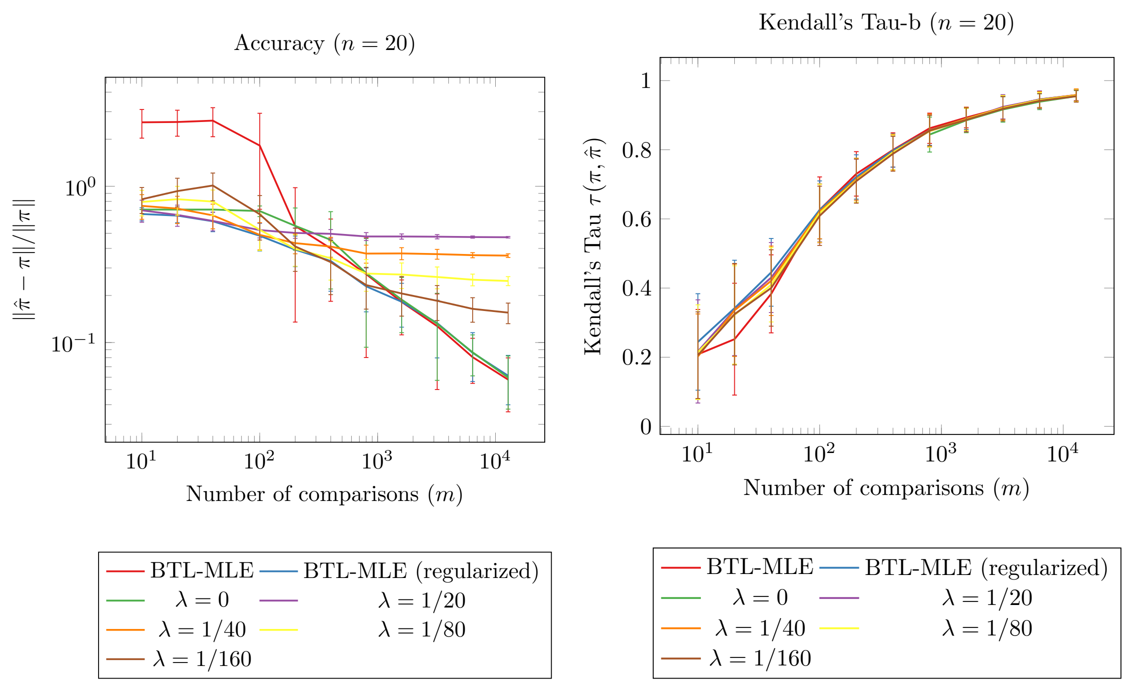
Empirical evidence suggests that we can use values of λ much larger than $\frac{\gamma}{2}$ and still get very good results. We believe that bridging the gap between the theory and application is worthy of further inquiry in the future.



(a) ℓ_2 error

(b) Kendall's Tau-b metric

Figure 3.1: Performance of algorithms with $n = 200$



(a) ℓ_2 error

(b) Kendall's Tau-b metric

Figure 3.2: Performance of algorithms with $n = 20$

3.4 Empirical Results

Having analyzed RankCentrality and λ-regularized RankCentrality in theory, we now evaluate their performance in empirical experiments.

Our main experiments was to evaluate convergence of these algorithms with synthetic BTL scores and comparisons. We compared (unregularized) RankCentrality, λ-regularized RankCentrality (with values of λ from $\frac{1}{n}$, $\frac{1}{2n}$, $\frac{1}{4n}$, and $\frac{1}{8n}$), the BTL maximum likelihood estimation (see (3.2)), and regularized BTL-MLE (using the Scikit-Learn [Ped+11] implementation of logistic regression). The BTL score w_i for each item i was assigned by choosing v_i uniformly at random from $[0, 5]$ and setting $w_i = \exp(v_i)$. Then, for various values of m, we generated m comparisons (first choose m pairs of items, uniformly at random from all possible pairs, then assign winners via the BTL model) and ran each algorithm on the same set of comparisons. In each of these cases, we store the ℓ_2 error and the Kendall's Tau correlation metric. We repeat this process of generating comparisons and evaluating algorithms for a total of 40 times and record the mean and standard deviations of the ℓ_2 error and the Kendall-Tau correlation metric. The results for these experiments for a few different values of n are shown in Figures 3.1 and 3.2.

The Kendall-Tau correlation metric we use in our experiments is also know as Kendall's Tau-b, defined as

$$\tau(\alpha, \beta) = \frac{P - Q}{\sqrt{(P + Q + T) * (P + Q + U)}}, \tag{3.10}$$

where P is the number of concordant pairs (i.e., the number of pairs i, j such that the relative ordering of α_i and α_j is the same as that of β_i and β_j), Q the number of discordant pairs, T the number of ties only in α, and U the number of ties only in β.

The first observation we make is that (unregularized) RankCentrality does not return non-trivial results until many comparisons have been made. As we have noted many times above, this is because the Markov chain $\hat{Q}$ is not ergodic and hence the algorithm is forced to return a uniform distribution $\hat{\pi} = \frac{1}{n}\mathbf{1} = [\frac{1}{n}]_{i=1}^{n}$. Because of the way that Kendall's Tau-b handles ties in $\hat{\pi}$, it is undefined when $\hat{\pi}$ is uniform.

