# Statistical techniques for exploratory analysis of structured three-way and dynamic network data

by

Shawn Pankaj Mankad

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2013

Doctoral Committee:

Professor George Michailidis, Chair
Associate Professor Lada Adamic
Professor Kerby Shedden
Professor Ji Zhu

# TABLE OF CONTENTS

# LIST OF FIGURES

**Figure**

# LIST OF TABLES

**Table**

# ABSTRACT

Statistical techniques for exploratory analysis of structured
three-way and dynamic network data

by

Shawn Mankad

Advisor: George Michailidis

In this thesis, I develop different techniques for the pattern extraction and visual exploration of a collection of data matrices. Specifically, I present methods to help home in on and visualize an underlying structure and its evolution over ordered (e.g., time) or unordered (e.g., experimental conditions) index sets. The first part of the thesis introduces a biclustering technique for such three dimensional data arrays. This technique is capable of discovering potentially overlapping groups of samples and variables that evolve similarly with respect to a subset of conditions. To facilitate and enhance visual exploration, I introduce a framework that utilizes kernel smoothing to guide the estimation of bicluster responses over the array. In the second part of the thesis, I introduce two matrix factorization models. The first is a data integration model that decomposes the data into two factors: a basis common to all data matrices, and a coefficient matrix that varies for each data matrix. The second model is meant for visual clustering of nodes in dynamic network data, which often contains complex evolving structure. Hence, this approach is more flexible and additionally lets the basis evolve for each matrix in the array. Both models utilize a regularization within

the framework of non-negative matrix factorization to encourage local smoothness of the basis and coefficient matrices, which improves interpretability and highlights the structural patterns underlying the data, while mitigating noise effects. I also address computational aspects of applying regularized non-negative matrix factorization models to large data arrays by presenting multiple algorithms, including an approximation algorithm based on alternating least squares.

# CHAPTER I

# Introduction and Literature Review

## 1.1 Overview

An important problem underlying many emerging statistical applications is to discover how components interact with each other in massive, complex systems, where, due to technological advances, researchers can collect data over time or in different conditions at the component level. Given the complex nature of the data, visual exploration and pattern extraction arguably have increased importance to decision making processes, and can contribute towards performing a number of critical tasks. For instance, learning the underlying structure and summarizing its evolution can be used to compress and organize large sized data. Exploratory and clustering techniques can also facilitate decision making by simplifying the complex structure of the data and pinpointing important patterns. As an example, in gene expression data, identifying groups of genes that are co-expressed under different conditions can improve disease diagnosis and further our understanding of gene regulatory networks. With economic data, uncovering sectors that respond similarly to the ebbs and flows of the larger economy can improve resource allocation and policy decisions.

In this dissertation, it is assumed the data is structured, that is, I observe many samples for a number of variables across different time points or experimental conditions. Such data can be organized into three-dimensional (three-way) arrays, with

the first two dimensions corresponding as usual to samples and variables, respectively, while the third dimension to time or experimental conditions. This is a reasonable assumption for many real world problems. For instance, in genomics and economics, data arrays commonly feature samples (genes or individuals, firms, etc.) on the rows and covariates on the columns. The third depth dimension corresponds to time, so that the data array can be conceptualized as a time-series of individual data matrices. Data arrays in pharmacology commonly feature a third dimension that indexes dose levels (experimental conditions).

A major challenge in analyzing such data arrays is that the additional dimension does not allow straight-forward application of most statistical techniques. For instance, consider the following approaches. One may proceed by collapsing the third dimension and working with a single two dimensional matrix. However, this coarse graining step masks the finer structure in the data that may be important for certain analyses. Another option could be to analyze each data matrix independently and then look for concordant patterns and features. However, this strategy ignores the additional structure like time or conditions, and it may also be hard to interpret the results.

This dissertation takes an approach that is between these direct strategies by utilizing kernel smoothing and regularization for a variety of applications in clustering, data integration, and visualization of three-dimensional data arrays.

Chapter II of this dissertation introduces a biclustering technique for three-way data arrays. The main idea is to decompose each data slice into a series of additive layers that capture the underlying structure of the data. The new technique is especially useful for discovering and characterizing the evolution through the data of unknown and potentially overlapping groups of samples and variables. To enhance visual exploration and robustness, I introduce an algorithm that utilizes kernel smoothing to guide the estimation of bicluster responses over the array. I also discuss

computational aspects by developing an estimation algorithm, and show it is capable of handling large size data through numerical experiments.

In Chapter III, I introduce a regularized non-negative matrix factorization model for a variety of applications in data integration and visualization of three-way data. The goal is to find low rank representations of the data, where a common basis captures the most persistent structure, and factors at different times or conditions are close together if they are in neighboring data slices. This local smoothness is encouraged through a penalization framework, where the size and amount of smoothing are set by the user to influence the analysis.

Chapter IV introduces a variant of the model from Chapter III for visual clustering of nodes in dynamic network data. Such data often contains complex evolving structure, and hence, this approach is more flexible and additionally lets the basis evolve for each matrix in the array. A variant of the regularization and estimation algorithm from the previous chapter are developed and illustrated on a variety of synthetic and real world network data sets.

Possible extensions and future work are discussed in Chapter V.

## 1.2 Non-negative Matrix Factorization

Matrix factorizations have become part of the standard repertoire for pattern identification and dimension reduction. The most common one is the Singular Value Decomposition (SVD), which has fundamental connections to principal component analysis (PCA), multi-dimensional scaling (MDS), among others, and is commonly applied for such low-rank representations and analyses (*Hastie et al.*, 2001).

The non-negative matrix factorization (NMF) is an alternative that has been shown to be advantageous for visualization of non-negative data. Non-negative data commonly occur in networks, as edges commonly correspond to flows, capacity, or binary relationships. Image and text processing, and other applications in the social

sciences also frequently feature non-negative data.

NMF has been successfully employed in a diverse set of areas, including computer vision (*Lee and Seung*, 1999), environmetrics (*Paatero and Tapper*, 1994), and computational biology (*Devarajan*, 2008). There has been work that explains its usefulness by posing it as a relaxed version of k-means and other spectral clustering methods (*Ding et al.*, 2005, 2008). It shares a common algebraic form with SVD, since both factorizations approximate a given data matrix $X \in \mathbb{R}^{n \times p}$ with an outer product

$$X \approx UV^T, \tag{1.1}$$

of two matrices $U \in \mathbb{R}^{n \times K}, V \in \mathbb{R}^{p \times K}$ for $K \leq \min\{n, p\}$. The rank of the approximation is chosen similarly to other matrix factorizations to obtain a good fit to the data while achieving interpretability.

The key difference between SVD and NMF are the constraints that are placed on $U$ and $V$. SVD imposes a particular geometry on the factorization, so that $U$ and $V$ can each be viewed as coordinate systems that fit the data. In particular, each (eigen)vector $U_i$ is perpendicular to every other vector $U_j$, so that the collection $\{U_1, ..., U_K\}$ forms a lower dimensional orthonormal space that the data can be visualized in. In addition, $U$ satisfies $U^T U = I$ (orthonormality constraints). A similar characterization holds for $\{V_1, ..., V_K\}$. In contrast, with NMF the orthogonality constraints are replaced with a restriction of non-negativity of the factorized matrices (*Lee and Seung*, 1999, 2001). That is, every element of $U$ and $V$ is greater than or equal to zero. The geometric characterization of SVD is traded for the enhanced interpretability that comes from strictly additive combinations. For instance, consider (1.1) in element form $X_{ij} = \sum_{k=1}^{K} U_{ik} V_{kj}$. Since each term is non-negative, each term of the sum can be thought of as the contribution of cluster $k$ to element $X_{ij}$.

A standard objective function for NMF minimizes the Frobenius distance between the given data and a lower dimensional reconstruction of it. In particular, the objec-

tive function is

$$\min_{U,V} ||X - UV^T||_F^2 \text{ such that } U_{ik}, V_{jk} \geq 0. \tag{1.2}$$

The optimization problem is a challenging one, since the objective function is convex in $U$ only or $V$ only, not in both simultaneously. Further, the non-negativity constraint is not enough to guarantee uniqueness of the factors, so that the estimates are always rescalable (scale invariant). For further discussion on the issue of identifiability, see *Wang and Zhang* (2012) and references therein.

The benchmark algorithm for NMF was proposed by *Lee and Seung* (1999, 2001), and is known as 'multiplicative updating'. The algorithm can be viewed as an adaptive gradient descent, and was shown to find local minima of the objective function. It is relatively simple to implement, but can converge slowly due to its linear convergence rate (*Chu et al.*, 2004). Even though this approach can be slow to converge, especially as the algorithm approaches a limit point, in practice my extensive numerical work shows that after a handful of iterations, the algorithm results in visually meaningful factorizations. In Chapters III and IV, a multiplicative updating algorithm and corresponding convergence results are developed.

There has been work indicating that this class of algorithms converges to less satisfactory solutions (see *Berry et al.* (2006), *Wang and Zhang* (2012) and references therein). Another popular and more flexible class of NMF algorithms is the alternating non-negative least squares algorithm (ANLS), first proposed for NMF in *Paatero and Tapper* (1994) (see *Kim and Park* (2008) for a more recent reference). This type of iterative algorithm exploits the biconvexity of the objective function by holding one argument fixed at each step and solving for the other using constrained least squares. The ANLS algorithm will converge to a local minimum of the objective function, and at a faster convergence rate. However, the cost per iteration is higher than multiplicative updating. In Chapter III, I discuss some technical challenges of ANLS and develop an approximation algorithm that has been shown to work well in

practice.

The methods introduced in Chapters III and IV rely on examining how lower dimensional matrix representations evolve through the data, and controlling their evolution using constrained optimization. The constraint strengths that control how sensitive the matrix representations are to short term fluctuations are set by the user to steer the analysis. The use of additional constraints in matrix factorizations is a common technique to reveal additional structure within the data. I refer to this class of models as penalized matrix factorizations, since usually the constraints are represented as penalties using the Lagrangian form of an objective function. In penalized matrix factorizations, the factorized matrices are obtained through minimizing an objective function that consists of a goodness of fit component and a roughness penalty. The strength of the penalty is set by the user, where a larger penalty encourages smoother $U$ and $V$. Penalized NMF has been explored extensively in previous works to encourage sparsity or smoothness of the factors (see *Berry et al.* (2006); *Chen and Cichocki* (2005); *Hoyer* (2002, 2004); *Cai et al.* (2011) and references therein).

Previous works usually consider a static setting, that is, applying factorization to a single data matrix. This dissertation uses penalties as a way to extend matrix factorization to a collection of matrices. Thus, the problem I consider poses additional modeling challenges, because I observe many matrices, and does not directly fit into existing approaches due to either the time series component or multiple, correlated variables at each time point.

Non-negative tensor factorizations are also closely related to Chapters III and IV (for overviews, see *Cichocki and Zdunek* (2007); *Hazan et al.* (2005); *Welling and Weber* (2001); *Cichocki et al.* (2007)). In fact, NMF can be seen as a special case of non-negative 2-dimensional tensor factorization. As a consequence of the generalization, the optimization problems associated with tensor models are usually challenging, and practical matters like displaying the estimates become nontrivial. Existing tensor

factorizations differ from Chapter III in terms of the model and regularization framework. For instance, the model in Chapter III uses a common basis. This reduces the number of estimable parameters and is particularly suitable for data integration. The regularization framework allows the detection of nonlinear and hidden structures in the data by encouraging local smoothness. In Chapter IV, I utilize a new penalized tensor approach to visualization and pattern extraction in dynamic networks.

# CHAPTER II

# Biclustering Three-Dimensional Data Arrays with Plaid Models

## 2.1 Introduction

A main focus in the literature about clustering has been on partitioning samples into interpretable groups. However, in many applications it is more realistic to discover groups of both samples and variables due to the heterogeneity of the data and number of variables measured. Biclustering provides such flexibility by selecting important variables and relaxing 'hard' partitioning of samples, e.g., allowing samples to be in more than one cluster, or in none at all; Variables in the cluster can be defined with respect to only a subset of samples, not necessarily with respect to all of them. In the context of three-way data, it is natural to define these potentially overlapping groups of samples and variables (henceforth referred to as *biclusters*) that evolve similarly with respect to a subset of conditions.

In this chapter, I introduce bicluster in three-way data by modeling a bicluster's evolution through the data using a variety of curve estimation techniques within the plaid model of *Lazzeroni and Owen* (2000). For instance, in one application I examine bilateral trade data between countries in the United Nations over time. As shown in Figure 2.1, the proposed model can model a global growth curve describing the

average world trade over time. Given this growth curve (global mean), particular groups of countries, acting as importers and exporters, with mean trade levels that change smoothly over time are discovered in the biclusters.

The main idea is to first account for system-wide behavior by modeling the full data's mean structure over conditions (or time). Then, detect biclusters that exhibit deviations for some conditions, and estimate each bicluster's dependence over conditions using methods from functional data analysis.

There are many benefits of the proposed approach. First, the model helps users identify and visualize interesting patterns in complex structured data, while incorporating knowledge of the underlying generating process. This facilitates data exploration by displaying different types of curves and characteristics, and ultimately facilitates information extraction and decision making. Second, it allows for inference at unobserved conditions, which may be of interest in dose response (*Rosenberger and Haines*, 2002) and statistical calibration studies (*Osborne*, 1991), among others. Lastly, our algorithmic framework is computationally scalable and easy to implement.

The remainder of this chapter is organized as follows: In the next section, I briefly review related approaches for similarly structured data; Section 2.3 reviews the plaid model for static (cross-sectional) data, and the proposed extension for three-dimensional data. Section 2.4 contains a simulation study of the proposed and existing techniques. Section 2.5 illustrates the proposed model on gene expression and economic trade data, and the chapter concludes with a brief discussion in Section 2.6.

## 2.2 Related Approaches

Problems arising from genomics have motivated the development of many model based clustering approaches for longitudinal data. For example, curve-based clustering algorithms have been proposed to analyze data from time course microarray experiments, where thousands of genes are repeatedly measured over time (*Luan and*

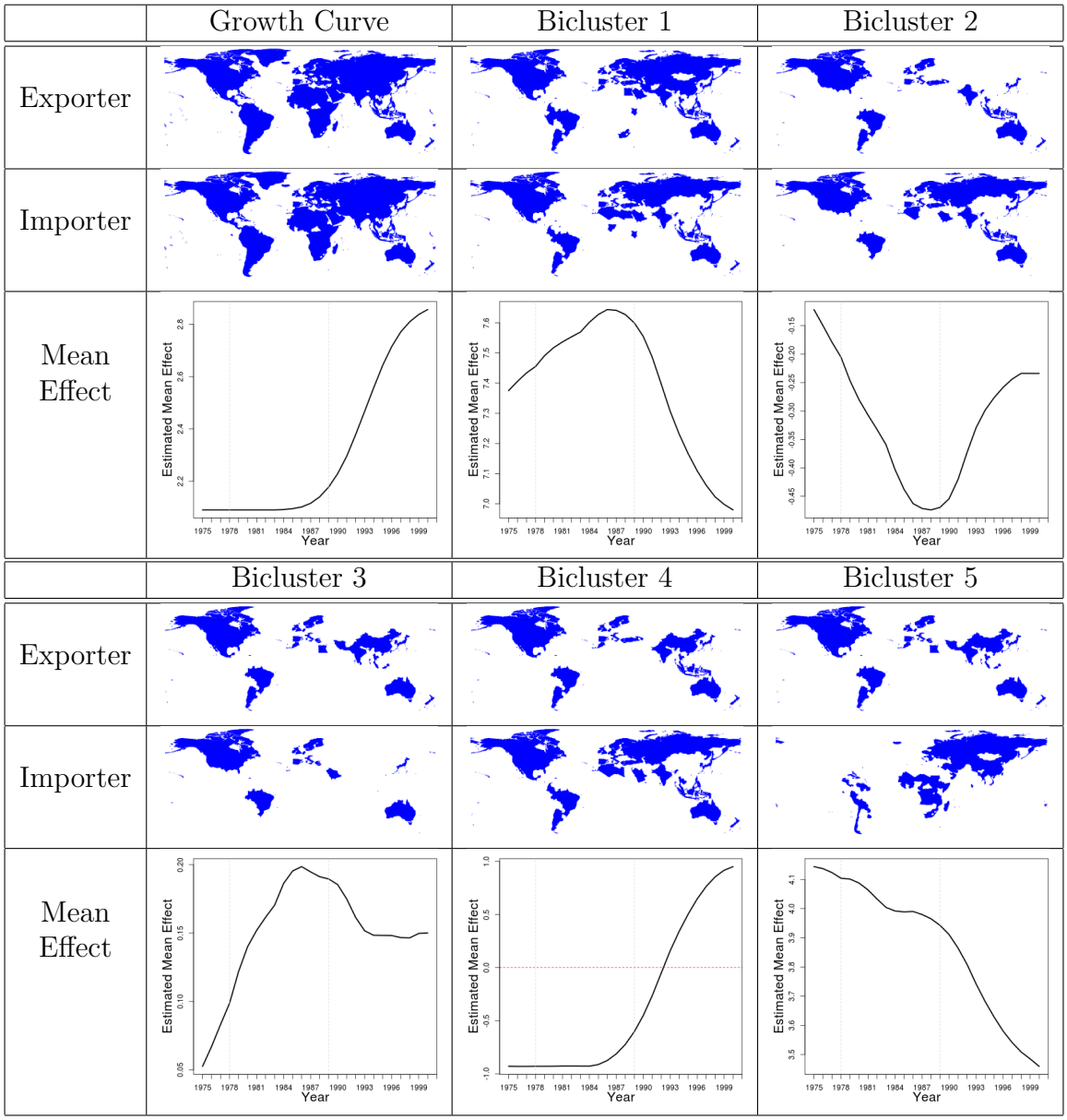



Figure 2.1: World bilateral trade results using a shape constrained growth curve for the global mean and time-smoothed bicluster mean effects.

*Li*, 2003; *Ma et al.*, 2006). In these approaches, the mean gene expressions are approximated with a linear combination of spline bases. In *Qin and Self* (2006) and *Eng et al.* (2008), a more parsimonious model is employed that assumes genes in the same cluster share the same mean and dependence structure over time. By defining the model at the cluster level, fewer parameters are required to be fitted, resulting in more stable estimates.

Even with these refinements, model based clustering for longitudinal data tends to require iterative algorithms for estimation (usually variants of EM) that become computationally expensive for large data. Moreover, these approaches tend to partition the samples into disjoint groups, which can mischaracterize finer structure in the data. Biclustering, a term first used by *Cheng and Church* (2000) in gene expression data analysis, provides additional flexibility by allowing samples to be in more than one cluster, or in none at all. Variables in the bicluster can be defined with respect to only a subset of samples, not necessarily with respect to all of them; further, biclusters may evolve with conditions.

Past works on biclustering mostly focus on extracting patterns within a single (static) data matrix when both the rows and columns are of scientific interest (see *Madeira and Oliveira* (2004) for a survey of past works). In this chapter, I extend the plaid model of *Lazzeroni and Owen* (2000), which decomposes data into a series of additive biclusters that capture the underlying structure of the data. Additional details will be provided in the next section. *Turner et al.* (2005b) provide an extension of the plaid model for repeated measures data. We further generalize this approach to handle data over experimental conditions or time, and interpolation at unobserved points. Further discussion, including an extensive simulation study involving the methodology in *Turner et al.* (2005b), are provided in Section 2.4.

A related approach to biclustering is matrix factorization, which can be extended to a three-dimensional data arrays with penalized matrix decompositions and tensor

factorizations. *Zou et al.* (2006); *Witten et al.* (2009) and *Guo et al.* (2010) relax the orthogonality constraints in singular value decomposition (SVD) and principal component analysis through $l_1$ and $l_2$ penalties. Extending such factorizations to three-way data leads to challenging optimizations problems. Further, practical matters like displaying and interpreting estimates become cumbersome. For instance, an extension of SVD that approximates each matrix observation with an outer product relies on an underlying model that is multiplicative in nature. This causes the number of parameters to grow rapidly with the number of matrix observations. In contrast, the proposed approach is built on an additive model that yields interpretable estimates, and an algorithmic framework that is computationally inexpensive.

## 2.3 The Plaid Model for Three Dimensional Arrays

This section begins with some background material on the plaid model. In particular, I introduce an important concept to the plaid model, namely that of a "layer". A layer is a canonical matrix matching the dimensions of the given data matrix, with zeros everywhere except the biclustered elements. In the plaid model, the data is decomposed into a series of additive layers that capture the underlying structure of the data. It includes a background layer consisting of all rows and columns to account for global effects in the data. Subsequent layers, which can overlap, represent additional effects corresponding to specific rows and columns that exhibit a strong pattern not explained by previous layers. Formally, the data matrix $X$ can be represented as

$$X_{ij} = \mu_0 + \sum_{k=1}^{K} \theta_{ijk} r_{ik} c_{jk} \tag{2.1}$$

where $\mu_0$ captures the uniform background, and $\theta_{ijk}$ describes the bicluster effects, with $k$ being a layer index running to $K$, the number of biclusters. The parameters $r_{ik}$ and $c_{jk}$ are indicator variables denoting bicluster membership for, respectively,

samples and variables.

There are several modeling choices for the form of $\theta_{ijk}$, the most common being

$$\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}, \tag{2.2}$$

where each bicluster has a mean, row, and column effect. Hence, each bicluster is assumed to be the sum of a mean background level plus row and column specific effects that capture additional structure. If only a mean effect is included, $\theta_{ijk} = \mu_k$, then one can write the model as a relaxed Singular Value Decomposition. In particular, $X = UDV$, where $U$ and $V$ are binary matrices of rank $K$ with each column denoting bicluster memberships. $D$ is square diagonal with elements $\mu_1, \ldots, \mu_K$. If row or column effects are included, the plaid model cannot be written in the SVD algebraic form (*Lazzeroni and Owen*, 2000).

The estimation procedure is an iterative algorithm based on minimizing the sum of squares of the data minus the fitted bicluster values. Suppose $K-1$ layers have been estimated in addition to the background layer. The residual data matrix is given by

$$\hat{Z}_{ij} = X_{ij} - \hat{\mu}_0 - \sum_{k=1}^{K-1} \hat{\theta}_{ijk}\hat{r}_{ik}\hat{c}_{jk}. \tag{2.3}$$

The $K$th bicluster is found by minimizing the residual sum of squares

$$\min_{\{\theta_{ijK}, r_{iK}, c_{jK}\}} \sum_{i=1}^{n}\sum_{j=1}^{p}(\hat{Z}_{ij} - \theta_{ijK}r_{iK}c_{jK})^2. \tag{2.4}$$

Estimates of the bicluster memberships ($\hat{r}_{iK}$,$\hat{c}_{jK}$) are obtained with a numerical search. When given bicluster memberships, the estimation of the bicluster-specific effects ($\theta_{ijK}$) is straightforward, as one can use the usual two-way Analysis of Variance estimators (*Turner et al.*, 2005a).

First the background layer is fitted, then biclusters are added one at a time until

no more statistically significant biclusters can be found, as determined by a permutation test. The permutation test relies on resampling the residual data matrix to approximate the significance of the bicluster. The basic idea is that the data values are independent of biclusters after permuting the rows and columns. Thus, comparing the candidate bicluster against biclusters obtained after randomizing the data, allows one to accept a bicluster only if it is significantly larger than what one would find in noise. More comprehensive discussion on this idea can be found in *Lazzeroni and Owen* (2000) and references therein.

I now present the proposed plaid model for three dimensional data arrays. Suppose I observe $\{X_m, m = 1, .., M\}$, where $X_m \in \mathbb{R}^{n \times p}$ and the third "depth" dimension indexed by $m$ corresponds to time, experimental conditions or factors. The data matrix $X_m$ can be represented as

$$X_{ijm} = \mu_{m0} + \sum_{k=1}^{K} \theta_{ijmk} r_{ik} c_{jk}. \tag{2.5}$$

In this model, the same biclustering structure applies to each data slice. In other words, the row (sample) and column (variable) memberships do not change over $m$, and the total number of biclusters $K$ is also the same for different $m$. These are fairly strong assumptions and may not be realistic for some real world applications, where the biclustering structures, including the number of biclusters, varies over time or under different conditions. However, relax these rigid assumptions can be relaxed by first partitioning the data over $m$ and then estimating the model in Equation 2.5 on each partition separately. This type of strategy assumes that biclustering structures are the same only in adjacent time points or similar conditions. It is also worth noting that without partitioning, the model in Equation 2.5 lets the bicluster effect vary with $m$, so that a bicluster can be effectively absent for some conditions or time points, allowing the total number of biclusters $K$ to effectively change for different $m$.

In the proposed plaid model, the bicluster effect is modeled as

$$\theta_{ijmk} = f_k(m) + \alpha_{ik}(m) + \beta_{jk}(m), \tag{2.6}$$

where $f_k(\cdot)$ is a functional mean effect of the bicluster over conditions. $\alpha_{ik}(\cdot)$ and $\beta_{jk}(\cdot)$ are row and column effects as before, and to avoid over-parameterization satisfy constraints $\sum_i r_{ik}\alpha_{ik}(m) = \sum_j c_{ik}\beta_{ik}(m) = 0$. Thus, the full model is similar to functional analysis of variance defined at the bicluster level (see Chapter 13 of *Ramsay and Silverman* (2005)).

The function $f(\cdot)$ can be modeled using parametric curves, or more general smoothing, shape constrained curves, and so on. Sufficient data is available to make these complex models practically relevant, since the same mean structure applies to the entire bicluster. I focus in this chapter on modeling mean functions, since they are critical in most contexts. Note that in principle, row and column specific effects can also be modeled with a similar approach. Next, the fitting procedure is discussed, in which estimates the row and column specific effects without smoothness or other modeling constraints.