These plots demonstrate that the bias for λ-regularized RankCentrality is most visible in the ℓ_2 error rather than in Kendall's Tau-b, which suggests that if practitioner is more concerned with recovering a ranking of the n items rather than the BTL scores, the need for careful tuning of λ is diminished.

It is also worth noting that we have used values of λ larger than $\gamma = \frac{n\mu_{\min}}{2(1+\sqrt{2})b^{3/2}}$, which violates the hypotheses of Theorem 3.11. By comparing the bias in Figures 3.1 and 3.2, we see that λ may not need to scale as $O(n^{-1})$. It is clear that the bias, when $\lambda \in \left\{\frac{1}{n}, \frac{1}{2n}, \frac{1}{4n}, \frac{1}{8n}\right\}$, is much greater when $n = 20$ as opposed to $n = 200$.

3.5 Similarity-based Regularization

In this section we illustrate creative approaches to regularization that can significantly improve the performance of RankCentrality. Intuitively, we seek to exploit other data about the n items with the goal of improving the output ranking even when we have very few pairwise comparisons.

Thus far, we have considered the case when we have n items and m pairwise comparisons. The other end of the spectrum is the case when we have features (i.e., an embedding $[n] \to \mathbb{R}^d$) for the items we are comparing. When we have features $x_i \in \mathbb{R}^d$ for each item $i \in [n]$, it may be reasonable to assume that there is a function $f : \mathbb{R}^d \to \mathbb{R}$ such that $\operatorname{sign}(f(x_i), f(x_j)) = \operatorname{sign}(w_i - w_j)$. If we assume the BTL model, we may (but not necessarily) ask that the function f output BTL scores. The literature has various methods for learning such functions f. The goal in these pursuits is typically not to learn a ranking of the n items given but rather to learn a function f to rank new (unseen) items.

We can adapt our method of regularizing RankCentrality to solve a problem in the middle of this spectrum. We do not seek to use the features x_i to learn a scoring function f, but rather to augment the pairwise comparison data and improve our estimation of the ranking on the n items. To do so, we construct a regularizing matrix D that incorporates information that we have about features and/or similarity of the n items.

We give the practitioner two examples of regularizing matrices D. If we have continuous features $x_i \in \mathbb{R}^d$, we can begin by constructing an $n \times n$ affinity matrix A by setting

$$A_{ij} := \exp\left(\frac{-(x_i - x_j)^2}{\sigma^2}\right),$$

where σ, the kernel width, is an appropriately chosen hyperparameter. We make the matrix row stochastic by dividing each row by its row sum, i.e.,

$$D'_{ij} := \frac{A_{ij}}{\sum_{\ell=1}^{n} A_{i\ell}}.$$

Finally, our regularization matrix is $D_\lambda D'$ (here D_λ ensures that the resulting $\hat{Q}$ is irreducible). Note that $D' \to D_0 = I_n$ as $\sigma \to 0$ (assuming all x_i are distinct) and $D' \to D_1$ as $\sigma \to \infty$.

If we have discrete features, we let A_{ij} be the total number of features on which x_i and x_j agree (i.e., d minus the Hamming distance between x_i and x_j). Then

$$D'_{ij} = \begin{cases} A_{ij}/\tau & \text{if } i \neq j \\ 1 - \sum_{\ell=1,\ell\neq i}^{n} A_{ij}/\tau & \text{if } i = j \end{cases}$$

where τ, the tightness, is an appropriately chosen hyperparameter such that $\sum_{\ell=1}^{n} A_{ij} < \tau$ for all i. Again, our regularization matrix is $D_\lambda D'$.

We expect that there are many other ways to construct D and encourage the practitioner to explore regularizations that are appropriate for their purposes. We think of the role that D plays as local averaging. In doing so, multiplication by D imputes comparisons. To see this, consider the following example where $n = 4$. Let

$$D = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

If we make a single observation (say that item 1 beat item 3), then

$$\hat{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

but

$$\hat{Q}D = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 \\ 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

As we can see, we have not only inferred a preference for 1 over 3, but imputed preferences for 1 and 2 over 3. Further, allowing a similarity based transition from state 4 to 3, we have created paths from state 4 to states 1 and 2, thus inferring a preference for items 1 and 2 over items 3 and 4. The reader might notice that the low preference for item 4 is more explicitly observed if we were to compute $D\hat{Q}D$. In practice, however, the weight of self-loops in $\hat{Q}$ are very large (asymptotically at least $1 - (n-1)\binom{n}{2}^{-1} = 1 - \frac{n}{2}$), which makes $\hat{Q}$ close to I_n, and $D\hat{Q}D$, $D^2\hat{Q}$, and $\hat{Q}D^2$ are all approximately equal.

For the remainder of this section, we present the results of experiments using real and synthetic datasets.

3.5.1 Car Preferences Dataset

The Car Preferences dataset [Abb+13] is a publicly available dataset of pairwise comparisons. Abbasnejad, et. al., the investigators of the experiment, gave each of 60 users pairwise comparisons between 10 cars with four different attributes:

- body type (sedan or SUV)
- transmission (automatic or manual)
- engine capacity (2.5L, 3.5L, 4.5L, 5.5L, or 6.2L)
- fuel consumed (hybrid or non-hybrid)

Each user was given all 45 pairs to compare and five duplicate pairs (as a control). All but two users compared at least 48 pairs, and 46 users compared all 50 pairs given. We discarded the observations that were intended to be controls (one user had two consistent answers to controls, six users had three consistent answers, 22 users had four consistent answers, and 31 had all five correct). The primary reason for discarding these controls was that the control pairs seemed to be chosen from a very limited set of pairs, skewing the distribution μ of pairs. Once we discarded control observations, the distribution of pairs seemed to be almost uniform (with each pair being observed once per user, except for a small number of missing observations). We were not concerned by the small number of users with multiple inconsistent comparisons and did not take any steps to filter those comparisons out. In our experiments, we ignore which user made each observation. We are implicitly assuming that the users are our source of randomness.

We randomly selected 1600 observations for training and the remaining 1075 to evaluate the algorithms. For various values of m, we choose m observations 20 times uniformly at random without replacement from the 1600 training observations and ran each algorithm on each of the 20 sets of m observations. We recorded the mean and standard deviation of the test error (the fraction of comparisons in the test dataset in which the item with the lower predicted score was preferred). The results of this experiment are summarized in Figure 3.3. That similarity based regularization is competitive with RankSVM when the number of samples is small demonstrates the strength of this method, considering that the similarity-based regularization only considered the number of features that cars had in common but did not access the features directly.

3.5.2 Synthetic Data

To evaluate the empirical performance of similarity-based regularization on data with continuous features, we constructed three synthetic datasets.

- In Experiment 1, we generated 1000 points $\{x_i\}_{i=1}^{1000}$ chosen uniformly at random from $[0, 4]$ and chose $\omega \in \mathbb{R}$ at random from a Gaussian. To each $i \in [1000]$ we associate a score $w_i = \exp(\cos(5\omega x_i))$.

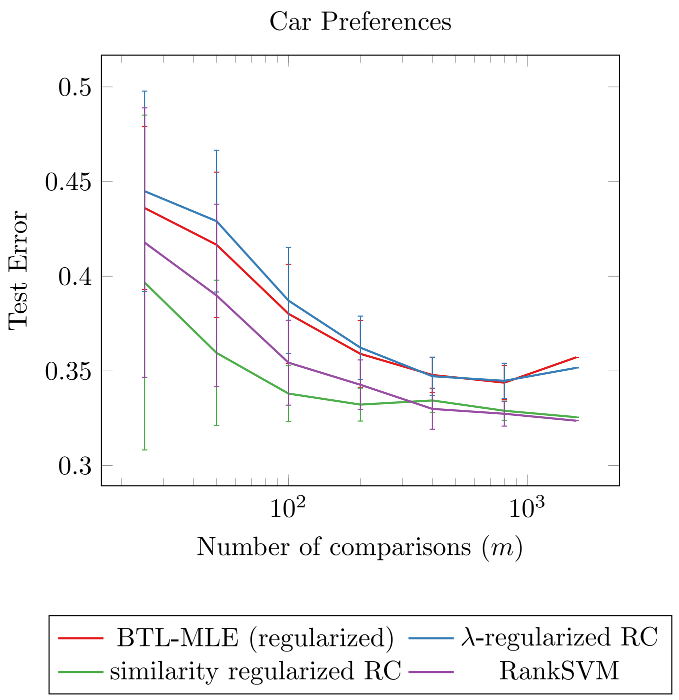


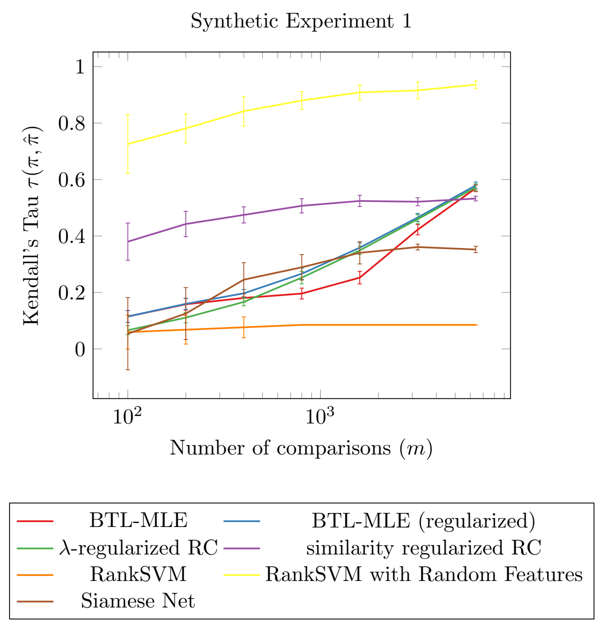
Figure 3.3: Performance of various algorithms when given a small number of observations from the Car Preferences dataset. Similarity-based regularization was run with tightness parameter $\tau = 21$.

- In Experiment 2, we generated 1600 points $\{x_i\}_{i=1}^{1600}$ chosen uniformly at random from $[0, 4]^2$, we chose $\omega_1, \omega_2, \ldots, \omega_4 \in \mathbb{R}^2$ at random, each entry chosen independently from a Guassian. To each $i \in [1600]$ we associate a score $w_i = w_i = \sum_{h=1}^{2} \exp(\cos(5\omega_h^T x_i)) + \sum_{h=3}^{4} \exp(\omega_h^T x_i/10)$.