The $K$th layer is found by minimizing

$$\min_{\{\theta_{ijmk}, r_{iK}, c_{jK}\}} \sum_{m=1}^{M}\sum_{i=1}^{n}\sum_{j=1}^{p}(\hat{Z}_{ijm} - \theta_{ijmk} r_{iK} c_{jK})^2, \tag{2.7}$$

where

$$\hat{Z}_{ijm} = X_{ijm} - \hat{\mu}_{m0} - \sum_{k=1}^{K-1} \hat{\theta}_{ijmk}\hat{r}_{ik}\hat{c}_{jk}. \tag{2.8}$$

The algorithm is shown in Algorithm II.1. The main idea is to detect biclusters that exhibit deviations in some conditions, and then estimate each bicluster's response for every condition. With most data sets, a background layer modeling the global mean that all samples and variables follow is estimated before searching for

biclusters. Bicluster effects are estimated sequentially. First, the mean effect is found by minimizing

$$\min_{f_K(m)} \sum_{m=1}^{M} \sum_{i=1}^{n} \sum_{j=1}^{p} (\hat{Z}_{ijm} - f_K(m)\hat{r}_{iK}\hat{c}_{jK})^2, \tag{2.9}$$

subject to smoothness or other constraints (discussed in Section 2.3.2). Then, if desired, row and column effects are computed.

Pruning strategies, backfitting, and other heuristics were proposed in *Lazzeroni and Owen* (2000) and *Turner et al.* (2005a) to obtain more interpretable and parsimonious structure. These strategies may also be employed with the proposed framework, especially if a large number of layers are found that are statistically significant, but not interpretable. In the data examples below, this does not appear to be an issue.

### 2.3.1 Implementation Issues

Next, I address issues pertaining to the implementation of Algorithm II.1. Specifically, (i) the permutation test to accept or reject a candidate bicluster, (ii) forming a candidate bicluster (line 1 of Algorithm II.1).

**Permutation test.** The standard permutation test is modified to accommodate structure along $m$. Matrix observations corresponding to different experimental conditions (or time points) should be permuted separately, so that the evolution over $m$ is maintained. After permuting the rows and columns of each given matrix, the standard approaches are followed to compare the candidate bicluster with what one would expect to find in noise.

I use the sum of squares $\sum_{i,j,m} r_i c_j \theta_{ij}^{(m)2}$ as proposed in the original paper (*Lazzeroni and Owen*, 2000) to measure the importance of a particular bicluster. The permutation test is shown in Algorithm II.2, where $\sigma^2$ is the sum of squares of the candidate layer, and $\sigma_{nr}^2$ is the sum of squares of noise layers $r$. The permutation test requires the candidate layer to have more information than the noise layers.

**Algorithm II.1** Plaid model for 3-way data

1: Apply one-way K-means clustering on the rows and columns of each $\hat{Z}_m$ to obtain $M$ biclusters, and combine them to form $\hat{r}_i, \hat{c}_j$ a candidate bicluster.
2: Estimate $f$ by minimizing equation 2.9.
3: **repeat**
4: $\quad \delta_{old} = \sum_{m=1}^{M} \sum_{i=1}^{n} \sum_{j=1}^{p} (\hat{Z}_{ijm} - \hat{f}_K(m) \hat{r}_{iK} \hat{c}_{jK})^2$
5: $\quad$ Update row and column memberships:

$$\hat{r}_{iK} = \begin{cases} 1, & \sum_{m,j} (\hat{Z}_{ijm} - \hat{f}(m) \hat{c}_{jK})^2 < \sum_{m,j} \hat{Z}_{ijm}^2 \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

$$\hat{c}_{jK} = \begin{cases} 1, & \sum_{m,i} (\hat{Z}_{ijm} - \hat{f}(m) \hat{r}_{iK})^2 < \sum_{m,i} \hat{Z}_{ijm}^2 \\ 0 & \text{otherwise} \end{cases} . \tag{2.11}$$

6: $\quad$ Estimate $f$ by minimizing equation 2.9.
7: $\quad \delta = \sum_{m=1}^{M} \sum_{i=1}^{n} \sum_{j=1}^{p} (\hat{Z}_{ijm} - \hat{f}_K(m) \hat{r}_{iK} \hat{c}_{jK})^2$
8: **until** $|\delta - \delta_{old}| / \delta_{old} <$ convergence threshold
9: Estimate row and column effects, if desired:

$$\alpha_{iK}(m) = (\sum_{j=1}^{p} \hat{c}_{jK})^{-1} \sum_{j=1}^{p} \hat{Z}_{ijm} \hat{r}_{iK} \hat{c}_{jK} - \hat{f}(m) \hat{r}_{iK} \hat{c}_{jK}, m = 1, ..., M \tag{2.12}$$

$$\beta_{jK}(m) = (\sum_{i=1}^{n} \hat{r}_{iK})^{-1} \sum_{i=1}^{n} \hat{Z}_{ijm} \hat{r}_{iK} \hat{c}_{jK} - \hat{f}(m) \hat{r}_{iK} \hat{c}_{jK}, m = 1, ..., M. \tag{2.13}$$

10: Keep or reject $\{\hat{r}_{iK}, \hat{c}_{jK}, \hat{\theta}_{ijmk}\}$ according to a permutation test.

**Algorithm II.2** Permutation test to assess the significance of a candidate layer $\{\hat{r}_{iK}, \hat{c}_{jK}, \hat{\theta}_{ijmk}\}$

1: **for** r=1,...,R **do**
2: $\quad$ Compute the residual data matrix $\hat{Z}_m$, including the candidate layer.
3: $\quad$ Permute the rows and columns of each residual data matrix.
4: $\quad$ Estimate a noise layer $\hat{\sigma}_{n_r}$ from the permuted $\{\hat{Z}_m\}$.
5: **end for**
6: **if** $\hat{\sigma}^2 > \max\{\hat{\sigma}_{n_1}^2, ..., \hat{\sigma}_{n_R}^2\}$ **then**
7: $\quad$ Accept candidate layer.
8: **else**
9: $\quad$ Reject candidate layer.
10: **end if**

It is argued in *Lazzeroni and Owen* (2000) that, since after permuting rows and columns the data values are independent of row and column labels, the approximate probability of accepting $k$ or more false biclusters is $(R+1)^{-k}$, where $R$ is the total number of noise biclusters. The authors suggest four or fewer noise biclusters for each permutation test, though this is highly dependent on the size of the data and available computing power (costs are proportional to the number of noise biclusters).

**Extracting candidate biclusters.** There are many possible ways of combining the $M$ initial candidates to form a final candidate bicluster. I present a numerical comparison of three different techniques shows that they all lead to similar clustering results. The three methods are denoted as 'Average Data', 'Majority Vote', and 'Similarity'.

Average Data follows the simplest strategy of first averaging data matrices over $m$, then applying to the result one-way K-means clustering separately on the rows and columns to form the candidate bicluster. Majority Vote applies one-way K-means clustering separately to each of the $m$ data matrices, then takes the rows (columns) that are clustered most often for the candidate bicluster. Similarity refers to taking the intersection of the two most overlapping biclusters after applying one-way K-means clustering separately to each of the $m$ data matrices. I first define a similarity measure between the biclusters identified at $m$ and $m'$ as

$$S(m, m') = \frac{\sum_{i,j} r_{im} c_{jm} r_{im'} c_{jm'}}{\min\{\sum_{i,j} r_{im} c_{jm}, \sum_{i,j} r_{im'} c_{jm'}\}}. \tag{2.14}$$

This measure computes the amount of overlap relative to the size of the smaller bicluster. Finally, to choose the elements of the candidate bicluster, take the intersection of the two most similar biclusters.

The Majority Vote and Similarity heuristics are more complex, and indeed do a better job at forming candidate biclusters that exhibit deviations in only some con-

ditions. In other words, the simpler strategy of Average Data identifies less accurate candidates since critical information is lost when the three-dimensional data array is collapsed to two dimensions. However, due to the optimization that follows (lines 3 - 8 in Algorithm 1), the different strategies lead to similar clustering results. Table 2.1 shows nearly identical detection and false positive rates for the three different initialization techniques under two different biclustering simulations (details for the generating process are provided in Section 2.4.1). Thus, the investigations indicate that one can follow the simpler and computationally efficient strategy of first averaging data matrices over $m$, without compromising the overall accuracy of the overall biclustering procedure.

### 2.3.2 Modeling the mean effect

I present two simple modeling approaches that facilitate exploratory analysis and data visualization. The first approach estimates bicluster $K$'s mean effect with

$$\hat{f}_K(m) = (\sum_{i=1}^{n}\sum_{j=1}^{p} \hat{Z}_{ijm}\hat{r}_{iK}\hat{c}_{jK}) / \sum_{i=1}^{n}\sum_{j=1}^{p} \hat{r}_{iK}\hat{c}_{jK}, \tag{2.15}$$

so that each bicluster mean is modeled by its cross-sectional average at $m$. This simple approach is computationally inexpensive, and hence facilitates a quick decomposition of the data.

The second approach estimates bicluster $K$'s mean effect with kernel smoothing to enhance visual interpretation and provide insights into bicluster behavior at unobserved points. For a potentially unobserved point $m'$, the mean effect is estimated with

$$\hat{f}_K(m') = (\sum_{m} W(m) \sum_{i=1}^{n}\sum_{j=1}^{p} \hat{Z}_{ijm}\hat{r}_{iK}\hat{c}_{jK}) / \sum_{m} W(m), \tag{2.16}$$

where $W(m) = Q((m - m')/h_n)$, $Q(\cdot)$ the kernel function, and $h_n$ the bandwidth. In

Panel A: Change Point

| $\sigma$ | Method | % Bicluster Detected | % False Positive | Number Detected |
|---|---|---|---|---|
| 0.3 | Average Data | 1.00 (0.00) | 0.09 (0.03) | 1.36 (0.06) |
| | Majority Vote | 1.00 (0.00) | 0.10 (0.03) | 1.36 (0.07) |
| | Similarity | 1.00 (0.00) | 0.07 (0.03) | 1.20 (0.06) |
| 0.5 | Average Data | 1.00 (0.00) | 0.10 (0.03) | 1.34 (0.07) |
| | Majority Vote | 1.00 (0.00) | 0.10 (0.03) | 1.32 (0.06) |
| | Similarity | 1.00 (0.00) | 0.09 (0.03) | 1.31 (0.07) |
| 0.7 | Average Data | 0.78 (0.03) | 0.35 (0.04) | 1.05 (0.08) |
| | Majority Vote | 0.76 (0.04) | 0.39 (0.03) | 0.98 (0.09) |
| | Similarity | 0.73 (0.04) | 0.35 (0.03) | 1.03 (0.09) |

Panel B: Isotonic Sine

| $\sigma$ | Method | % Bicluster Detected | % False Positive | Number Detected |
|---|---|---|---|---|
| 0.3 | Average Data | 1.00 (0.00) | 0.06 (0.02) | 1.12 (0.04) |
| | Majority Vote | 1.00 (0.00) | 0.09 (0.02) | 1.12 (0.04) |
| | Similarity | 1.00 (0.00) | 0.08 (0.02) | 1.09 (0.03) |
| 0.5 | Average Data | 0.99 (0.00) | 0.08 (0.02) | 1.14 (0.05) |
| | Majority Vote | 0.97 (0.00) | 0.08 (0.02) | 1.01 (0.04) |
| | Similarity | 0.98 (0.00) | 0.06 (0.02) | 1.08 (0.03) |
| 0.7 | Average Data | 1.00 (0.00) | 0.10 (0.03) | 1.12 (0.04) |
| | Majority Vote | 1.00 (0.00) | 0.10 (0.03) | 1.14 (0.04) |
| | Similarity | 1.00 (0.00) | 0.13 (0.03) | 1.15 (0.04) |

Table 2.1: Average (standard error) recovery results for the proposed plaid procedure implemented with different methods of extracting candidate biclusters. % Bicluster Detected measures the proportion of the single underlying that was detected. % False Positive measures the proportion of all biclustered elements that were false positives. Number Detected reports the number of biclusters detected.

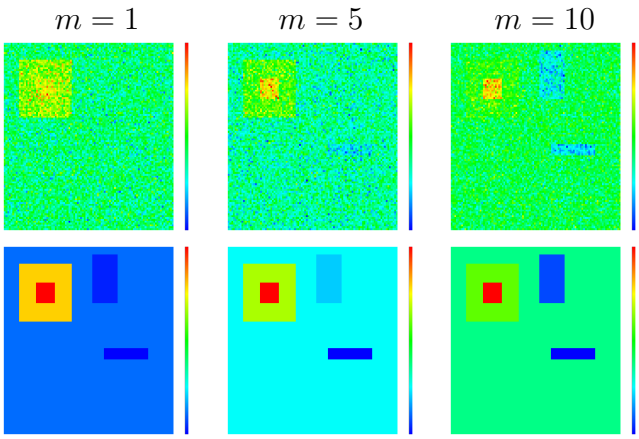



Figure 2.2: The top row shows examples of raw data. The bottom row shows examples of the filtered data.

my implementation, I employ the triangular kernel

$$Q(x) = (1 - |x|)\mathbb{I}\{x \in (-1, 1)\}. \tag{2.17}$$

The bandwidth $h_n$ is chosen by exploring estimates over a range of bandwidths and selecting the one that emphasizes the structure change most.

A practitioner could follow an iterative strategy that first starts with the cross sectional averages for the bicluster means for an initial exploration of the data. The results may then be enhanced by re-estimating with smoothing over a range of bandwidths. Alternatively, if strong evidence for a particular pattern is observed, a parametric model may then be postulated for the bicluster mean effects.

### 2.3.3 An Illustrative Example

The proposed methodology is illustrated with simulated data. In particular, I set $X_m \in \mathbb{R}^{100\times100}$, where $X_{ijm} \sim N(10, 1)$. In other words, the background layer has constant mean $\mu_{m0} = 10$, and there are four biclusters with structure summarized in Table 2.2. There are 10 observed slices and the third dimension is sampled uniformly between 1 and 10 ($m = 1, 2, .., 10$). The biclusters are fixed over $m$.

| Bicluster | $\mu_{mk}$ | Size | Rows | Columns |
|---|---|---|---|---|
| 1 | $2 + cos(m)$ | $10 \times 10$ | 10-20 | 10-20 |
| 2 | $-2\mathbb{I}\{m > 5\}$ | $30 \times 30$ | 10-40 | 10-40 |
| 3 | $\sqrt{m}$ | $5 \times 15$ | 55-60 | 60-85 |
| 4 | $-m/4$ | $25 \times 15$ | 5-30 | 53-68 |

Table 2.2: Summary of bicluster structure for the illustrative example.

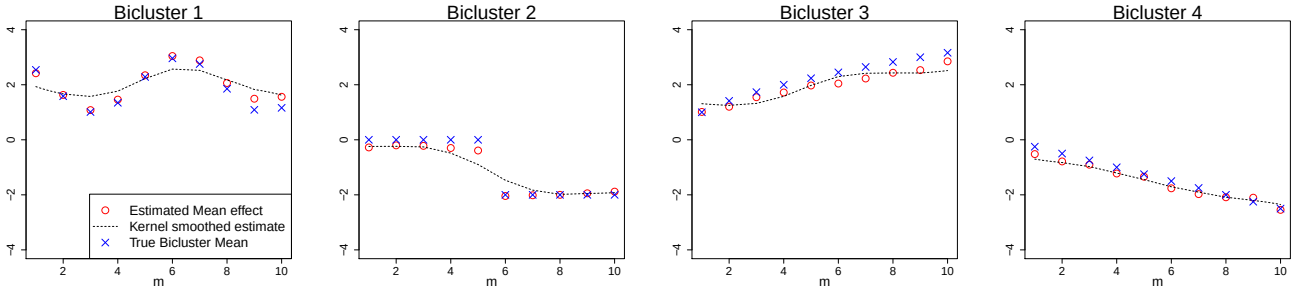



Figure 2.3: Estimated bicluster effects for the illustrative example.

The background layer is estimated using a simple cross-sectional mean that varies along $m$. The bicluster mean effects are modeled using the two forms discussed above: (i) cross-sectional averages, and (ii) kernel smoothing.

Figure 2.2 shows examples of the raw and estimated data, and Figure 2.3 shows the true and estimated biclusters effects. The proposed procedure is able to identify the correct matrix groups, and then estimate the different expression patterns accurately. The kernel smoothed version appears more satisfactory for visualization and interpolation. Though, if the mean structure is over simplified (e.g., smooth too heavily) features are masked in the reconstruction, as shown in Figure 2.4. In this example, there is a range of bandwidths that perform well, given the true functional forms of the biclusters. Using a simple cross-sectional mean recovers the true values at the observed points accurately.

Figure 2.5 shows competing approaches that directly apply the plaid model or independent K-means clustering to each data matrix tend to miss the true functions and biclusters governing the data generation mechanism. On the other hand, the proposed approach utilizes information from neighboring data matrices to obtain

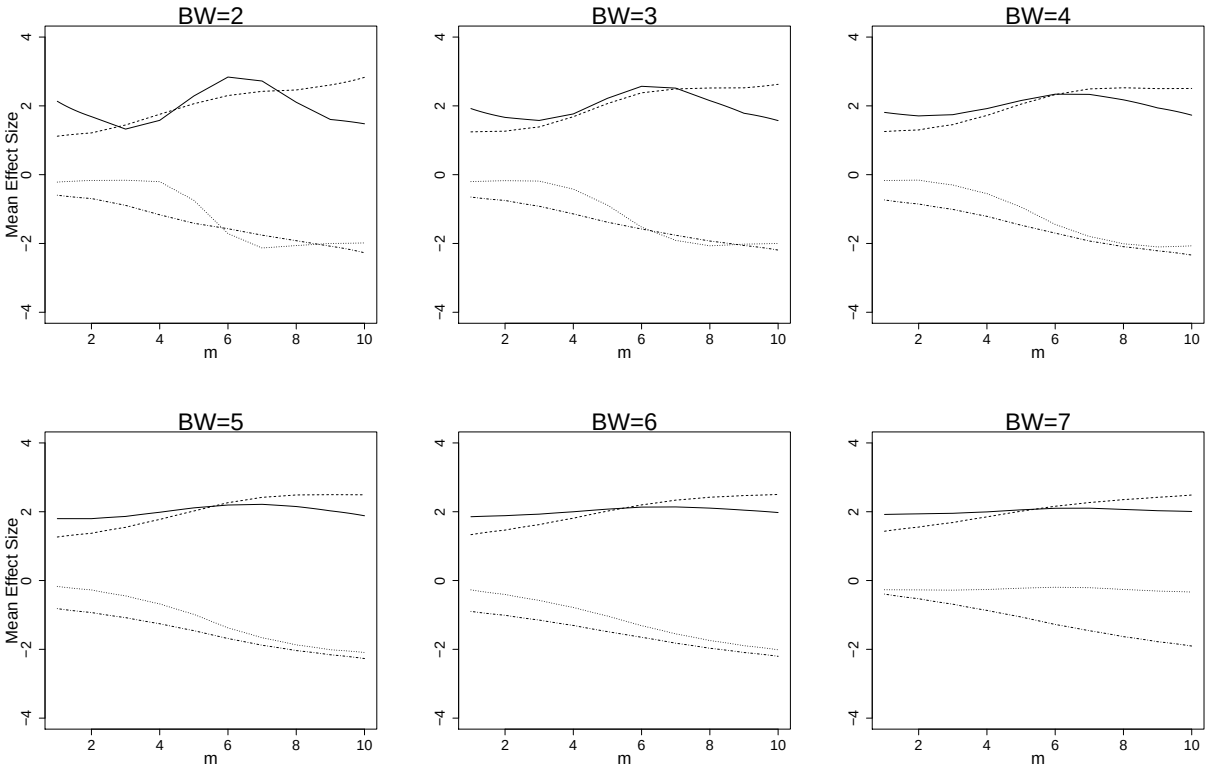



Figure 2.4: Bicluster mean effects with different bandwidths.

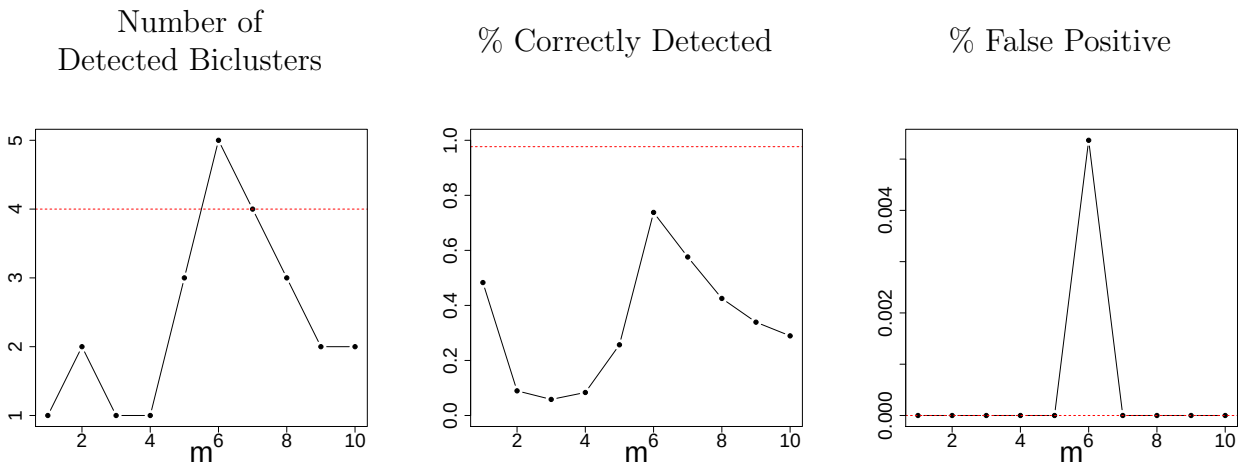



Figure 2.5: Comparison with a direct plaid approach. The dashed line shows results from the proposed method; the solid line shows results using the plaid model applied to each data matrix separately.

accurate and interpretable results.

Figure 2.6 plots reconstruction accuracy over each data matrix. A null model that includes only a global mean and absence of biclusters explains a large amount of variance. Though more complex models with estimated biclusters are both accurate and interpretable, it appears the bicluster contribution to explained variance is limited. This pattern is expected when the global mean is large and biclusters are relatively small in size. This highlights the fact that biclustering approaches are advantageous for finding 'needles in a haystack', and closely related to anomaly detection. Biclustering in these contexts uncovers unusual structure, and facilitates exploratory analysis and visualization. For instance, the null model has a clear pattern reflecting the missing structure. A cross-sectional mean or light smoothing uncovers the biclustering structure and maintains a stable reconstruction error.

## 2.4 Comparing Two Models for Three-way Data

As mentioned in Section 2.2, *Turner et al.* (2005b) also provide an extension of the plaid model for replicated longitudinal data. Their postulated model adds a main

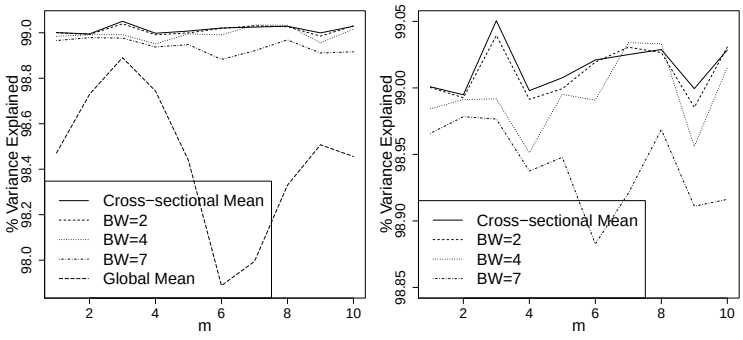



Figure 2.6: The left panel shows percent of variance explained ($1 - ||X_m - \hat{X}_m||_F^2/||X_m||_F^2$) for different models. Global mean refers to setting the estimates to be the cross sectional means without any biclustering. The right panel provides a zoomed-in version.

time effects to each layer to account for changes in expression levels over time

$$X_{ij}^{(t)} = \mu_0 + \alpha_{i0} + \beta_{i0} + \tau_0(t) + \sum_{k=1}^{K}(\mu_k + \alpha_{ik} + \beta_{jk} + \tau_k(t))r_{ik}c_{jk} + \epsilon_{ij}. \tag{2.18}$$

The focus of this model is to detect changes in expression levels over time with $\tau_k(t)$.

In comparison, the proposed model is

$$X_{ijm} = \mu_{m0} + \sum_{k=1}^{K}(f_k(m) + \alpha_{ik}(m) + \beta_{jk}(m))r_{ik}c_{jk} + \epsilon_{ijm}, \tag{2.19}$$

where $f_k(\cdot)$ is a functional mean effect of the bicluster over conditions, and $\alpha_{ik}(\cdot)$ and $\beta_{jk}(\cdot)$ are row and column effects that can also be modeled as functions over conditions.

If $f_k(m) = \mu_k + \tau_k(m)$ and fix the row and column effects over $m$, then the models are essentially equivalent. However, I will illustrate below that there are situations in which a simple main time effect is not flexible enough for detection and visualization of the expression level changes. For instance, (i) if the underlying mean effect has strong nonlinearities or (ii) if the row and column effects change over conditions, then the proposed model performs significantly better.