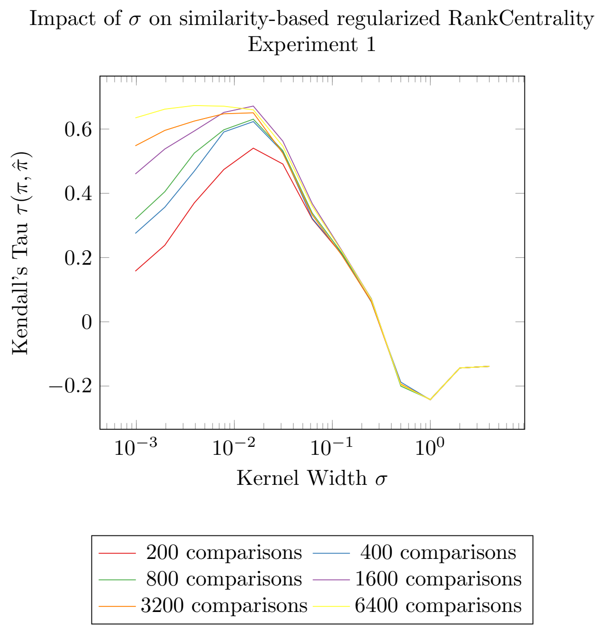
- In Experiment 3, we generated 1000 points $\{x_i\}_{i=1}^{1000}$ chosen uniformly at random from $[0, 4]^{10}$. We chose $\omega_1, \omega_2, \ldots, \omega_5 \in \mathbb{R}^{10}$ at random, each entry chosen independently from a Guassian. To each $i \in [1000]$ we associate a score $w_i = \sum_{h=1}^{5} \exp(\omega_h^T x_i/10)$.

Similar to the experiment on Car Preferences, for various values of m, we simulated m observations 20 times (by choosing a pair uniformly from all possibilities and choosing a winner according to the BTL model) and ran each algorithm on each of the 20 sets of m observations. We recorded the mean and standard deviation of the Kendal-tau correlation metric (with respect to the synthetic scores we generated). The results of these experiments are summarized in Figures 3.4a, 3.5a, and 3.6a.

The impact of similarity-based regularization in Experiment 1 is dramatic when compared to λ-regularized RankCentrality. While it is true that RankSVM with random features far outperforms other algorithms, it should not come as a surprise given that the BTL scores w_i, as a function of x_i, come from precisely the kind of function that random features are supposed to help learn.



(a) Performance of various algorithms when given a small number of observations from synthetic experiment 1. Similarity-based regularization was run with kernel width $\sigma = 2^{-5}$.



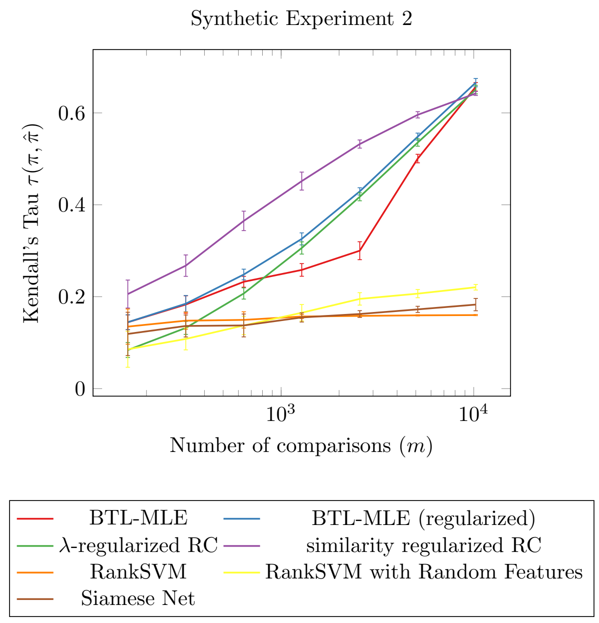
(b) Impact of kernel width on performance of similarity based regularized RankCentrality.

Figure 3.4: Synthetic Experiment 1

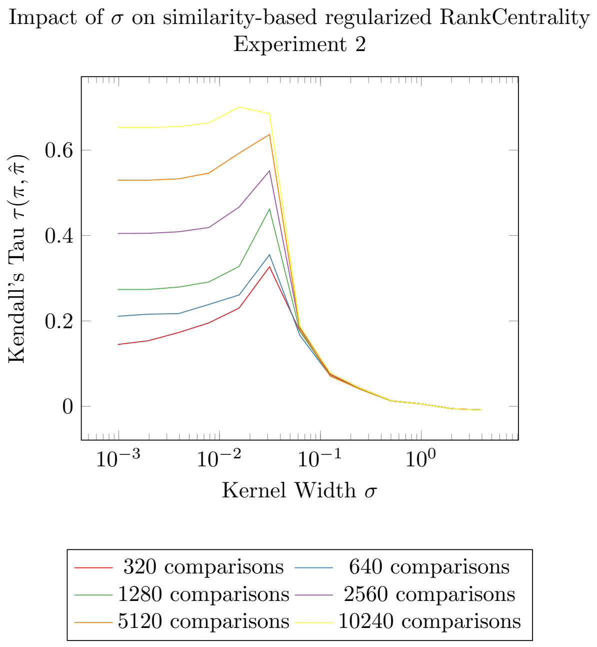
In Experiment 2, similarity-based regularization proves to be the best method when the number of comparisons is small, but the regularization biases the estimator sufficiently that algorithms that do not consider the features (e.g., the BTL-MLE and λ-regularized RankCentrality) eventually out-perform it.