### 2.4.1 Simulation Study

I employ the $R$ code provided in the Supplementary material of *Turner et al.* (2005b) to test the model in Equation 2.18. Results are reported below for two different parameter settings. The first, labeled as Turner-1 below, sets the *row release* and *column release* parameters to equal 0.5, the recommended setting in *Turner et al.* (2005b). However at times, this results in no detected biclusters. Thus, even though it is not recommended in *Turner et al.* (2005b), I also present results from setting the *row release* and *column release* parameters to equal 0.1. These results are labeled as Turner-2.

**Nonlinear Mean Effects.** Let $X_m \in \mathbb{R}^{100\times 100}$, where $X_{ijm} \sim N(0, \sigma)$. Rows 20 through 30 and columns 20 through 30 form a single bicluster, where the mean effect follows the functions described below. The design space $m$ is the $[0, 1]$ interval, with 20 uniformly spaced points.

In one simulation setting, the bicluster mean exhibits a change-point according to $f_1(m) = \mathbb{I}\{m > 0.75\}$. In the second simulation setting, the bicluster mean follows an isotonic sine function $f_1(m) = (1/40)\sin(6\pi m) + 1/4 + (1/2)m + (1/4)m^2$, shown in the left panel of Figure 2.7. The critical features of the isotonic sine function are that it is non-decreasing in $m$ and has local oscillations.

Results, averaged over 1000 replications, in Table 2.3 show that for both mean functions and at most noise levels, the proposed procedure has better success at identifying the bicluster. The recommended model of *Turner et al.* (2005b) struggles to detect the single underlying bicluster in the change point case, and for $\sigma > 0.3$ in the isotonic sine case. Lower row and column release parameters improve results, though the final setting may be unrealistic and difficult to tune with real data. The more general and proposed model improves the ability to detect the underlying structure, though with more falsely biclustered matrix elements. The false positive rate could be reduced by utilizing a permutation test with additional noise layers.

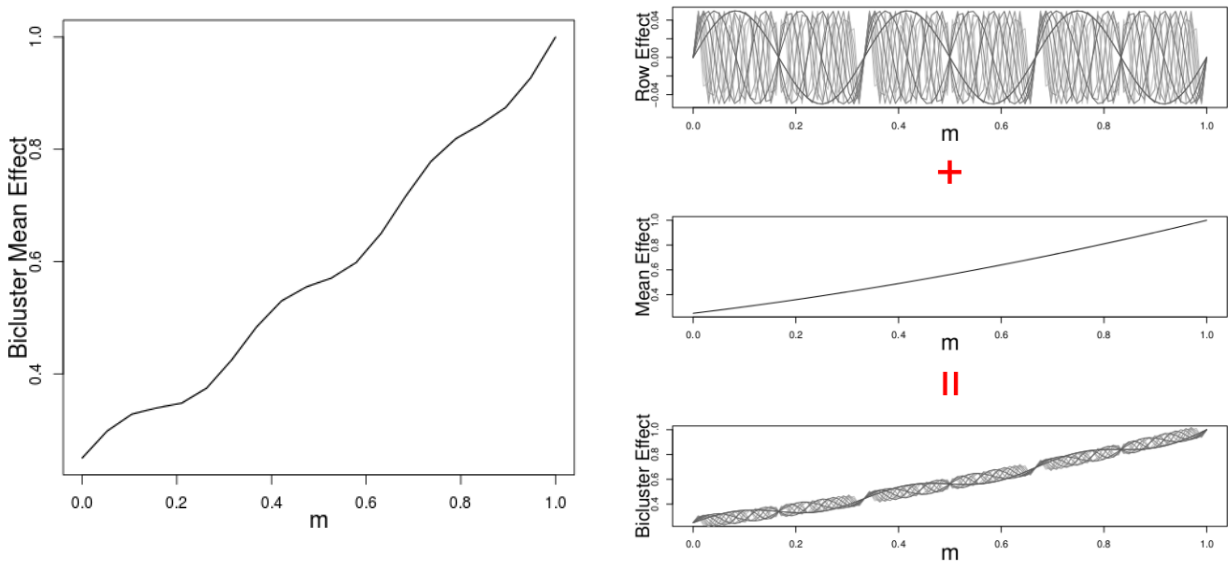



Figure 2.7: The left panel shows for the first simulation setting the true bicluster mean effect, which follows the isotonic sine function. The right panel shows for the second simulation setting the curves that comprise the bicluster effect, where the oscillations are controlled by row effects.

The local smoothing that the proposed procedure accommodates improves the estimate, especially at higher noise levels, as shown in Figure 2.8 for the isotonic sine case. The isotonic property and wiggly nature of the function are preserved in the kernel smoothed fits, but not the simpler cross-sectional means of the recovered bicluster. Estimates of $\tau_{tk}$ are not shown in the figure, because the $R$ code provided in the Supplementary material of *Turner et al.* (2005b) does not output it. However, $\hat{\tau}_{tk}$ would suffer in higher noise settings, just as the cross-sectional mean estimates do in Figure 2.8.

**m-varying Row/Columns Effects.** Let $X_m \in \mathbb{R}^{100\times 100}$, where $X_{ijm} \sim N(0, \sigma)$. Rows 20 through 30 and columns 20 through 30 form a single bicluster. The mean

Panel A: Change Point

| $\sigma$ | Method | % Bicluster Detected | % False Positive | Number Detected |
|---|---|---|---|---|
| 0.3 | Turner-1 | 0.00 (0.00) | - (-) | 0.00 (0.00) |
| | Turner-2 | 0.62 (0.02) | 0.00 (0.00) | 1.27 (0.05) |
| | Proposed Plaid | 1.00 (0.00) | 0.06 (0.02) | 1.09 (0.03) |
| 0.5 | Turner-1 | 0.00 (0.00) | - (-) | 0.00 (0.00) |
| | Turner-2 | 0.06 (0.01) | 0.04 (0.01) | 0.77 (0.07) |
| | Proposed Plaid | 0.99 (0.00) | 0.08 (0.03) | 1.11 (0.04) |
| 0.7 | Turner-1 | 0.00 (0.00) | - (-) | 0.00 (0.00) |
| | Turner-2 | 0.00 (0.00) | - (-) | 0.00 (0.00) |
| | Proposed Plaid | 0.88 (0.03) | 0.35 (0.03) | 1.07 (0.07) |

Panel B: Isotonic Sine

| $\sigma$ | Method | % Bicluster Detected | % False Positive | Number Detected |
|---|---|---|---|---|
| 0.3 | Turner-1 | 0.99 (0.00) | 0.00 (0.00) | 1.00 (0.00) |
| | Turner-2 | 1.00 (0.00) | 0.00 (0.00) | 1.01 (0.01) |
| | Proposed Plaid | 1.00 (0.00) | 0.09 (0.03) | 1.13 (0.04) |
| 0.5 | Turner-1 | 0.01 (0.02) | 0.00 (0.00) | 0.15 (0.04) |
| | Turner-2 | 1.00 (0.00) | 0.00 (0.00) | 1.00 (0.00) |
| | Proposed Plaid | 1.00 (0.00) | 0.10 (0.03) | 1.10 (0.03) |
| 0.7 | Turner-1 | 0.00 (0.00) | - (-) | 0.00 (0.00) |
| | Turner-2 | 0.87 (0.03) | 0.00 (0.00) | 0.91 (0.04) |
| | Proposed Plaid | 1.00 (0.00) | 0.11 (0.03) | 1.14 (0.04) |

Table 2.3: Average (standard errors) recovery results under the two different m-varying mean effect scenarios. % Bicluster Detected measures the proportion of the single underlying that was detected. % False Positive measures the proportion of all biclustered elements that were false positives. Number Detected reports the number of biclusters detected.

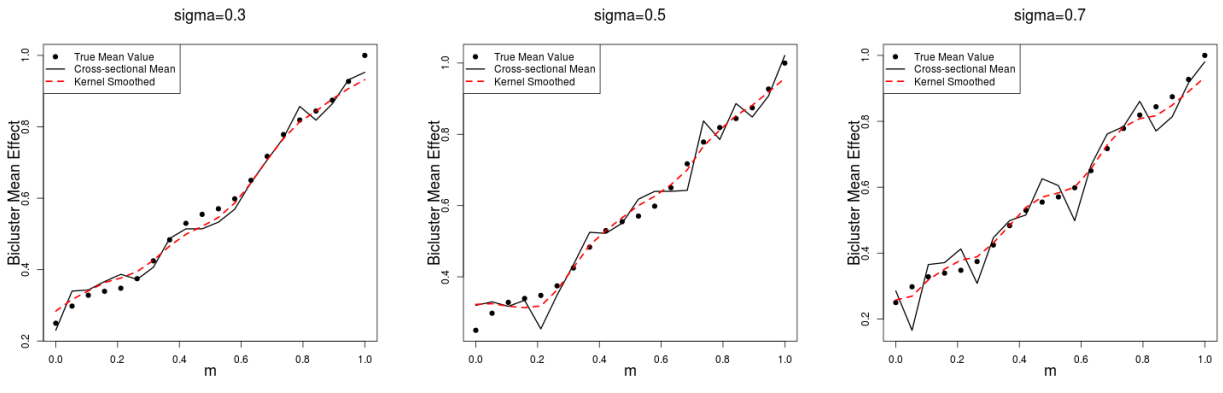



Figure 2.8: The estimated mean function from the proposed procedure under different noise levels for the isotonic sine case.

and row effects follow

$$f_1(m) = 1/4 + m/2 + (1/4) * m^2 \tag{2.20}$$

$$\alpha_{i1}(m) = (1/40)\sin(6\pi m i). \tag{2.21}$$

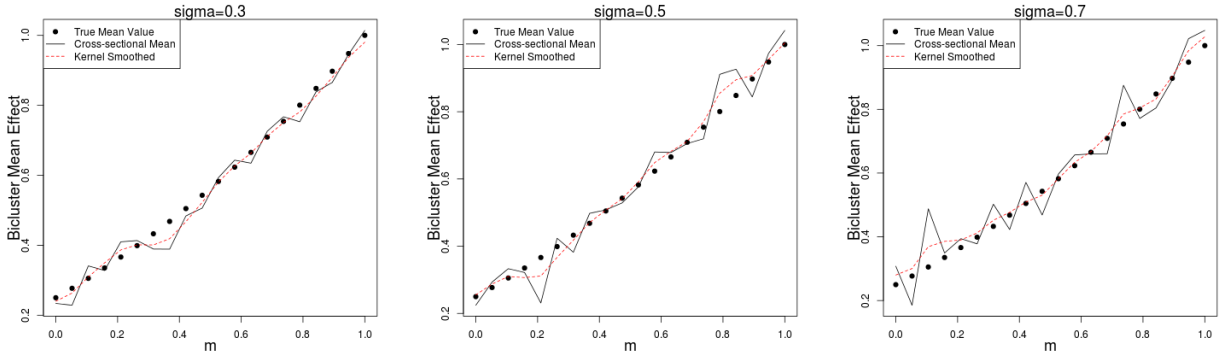
There are no column effects. The bicluster effects are similar to an isotonic sine function, where the amount of oscillation is controlled by the row. The different bicluster effects,including the different row effects, are shown in the right panel of Figure 2.7. The design space $m$ is again the $[0, 1]$ interval, with 20 uniformly spaced points.

Results in Table 2.4 show that the recommended model of *Turner et al.* (2005b) struggles to detect the single underlying bicluster in higher noise settings. Again tuning the row and column release parameters improves results, though when $\sigma = 0.7$ Turner-2 also struggles to detect the bicluster. Even if the structure is correctly estimated, the model in *Turner et al.* (2005b) will provide estimates that are more sensitive to noise and misspecified. As in the previous example, Figure 2.9 shows the proposed model is able to take advantage of the well-known benefits of local smoothing, resulting in estimates that are interpretable and accurate.

| $\sigma$ | Method | % Bicluster Detected | % False Positive | Number Detected |
|---|---|---|---|---|
| 0.3 | Turner-1 | 0.99 (0.00) | 0.00 (0.00) | 1.00 (0.00) |
| | Turner-2 | 1.00 (0.00) | 0.00 (0.00) | 1.00 (0.00) |
| | Proposed Plaid | 1.00 (0.00) | 0.07 (0.04) | 1.20 (0.04) |
| 0.5 | Turner-1 | 0.02 (0.00) | 0.00 (0.00) | 0.29 (0.04) |
| | Turner-2 | 1.00 (0.00) | 0.00 (0.00) | 1.00 (0.00) |
| | Proposed Plaid | 1.00 (0.00) | 0.09 (0.03) | 1.17 (0.04) |
| 0.7 | Turner-1 | 0.00 (0.00) | - (-) | 0.00 (0.00) |
| | Turner-2 | 0.66 (0.04) | 0.00 (0.00) | 0.78 (0.04) |
| | Proposed Plaid | 1.00 (0.00) | 0.11 (0.03) | 1.17 (0.04) |

Table 2.4: Average (standard error) recovery results under m-varying mean and row effect scenarios. % Bicluster Detected measures the proportion of the single underlying that was detected. % False Positive measures the proportion of all biclustered elements that were false positives. Number Detected reports the number of biclusters detected.

Panel A: Mean Effects




Panel B: Row Effects

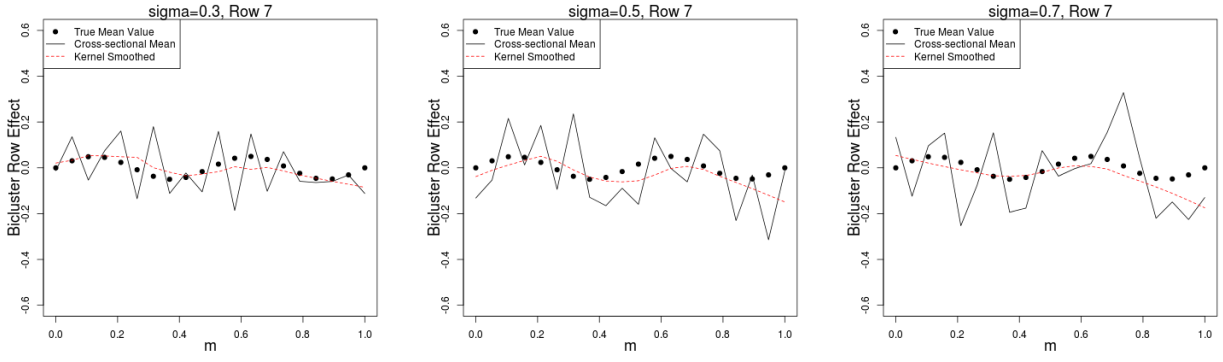
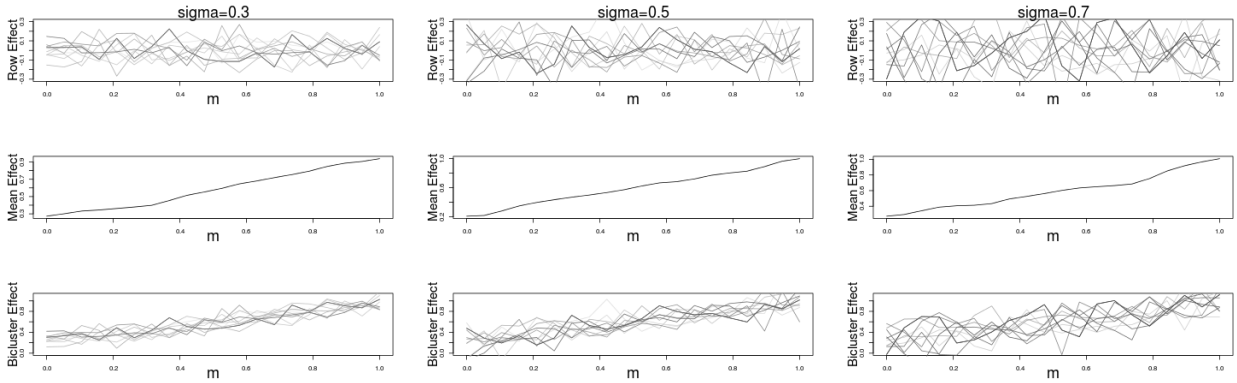



Figure 2.9: The top panel shows the estimated mean function, the second panel shows an estimated row effect for a single row.

Panel A: Estimates with smooth mean functions




Panel B: Estimates with smooth mean and row functions

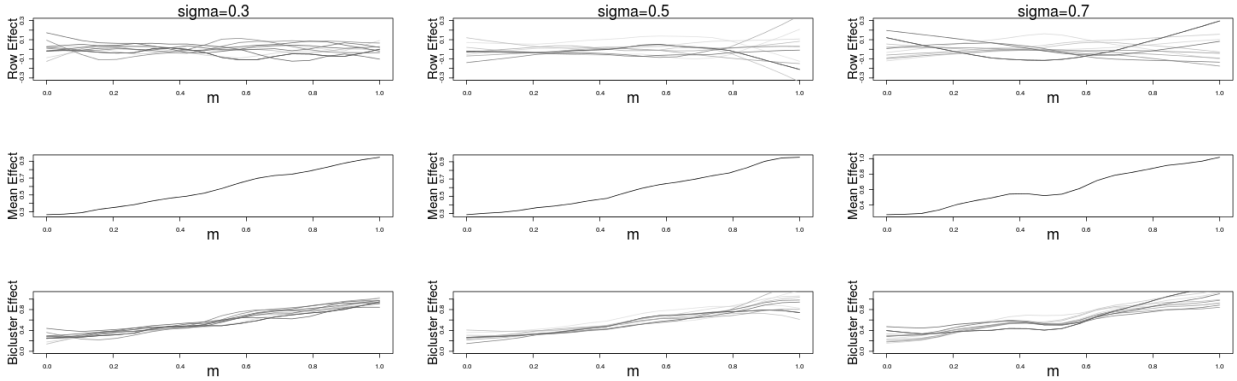



Figure 2.10: The Panel A shows all mean and row effects estimates without any smoothing, and Panel B shows corresponding smoothed estimates.

To illustrate the general framework, the bottom panel of Figure 2.9 shows smooth row effect estimates, which were estimated again with kernel smoothing

$$\hat{\alpha}_{iK}(m) = (\sum_{m} W(m)(\sum_{j=1}^{p} \hat{c}_{jK})^{-1} \sum_{j=1}^{p} (\hat{Z}_{ijm} - f(m)) r_{iK} c_{jK}) / \sum_{m} W(m), \qquad (2.22)$$

instead of Equation 2.12 in Algorithm II.1. Figure 2.10 contains estimates for all row effects in the bicluster, and shows that smoothing both mean and row effects improve interpretability, especially in high noise settings.

## 2.5 Applications

### 2.5.1 Interpolating gene expression biclusters

Time course gene expression data may be modeled as a discrete sampling from continuous processes over time. The aim is to identify groups of co-regulated genes with respect to a subset of samples, and estimate the underlying, evolving processes.

**T-cell data.** I illustrate the proposed model by smoothing genetic regulatory activations using the time-course gene expression data of *Rangel et al.* (2004) on T-cell activation. The activation of T-cells are a central event in the generation of an immune response.

The data is available in the R package Genenet (*Schafer et al.* (2006)), and consists of 44 gene expression samples of 58 genes, measured over 10 time points. In this study, the gene activity levels are measured at $t = 0, 2, 4, 6, 8, 18, 24, 32, 48, 72$ hours after stimulation.

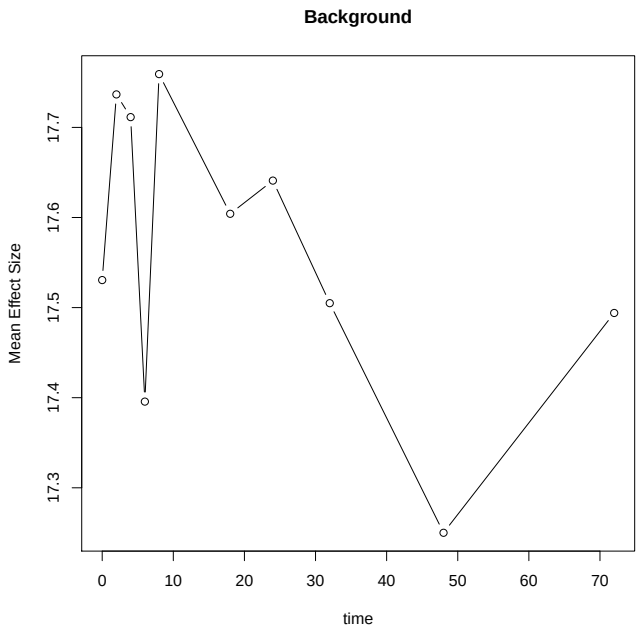



Figure 2.11: Background layer for T-cell data.

The background layer, shown in Figure 2.11, was estimated using a cross-sectional

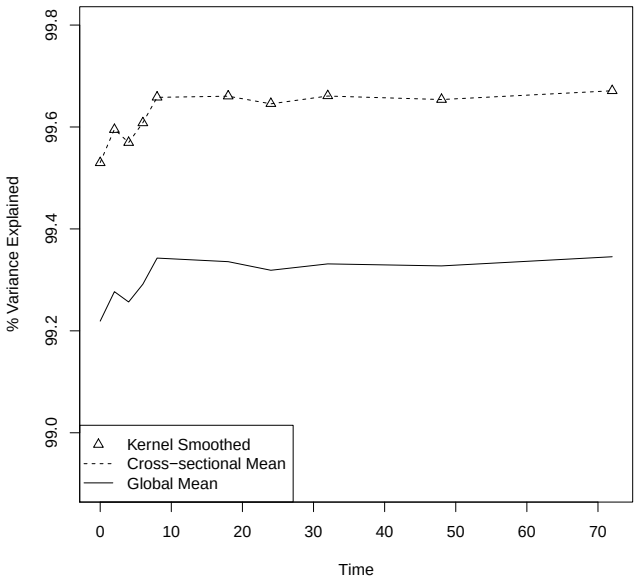



Figure 2.12:
Percent of variance explained $(1 - ||X_m - \hat{X}_m||_F^2/||X_m||_F^2)$ for different models for the T-cell data.

mean that varies with time. The global mean is fairly stable, and does not provide evidence for more complex models, e.g., smoothing. Figure 2.12 shows, much like the illustrative example, that the global mean captures a large amount of variance and biclusters have limited contribution. The additional biclustering highlights small groups of samples and genes that exhibit unusual behavior.

I model each bicluster with row (sample) specific effects $(f_k(m) + \alpha_{ik}(m))$. Biclusters were chosen until the stopping criterion was met using three noise layers. Forcing the algorithm to recover additional biclusters found ones with minuscule mean effects that were uninterpretable.

Figure 2.13 contrasts the kernel smoothed with cross-sectional mean effects. The kernel smoothed version indicates an inflection point for bicluster 3 around $t = 48$, while both types of mean functions shows that bicluster 1 reaches a local minimum and biclusters 2 peaks around that time. In other words, 48 hours after stimulation may be a point of interest for further biological analysis.

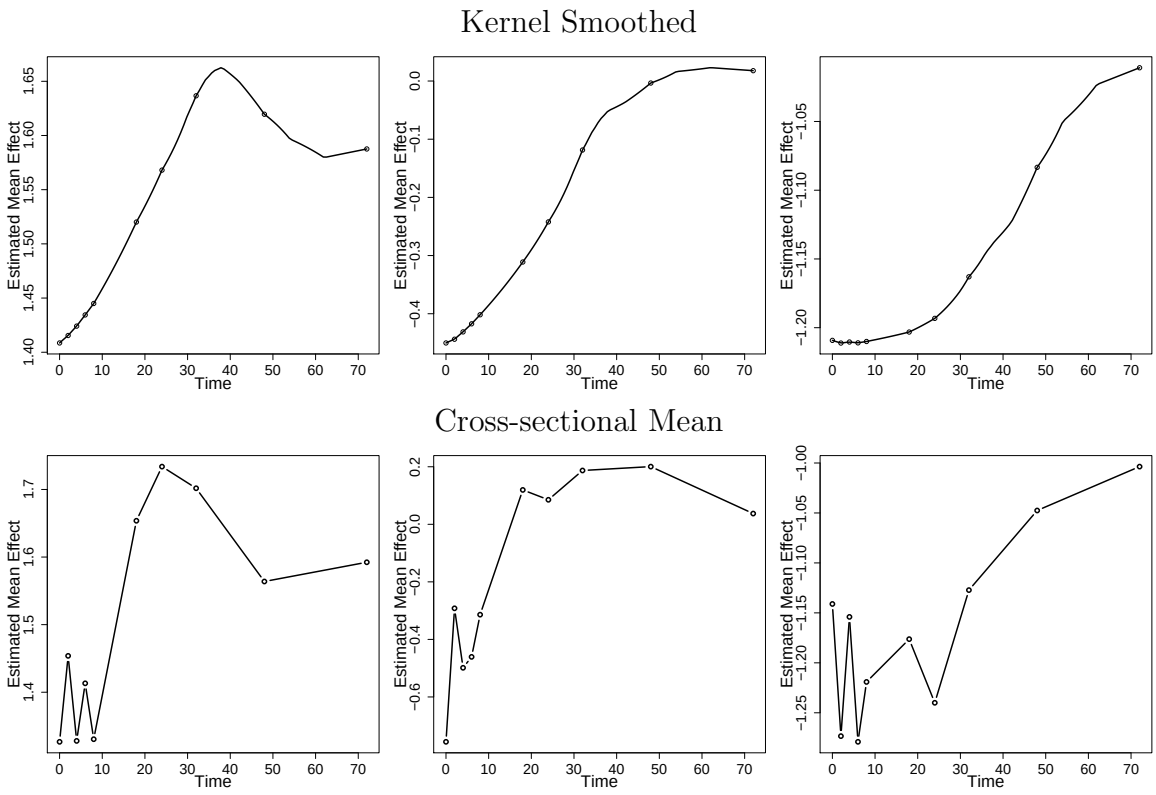



Figure 2.13: Estimated bicluster mean effects on T-cell activation.

Table 2.5 shows that 30 genes were identified in 3 biclusters, with a number of common genes in the first two biclusters (FYB,ZNFN1A1,CTNNB1,SKIIP,IL2RG). These key genes are consistent with the results of the state space model in *Rangel et al.* (2004), and are likely high in the hierarchy of events downstream of cell activation. For example, many of the genes common among the first two biclusters are located in close proximity to FYB, which is known to be an important molecule in T-cell systems.

### 2.5.2 Time-varying Community Detection in Weighted and Directed Graph Sequences

There has been tremendous interest in community discovery within networked systems, including biological, social and technological networks, where groups of nodes (vertices) feature relatively dense within group connectivity and sparser be-

| Bicluster | # Samples | # Genes | Effect | Downstream Genes |
|---|---|---|---|---|
| 1 | 10 | 9 | + | LCK,RPS6KA1,EGR1,SOD1 |
| 2 | 10 | 12 | − | CD69,MAP2K4,ITGAM,ID3,GATA3, CYP19,JUNB |
| 3 | 10 | 14 | − | CCNG1,CLU,IL4R,SCYA2,PDE4B,PIG3, IRAK1,MYD88,RBL2,C3X1,IFNAR1, CIR,MAP3K8,IL3RA |

Table 2.5: Bicluster summary statistics for T-cell data. 'Downstream genes' identifies genes that are unique to each bicluster.

tween group connections (*Newman*, 2010; *Newman et al.*, 2006). Due to advances in data collection technologies, it is becoming increasingly common to study patterns of behavior within a time series of networks (*Li et al.*, 2011; *Gong et al.*, 2011). The analysis of such data is challenging, since time dependent changes may simultaneously affect network topology and node/edge features. Moreover, many existing analysis tools are arguably only compatible with networks of binary relations. In this section, I apply the proposed model to extract communities (biclusters), whose similarity moves beyond dense clumps of connected nodes by utilizing information over time within a sequence of weighted and directed networks.

A network can be equivalently represented using an adjacency matrix, which is a square matrix of size $n$, where $n$ is the number of nodes and the $i, j$ element is zero if there is no edge between node $i$ and node $j$. In this setting, $\{X^{(t)}\}$ are now square matrices observed over time. Instead of the sample-variable interpretation to the rows-columns, each row and column now corresponds to a node on the graph. Thus, a bicluster on the adjacency matrix corresponds to a densely connected subgraph. The proposed plaid model extracts potentially overlapping subgraphs and estimates their strength over time. Next, I apply the model on global trade flow data to identify groups of countries with interesting growth patterns.

**World trade data.** The data consists of annual, total bilateral trade flows between all two hundred countries in the United Nations from 1975-2000 (*Feenstra*

*et al.*, 2004) and is available at http://cid.econ.ucdavis.edu/. Thus, I observe a dynamic, weighted graph at 26 time points, where each directional edge denotes the total value of exports from one country to another. Since trade flows can differ in size by orders of magnitude, I work with trade values that are expressed in log nominal dollars. The $i, j$ element of each data matrix corresponds to the amount country $i$ exported to country $j$. Then, the rows of each bicluster identify a set of exporting countries, and the columns identify corresponding importers.

Often economic output and trade are modeled with growth curves intended to capture the idea that continued innovation results in constant economic expansion (*Rodriguez and Rodrik*, 2001; *Bernanke and Rogoff*, 2001). Hence, in the context of bilateral trade data, I estimate for the global mean a growth curve that characterizes world trade over time. To avoid the well-known issue of choosing a particular form out of many possible growth curve models, the global mean is modeled with isotonic regression (IR), which can be used to overcome errors that occur from parametric growth models. For times $\{t_i\}_{i=1}^T$ and corresponding bilateral trades $\{Y_i\}_{i=1}^T$, the IR estimate $f_I(\cdot)$ is given by

$$f_I(x) = \begin{cases} f_1^\star & \text{if } x \in [a, t_1], \\ f_i^\star & \text{if } x \in [t_i, t_{i+1}), i = 1, 2, \cdots, T-1, \\ f_T^\star & \text{if } x \in [t_T, b]\,, \end{cases} \tag{2.23}$$

where

$$\{f_i^\star\}_{i=1}^T = \underset{f_1 \le f_2 \le \cdots \le f_T}{\arg\min} \sum_{i=1}^T (Y_i - f_i)^2.$$

This minimizer exists uniquely and has a geometric characterization as the slope of the greatest convex minorant. The pool adjacent violators algorithm is used for computation of the IR estimate (see, for example, *de Leeuw et al.* (2009); *Robertson et al.* (1988)). Since the global mean covers all trades, there are $n^2 - n$ trade flows

(responses) at each time $t$. The IR estimate results in an increasing function to the set of bilateral trades over times $t$.

Smoothing the isotonic regression estimate (SIR) facilitates visual interpretability and has been shown to achieve better asymptotic rates (see *Mukerjee* (1988) and references therein). Specifically, the SIR estimate for observed $\{t_i, Y_i\}_{i=1}^T$ is given by

$$f_{Is}(x) = \sum_{i=1}^{T} W(t_i) f_I(t_i) / \sum_{i=1}^{T} W(t_i) \tag{2.24}$$

where $W(t_i) = Q((t_i - x)/h_n)$, $Q(\cdot)$ the kernel function, and $h_n$ the bandwidth. SIR is an appropriate alternative to parametric growth curves, since for log-concave kernels the estimate $m_{Is}$ is non-decreasing (see Remark 2.1 in *Mukerjee* (1988)).

As with the bicluster mean effects, I employ the triangular kernel and choose the bandwidth by visual inspection. Row and column specific effects are also included for each bicluster to account for country-specific effects resulting from political events, climate, military interventions, among others, that are not captured in the data. A stopping criterion with three noise layers is utilized.

The estimation results are shown in Figure 2.1. The world trade growth curve (global mean), estimated with SIR, shows that economic growth expanded heavily in the late 1980's and 1990's. The first bicluster represents additional growth patterns excluding notably China. The second bicluster identifies imports from the former Soviet Union. The dip around 1989 corresponds with the fall of the former Soviet Union. Bicluster 4 recovers the growing economic strength of east Asian countries and the so-called 'Asian miracles': countries in east Asia that experienced persistent and rapid economic growth in the 1990's (*Stiglitz*, 1996; *Nelson and Pack*, 1998). The mean effects are smooth functions that are visually interpretable and provide reasonable estimates between time points. Moreover, a naive approach may become overly focused on global changes. The proposed approach recovers biclusters from

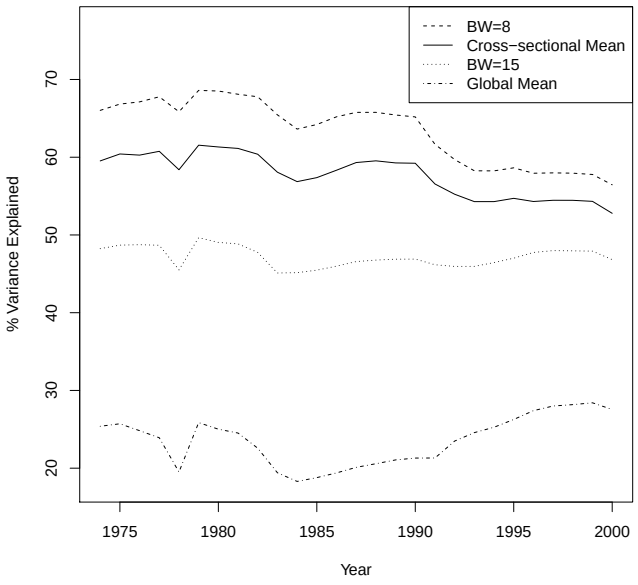



Figure 2.14: Percent of variance explained for different models for the world trade data..

the data after removing the world growth curve.

For a closer inspection of the data, the terms $\alpha_{ik}(m)$ and $\beta_{jk}(m)$ identify the countries that are most important to each bicluster. Figure 2.15 shows estimates of $f_k(m) + \alpha_{ik}(m)$ and $f_k(m) + \beta_{jk}(m)$ for each country in bicluster 4, where the largest exporters include the so-called Asian miracles, in addition to the USA and some European countries. The corresponding group of importers are more heterogeneous, with the largest importers consisting of Russia and eastern European countries.

Figure 2.14 shows that the biclusters substantially improve the percent of variance explained. Moreover, a moderate amount of smoothing achieves the highest accuracy, while also improving interpretability of the bicluster mean effects. Table 2.6 shows the proposed plaid procedure outperforms competing approaches in this respect, including matrix factorizations.

Figure 2.16 shows the estimated import and export levels without the global mean. These statistics can be computed with row and column sums of the estimated data

| | Proposed Plaid | Direct SVD | Joint SVD | Direct Plaid |
|---|---|---|---|---|
| % Variance Explained | 58.3 | 40.7 | 31.1 | 25.6 |

Table 2.6: % Variance Explained of different approaches for world trade data. 'Direct' denotes applying Singular Value Decomposition or the plaid model to each data matrix separately. Joint SVD uses a common basis ($X_t = UV_t^T$). The first 10 components are kept in the matrix factorizations.

after subtracting the global mean. These figures denote the trade performance of each country relative to global growth, and helps answer whether a country was trading at higher levels than expected during global recessions/booms. For example, the estimated export levels show the relative decline of US and rise of Chinese exporting. In 2000, Russian exports and Indian imports underperformed the global market. In general, African and Central American nations tend to follow the global trend.

Now that we have seen the benefits of the plaid model to exploratory and visual analysis, the reader may wonder if similar information can be extracted by inspecting heatmaps of the raw data. Shown in Figure 8 of the Supplementary material, displays of the raw data do convey the strongest patterns, such as the rise of Chinese exporting by the year 2000. However, insights about a country's trade status relative to the underlying time-varying mean are difficult to gain by displaying the raw data.

## 2.6 Conclusion

As with other exploratory and visualization tools, plaid models are sensitive to the scaling of the data. For instance, if investigating annual world trade values that are expressed in nominal dollars instead of log-nominal dollars, then results are dominated by the United States, because that country has by far the largest variance. Just as in principal components analysis and many other multivariate methods, the analyst should make a decision on standardizing observations based on what aspect of the data is of interest. Similarly, the results can change in response to the algorithm and

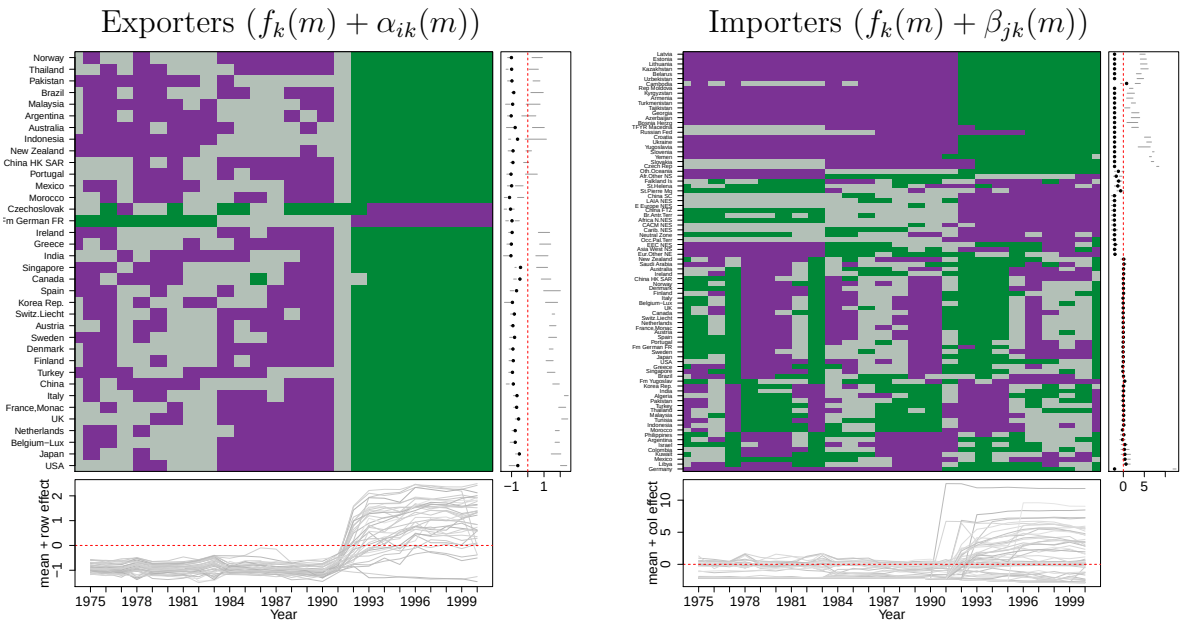



Figure 2.15: The left panel shows $f_k(m) + \alpha_{ik}(m)$ for $k = 4$, identifying the importers that are affected most under bicluster 4. The right panel shows $f_k(m) + \beta_{jk}(m)$, identifying exporting countries are affected most under the bicluster. Figures are created using code from *Peng* (2008).

model parameterization. For example, a greater number of overlapping biclusters are typically discovered when utilizing just a mean effect for the T-cell data. Many of the genes reported above are biclustered in either bicluster specification, and are consistent with the previous findings of *Rangel et al.* (2004). The main results I present in the numerical work above are consistently found in repeated analyses of the data, thus supporting the notion that they are not noise artifacts.

Runtimes are also provided in Table 2.7, where I add noisy columns to the illustrative example's data generating process to investigate the performance of the model when the number of samples and variables are each in the order of thousands. Though smoothing does add computational cost, the algorithms can produce estimates in a practically useful amount of time for data sets with thousands of rows and columns.

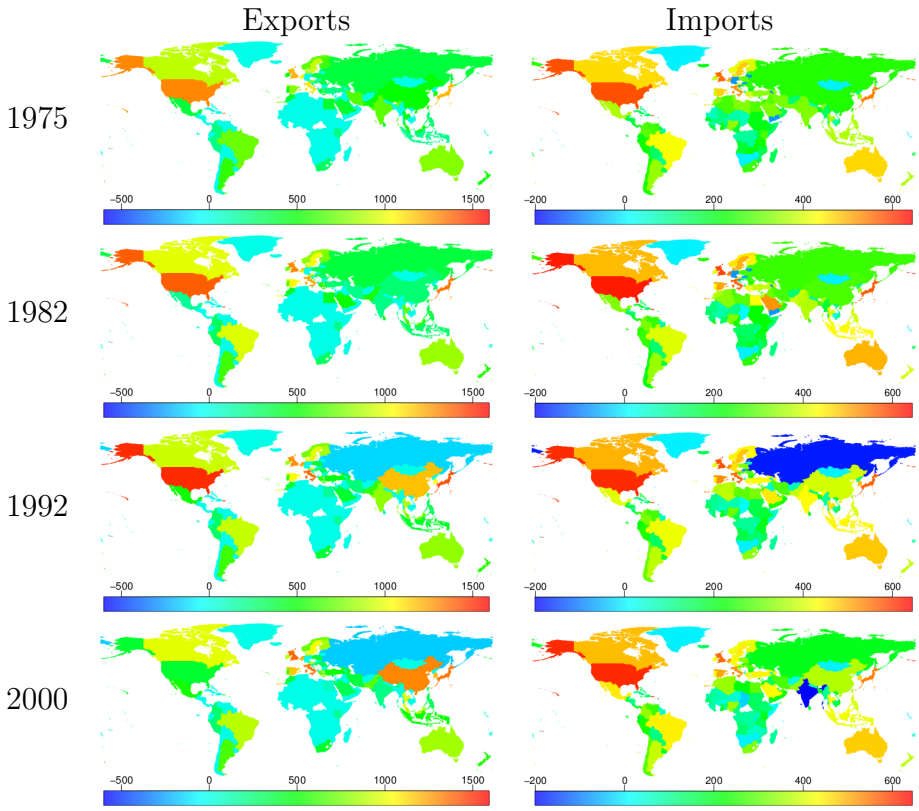



Figure 2.16: Heatmaps of estimated import and export levels without the global mean.

| | Illustrative | Illustrative (extra Columns) | Illustrative (extra Rows/Columns) | T-cell | World Trade |
|---|---|---|---|---|---|
| Cross-sectional | 1.7 | 12.8 | 203.9 | 1.8 | 17.2 |
| Kernel Smoothed | 13.0 | 60.5 | 769.4 | 4.6 | 103.6 |
| Turner-2 | 1.2 | 17.2 | 332.5 | 0.72 | 12.1 |
| Dimensions | 100 × 100 × 10 | 100 × 1000 × 10 | 1000 × 1000 × 10 | 58 × 58 × 10 | 200 × 200 × 27 |

Table 2.7: Average runtimes (seconds) on a Linux netbook with 4GB Ram and 1.7 GHz AMD Athlon Neo K125 Processor. The number of layers is fixed at five with bicluster means given by their cross-sectional mean or kernel smoothed.

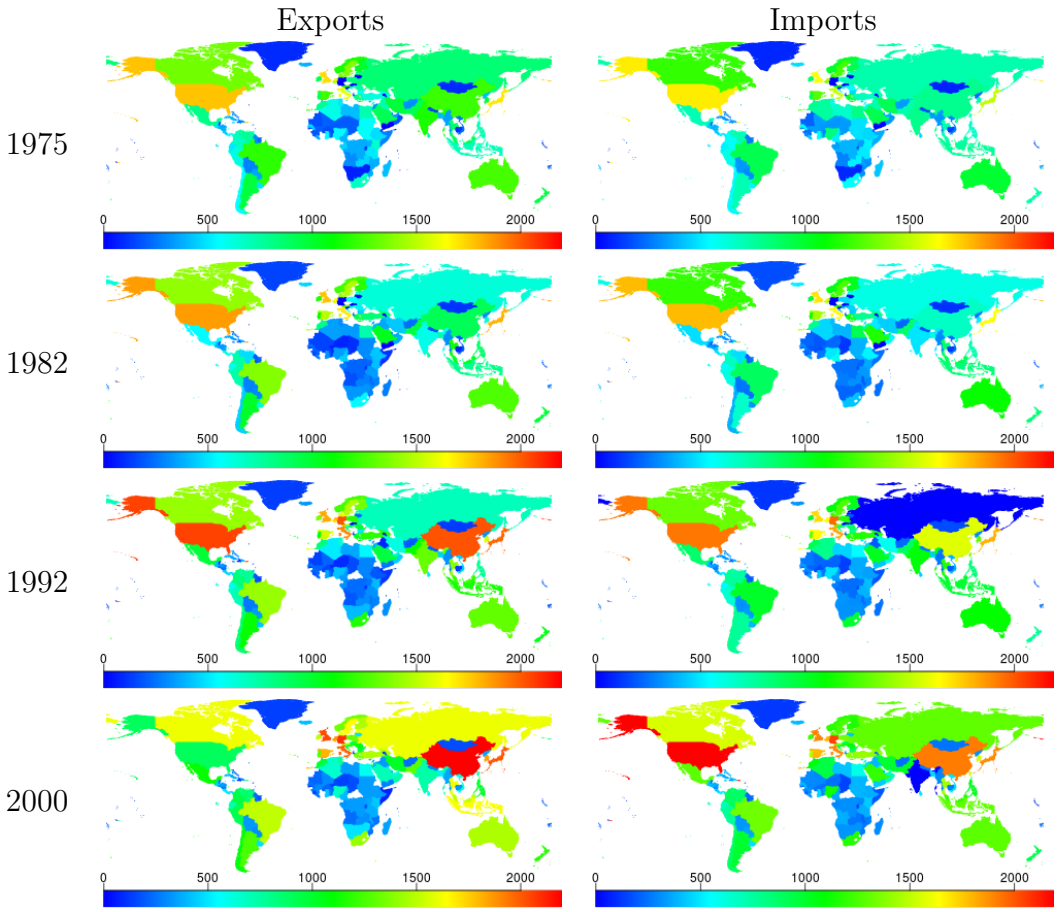



Figure 2.17: Heatmaps of raw import and export levels.

# CHAPTER III

# Integrative Analysis of Three-Dimensional Data Arrays with Non-negative Matrix Factorization

## 3.1 Introduction

As mentioned in Chapter I, it is increasingly common to collect data over time or in different conditions on components of a large, complex system. A challenging, yet important task in this context is to separate stable from dynamic structure, followed by visualizations that communicate the results and help improve decision making.

The goal of this chapter is to utilize a regularization within the framework of non-negative matrix factorization for such data integration tasks. I introduce a model that identifies stable structure by integrating the matrix observations through a common basis. After computing the matrix factorization, the basis and factors can be used to represent, respectively, the stable and dynamic patterns in the data.

Next, I note some aspects of the proposed model. The proposed factorization is a general one, where the depth dimension can correspond to time, experimental condition, dose level, or geographic location. Moreover, other knowledge such as grouping structure on the features (columns), label information on the samples (rows), etc., can be accommodated when constructing the smoothing penalty. Thus, in addition to data integration, the model in this chapter can also utilized for clustering and

semi-supervised learning in sets of data matrices.

The remainder of this chapter is organized as follows: I describe the proposed approach in the next section and present the core algorithms with convergence results in Section 3.3. In Section 3.4, I provide a systematic and theoretically sound approach to parameter selection, including a statistically consistency cross validation procedure for selecting the approximation rank. The approach is illustrated on real-world data sets derived from citation networks and trade data in Section 3.5. I close the chapter with a short discussion in Section 3.6.

## 3.2 Integrative NMF for Data Arrays

Given a collection of data matrices $\{X_m \in \mathbb{R}_+^{n\times p}, m = 1, \ldots, M\}$, the goal is to produce a sequence of smooth, low rank factors for data representation and pattern discovery. The first component of the proposed objective function is

$$\min_{U\geq 0,\{V_m\geq 0\}} \sum_{m=1}^{M} ||X_m - UV_m^T||_F^2. \tag{3.1}$$

The basis $U$ captures information common in all data slices, and the coefficient matrices $\{V_m\}$ vary by the third dimension (time/experimental condition). A common basis allows the model to integrate information from all data matrices and mitigate the influence of transient patterns.

The local smoothness of $V_m$ is a fundamental concern to reduce noise effects, and maintain the interpretability and effectiveness of visual representations. I add a regularization term to encourage smoothness that penalizes changes in $V$ depending on the distance between adjacent data slices. The penalty strength is controlled by a weighting function $W(\cdot, \cdot)$. I also add an optional group penalty that controls the

fluctuations of a given group within the factors. The objective function becomes

$$\min_{U\geq 0,\{V_m\geq 0\}} \quad \sum_{m=1}^{M} ||X_m - UV_m^T||_F^2 + \sum_{m,\tilde{m}=1}^{M} W(m,\tilde{m})||V_m - V_{\tilde{m}}||_F^2 + \lambda_g \sum_{m=1}^{T} Tr(V_m^T \mathcal{L}_m V_m), \tag{3.2}$$

where $\mathcal{L}_m$ corresponds to the Laplacian of a graph induced by pairwise group relations. Without prior knowledge of group structure, $\lambda_g$ is set to zero so that the penalty is optional. Otherwise if such groupings are known, larger values of $\lambda_g$ more strongly encourage groups to evolve similarly in $V_t$. The weight function $W(q,\tilde{m})$ can take a variety of shapes. For instance, *Cai et al.* (2011) investigate weights given by the heat kernel, which has fundamental connections to the Laplace-Beltrami operator on differentiable functions on manifolds. Gaussian weighting, triangular kernels, uniform kernels, among many others, have been proposed and extensively studied in the vast literature on kernel methods. The weight function is discussed further in Section 3.4.1.

This objective function can be written succinctly as follows

$$\min_{U\geq 0,\{V_m\geq 0\}} \sum_{m=1}^{M} ||X_m - UV_m^T||_F^2 + Tr(V^T \mathcal{L} V), \tag{3.3}$$

where $V = [V_1, V_2, ..., V_M]^T$. The smoothing matrix $\mathcal{L}$ is constructed as a function of the penalty weights. It can be seen that the smoothing matrix corresponds to the Laplacian of a graph (network) with a node for each column in each data matrix. The graph Laplacian matrix is defined as

$$\mathcal{L} = \mathcal{D} - \mathcal{W}, \tag{3.4}$$

where $\mathcal{D}$ is a diagonal degree matrix whose (j,j) entry is $\sum_i (\mathcal{W})_{ij}$ and $\mathcal{W}$ is an adjacency matrix. An overview of its properties can be found in *Chung* (1997) and references therein. The representation in (3.3) is a consequence of the following fact established in Chapter 1 of *Chung* (1997): For every vector $f$,

$$f^T \mathcal{L} f = \frac{1}{2} \sum_{i \sim j} (\mathcal{W})_{ij} (f_i - f_j)^2, \tag{3.5}$$

where $i \sim j$ denotes node $i$ and $j$ are connected by an edge.