Finally, in Experiment 3, the dataset is generated from precisely the kind of function we would expect RankSVM and Siamese Nets to learn and this is reflected in the plots. While similarity-based regularization outperforms methods that do not use any feature information it should not surprise the reader that in this case, it does not perform as well as the methods that designed to learn scoring functions such as the one we used in this experiment.

Figures 3.4b, 3.5b, and 3.6b demonstrate the role that the kernel width σ plays. Recall that as $\sigma \to 0$, we have $D \to D_\lambda$. The left end of these plots represents the case when we have hardly regularized and the right end the case when we have over-regularized. As one should expect, when we have only a small number of comparisons, regularization can have a beneficial impact. However, when we are given sufficiently many pairwise comparisons, we see that λ-regularized RankCentrality already performs very well and the bias introduced by similarity-based regularization becomes apparent, especially in 3.4b.

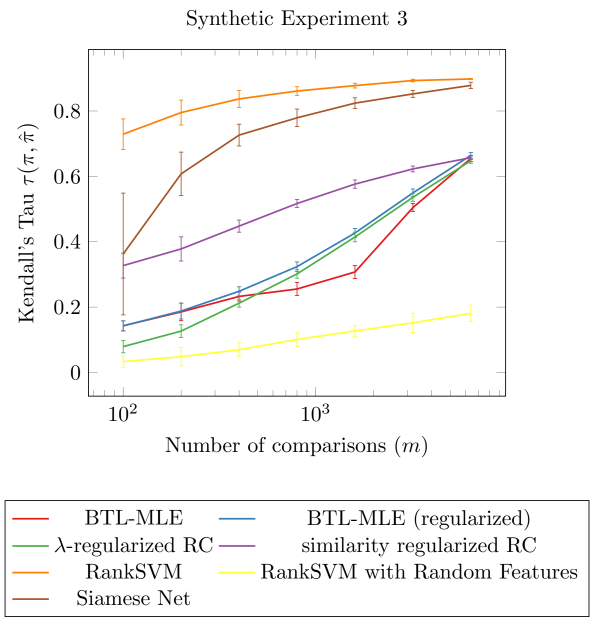


(a) Performance of various algorithms when given a small number of observations from synthetic experiment 2. Similarity-based regularization was run with kernel width $\sigma = 2^{-4}$.

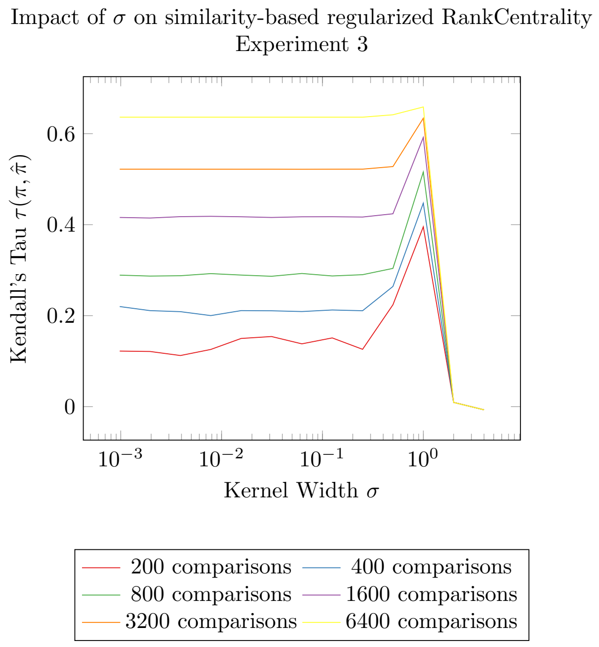


(b) Impact of kernel width on performance of similarity based regularized RankCentrality.

Figure 3.5: Synthetic Experiment 2



(a) Performance of various algorithms when given a small number of observations from synthetic experiment 3. Similarity-based regularization was run with kernel width $\sigma = 2^{-5}$.



(b) Impact of kernel width on performance of similarity based regularized RankCentrality.

Figure 3.6: Synthetic Experiment 3

3.5.3 Baseline methods

In the following paragraphs, we describe the baseline methods that we have compared our method against. In each of these algorithms, it is assumed that associated to each item $i \in [n]$ there is a vector $x_i \in \mathbb{R}^d$ and the goal is to learn a ranking function $f : \mathbb{R}^d \to R$ such that item i is ranked lower than item j if and only if $f(i) < f(j)$. The methods vary in the assumptions they make about f and how they learn such a function.