Next, I will discuss the form of the regularization graph. Some additional notation is introduced to simplify the presentation. Let double subscripts identify submatrices for the smoothing matrix $\mathcal{L}$, and its components $\mathcal{D}$ and $\mathcal{W}$. For example, $\mathcal{W}_{kj}$ denotes the square submatrix corresponding to edges from data matrix $k$ to data matrix $j$. The structure for $\mathcal{W}$ is shown next. The same block structure applies to $\mathcal{L}$ and $\mathcal{D}$.

$$\mathcal{W} = \begin{bmatrix} \mathcal{W}_{11} & . & ... & \mathcal{W}_{1M} \\ . & \mathcal{W}_{22} & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ \mathcal{W}_{M1} & . & ... & \mathcal{W}_{MM} \end{bmatrix}, \tag{3.6}$$

where $\mathcal{W}_{m\tilde{m}}$ is a square submatrix. The weighted adjacency matrix underlying the regularization graph has the following form

$$\mathcal{W}_{m\tilde{m}} = \begin{cases} W(m, \tilde{m}) \cdot I & \text{if } m \neq \tilde{m}, \\ \lambda_g A_m & \text{if } m = \tilde{m}, \end{cases} \tag{3.7}$$

where $I$ is the identity matrix and $A_m$ is the adjacency matrix of the graph induced by the given groups. Thus, the full graph and hence Laplacian smoothing matrix have a highly structured, sparse form. With the weighting functions described above,

'nearby' data have a larger effect on the current factor; this effect decreases as the distance between the data matrices grows.

The form of $\mathcal{L}$ given in (3.7) enforces and enhances similarity between adjacent data slices. However, in other contexts $\mathcal{L}$ would take different forms. For example, factorial experimental designs, which are common throughout the social and health sciences, correspond to tree structures. Other particular graph forms have a one-to-one correspondence with different experimental designs. Moreover, additional knowledge, such as group or label information, can be accommodated through additional graph structure.

Lastly, I briefly discuss another approach that addresses higher order array structure is non-negative tensor factorizations (*Cichocki et al.*, 2009; *Hazan et al.*, 2005; *Welling and Weber*, 2001), where both the $U$ and $V$ vary by condition or time ($X_m \approx U_m V_m^T$). This type of model is introduced in Chapter IV and is more robust to sharp changes in the data. The model in this chapter uses a global basis, which assumes there is common information across data slices and can struggle with large structural changes. However, NMF is already an under-constrained model and additionally allowing $U$ to vary would make it *massively* under-constrained. Moreover, due to the multiplicative nature and rotational indeterminancy, it becomes challenging without a common basis to compare factors across conditions. An even more complex regularization would be required to control the factor evolutions through the data. In contrast, utilizing a common basis provides parsimony, and more efficiently facilitates visualization, analysis and interpretation.

### 3.2.1 Illustrative Example

Before discussing algorithmic issues, I illustrate the model with simulated data. In particular, I set $X_m \in \mathbb{R}^{100\times 100}$, where $(X_m)_{ij} \sim \mathcal{N}(10, 1)$. There are three embedded

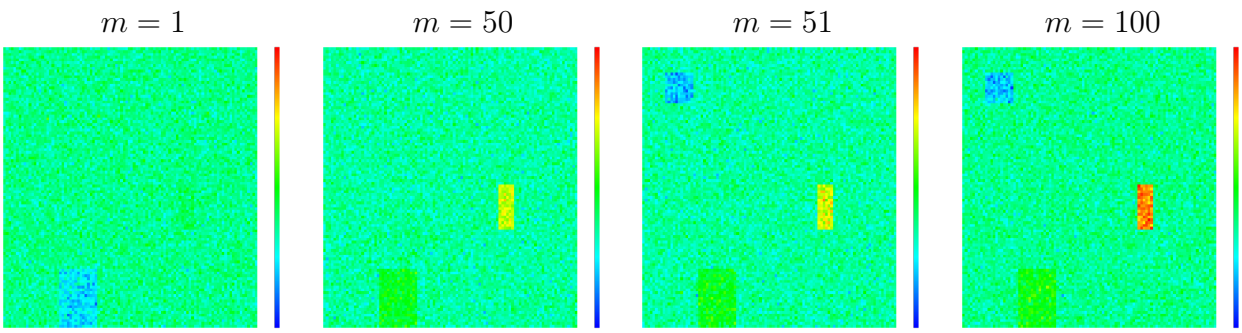



Figure 3.1: Examples of Raw Data for the illustrative example.

submatrices with evolving mean structures:

$$\mu_1(m) = 10 + \frac{2.5(m-30)}{\sqrt{1+(m-30)^2}} \tag{3.8}$$

$$\mu_2(m) = 10 - 3\mathbb{I}\{m > 50\} \tag{3.9}$$

$$\mu_3(m) = 10 + \sqrt{m}. \tag{3.10}$$

The first submatrix is composed of rows 80 to 100 and columns from 23 to 37. The second submatrix is square and is composed of rows/columns 10 to 20. The third submatrix contains rows 50 to 65 and columns 70 to 75. There are 100 observed data matrices and the third dimension is sampled uniformly between 1 and 100, that is, $m = 1, 2, .., 100$. Figure 3.1 shows examples of the input data.

I will use this data to compare three models:

1. The **direct model** applies classical NMF to each data slice separately: $X_m \approx U_m V_m^T$.

2. **Common** $U$ applies NMF to each data slice with a common basis: $X_m \approx U V_m^T$.

3. **Common** $U$ **with penalty** applies the proposed NMF model by minimizing (3.3).

For each model, I fit a series of rank one approximations to facilitate visualization of the factorizations. The factors are plotted as time-series in Figure 3.2. First,

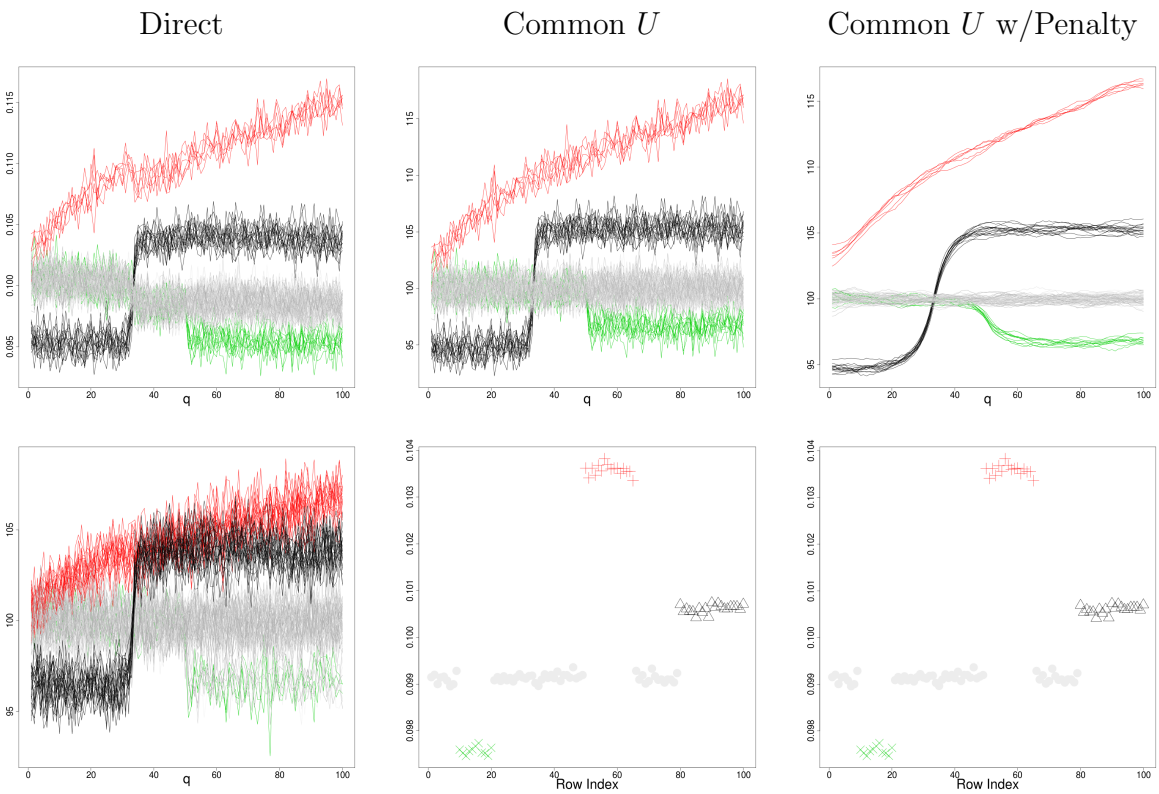



Figure 3.2: Estimates for the illustrative example under different model specifications. The first row shows estimates for $V$. Each line (trajectory) corresponds to a column in the data. The second row shows estimates of $U$. The colors identify the true columns/rows that belong to the submatrices.

all three models do a reasonable job at representing the columns comprising the submatrices. However, without smoothness penalties the cluster memberships and trajectories are difficult to identify. The second row shows a user would encounter additional difficulty in identifying the rows of the submatrices with the naive model. It is easier to identify row memberships with the other models. Altogether, the smoothed version appears more satisfactory, as the main structural and functional patterns underlying the data are clearly represented.

## 3.3 Algorithms

### 3.3.1 Multiplicative Updating

One can derive multiplicative updating rules similar to those discussed in *Lee and Seung* (1999, 2001) by following the standard argument of forming the Lagrangian and deriving the corresponding KKT conditions.

The objective function in (3.3) can be written as

$$\begin{aligned} \mathcal{O} &= \sum_{m=1}^{M} Tr(X_m - UV_m)(X_m - UV_m)^T \qquad (3.11) \\ &\quad + Tr[V_1, V_2, ..., V_m]\mathcal{L}[V_1, V_2, ..., V_m]^T \\ &= \sum_{m=1}^{M} \{Tr(X_m X_m^T) - 2Tr(X_m V_m U^T) \qquad (3.12) \\ &\quad + Tr(UV_m^T V_m U^T)\} \\ &\quad + Tr[V_1, V_2, ..., V_m]\mathcal{L}[V_1, V_2, ..., V_m]^T \end{aligned}$$

To enforce the non-negativity constraints, consider the Lagrangian

$$\begin{aligned} L &= \sum_{m=1}^{M}\{Tr(X_m X_m^T) - 2Tr(X_m V_m U^T) \\ &\quad + Tr(U V_m^T V_m U^T)\} \\ &\quad + Tr[V_1, V_2, ..., V_m]\mathcal{L}[V_1, V_2, ..., V_m]^T \\ &\quad + Tr(\Phi U^T) + Tr(\Psi V^T), \end{aligned} \tag{3.13}$$

where $\Phi, \Psi$ are Lagrange multipliers.

I get the following KKT optimality conditions by setting $\frac{\partial L}{\partial U} = \frac{\partial L}{\partial V_m} = 0$.

$$-2\sum_{m=1}^{M} X_m V_m + 2\sum_{m=1}^{M} U V_m^T V_m = \Phi \tag{3.14}$$

$$-2X_m^T U + 2V_m U^T U + 2\sum_{j=1}^{M} \mathcal{L}_{mj} V_j = \Psi. \tag{3.15}$$

Then, the KKT complimentary slackness conditions yield

$$(-2\sum_{m=1}^{M} X_m V_m + 2\sum_{m=1}^{M} U V_m^T V_m)_{ij}(U)_{ij} = 0 \tag{3.16}$$

$$(-2X_m^T U + 2V_m U^T U + 2\sum_{j=1}^{M} \mathcal{L}_{mj} V_j)_{ij}(V_m)_{ij} = 0. \tag{3.17}$$

These relations lead to the following multiplicative update rules

$$(U)_{ij} \leftarrow (U)_{ij}\frac{\sum_{m=1}^{M}(X_m V_m)_{ij}}{\sum_{m=1}^{M}(U V_m^T V_m)_{ij}} \tag{3.18}$$

$$(V_m)_{ij} \leftarrow (V_m)_{ij}\frac{(X_m^T U + \sum_{j=1}^{M} \mathcal{W}_{mj} V_j)_{ij}}{(V_m U^T U + \mathcal{D}_{mm} V_m)_{ij}}, \tag{3.19}$$

where I use the fact that $\mathcal{L} = \mathcal{D} - \mathcal{W}$, and $\mathcal{D}_{jm} = 0$ for all $j \neq m$.

The algorithm for NMF is shown in Algorithm III.1.

I establish the following result for the updating algorithm.

**Algorithm III.1** Multiplicative Updating for Integrative NMF

1: Construct the smoothing matrix, $\mathcal{L} = \mathcal{D} - \mathcal{W}$
2: Initialize $U, \{V_m\}$ as a dense, positive random matrices
3: **repeat**
4: Set

$$(U)_{ij} \leftarrow (U)_{ij} \frac{\sum_{m=1}^{M} (X_m V_m)_{ij}}{\sum_{m=1}^{M} (U V_m^T V_m)_{ij}}$$

5: **for** m=1..M **do**
6: Set $V_m$

$$(V_m)_{ij} \leftarrow (V_m)_{ij} \frac{(X_m^T U + \sum_{j=1}^{M} W_{mj} V_j)_{ij}}{(V_m U^T U + D_{mm} V_m)_{ij}}$$

7: **end for**
8: **until** Convergence

**Theorem III.1.** *The objective function in (3.3) is non-increasing under the multiplicative updates rules.*

A proof is given in the Appendix 3.7. The argument makes use of auxiliary functions and is similar to the one used by *Lee and Seung* (1999, 2001). Minor modifications provided by *Lin* (2007) can be employed to guarantee convergence to a stationary point.

### 3.3.2 Alternating Least Squares

As mentioned in the Introduction, it has been suggested that this class of algorithms converges to less satisfactory solutions due to the fact that any element that is zero must remain zero in subsequent updates. Thus, the algorithm can get 'stuck' on a particular fixed point. Another more flexible class of NMF algorithm is the alternating non-negative least squares algorithm (ANLS).

To develop an alternating least squares algorithm for the model, I first obtain the

partial derivatives of Equation 3.3. The objective function can be written as

$$\begin{aligned} \mathcal{O} &= ||X - UV^T||_F^2 + Tr(V^T \mathcal{L} V) && (3.20) \\ &= Tr(XX^T) - 2Tr(XVU^T) && (3.21) \\ &\quad + Tr(UV^T VU^T) + Tr(V^T \mathcal{L} V). \end{aligned}$$

Taking partial derivatives

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial U} &= -2XV + 2UV^T V && (3.22) \\ \frac{\partial \mathcal{O}}{\partial V} &= -2X^T U + 2VU^T U + 2\mathcal{L}V, && (3.23) \end{aligned}$$

and solving respectively for $U$ and $V$ after setting equal to zero yields update relations for the algorithm. In particular, $U$ can be solved for using the usual least squares estimator

$$U = XV(V^T V)^{-1} \tag{3.24}$$

or using active set methods to solve subject to non-negativity (see *Kim and Park* (2008)).

Setting $\frac{\partial \mathcal{O}}{\partial V}$ equal to zero and isolating $V$ yields

$$VU^T U + \mathcal{L}V = X^T U. \tag{3.25}$$

Hence, updating $V$ requires solving an important matrix equation called Sylvester's equation, which is of the form

$$VA + BV = C, \tag{3.26}$$

where $A = U^T U$ is $K \times K$, $B = \mathcal{L}$ is $n \times n$, $C = X^T U$ is $n \times K$ and $V$ is solved for. A classical algorithm for the numerical solution of Sylvester's equation is the

Bartels-Stewart algorithm, which transforms $A$ and $B$ into Hessenberg form, then solves the resulting system via back-substitution *Bartels and Stewart* (1972); *Golub et al.* (1979). This leads to Algorithm III.2.

**Algorithm III.2** Alternating Least Squares for Integrative NMF

1: Construct the smoothing matrix, $\mathcal{L} = \mathcal{D} - \mathcal{W}$
2: Initialize $V = [V_1, \ldots, V_Q]$ as a dense, positive random matrices
3: **repeat**
4: Set $U = XV(V^TV)^{-1}$
5: Set all negative elements in $U$ to 0
6: Solve for $V$ using Bartels-Stewart in
$VU^TU + \mathcal{L}V = X^TU$
7: Set all negative values in $V$ to 0
8: **until** Convergence

Projection steps are included in the algorithm because the Bartels-Stewart approach solves Sylvester's equation *without* non-negativity constraints. To my knowledge, a procedure to find the constrained solution is an open problem and technically challenging.

*Berry et al.* (2006) propose an NMF algorithm for such an unconstrained solution that employs projection steps to enforce non-negativity (henceforth referred to as ALS). While there exists numerical support for approximate ALS algorithms, convergence theory is lacking, since, for instance, projecting onto the non-negative orthant could increase the objective value. This would negate the non-increasing update theorem presented in the Appendix for multiplicative updating.

The computational cost of Bartels-Stewart for solving Equation 3.26 is conservatively estimated in the original paper (*Bartels and Stewart*, 1972) to be $O(K^3+(nT)^3)$. The cost per iteration of multiplicative updating is $O(n^2KT)$. Thus, the cost per iteration for the ALS algorithm is higher. On the other hand, the overall convergence rate is quadratic, which is faster than that of multiplicative updating. Multiplicative updating attains a linear convergence rate and can be especially slow near limit points *Chu et al.* (2004).

## 3.4 Parameter Selection

### 3.4.1 Choosing the Weight Function

I use a weight function that is proportional to the triangular kernel, since it results in sparse smoothing matrices, and gives larger weight to nearby slices:

$$W(m, \tilde{m}) = \frac{\lambda}{h_m} K(\frac{m - \tilde{m}}{h_m}), \tag{3.27}$$

$$K(x) = (1 - |x|)\mathbb{I}\{x \in (-1, 1)\}, \tag{3.28}$$

where $\lambda$ controls the strength of the penalty, and $h_m$ (the bandwidth) is a parameter used to adjust the penalty to the scale of the data.

The bandwidth controls the number of neighboring matrices to average over. Larger values of $h_m$ mean that the model has more memory, so it incorporates more points for estimation. This risks missing sharper changes in the data and only detecting the most persistent patterns. On the other hand, small values of $h_m$ make the fitting more sensitive to sharp changes, but increase variance due to smaller number of observations. I find setting $h_m$ to include the closest two or three data matrices is sufficient for the data considered in this chapter. Larger values could be used in noisier settings to smooth results.

The selection of $\lambda$ is again highly contextual. If the goal is visual exploration of the data, it can be satisfactory to choose the penalty strength subjectively by eye. This involves looking at several estimates over a range of strengths and selecting the one that emphasizes the structure most. For other purposes, such as clustering or prediction, the cross validation based approach discussed below can be extended to select $\lambda$ in addition to the estimation rank.

### 3.4.2 Choosing the Estimation Rank

For visualization, lower rank ($\leq 3$) representations are preferred for practical reasons. Figures similar to Figure 3.2 can be constructed to visualize the underlying dynamic structure in the data.

For the goal of clustering, the rank should be equal to the number of underlying groups. Then, $(U)_{ik}(V_m)_{kj}$ can be interpreted as the contribution of the $k-th$ cluster to the data element $(X_m)_{ij}$. The rank can be ascertained by examining the accuracy of the reconstruction as a function of rank. However, this tends to rely on subjective judgments and overfit the given data. Cross validation based approaches are theoretically preferable and follow the same intuition.

The idea behind cross validation is to use random subsets of the data from each data slice to fit the model, and another subset from each data slice to assess accuracy. Different values of $K$ are then cycled over and the one that corresponds to the lowest test error is chosen.

Due to the data structure, I employ two-dimensional cross validation. Two-dimensional refers to the selection of *submatrices* for the training and test data. Special care is taken to ensure that the same rows and columns are held out of every data slice, and the dimensions of the training and test sets are identical.

The hold out pattern divides the rows into $k$ groups, the columns into $l$ groups, then uses the corresponding $kl$ submatrices to fit and test the model. In each submatrix, the given row and column group identifies a held out submatrix that is used as test data, while the remaining cells are used for training. The algorithm is shown in Algorithm III.3. The notation in the algorithm uses $\mathcal{I}_l$ and $\mathcal{I}_J$ as index sets to identify submatrices in the each data matrix.

I then cycle over different values of $K$ to choose the one that minimizes average test error. Consistency results are developed in *Perry and Owen* (2009) to provide theoretical foundations for this approach.

**Algorithm III.3** Cross-validation for choosing the approximation rank

1: Form row holdout set: $\mathcal{I}_l \subset \{1, .., n\}$

2: Form column holdout set: $\mathcal{I}_J = \subset \{1, .., p\}$

3: Set

$$(\tilde{U}, \tilde{V}_m) = \underset{U, V_m \geq 0}{\arg\min} \sum_m ||(X_m)_{-\mathcal{I}_l, -\mathcal{I}_J} - U V_m^T||_F^2$$

4: Set

$$\breve{U} = \underset{U \geq 0}{\arg\min} \sum_m ||(X_m)_{\mathcal{I}_l, -\mathcal{I}_J} - U \tilde{V}_m^T||_F^2$$

5: Set

$$\breve{V}_m = \underset{V \geq 0}{\arg\min} \sum_m ||(X_m)_{-\mathcal{I}_l, \mathcal{I}_J} - \tilde{U} V_m^T||_F^2$$

6: Set

$$(\hat{X_m})_{\mathcal{I}_l, \mathcal{I}_J} = \breve{U} \breve{V}_m^T$$

7: Compute Test error

$$\text{Test Error} = \sum_m ||(X_m)_{\mathcal{I}_l, \mathcal{I}_J} - (\hat{X_m})_{\mathcal{I}_l, \mathcal{I}_J}||_F^2$$

## 3.5 Applications

### 3.5.1 World Trade Data

I use the model to extract and compare decompositions of global trade for different types of goods. The data consists of bilateral trade flows between 202 countries from 1980 to 2000 (*Feenstra et al.*, 2004) and is available at `http://cid.econ.ucdavis.edu/`. The raw data contains annual trade values classified into approximately one thousand categories according to the Standard International Trade Classification (SITC) codes. We aggregate SITC codes to construct the following trade types.

1. **Agriculture** includes trade with SITC codes mentioning rice, wheat, fish, meat, milk, butter, fruits, or vegetables in its description.

2. **Ore** includes iron, copper, aluminum, nickel, copper, lead, zinc, tin, manganese,

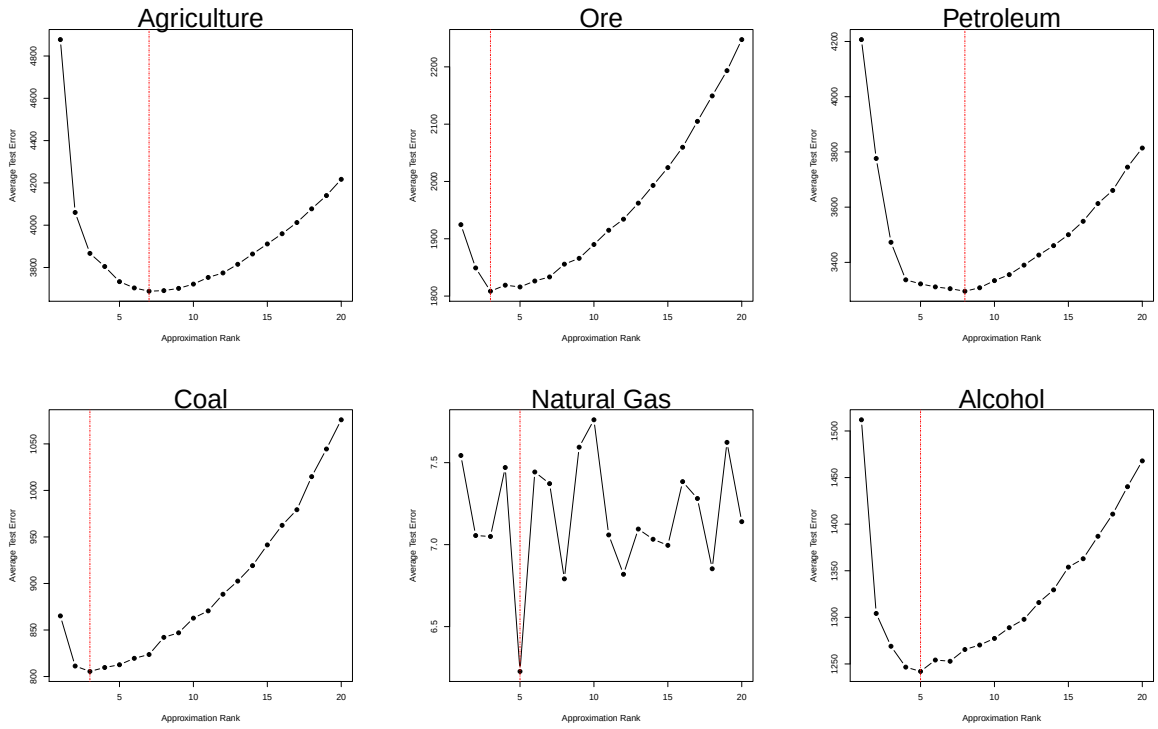



Figure 3.3: Average test errors obtained by cross validation with 5 partitions of row and column sets (25 submatrices in each data matrix). The vertical line identifies the minimum.

and other concentrates of base and precious metals.

3. **Petroleum** includes trade with SITC codes mentioning petroleum or oils in its description.

4. **Coal** includes trade with SITC codes mentioning coal, lignite or peat in its description.

5. **Natural Gas** includes trade with SITC codes mentioning natural gas in its description.

6. **Alcohol** includes general alcoholic beverages, spirits, liqueurs, beer, wine, and other fermented beverages.

Thus, we observe six separate data series over 21 time points, where each $i, j$ entry in a data matrix denotes the exports from country $i$ to country $j$. Since trade flows can differ in size by orders of magnitude, we work with values that are expressed in nominal log dollars.

I apply the model for a number of tasks. First, the basis vectors in $U$ decompose the data arrays into interpretable parts by integrating over all time points in a systematic way. In context, basis vectors are useful for identifying countries that represent the most persistent driving forces in global trade. For more detailed analysis, the time-varying expressions ($V_t$) can be examined to capture dynamics.