RankSVM was first introduced by Joachims [Joa02] to rank items with features (Joachims generated features as a function of the search query and document/item, our experiments can be considered as a special case where there is only one query). RankSVM tries to learn a ranking function f that is linear, i.e., $f : x \mapsto w^T x$. Note that this can be solved with standard implementations of the SVM algorithm by treating every pairwise comparison observation (i, j, y) as a point $x_i - x_j$ with label y. In this case, an observation (i, j, y) conveys information about the sign of $f(x_i) - f(x_j)$ should be higher and $f(x_i) - f(x_j) = w^T(x_i - x_j)$.

The RankSVM cannot naïvely be kernelized to learn non-linear functions because the distributive property $f(x_i) - f(x_j) = f(x_i - x_j)$ we used above no longer holds. To overcome this problem, we transformed the features for each point using random features, as described in [RR08] and implemented in [Ped+11]. These random features can give a finite-dimension approximation to the infinite dimension inner product space that the RBF kernel is equivalent to. Since we are fitting a linear function on this space, the distributive property now holds and we can run RankSVM on the random features space to learn a non-linear function $f : \mathbb{R}^d \to \mathbb{R}$.

Siamese Nets were originally introduced by Bromley, et. al. in [Bro+94] to detect forged signatures and now have numerous applications, especially in one-shot learning. A Siamese network consists of two identical and simultaneously updated copies (the "left" and "right" copies) of a base neural network and whose outputs are combined appropriately (for example, in one-shot image recognition, it is common to measure a norm of the differences in output and apply an appropriate transformation). We implemented a Siamese network using Keras [Cho+15] with two hidden dense layers, each with 20 nodes and a dropout factor of 0.1, and an output dimension of 1. Each layer in the base network used a ReLU activation. The outputs of the right network is subtracted from that of the left and a sigmoid transformation is applied to this difference. Thus, the Siamese Net is a function $g : \mathbb{R}^d \times \mathbb{R}^d \to (0, 1)$ such that $g(x, y) = 1 - g(y, x)$.

3.6 Comparison of Algorithms

In this chapter, we have summarized two versions of the RankCentrality algorithm from the literature and presented a version of the algorithm that is easier to analyze and gives a stronger sample complexity bound for a natural sampling scheme. We proposed a λ-regularized RankCentrality algorithm that is able to give meaningful non-trivial output when the number of comparisons is small. Finally, we also demonstrate other ways of regularizing RankCentrality that can incorporate feature information and we give an empirical analysis.

3.6.1 Sampling Schemes

Negahban, et. al., in their original paper [NOS16], assume a sampling scheme that we find somewhat unnatural. Their analysis assumes that the set of pairs compared form an Erdős–Rényi graph $G(n, p)$ and that each edge (pair) in the graph is compared a fixed k times. Rajkumar and Agarwal [RA14] give a more natural sampling scheme: they assume that each comparison is independent of the previous comparisons and the pair (i, j) is compared with probability μ_{ij} (where $i < j$). We consider this sampling scheme to be more natural.

3.6.2 Importance Sampling

An important difference between Rajkumar and Agarwal's transition matrix $\hat{Q}^{\mathrm{RA}}$ defined in (3.4) and our transition matrix $\hat{Q}$ defined in (3.5) is that while $\hat{Q}^{\mathrm{RA}}$ is constructed using the empirical comparison matrix $\hat{P}$, we build $\hat{Q}$ directly from the comparisons $S = \{(i_k, j_k, y_k)\}_{k=1}^{m}$. The $\hat{Q}^{\mathrm{RA}}$ matrix essentially weights each observation by the reciprocal of the number of times the corresponding pair was observed. While this helps to handle the case when μ_{ij} is not known, it means that entries in $\hat{Q}^{\mathrm{RA}}$ cannot be written as a sum of independent random variables. On the other hand, $\hat{Q}$, the empirical markov chain that we construct in this chapter, is a sum of m independent random matrices where each matrix encodes the outcome of a comparison, which is possible only when μ_{ij} is known. This discussion is not as relevant to Negahban, et. al.'s analysis as they assume that each observed pair is observed a constant number of times.

3.6.3 Ergodicity of Empirical Markov Chain

A Markov chain is ergodic if it is irreducible (underlying directed graph is strongly connected) and aperiodic (gcd of length of paths from any i to j is 1). Because $\hat{Q}$, $\hat{Q}^{\mathrm{NOS}}$, and $\hat{Q}^{\mathrm{RA}}$ all have self-loops, irreducibility implies aperiodicity, which leaves irreducibility as the only condition required to guarantee a unique stationary distribution.

Negahban, et. al. require that the graph of comparisons is connected and hence, for large enough k, with high probability the Markov chain $\hat{Q}^{\mathrm{NOS}}$ is ergodic (irreducible and aperiodic). The RankCentrality algorithm in Rajkumar and Agarwal outputs the unique stationary distribution only if $\hat{Q}^{\mathrm{RA}}$ is irreducible (and outputs $\mathbf{0}$ otherwise). Unfortunately, when m, the number of comparisons, is small, $\hat{Q}^{\mathrm{RA}}$ is unusually not irreducible. The λ-regularized RankCentrality algorithm we propose guarantees that the Markov chain $\hat{Q}D_{\lambda}$ is irreducible and hence is appropriate to use even when the number of comparisons is small. We give a detailed analysis of the bias-variance trade-off of λ-regularized RankCentrality.