The number of components for each data array are chosen according to cross validation, as shown in Figure 3.3. The figures seem reasonable, except perhaps natural gas, where the average test error is near zero and the shape of the plot is irregular. As shown in the Appendix 3.7, a single component yields a very accurate reconstruction of the natural gas data. Four to five components yields a nearly perfect reconstruction. Hence, the choice of five components from cross validation seems reasonable, but a smaller number could suffice. Altogether, the six data arrays are processed using between three and eight components.

Figure 3.4 shows the basis vectors learned for each set of goods. Coal, natural gas, and alcohol appear to be commonly driven by the United States, Russia, Australia, and parts of Europe, since they former and latter due to their level industrialization are net importers, while the other two are significant net exporters. In contrast, Agriculture, ore, and petroleum are more diversified markets, with all continents showing presence. The ore bases show Australia, Brazil, and South Africa as important to each component, due to their exporting strength on the basis of their vast deposits. The United States, due to its importing, appears weakly in all three components. The petroleum bases shows the importance of Saudi Arabia, Iran, Venezuela, Columbia, Libya, Algeria and Morocco to the market. These countries largely appear only in the petroleum bases, indicating that oil represents their main tradable good.

Countries in southeast Asia, such as China, South Korea and Singapore, that experienced rapid growth in the 1990's (see *Stiglitz* (1996); *Nelson and Pack* (1998)) are not readily observed in the learned bases. Their absence is due to the fact that $U$ combines information from all times to highlight the most *persistent* countries.

Another contributing factor for China's absence is that for the years 1988 - 2000, the raw data feature export values that were adjusted (lowered) to account for Chinese goods that are re-exported through Hong Kong (see page 5 of *Feenstra et al.* (2004)). As a result, the import value from Hong Kong increases, while the the export value from China decreases by the same amount. We anticipate that with additional data from 2001 and beyond, China's influence in particular would grow enormously. Nonetheless, the time-varying expressions ($V_t$) capture dynamic behavior and many of the Asian countries discussed above are more visible in them.

Figure 3.5 displays the estimated expressions ($V_t$) as time series. The top countries' expressions reflect global economic growth. The curves appear roughly sigmoid shaped and are consistent with economic models that utilize sigmoid shaped curves to model global trade and economic growth (*Rodriguez and Rodrik*, 2001; *Bernanke*

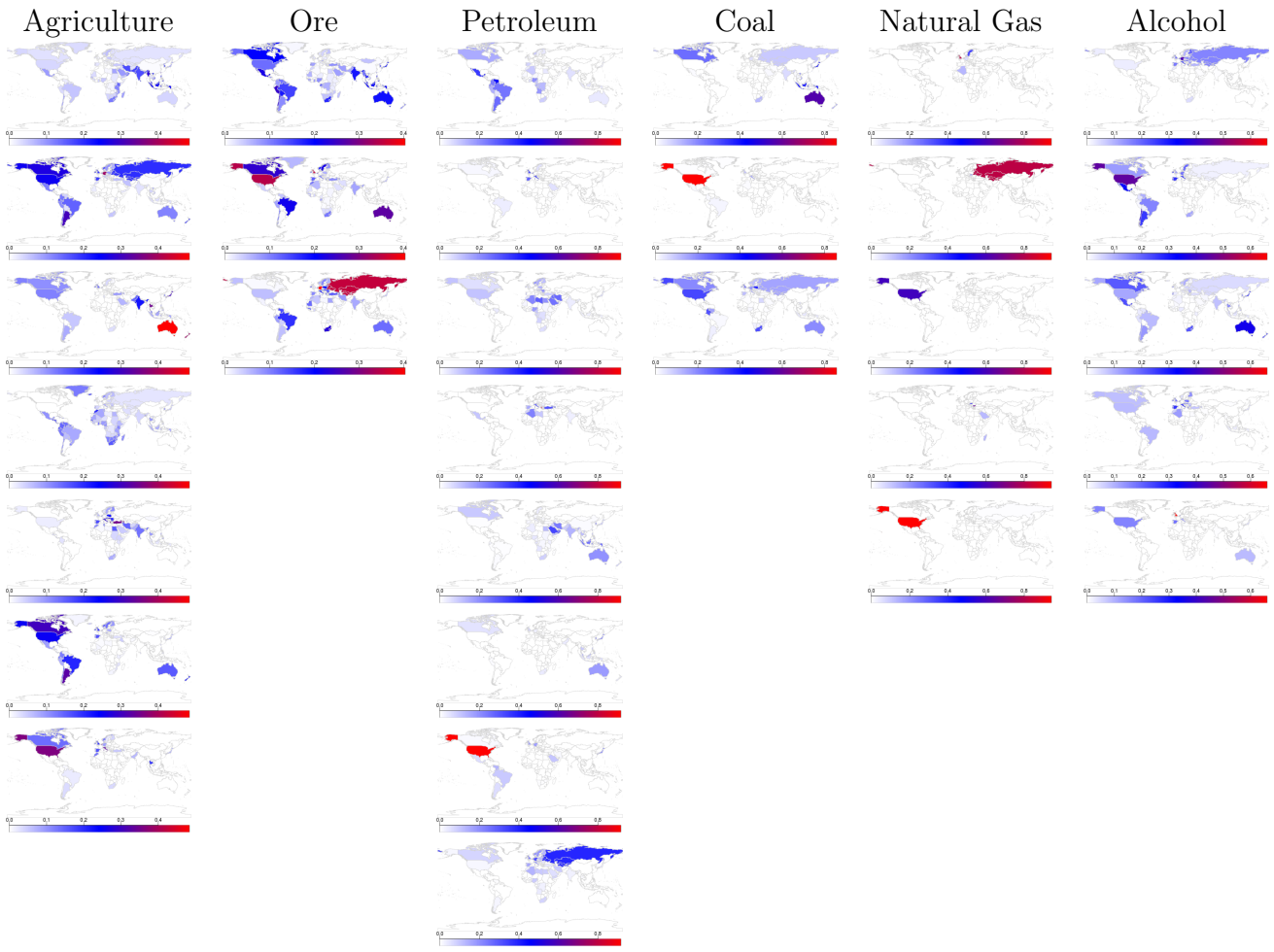



Figure 3.4: Basis vectors learned from the world trade data arrays.

| | Agriculture | Ore | Petroleum | Coal | Natural Gas | Alcohol |
|---|---|---|---|---|---|---|
| Number Components | 7 | 3 | 8 | 3 | 5 | 5 |
| % Var. Explained (Penalty) | 74.0 | 54.6 | 66.7 | 49.8 | 36.1 | 70.1 |
| % Var. Explained (no Penalty) | 75.3 | 57.1 | 68.0 | 57.6 | 78.5 | 71.1 |

Table 3.1: Summary statistics for the decompositions. The penalized fit corresponds to $\lambda = 1000$, with $h_m = 2$ years. Percent of Variance Explained is defined as $1 - ||X - \hat{U}\hat{V}^T||_F / ||X||_F$.

*and Rogoff*, 2001).

Figures in the Appendix show the expressions without any penalty projected onto a world map. The main pattern conveyed is that countries in North and South America, Australia, and South Africa increased their coal trade levels over time. However, a closer inspection shows some strange features. For instance, India and Libya appear active only in 1990. Similarly, Russia appears only in year 2000. The third component even indicates a persistent decreasing trend in trade levels throughout Europe and other parts of the world. These features are not consistent with growth (sigmoid shaped) curve models.

Employing the smoothness penalty, as in Figure 3.6, removes the unwanted features while still conveying the main pattern of growth in North and South America, Australia, and South Africa. The odd change point-like behavior is smoothed out, and global growth appears to trend upwards for most countries. Additional figures in the Appendix show the factors over larger penalty strengths.

Table 3.1 compares the NMF model with and without penalties. We find that adding smoothness penalties causes a minor loss in reconstruction accuracy. The only large drop was for natural gas, which may indicate that factors were over-smoothed. Nonetheless, it appears that the benefits of smoothing outweigh the small average loss in reconstruction accuracy.

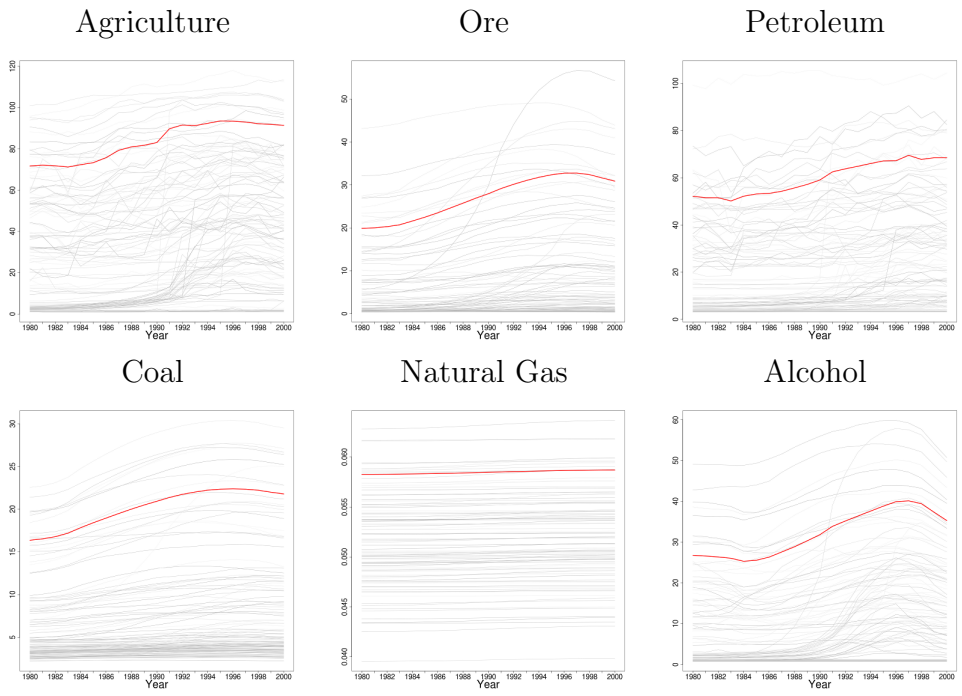



Figure 3.5: Smooth expressions (sum of $V_t$ components) learned from the data arrays. Each grey line represents a country, the bold line shows the mean of the top 20 countries.

Lastly, we note that growth curves are commonly derived from multidimensional time-series to facilitate decision making in many applications in a diverse set of scientific fields (*Pan et al.*, 2002). For instance, logistic curves have been used to model cellular growth rates (*Airoldi et al.*, 2009) and specie population levels (*Fath et al.*, 2004). Thus, the smoothing encouraged with the proposed model should be useful for analyzing data collected in many areas.

### 3.5.2 arXiv Citations

Citation networks, composed of references (edges) between documents (nodes), have a long history of study in bibliometrics, going back to *de Solla Price* (1965), where it was posited that in the world of research "success breeds success", that is, a popular research article is more likely to be referenced than less cited ones. Accordingly, citation networks have been shown to feature heavy-tailed degree dis-

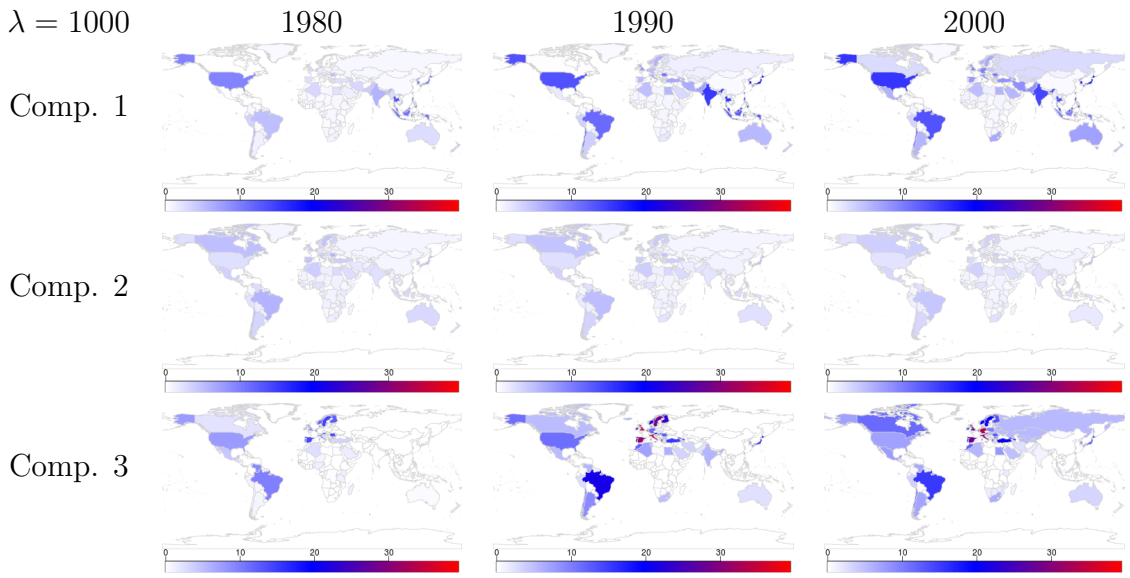



Figure 3.6: Time-varying expression vectors ($V_t$) learned from the Coal data array, with $h_m = 2$ years and $\lambda = 1000$. Three years of estimates are shown instead of all years due to space constraints.

tributions (*Clauset et al.*, 2009). Other works, such as *Girvan and Newman* (2002b), find grouping structure that correspond to research topic and methodology. Yet, these and most other empirical studies of citation networks treat the graph object as static, even though network structure can evolve over time as documents are created and content focus shifts.

Recently, as data collection technologies have improved, researchers have begun to investigate citation graphs as dynamic objects. *Leskovec et al.* (2005) find that growth patterns in citation networks feature some surprising empirical characteristics. In particular, the number of references grows faster, becomes more dense, and exhibits a shrinking diameter over time – all empirical patterns that challenge the dynamical assumptions of preferential attachment models (*Barabsi and Albert*, 1999; *Newman et al.*, 2006). *Leicht et al.* (2007) find that the relevance of document communities rise and fall as content focus and semantics shift within a corpus of US Supreme Court opinions. As such, a remaining and important goal is to characterize the time-varying complexity of citation networks in terms of the number and different types

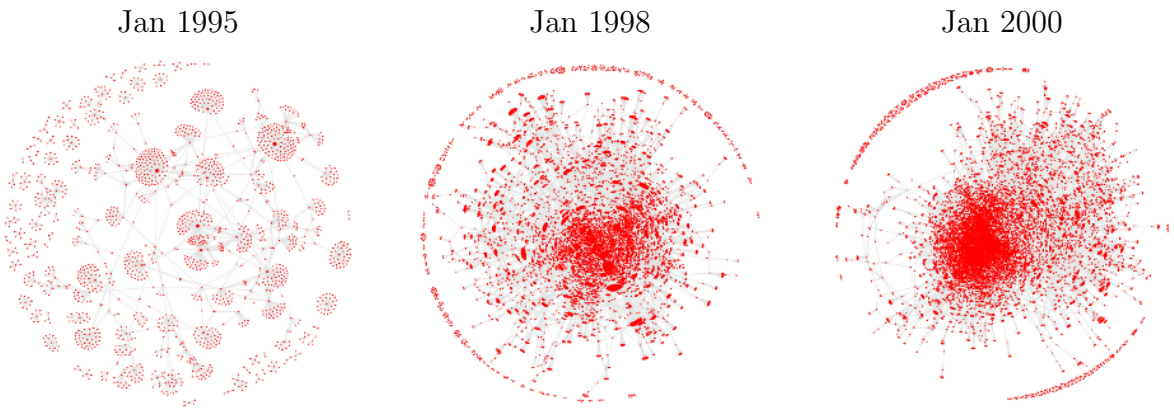



Figure 3.7: Graph layouts of the raw data at three different time points. Due to the size of the networks, it quickly becomes difficult to discern paper (node) properties.

of evolutions that papers follow in such data.

We investigate a sequence of monthly citation networks from October 1993 to December 2002 for the e-print service arXiv. We find that existing techniques are better suited for characterizing global changes to network topology and inadequate for uncovering and representing paper dynamics. The proposed regularized non-negative matrix factorization is utilized that captures the different life-cycles of research articles through interpretable visual representations. For example, we discover the rapid and sustained rise in popularity of fundamental papers, as well as the more common dynamics of lower impact articles. Further, the results allow us to infer the growth of an important research topic in theoretical physics, followed by a shift to other problems as the first topic matures.

The citation networks we analyze are from the e-print service arXiv for the 'high energy physics theory' section. The data covers papers from October 1993 to December 2002, and was originally provided as part of the 2003 KDD Cup (*Gehrke et al.*, 2003).

The data is organized into monthly networks. In particular, if paper $i$ cites paper $j$, then the graph contains a directed edge from $i$ to $j$. Citations to or from papers

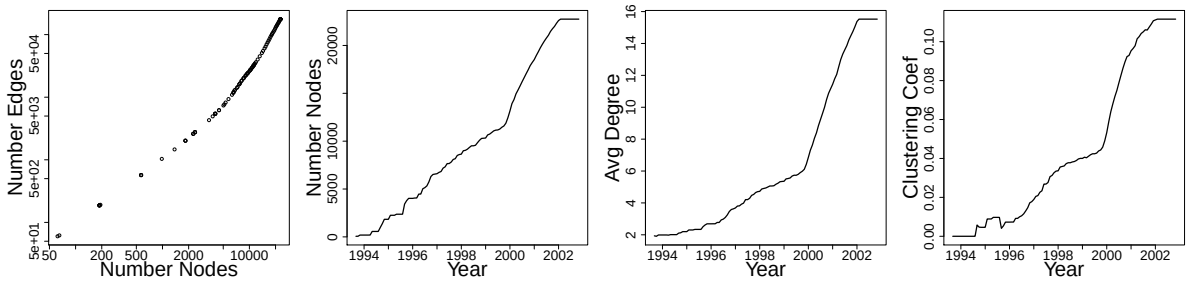



Figure 3.8: The kink near near Jan 2000 indicates sudden, rapid growth. The network statistics also indicate the average length of bibliographies increased over time (*Leskovec et al.*, 2005). The top-left plot is on a log-log scale.

outside the dataset are not included. Edges are aggregated, that is, the graph for a given month will contain all edges from the beginning of the data up to, and including, the current month. Altogether, there are 22750 papers (nodes) with 176602 edges over 112 months. Graph layouts are shown in Figure 3.7, where we see that even when considering a single time point, it quickly becomes difficult to discern paper (node) properties due to the large network size. Thus, the data requires network statistics and other analytical tools to extract structure and infer dynamics in the network sequence.

Figure 3.8 shows a noticeable kink in the network statistics around the year 2000, after which the network grew faster. This pattern is commonly attributed to papers that reference other works before the start of the observation period (see *Leskovec et al.* (2005)). As we move away from the beginning of the data, papers primarily reference other papers belonging to the data set. To better understand the dynamics underlying the citation networks, in the next section we discuss and apply several popular methodologies.

We discuss next popular approaches to extracting structure and inferring dynamics within a sequence of networks. Specifically, we address (i) animated network drawings, (ii) network statistic time-series, and (iii) a likelihood-based approach to extracting communities. We note that the two latter approaches are applied to a citation network

of US Supreme Court opinions in *Leicht et al.* (2007).

#### 3.5.2.1 Animated Layouts

There are two main extensions of the graph drawing techniques that were used to create Figure 3.7. The first and most popular extension relies on animation. The alternative is to view all time periods simultaneously using a matrix of images. With either technique, the main challenge is to preserve the overall shape and attributes of the network, so that nodes are moved as little as possible between time steps to facilitate readability.

However the data features 112 time points and is challenging for either approach if the user is interested in detailed analysis. *Ghani et al.* (2012) find that the effectiveness of animation is strongly predicted by node speed and target separation. Thus, there exists a bottleneck stemming from the analyst's cognitive load, as the analyst must remember patterns over a large time span or time points must be traversed quickly increasing node speed. Screen space acts as a bottleneck with a matrix of images. A pure visualization approach is unsuitable for the data, since it contains both a large number of nodes and time points.

#### 3.5.2.2 Connectivity Scores

Uncovering community structure by maximizing the modularity function is a widely adopted method (*Newman*, 2006b). The idea behind the modularity function is to measure, when given group assignments, whether a larger than expected number of edges exist within each group. In practice, maximizing the modularity over all possible partitions is NP hard. Approximate solutions are obtained by first using the leading eigenvector of the so-called modularity matrix to split the network into two, and then repeatedly dividing the network in two until the modularity decreases.

Following the analysis in *Leicht et al.* (2007), I partition the fully formed citation

network when $t = 112$ using the approximate modularity solution described above. The optimal number of groups is over two hundred. However, there are only four meaningful groups of papers, as the other groups contain only a handful of papers. The left panel of Figure 3.9 shows the degree of each paper over time, colored by the modularity group assignment. From the plot I can see a number of possible different time profiles, none of which are clearly captured in the modularity groupings. This finding is in contrast to the investigation in *Leicht et al.* (2007), which used different citation network data. The modularity approach finds groups of papers that are specifically linked together by edges, e.g., the temporal profile of each paper is not utilized, so that the groups are interpretable from a static connectivity point of view only.

I now compute the authority measure proposed by *Kleinberg* (1999), which derives a measure of importance from considering incoming edges only. Utilizing edge direction is useful in citation networks, since a paper that is cited by many other important papers is likely to be an authoritative one. The measure is computed with the leading eigenvector of $X^T X$, where $X$ is the asymmetric adjacency matrix corresponding to the directed citation network. The right panel of Figure 3.9 shows the average age of the most influential papers. Consistent with the other results, there is a drop in the average age around the year 2000 indicating perhaps a shift in current research topics. Similar to the modularity clustering, the temporal aspect of the data is not utilized when computing the authority scores. As a result, there are artificial numerical fluctuations in the authority scores before the drop.

#### 3.5.2.3 Mixture Model

I apply the mixture model in *Leicht et al.* (2007) to extract groups of papers according to their common temporal citation profiles. To briefly summarize, the model consists of two main sets of parameters. First, a set of time profiles for each

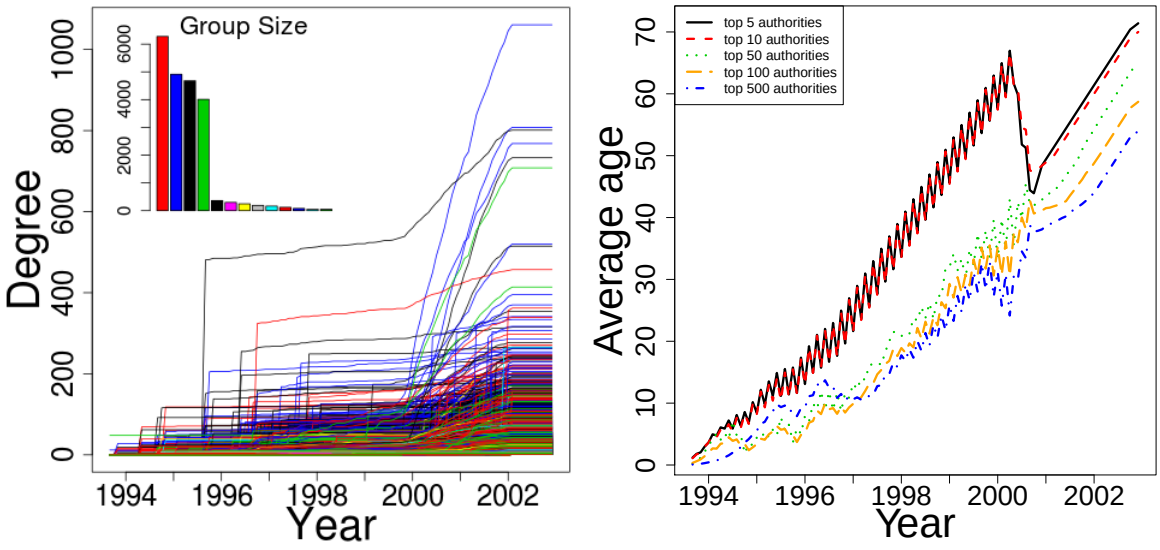



Figure 3.9: The left panel shows the degree of each node over all time points, colored by modularity groupings. The groupings are not interpretable in terms of the time-profile of each paper. The right panel shows the average age in months of the top authority paper over time.

group that represents the probability that a citation received in a given group is made during time $t$. The second parameter set consists of the probabilities that a randomly chosen document belongs to each group. Following standard derivations, an EM algorithm is utilized for estimation.

Figure 3.10 shows the time-profiles over different numbers of groups. Comparing against the degree plot in Figure 3.9, the temporal profiles are reasonable. One can clearly see at least two groups, one that grows slowly from the beginning of the observational period and another that experiences rapid growth starting just before the year 2000.

#### 3.5.2.4 A Unifying Framework for Node-level Analysis

The results above show that the authority scores of *Kleinberg* (1999) and the mixture model of *Leicht et al.* (2007) are useful. Combined, they can in principle be used to identify important papers, as well as characterize the data in terms of the

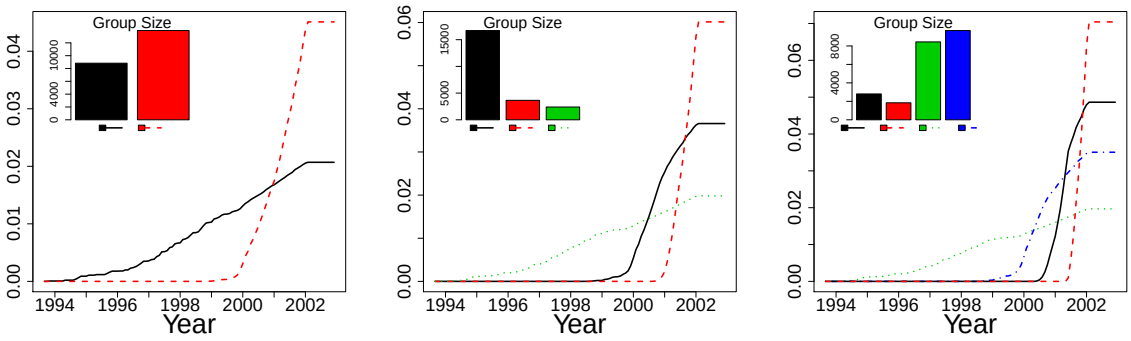



Figure 3.10: Time-profiles for each group based on the mixture model of *Leicht et al.* (2007).

number and types of different groups in the data.

However, the authority scores were computed separately at each time point creating irregularities in the time-series. Further, it is difficult to systematically combine them with the mixture results, which is required in order to identify particular papers and their interesting trajectories.

The proposed non-negative matrix factorization combines both approaches in a principled fashion by discovering groups of papers according to their trajectories in the data, while also providing time-series that are closely related to authority scores for each paper. Specifically, in a network setting, the $U$ vectors score nodes by their "interestingness", or distance from the average outgoing connectivity. The $V_t$ vectors yield similar scores based on incoming connections. Together, $U$ and $V_t$ are useful for highlighting nodes by their importance to connectivity. Moreover, due to the non-negativity constraint, it is straightforward to interpret the estimates. For instance, $\sum_k (V_t)_{kj}$ measures the total authority of paper $j$, and $(U)_{ik}(V_t)_{kj}$ can be interpreted as the contribution of the $k$th cluster to the edge $(X_t)_{ij}$. Thus, I characterize the time-varying complexity of the data in terms of the number and different types of authority evolutions that papers follow in such data.

I choose the parameters by following heuristics of varying parameters over a grid

of values and comparing results. Additional details and figures are given in the Appendix that show factors over different parameterizations. The results presented here use low rank embeddings and small smoothness penalties to obtain interpretable visual representations of the evolving structure within high energy physics theory.

Figure 3.11 shows the $V_t$ estimates using a series of two-dimensional approximations ($K = 2$). These time-varying factors are used to display the evolving impact of scientific articles. The paper trajectories are smoothed effectively and the important ones are highlighted by employing the penalty. Each component corresponds to a separate group in the data. With the exception of the highly popular outlier, the first component contains papers that mostly peak in their popularity by 1998. In other words, citations to these papers slowed dramatically around 1998, while research focus shifted to other topics and articles. The outlier continued to be cited throughout the data. The second component captures papers from 1998 onwards. Similarly, a small number of articles achieve massive impact.

The top papers from both $V_t$ components are identified in Tables 3.2 and 3.3. Most of these articles are about an extension of string theory called M-theory, which was first proposed in 1995 and led to new research in theoretical physics. It appears from the degree and citation counts that these papers were central to the develop of the theory. Notably, Witten is credited with naming M-theory, and appears often in the tables.

The results above show that existing methods discover similar patterns. However, authority scores do not utilize any temporal information creating difficulties in interpretation. The mixture approach does not provide a systematic way to identify important papers and trajectories that led each group. The proposed NMF is able to bridge the gap, by grouping papers according to their trajectories, while also providing temporal curves for each paper that are closely related to authority scores.

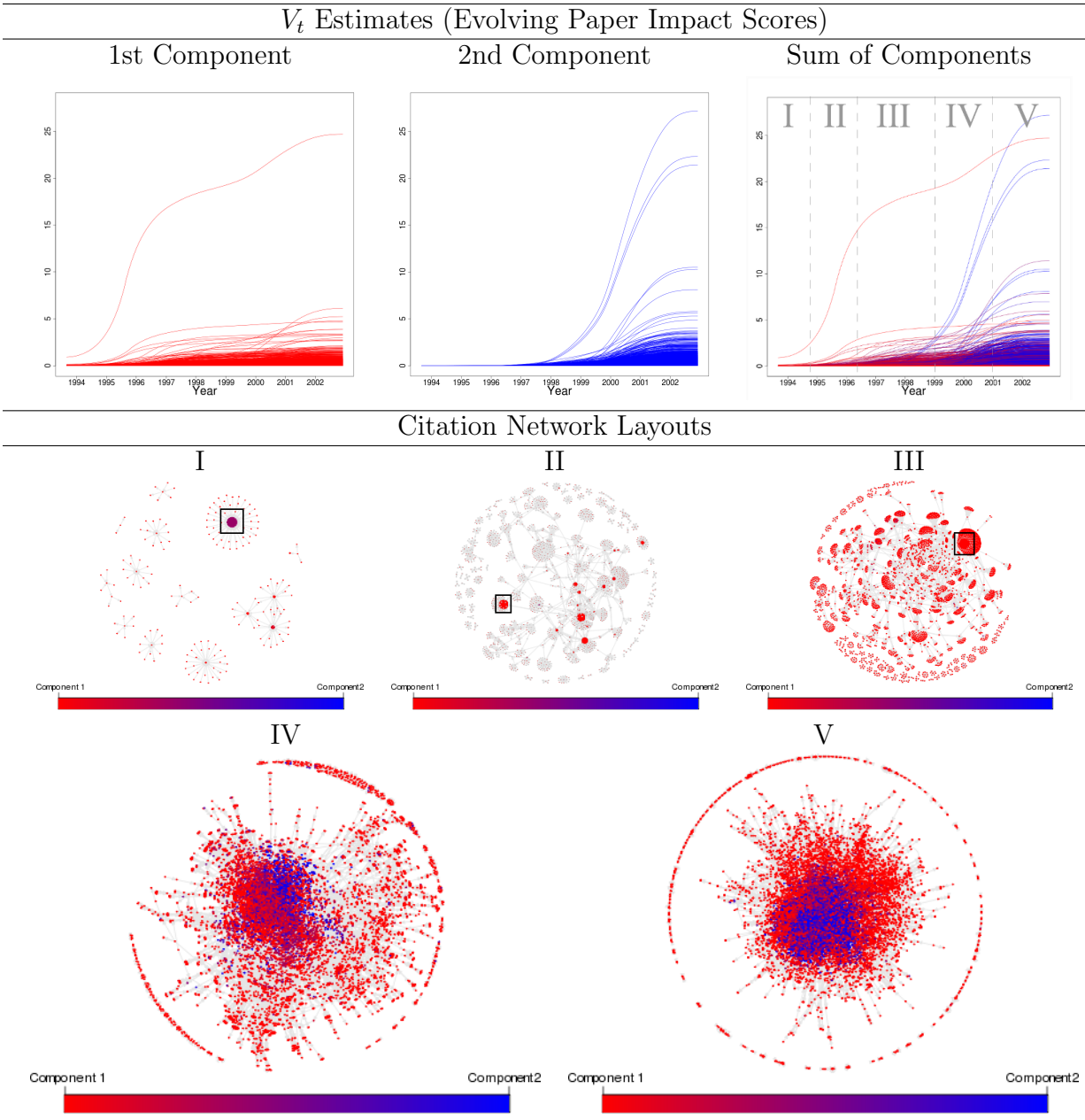



Figure 3.11: The top panel shows estimates of $V_t$ for the arXiv data with $\lambda = 4, h_m =$ 3 months. Each line corresponds to a paper (node) in the data. The bottom panel shows graph layouts of the raw data colored by the relative contribution of $V_t$ components with the node size proportional to the sum of components. The dominant paper in the first component is identified with a rectangle in the graph layouts in periods I,II,III.

| Title | Authors | In-Degree | Out-Degree | # citations (Google) |
|---|---|---|---|---|
| Heterotic and Type I String Dynamics from Eleven Dimensions | Horava and Witten | 783 | 18 | 2265 |
| Five-branes And $M$-Theory On An Orbifold | Witten | 169 | 15 | 249 |
| D-Branes and Topological Field Theories | Bershadsky, et. al | 271 | 15 | 457 |
| Lectures on Superstring and M Theory Dualities | Schwarz | 274 | 68 | 483 |
| Type IIB Superstrings, BPS Monopoles, And Three-Dimensional Gauge Dynamics | Hanany and Witten | 437 | 20 | 809 |

Table 3.2: The top 5 papers from the first component. # citations counts all references to the work, including by works outside of the data. These counts are obtained by Google.

| Title | Authors | In-Degree | Out-Degree | # citations (Google) |
|---|---|---|---|---|
| The Large N Limit of Superconformal Field Theories and Supergravity | Maldacena | 1059 | 2 | 9928 |
| Anti De Sitter Space And Holography | Witten | 766 | 2 | 6467 |
| Gauge Theory Correlators from Non-Critical String Theory | Klebanov and Polyakov | 708 | 0 | 5592 |
| Large N Field Theories, String Theory and Gravity | Aharony, et. al | 446 | 74 | 3131 |
| String Theory and Noncommutative Geometry | Seiberg and Witten | 796 | 12 | 3624 |

Table 3.3: The top 5 papers from the second component.

| Data | Rows | Columns | Time Points | Runtime (server) | Runtime (netbook) |
|---|---|---|---|---|---|
| Illustrative Example | 100 | 100 | 100 | 0.47 | 3.09 |
| | 1000 | 100 | 100 | 1.27 | 8.85 |
| | 10000 | 100 | 100 | 3.78 | 34.63 |
| | 100 | 1000 | 100 | 6.80 | 36.35 |
| | 100 | 10000 | 100 | 73.60 | 423.95 |
| | 100 | 100 | 1000 | 6.62 | 36.86 |
| | 100 | 100 | 10000 | 76.27 | 440.22 |
| arXiv | 22750 | 22750 | 112 | 35.67 | 240.10 |
| World Trade | 212 | 212 | 21 | 2.00 | 5.70 |

Table 3.4: Runtimes (seconds) in MATLAB for rank 1 NMFs from a University of Michigan dedicated computing server and a Linux netbook with 4GB Ram and 1.7 GHz AMD Athlon Neo K125 Processor. Mild temporal smoothing via the triangular kernel is utilized, with no group penalty.

## 3.6 Conclusion

Scalability (tens of thousands of rows, columns, and depth) is a highly desirable property. Since parameters have been through a grid search, the reader may wonder about the computational costs. As shown in Table 3.4, the factorizations are computed in a reasonable amount of time even on a modestly endowed computer. Further, the model achieves smoothness with a relatively small amount of penalization. As a consequence, the estimation algorithms are efficient.

An interesting application area not investigated in this work appears to be heaviest element searches. *Li et al.* (2011) propose a tensor-based framework for directed graphs called 'recurrent heavy subgraph' search. The goal is to identify the most common, largest weighted communities in biological networks observed over different experimental conditions. However the tensor-based framework requires solving a non-convex, continuous optimization as an approximate solution to an intractable discrete problem. The proposed approach should be easier to implement and computationally less demanding. Moreover, the proposed model and algorithm provides a systematic way of integrating information across data to find the 'heaviest elements'.

There are several questions to be investigated in future work. First, the approximation rank, penalty strength and scale $(K, \lambda, h_m)$ can be chosen by cross validation for smaller/moderately sized data sets. However, this is not satisfactory for large data sets, due to its computational cost. Theoretical guidelines for choosing the penalty, especially in large sized problems, remain to be developed. Second, I present a multiplicative updating, which is very similar to the original algorithm proposed by *Lee and Seung* (2001). Another popular and more flexible class of NMF algorithm is the alternating non-negative least squares algorithm (ANLS). A constrained solution to the classical Sylvester's equation is required to implement an ANLS algorithm with the proposed regularization framework. Such a solution seems a useful area of future work, but technically challenging.

## 3.7 Appendix

### 3.7.1 Multiplicative Updating

We establish the following Theorem III.1 by following an argument similar to the ones made by *Lee and Seung* (1999, 2001).

Concatenate $\{X_m\}$ as

$$X = [X_1, X_2, ..., X_M]. \tag{3.29}$$

Write $\{V_m\}$ in a similar manner.

The proposed objective function can then be written as

$$\mathcal{O} = ||X - UV^T||_F^2 + Tr(V^T \mathcal{L} V). \tag{3.30}$$

The argument makes use of an auxiliary function. We begin with the definition of auxiliary function.

**Definition III.2.** G(h,h') is an auxiliary function for F(h) if the conditions

$$G(h, h') \geq F(h), G(h, h) = F(h) \tag{3.31}$$

are satisfied.

**Lemma III.3.** *If G is an auxiliary function, then F is nonincreasing under the update*

$$h^{t+1} = argmin_h G(h, h^t) \tag{3.32}$$

*Proof.*

$$F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t) \tag{3.33}$$

□

Since the second term of the objective function is only related to $V$, we have exactly the same update formula for $U$ as in the original NMF. Thus, we can use the standard convergence proof of NMF for the $U$ update step. Now we will show that the update rules presented in the main text are exactly the update in (3.32) with a proper auxiliary function.

Considering any element $V_{ab}$ in $V$, we use $F_{ab}$ to denote the part of the objective function, which is only relevant to $V_{ab}$. It is easy to check that

$$F'_{ab} = (\frac{\partial \mathcal{O}}{\partial V})_{ab} = -2X^T U + 2VU^T U + 2\mathcal{L}V \tag{3.34}$$

$$F''_{ab} = 2U^T U + 2\mathcal{L}. \tag{3.35}$$

Since our update is essentially element wise, it is sufficient to show that each $F_{ab}$ is nonincreasing under the $V$ update step.

**Lemma III.4.** *Function $G(v, V_{ab}^{(i)})$ is an auxiliary function for $F_{ab}$, where*

$$G(v, V_{ab}^{(i)}) = F_{ab}(V_{ab}^{(i)}) + F'_{ab}(V_{ab}^{(i)})(v - V_{ab}^{(i)}) + \frac{(V_m U^T U)_{ab} + (DV_m)_{ab}}{V_{ab}^{(i)}}(v - V_{ab}^{(i)})^2. \tag{3.36}$$

*Proof.* Since $G(v, v) = F_{ab}(v)$ is obvious, we need only to show that $G(v, V_{ab}^{(i)}) \geq F_{ab}(v)$. We compare with the Taylor series expansion of $F_{ab}(v)$.

$$F_{ab}(v) = F_{ab}(V_{ab}^{(i)}) + F'_{ab}(V_{ab}^{(i)})(v - V_{ab}^{(i)}) + F''_{ab}\frac{(v - V_{ab}^{(i)})^2}{2} \tag{3.37}$$

Comparison with (3.36) shows that $G(v, V_{ab}^{(i)}) \geq F_{ab}(v)$ is equivalent to

$$\frac{(VU^T U)_{ab} + (DV)_{ab}}{V_{ab}^{(i)}} \geq (U^T U)_{bb} + (L)_{aa}. \tag{3.38}$$

We have

$$(VU^T U)_{ab} = \sum_{l=1}^{k} v_{al}^{(i)} (U^T U)_{lb} \geq V_{ab}^{(i)} (U^T U)_{bb} \tag{3.39}$$

which implies that

$$(VU^T U)_{ab} \geq (U^T U)_{bb}. \tag{3.40}$$

Then

$$(DV)_{ab} = \sum_{j=1}^{n} D_{aj} v_{jb}^{(i)} \geq D_{aa} V_{ab}^{(i)} \tag{3.41}$$

$$\geq (D - W)_{aa} V_{ab}^{(i)} = L_{aa} V_{ab}^{(i)} \tag{3.42}$$

Thus, we have shown that $G(v, V_{ab}^{(i)}) \geq F_{ab}(v)$. □

We can now demonstrate the convergence of Theorem III.1:

*Proof of Theorem III.1.* Replacing $G(v, V_{ab}^{(i)})$ in (3.32) by (3.36) results in the update

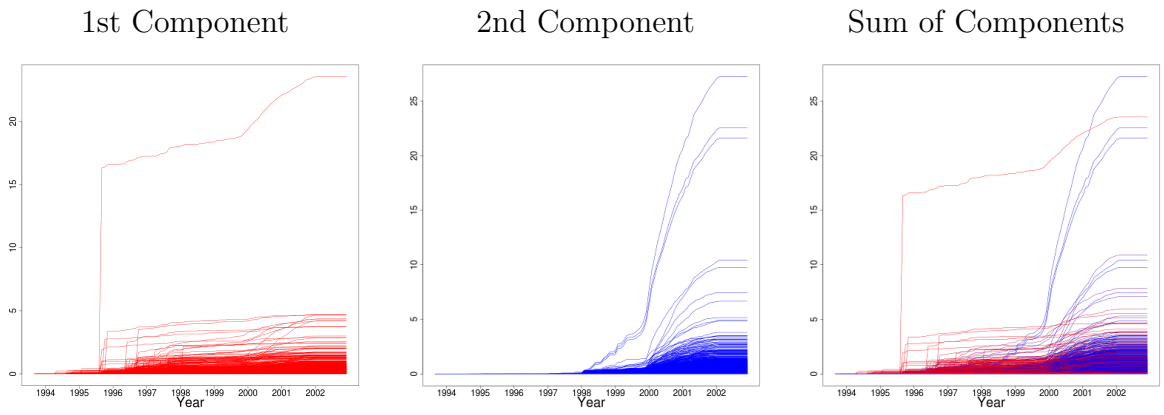



Figure 3.12: Fitted values for $V_t$ for the arXiv data with no penalty. Each line corresponds to a paper (node) in the data.

rule:

$$V_{ab}^{(i+1)} = V_{ab}^{(i)} - V_{ab}^{(i)} \frac{F'_{ab}(V_{ab}^{(i)})}{2(VU^TU)_{ab} + 2\lambda(DV)_{ab}} \tag{3.43}$$

$$= V_{ab}^{(i)} \frac{(X^TU)_{ab} + (WV)_{ab}}{(VU^TU)_{ab} + (DV)_{ab}} \tag{3.44}$$

Since (3.36) is an auxiliary function, $F_{ab}$ is nonincreasing under this update rule. □

### 3.7.2 arXiv Citations

Figure 3.12 shows the expression values ($V_t$) with no penalty framework. The overall pattern is the same as the smoothed components presented in the main text. The smooth curves help facilitate information extraction and is visually more satisfactory.

Figure 3.13 shows the expression values ($V_t$) with a bandwidth that is too small. The components appear numerically unstable, with local wiggles. Moreover, the first component features values that are minuscule.

### 3.7.3 World Trade Data

Figure 3.14 shows the reconstruction error by rank. As expected, adding additional components yields more accurate approximations. As mentioned in the main

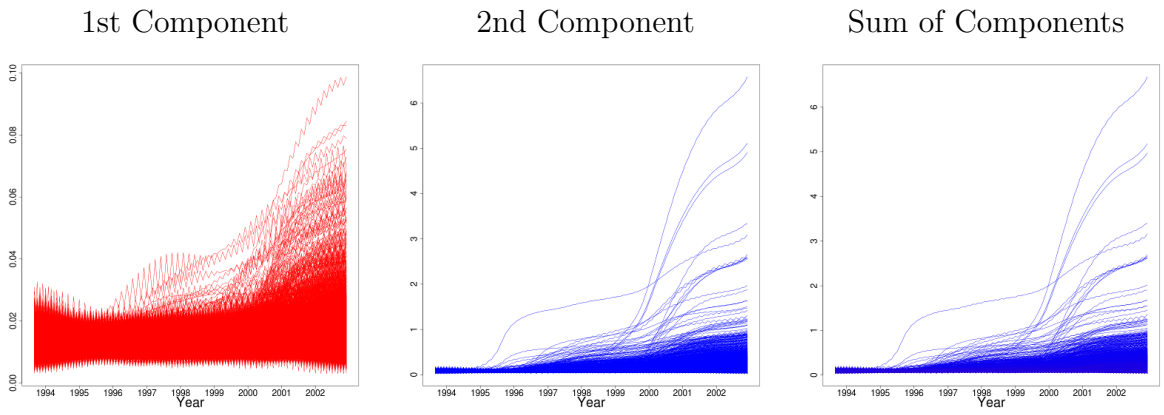



Figure 3.13: Fitted values for $V_t$ for the arXiv data with $h_M = 2$ months instead of three (as presented in the main text).

text, a small number of components yield a very accurate reconstruction of the natural gas data. Four to five components yields a nearly perfect reconstruction. Hence, the choice of five components from cross validation may be on the high side.

Figure 3.15 shows the time-varying expressions ($V_t$) with no penalties. The factors show some unrealistic patterns. For example, Russia appears only in year 2000, and India and Libya appear active only in 1990. The third component even indicates a persistent decreasing trend in trade levels throughout Europe and other parts of the world. These unwanted features are removed by employing the smoothness penalties, as shown in the main text.

Figure 3.16 shows the time-varying expressions ($V_t$) with a very large smoothness penalty. The factors become unstable, effectively collapsing into a single dimension. Further, the large smoothness penalty forces the factors to be constant and similar to the sum of basis vectors learned in $U$.

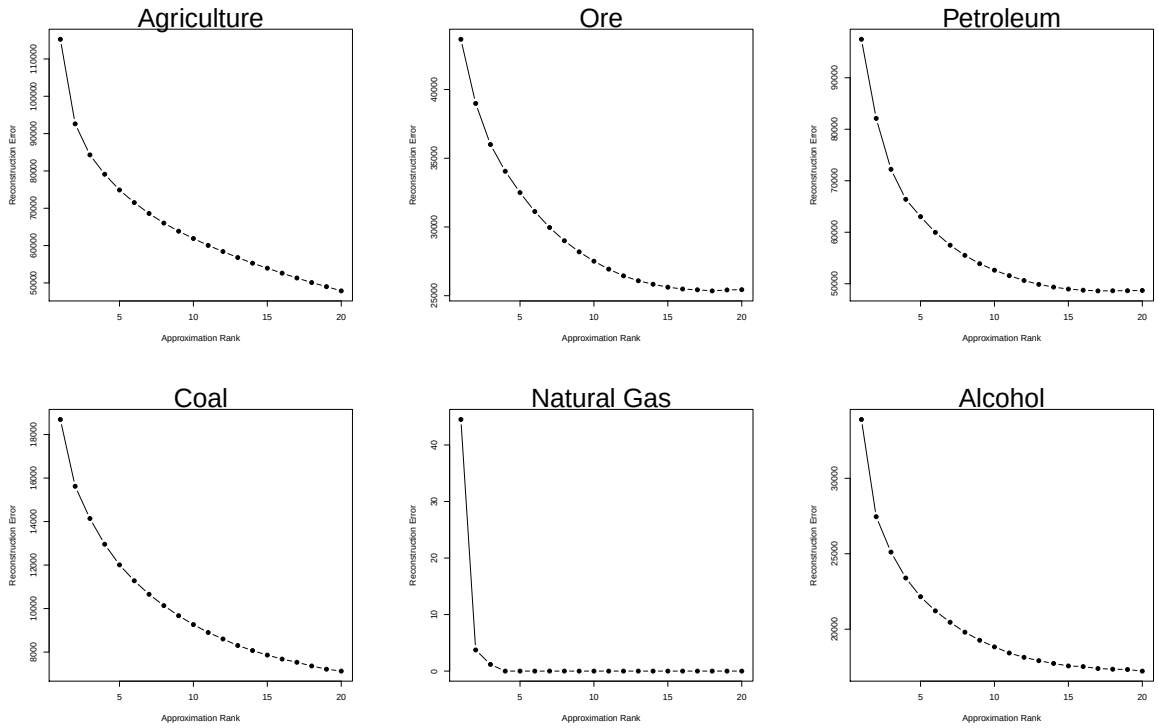



Figure 3.14: Average reconstruction errors as a function of rank.

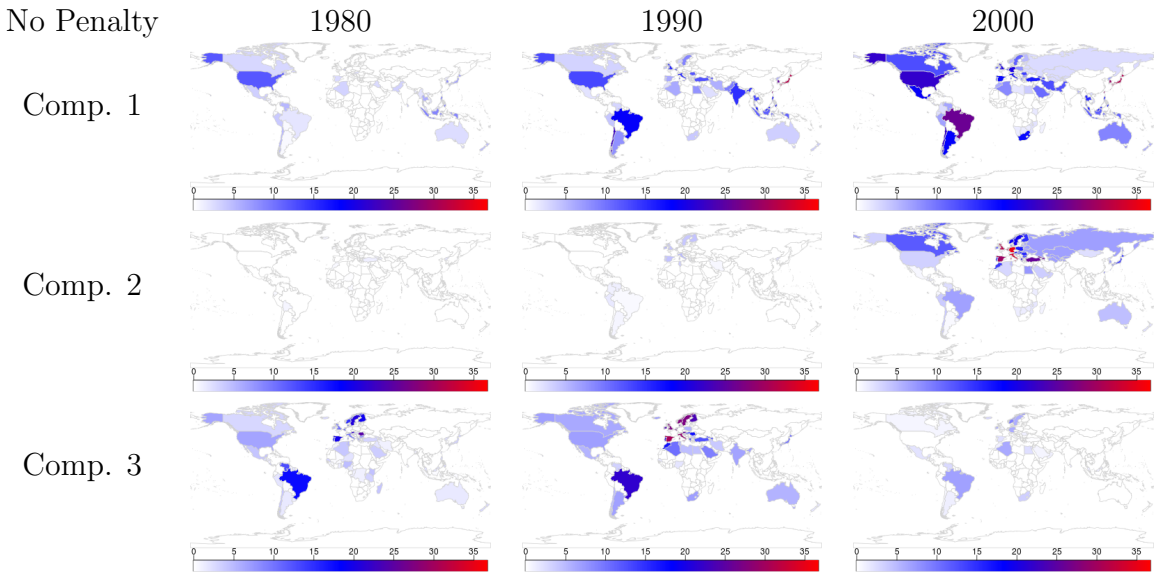



Figure 3.15: Time-varying expression vectors ($V_t$) learned from the Coal data array with no penalty.