3.6.4 Incorporating Feature or Similarity Information

The λ-regularized RankCentrality is useful even when there are no features available, but we extend the idea of multiplying the transition matrix $\hat{Q}$ by a matrix D to construct matrices D that can impute pairwise comparisons based on similarity information. By doing so, we solve a problem that is distinct from the two problems addressed the literature:

- learn a ranking on n items using pairwise comparisons between them, and
- learn a ranking function $f : \mathbb{R}^d \to \mathbb{R}$ using pairwise comparisons between n items where we have features x_i for each item $i \in [n]$.

Instead, we learn a ranking on n items using pairwise comparisons between them and use the associated features $x_i \in \mathbb{R}^d$ (or other similarity information) to offset an insufficient or small number of observations without learning a ranking function $f : \mathbb{R}^d \to \mathbb{R}$.

BIBLIOGRAPHY

[Abb+13] E. Abbasnejad et al. “Learning Community-based Preferences via Dirichlet Process Mixtures of Gaussian Processes”. In: *In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2013.

[AK98] Edoardo Amaldi and Viggo Kann. “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems”. In: *Theoretical Computer Science* 209.1 (1998), pp. 237–260. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(97)00115-1. URL: http://www.sciencedirect.com/science/article/pii/S0304397597001151.

[Bat94] R. Battiti. “Using mutual information for selecting features in supervised neural net learning”. In: *IEEE Transactions on Neural Networks* 5.4 (July 1994), pp. 537–550. ISSN: 1045-9227. DOI: 10.1109/72.298224.

[Ben+11] Sean C. Bendall et al. “Single-Cell Mass Cytometry of Differential Immune and Drug Responses Across a Human Hematopoietic Continuum”. In: *Science* 332.6030 (2011), pp. 687–696. ISSN: 0036-8075. DOI: 10.1126/science.1198704. eprint: https://science.sciencemag.org/content/332/6030/687.full.pdf. URL: https://science.sciencemag.org/content/332/6030/687.

[BKV] Robert M. Bell, Yehuda Koren, and Chris Volinsky. *The BellKor solution to the Netflix Prize*.

[BL] James Bennett and Stan Lanning. *The Netflix Prize*.

[Bro+12] Gavin Brown et al. “Conditional likelihood maximisation: a unifying framework for information theoretic feature selection”. In: *Journal of machine learning research* 13.Jan (2012), pp. 27–66.

[Bro+94] Jane Bromley et al. “Signature Verification using a “Siamese” Time Delay Neural Network”. In: *Advances in Neural Information Processing Systems 6*. Ed. by J. D. Cowan, G. Tesauro, and J. Alspector. Morgan-Kaufmann, 1994, pp. 737–744. URL: http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.pdf.

[Cho+15] François Chollet et al. *Keras*. https://keras.io. 2015.

[CT06] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 2006. ISBN: 9780471241959.

[DRS18] A Duò, MD Robinson, and C Soneson. “A systematic performance evaluation of clustering methods for single-cell RNA-seq data [version 2; peer review: 2 approved]”. In: *F1000Research* 7.1141 (2018). DOI: 10.12688/f1000research.15666.2.

[Dwo+01] Cynthia Dwork et al. "Rank Aggregation Methods for the Web". In: *Proceedings of the 10th International Conference on World Wide Web*. WWW '01. Hong Kong, Hong Kong: ACM, 2001, pp. 613–622. ISBN: 1-58113-348-0. DOI: 10.1145/371920.372165. URL: http://doi.acm.org.proxy.lib.umich.edu/10.1145/371920.372165.

[EEA08] Ali El Akadi, Abdeljalil El Ouardighi, and Driss Aboutajdine. "A Powerful Feature Selection approach based on Mutual Information". In: *International Journal of Computer Science and Network Security* 8.4 (Apr. 2008), pp. 116–121.

[Eng+19] Chee-Huat Linus Eng et al. "Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+". In: *Nature* 568.7751 (2019), pp. 235–239. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1049-y. URL: https://doi.org/10.1038/s41586-019-1049-y.

[ES08] A.E. Elo and S. Sloan. *The Rating of Chess Players, Past and Present*. Ishi Press International, 2008. ISBN: 9780923891275. URL: https://books.google.com/books?id=syjcPQAACAAJ.

[Fle04] François Fleuret. "Fast Binary Feature Selection with Conditional Mutual Information". In: *J. Mach. Learn. Res.* 5 (Dec. 2004), pp. 1531–1555. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1005332.1044711.

[GGL95] R.L. Graham, M. Grőtschel, and L. Lovász, eds. *Handbook of Combinatorics: Vol. 2*. Elsevier, 1995. ISBN: 9780444880024.

[Gie+14] Charlotte Giesen et al. "Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry". In: *Nature Methods* 11 (Mar. 2014). Article, p. 417. URL: https://doi.org/10.1038/nmeth.2869.

[Gie+17] Todd M. Gierahn et al. "Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput". In: *Nature Methods* 14 (Feb. 2017), p. 395. URL: https://doi.org/10.1038/nmeth.4179.

[Gre+18] Christopher Daniel Green et al. "A Comprehensive Roadmap of Murine Spermatogenesis Defined by Single-Cell RNA-Seq". In: *Developmental Cell* 46.5 (2018), 651–667.e10. ISSN: 1534-5807. DOI: https://doi.org/10.1016/j.devcel.2018.07.025. URL: http://www.sciencedirect.com/science/article/pii/S1534580718306361.

[Han75] Te Sun Han. "Linear dependence structure of the entropy space". In: *Information and Control* 29.4 (1975), pp. 337–368. ISSN: 0019-9958.

[HTF13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013. ISBN: 9780387216065.

[JN11] Kevin G Jamieson and Robert Nowak. "Active Ranking using Pairwise Comparisons". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 2240–2248. URL: http://papers.nips.cc/paper/4427-active-ranking-using-pairwise-comparisons.pdf.

[Joa02] Thorsten Joachims. “Optimizing Search Engines Using Clickthrough Data”. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’02. Edmonton, Alberta, Canada: ACM, 2002, pp. 133–142. ISBN: 1-58113-567-X. DOI: 10.1145/775047.775067.

[Kat+18] Sumeet Katariya et al. “Adaptive Sampling for Coarse Ranking”. In: *International Conference on Artificial Intelligence and Statistics*. 2018, pp. 1839–1848.

[KS40] M. G. Kendall and B. Babington Smith. “On the Method of Paired Comparisons”. In: *Biometrika* 31.3-4 (Mar. 1940), pp. 324–345. ISSN: 0006-3444. DOI: 10.1093/biomet/31.3-4.324. eprint: http://oup.prod.sis.lan/biomet/article-pdf/31/3-4/324/498992/31-3-4-324.pdf. URL: https://doi.org/10.1093/biomet/31.3-4.324.

[LT06] Dahua Lin and Xiaoou Tang. “Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion”. In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 68–82.

[Mac+15] Evan?Z Macosko et al. “Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets”. In: *Cell* 161.5 (May 2015), pp. 1202–1214. ISSN: 0092-8674. DOI: 10.1016/j.cell.2015.05.002. URL: https://doi.org/10.1016/j.cell.2015.05.002.

[NOS16] Sahand Negahban, Sewoong Oh, and Devavrat Shah. “Rank centrality: Ranking from pairwise comparisons”. In: *Operations Research* 65.1 (2016), pp. 266–287.

[Ntr+18] Vasilis Ntranos et al. “Identification of transcriptional signatures for cell types from single-cell RNA-Seq”. In: *bioRxiv* (2018). DOI: 10.1101/258566.

[NWF78] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. “An analysis of approximations for maximizing submodular set functions-I”. In: *Mathematical programming* 14.1 (1978), pp. 265–294.

[Pau+15] Franziska Paul et al. “Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors”. In: *Cell* 163.7 (2015), pp. 1663–1677. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2015.11.013. URL: http://www.sciencedirect.com/science/article/pii/S0092867415014932.

[Ped+11] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[PLD05] Hanchuan Peng, Fuhui Long, and C. Ding. “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (Aug. 2005), pp. 1226–1238. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.159.

[RA14] Arun Rajkumar and Shivani Agarwal. “A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data”. In: *Proceedings of the 31st International Conference on Machine Learning*. 2014.

[Raj+08] Arjun Raj et al. "Imaging individual mRNA molecules using multiple singly labeled probes". In: *Nature Methods* 5 (Sept. 2008), p. 877. URL: https://doi.org/10.1038/nmeth.1253.

[RR08] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt et al. Curran Associates, Inc., 2008, pp. 1177–1184. URL: http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf.

[Sat+15] Rahul Satija et al. "Spatial reconstruction of single-cell gene expression data". In: *Nature Biotechnology* 33 (Apr. 2015), p. 495. URL: http://dx.doi.org/10.1038/nbt.3192.

[Thu45] L. L. Thurstone. "The prediction of choice." In: *Psychometrika* 10 (1945), pp. 237–253. DOI: 10.1007/BF02288891.

[Tro12] Joel A. Tropp. "User-Friendly Tail Bounds for Sums of Random Matrices". In: *Foundations of Computational Mathematics* 12.4 (Aug. 2012), pp. 389–434. ISSN: 1615-3383.

[Von07] Jan Vondrák. "Submodularity in Combinatorial Optimization". Ph.D. thesis. Charles University, Prague, Czech Republic, Nov. 2007. URL: https://theory.stanford.edu/~jvondrak/data/KAM_thesis.pdf.

[WAT18] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome Biology* 19.1 (Feb. 2018), p. 15. ISSN: 1474-760X. DOI: 10.1186/s13059-017-1382-0.

[YM00] Howard Hua Yang and John Moody. "Data Visualization and Feature Selection: New Algorithms for Nongaussian Data". In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 2000, pp. 687–693.

[Zei+15] Amit Zeisel et al. "Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq". In: *Science* 347.6226 (2015), pp. 1138–1142. ISSN: 0036-8075. DOI: 10.1126/science.aaa1934. eprint: http://science.sciencemag.org/content/347/6226/1138.full.pdf. URL: http://science.sciencemag.org/content/347/6226/1138.

[Zhe+17] Grace X. Y. Zheng et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nature Communications* 8 (Jan. 2017), p. 14049. URL: http://dx.doi.org/10.1038/ncomms14049.