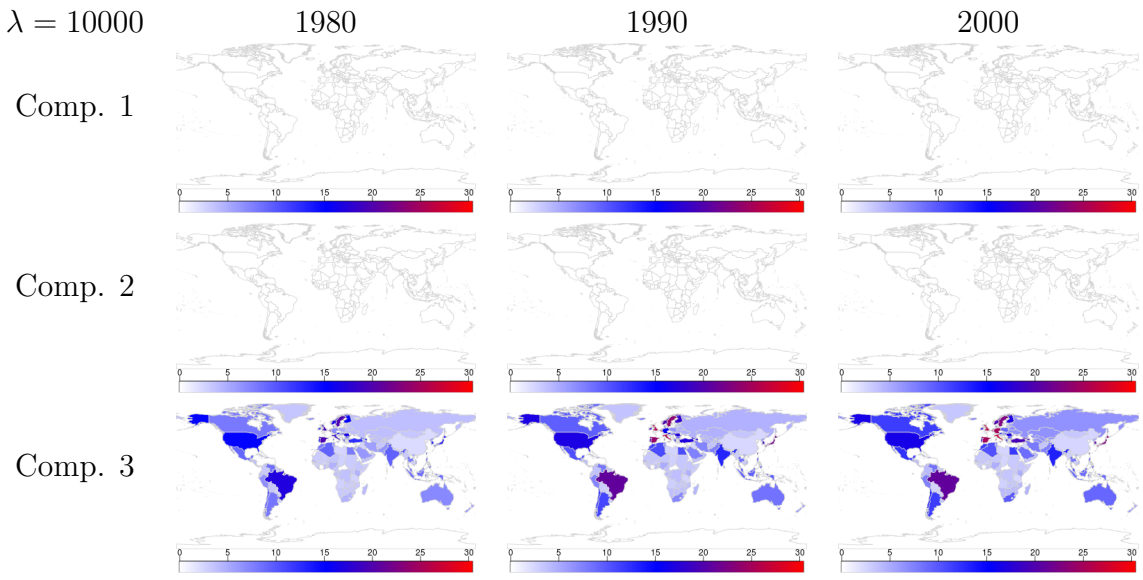



Figure 3.16: Time-varying expression vectors ($V_t$) learned from the Coal data array, with $h_M = 2$ months and $\lambda = 10000$.

# CHAPTER IV

# Structural and Functional Discovery in Dynamic Networks with Non-negative Matrix Factorization

## 4.1 Introduction

Due to advances in data collection technologies, it is becoming increasingly common to study time series of networks. An important research question is how to discover the underlying structure and dynamics in time-varying networked systems. In this chapter, I propose a new matrix factorization-based approach for community discovery and visual exploration within potentially weighted and directed network time-series. Next, I review and discuss this chapter in relation to popular approaches for analyzing a sequence of networks.

There have been many important contributions, extensively reviewed in *Fienberg* (2012); *Goldenberg et al.* (2009), from the fields of physics, computer science and statistics for community detection in network time-series. The basic goal of community detection is to extract groups of nodes that feature relatively dense within group connectivity and sparser between group connections (*Girvan and Newman*, 2002a; *Ball et al.*, 2011). A common strategy is to embed the graphs in low-dimensional latent spaces, then in a second stage extracting community structure from the latent spaces. For instance, *Leicht et al.* (2007) use latent variables to capture groups of

papers that evolve similarly in citation network data. *Sarkar and Moore* (2005) extend to the dynamic setting a popular latent space model for static data (*Hoff et al.*, 2002) by utilizing smoothness constraints to preserve the coordinates of the nodes in the latent space over time. This chapter also utilizes a similar low-dimensional embedding strategy. A key difference between this chapter and *Sarkar and Moore* (2005) is that community membership itself is subject to smoothness conditions in our approach, hence removing the need for a two stage procedure. This type of unified approach was first proposed in Facetnet (*Lin et al.*, 2008), a method based on a variant of non-negative matrix factorization (NMF) (*Lee and Seung*, 1999) for overlapping (soft) community detection in evolving and undirected graphs. In this chapter, I develop an alternative to Facetnet by proposing a different underlying NMF model and set of constraints for soft community detection. This chapter is also in contrast to previous works that use temporal smoothness constraints for non-overlapping (hard) community detection (*Sun et al.*, 2007), estimating time-varying network structure from covariate information (*Kolar et al.*, 2010), predicting network (link) structure (*Richard et al.*, 2012), or anomaly detection (*Asur et al.*, 2009; *Raginsky et al.*, 2012).

A sequence of non-negative factorizations discovers overlapping community structure, where node participation within each community is quantified and time-varying. Other works that consider a single network cross-section have shown advantages of NMF for community detection (*Psorakis et al.*, 2011; *Wang et al.*, 2011). In addition to a quantification of how strongly each node participates in each community, NMF does not suffer from the drawbacks of modularity optimization methods, such as the resolution limit (*Fortunato and Barthlemy*, 2007).

I also use the NMF to transform the time series of networks to a time-series for each node, which can be used to create an alternative to graph drawings for visualization of node dynamics. Much of the visualization literature aims to enhance static graph drawing methods with animations that move nodes (vertices) as little as possible

between time steps to facilitate readability (*Frishman and Tal*, 2008). However, the reliability of these methods rely on the human ability to perceive and remember changes (*Archambault et al.*, 2011). Moreover, experiments have discovered that the effectiveness of dynamic layouts are strongly predicted by node speed and target separation (*Ghani et al.*, 2012). Thus, dynamic graph drawings encounter difficulties when faced with a large number of time points, larger graphs that feature abrupt, non-smooth changes, or if the user is interested in detailed analysis, especially at the individual node level (see Section 3.2 of *von Landserber et al.* (2010), *von Landesberger et al.* (2011); *Yi et al.* (2010)). On the other hand, static displays facilitate detailed analysis and avoid difficulties associated with animated layouts. This highlights a main advantage our NMF model, namely creating static displays of node evolutions.

The remainder of this chapter is organized as follows: in the next section, I introduce a model for static network data in Section 4.2, followed by an extension for dynamic networks in Section 4.3. We then test the matrix factorization model on several synthetic and real-world data sets in Section 4.4. In Section 4.5, I close the chapter with a brief discussion.

## 4.2 NMF for Network Cross-sections

The most common factorization is the Singular Value Decomposition (SVD), which has important connections to community detection, graph drawing, and areas of statistics and signal processing (*Hastie et al.*, 2001). For instance in classical spectral layout, the coordinates of each node are given by the SVD of graph related matrices, and can be calculated efficiently using algorithms in *Koren* (2005); *Brandes et al.* (2006). Recently, there has been extensive interest in spectral clustering (*Rohe and Yu*, 2012; *Rohe et al.*, 2011; *Chung*, 1997), which aims to discover community structure in eigenvectors of the graph Laplacian matrix. The method proposed in this paper is similar in spirit, as it also relies on low rank approximations to adjacency

matrices (instead of Laplacian matrices). However, we search for low-rank approximations that satisfy different (relaxed) constraints than orthonormality, namely, that the approximating decompositions are composed of non-negative entries. Such factorizations, referred to as NMF, have been shown to be advantageous for visualization of non-negative data (*Lee and Seung*, 1999, 2001; *Paatero and Tapper*, 1994; *Devarajan*, 2008). Non-negativity is typically satisfied with networks, as edges commonly correspond to flows, capacity, or binary relationships, and hence are non-negative. NMF solutions do not have simple expressions in terms of eigenvectors. They can, however, be efficiently computed by formulating the problem as one of penalized optimization, and using modern gradient-descent algorithms. Recently, theoretical connections between NMF and important problems in data mining have been developed (*Ding et al.*, 2005, 2008), and accordingly, NMF has been proposed for overlapping community detection on static (*Psorakis et al.*, 2011; *Wang et al.*, 2011) and dynamic (*Lin et al.*, 2008) networks .

With NMF a given adjacency matrix is approximated with an outer product that is estimated through the following minimization

$$\min_{U\geq 0, V\geq 0} ||A - UV^T||_F^2, \tag{4.1}$$

where $A$ is the $n \times n$ adjacency matrix, and $U$ and $V$ are both $n \times K$ matrices with elements in $\mathbb{R}_+$. The rank or dimension of the approximation $K$ corresponds to the number of communities, and is chosen to obtain a good fit to the data while achieving interpretability. An interesting fact about NMF is that the estimates are always rescalable (scale invariant). For example, one can multiply $U$ by some constant $c$ and $V$ by $1/c$ to obtain different $U, V$ estimates without changing their product $UV^T$. Thus, as seen by the rotational indeterminancy and multiplicative nature of the factorization, NMF is an under-constrained model.

It is, however, straightforward to interpret the estimates due to non-negativity. For instance, $(U)_{ik}(V)_{jk}$ can be interpreted as the contribution of the $k$th cluster to the edge $(A)_{ij}$. In other words, the expected interaction $(\hat{A})_{ij} = \sum_{k=1}^{K}(U)_{ik}(V)_{jk}$ between nodes $i$ and $j$ is the result of their mutual participation in the same communities (*Psorakis et al.*, 2011). Such an edge decomposition can then be used to assign nodes to communities. For instance, one can proceed by first assigning all edges to the community with largest relative contribution. Then, nodes are assigned to communities according to the proportion of its edges that belong to each community. Note that with an NMF-based methodology, the adjacency matrix can be weighted (non-negatively), a potentially appealing feature since many existing analysis tools are arguably only compatible with networks of binary relations.

Though it is not explicitly controlled, standard NMF tends to estimate sparse components. Beyond the additional interpretability that sparsity provides, I find further motivation to encourage sparsity of the NMF estimate when working with networks. For instance, suppose $(A)_{ij}$=0 for some $i, j$, that is, there is an absense of an edge between nodes $i$ and $j$. In the low rank approximation there is no guarentee that $(\hat{A})_{ij} = 0$, though one expects it to be near zero. A straightforward way to force $(\hat{A})_{ij}$ *exactly* to zero is by anchoring $(U)_{ik} = (V)_{jk} = 0$ for all $k$, and estimating the remaining elements of $U$ and $V$ by the algorithm provided below (see *Buja et al.* (2008) for a similar strategy for multidimensional scaling). However, anchoring is not appropriate with repeated or sequential observations, as an edge can appear and disappear due to noise. Keeping in mind the extension to sequences of networks in the next section, I instead encourage sparsity in the form of an $l_1$ penalty.

The factorized matrices are obtained through minimizing an objective function that consists of a goodness of fit component and a roughness penalty

$$\min_{U\geq 0, V\geq 0} ||A - UV^T||_F^2 + \lambda_s \sum_{k=1}^{K} ||V_k||_1, \tag{4.2}$$

where the parameter $\lambda_s \geq 0$. The strength of the penalty is set by the user to steer the analysis, where a larger penalty encourages sparser $V$. Adding penalties to NMF is a common strategy, since they not only improve interpretability, but often improve numerical stability of the estimation by making the NMF optimization less under-constrained. *Berry et al.* (2006); *Chen and Cichocki* (2005); *Hoyer* (2002, 2004); *Cai et al.* (2011) and references therein review important penalized NMF models (see *Zou et al.* (2006); *Witten et al.* (2009); *Guo et al.* (2010) for similar approaches with SVD).

An advantage of an NMF-based approach is that it is easy to modify for particular datasets. For example, a similar $l_1$ penalty can be included on $U$ if the rowspace (typically out-going edges) are of interest.

The estimation algorithm I present is similar to the benchmark algorithm for NMF, known as 'multiplicative updating' (*Lee and Seung*, 1999, 2001). The algorithm can be viewed as an adaptive gradient descent. It is relatively simple to implement, but can converge slowly due to its linear convergence rate (*Chu et al.*, 2004). In practice I find that after a handful of iterations, the algorithm results in visually meaningful factorizations. The estimation algorithm for the penalized NMF in Eq. 4.2 is studied in *Hoyer* (2002) and *Hoyer* (2004), and the main derivation steps I present next follow these works.

First, to enforce the non-negativity constraints, consider the Lagrangian

$$\begin{aligned} L &= ||A - UV^T||_F^2 + \lambda_s \sum_{k=1}^{K} ||V_k||_1 \\ &+ Tr(\Phi U^T) + Tr(\Psi V^T), \end{aligned} \tag{4.3}$$

where $\Phi, \Psi$ are Lagrange multipliers.

To develop a modern gradient descent algorithm, I employ the following Karush-Kuhn-Tucker (KKT) optimality conditions, which provide necessary conditions for a local minimum (*Boyd and Vandenberghe*, 2004). The KKT optimality conditions are

obtained by setting $\frac{\partial L}{\partial U} = \frac{\partial L}{\partial V} = 0$.

$$\Phi = -2AV + 2UV^TV \tag{4.4}$$

$$\Psi = -2A^TU + 2VU^TU + 2\lambda_s. \tag{4.5}$$

Then, the KKT complimentary slackness conditions yield

$$0 = (-2AV + 2UV^TV)_{ij}(U)_{ij} \tag{4.6}$$

$$0 = (-2A^TU + 2VU^TU + 2\lambda_s)_{ij}(V)_{ij}, \tag{4.7}$$

which, after some algebraic manipulation, lead to the multiplicative update rules shown in Algo. IV.1. The algorithm has some notable theoretical properties. Specifically, each iteration of the algorithm will produce estimates that reduce the objective function value, e.g., the estimates improve at each iteration. Minor modifications provided in *Lin* (2007) can be employed to guarantee convergence to a stationary point.

**Algorithm IV.1** Sparse NMF

1: Set constant $\lambda_s$
2: Initialize $\{U, V\}$ as dense, positive random matrices
3: **repeat**
4: Set
$$(U)_{ij} \leftarrow (U)_{ij}\frac{(AV)_{ij}}{(UV^TV)_{ij}}$$
5: Set
$$(V)_{ij} \leftarrow (V)_{ij}\frac{(A^TU)_{ij}}{(VU^TU)_{ij} + \lambda_s}$$
6: **until** Convergence

Lastly, note that when the observed graph is undirected, due to symmetry of the adjacency matrix the factorization can be written as

$$A \approx U\Lambda U^T, \tag{4.8}$$

where $\Lambda$ is a non-negative diagonal matrix. This is the underlying model investigated in Facetnet (*Lin et al.*, 2008), with additional constraints on $U$ to satisfy an underlying probabilistic interpretation. The objective function considered in *Lin et al.* (2008) was also based on relative entropy or KL-divergence. I find that such symmetric NMF models are far more sensitive to additional constraints than its general counterpart, especially when dealing with sequences of networks as in the next section. Symmetric NMF has less flexibility, since additional constraints strongly influence the reconstruction accuracy of the estimation. On the other hand, without imposing symmetry, as $V$ changes, $U$ compensates (and vice versa) in order for the final product to reproduce the data as best as possible. Thus, for tasks of visualization of node evolution and community extraction in dynamic networks, I do not impose symmetry on the factorization.

### 4.2.1 Illustrative Examples

#### 4.2.1.1 Community Discovery on a Toy Example

I compare the following methods on a toy example shown in Fig. 4.1.

1. Leading eigenvector (modularity) based community discovery (*Newman*, 2006a)
2. Spectral clustering (*Rohe et al.*, 2011)
3. Clique percolation (*Palla et al.*, 2005) for overlapping community discovery
4. Classical NMF (Eq. 4.1)
5. Sparse NMF (Eq. 4.2)

The results of the alternative methodologies are provided in Fig. 4.2, where even on this toy example, there is disagreement in the recovered community structure. The leading eigenvector solution differs slightly from that of spectral clustering. Taken together, one may suspect a soft partitioning would result in overlap between the green

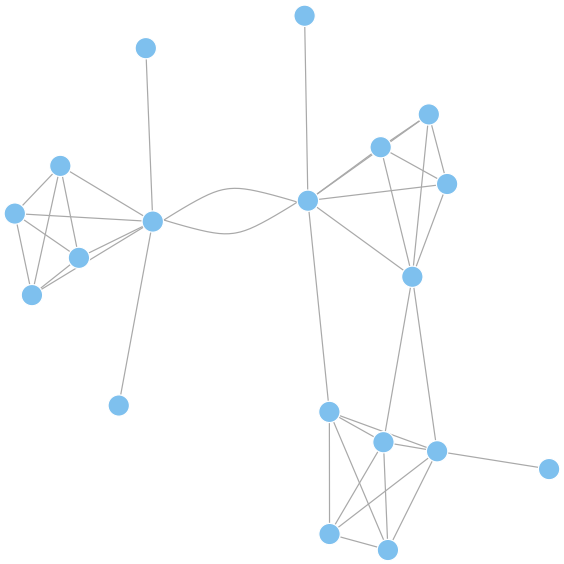


Figure 4.1: An undirected network with 19 nodes.

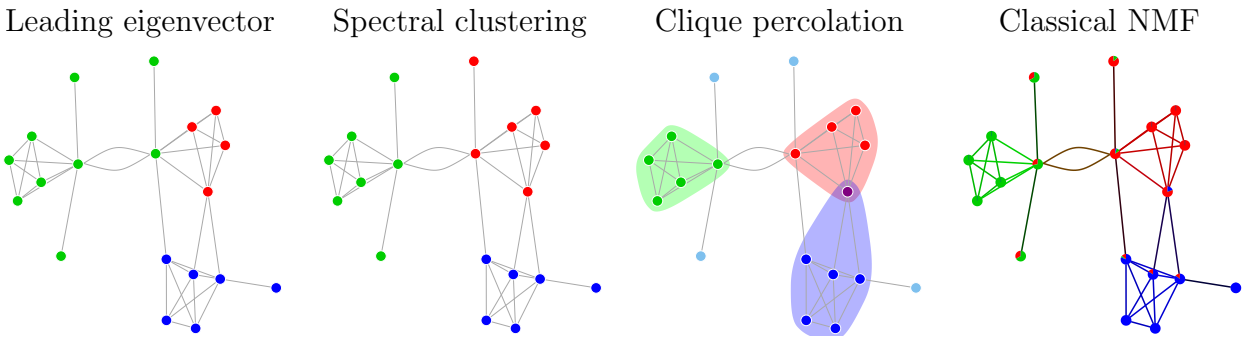



Figure 4.2: Results using alternative community discovery methods.

and red communities. Yet, clique percolation finds overlap between the blue and red communities. Classical NMF finds overlap between all three communities, quantifies the amount of overlap (denoted by the pie chart on each node), and decomposes each edge by community (colored as a mixture of red, green and blue). Fig. 4.3 shows that sparse NMF finds a cleaner structure compared to classical NMF. In particular, the sparse NMF solution has less overlap (mixing) between the three groups, while still quantifying community contribution to nodes and edges.

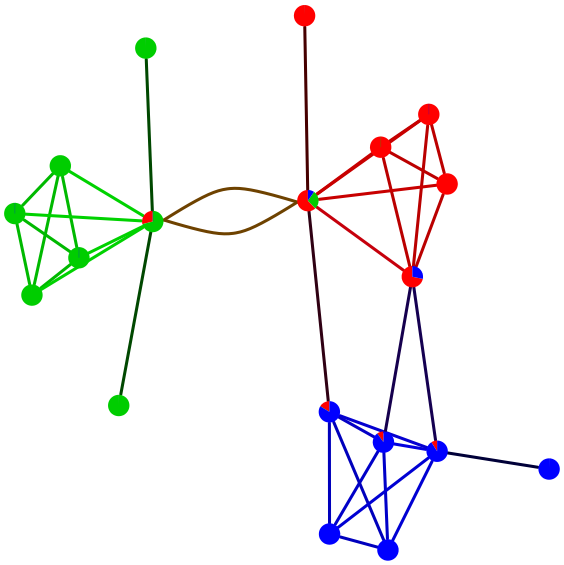


Figure 4.3: Results from applying sparse NMF (Algo. IV.1) with $\lambda_s = 5$. Nodes and edges are colored to denote the relative contribution of each community.

#### 4.2.1.2 Rank One Factorizations

I show in the experiments (Section 4.4) that a sequence of rank one matrix factorizations can be the basis for informative displays of time-varying node importance to connectivity. To provide some intuition as to why such a rank one factorization is informative, consider Fig. 4.4, which shows graph structures, corresponding NMFs, and Kleinberg's authority and hub scores (*Kleinberg*, 1999). Authority and hub scores are computed by the leading eigenvector of $A^T A$ and $AA^T$, respectively. Subject to rescaling of the NMF estimates, the results are identical. In fact, by the Perron-Frobenius theorem (see Chapter 8 of *Meyer* (2000)), the rank one NMF solution is always a rescaled version of authority and hub scores. This provides a natural interpretation for the rank one NMF. For instance, the $U$ vector on the Star Network highlights the hub node. The $V$ vector show that all peripheral nodes are equal in terms of their authority (incoming connections), and that the central node has no incoming connections. NMF vectors of the Ring Network show each node with an equal score for incoming (authority) and outgoing (hub) connectivity. The fact that

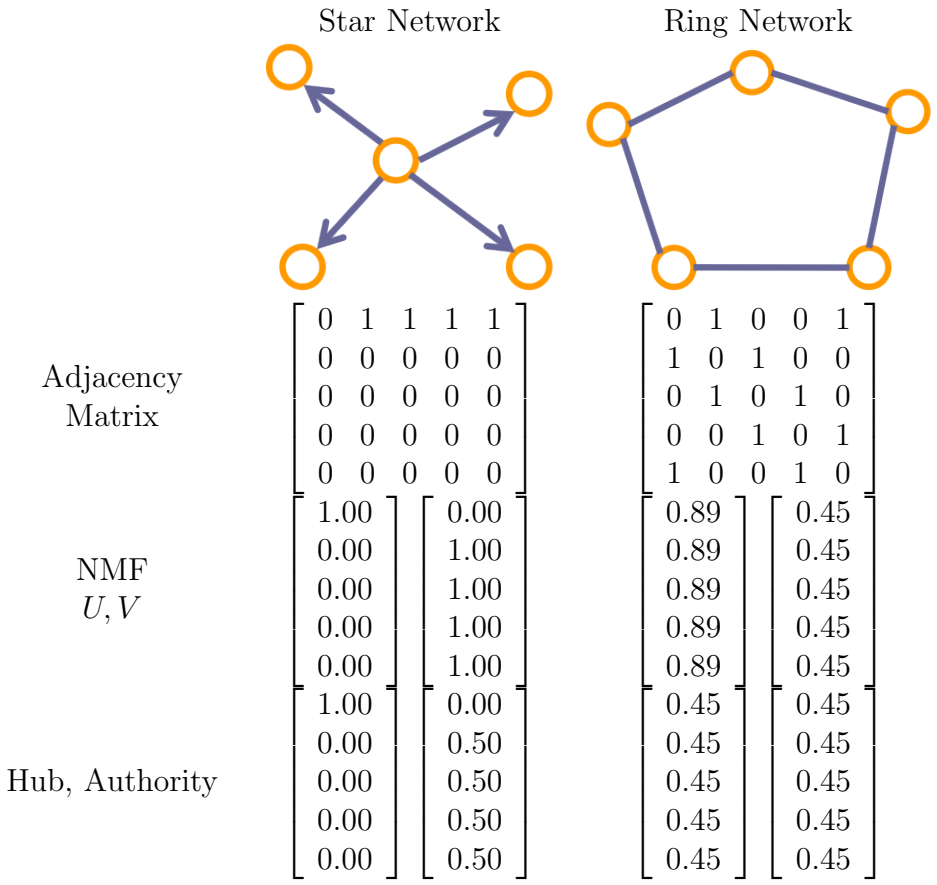



Figure 4.4: Rank 1 NMF without penalization and Kleinberg's authority/hub scores (*Kleinberg*, 1999).

$U$ contains larger elements than $V$ is arbitrary. However, the assignment of equal values within $U$ and $V$ shows each node is equally important to interconnectivity.

## 4.3 Model for Dynamic Networks

Given a time series of networks $\{\mathcal{G}_t = (V_t, E_t)\}_{t=1}^T$ with corresponding adjacency matrices $\{A_t\}_{t=1}^T$, the goal is to produce a sequence of low rank matrix factorizations $\{U_t, V_t\}_{t=1}^T$.

To extend the factorization from the previous section to the temporal setting, I impose a smoothness constraint on the basis $U_t$. This constraint forces new community structure to be similar to previous time points. Since individual node time-series given by $U_t$ are visually smooth, time plots for each node become informative and provide an alternative to graph drawings for visualizing node dynamics. Moreover, time plots are static displays, which facilitate detailed analysis and avoid difficulties associated with animated layouts when given a large number of time points or nodes.

The objective function becomes

$$\begin{aligned} \min_{\{U_t \geq 0, V_t \geq 0\}_{t=1}^T} \quad & \sum_{t=1}^{T} ||A_t - U_t V_t^T||_F^2 \\ & + \lambda_t \sum_{t=1}^{T} \sum_{\tilde{t}=t-\frac{W}{2}}^{t+\frac{W}{2}} ||U_t - U_{\tilde{t}}||_F^2 \\ & + \lambda_s \sum_{t=1}^{T} \sum_{k=1}^{K} ||V_{t,k}||_1, \end{aligned} \tag{4.9}$$

where $W$ is a small integer representing a time window. The parameters $\lambda_t, \lambda_s$ and $W$ are set by the user to steer the analysis.

The interpretations of $U_t, V_t$ extend naturally from the previous section, so that, for instance, $\sum_k (V_t)_{kj}$ measures the importance of node $j$ (typically corresponding to incoming edges), and $(U_t)_{ik}(V_t)_{jk} / \sum_{k=1}^K (U_t)_{ik}(V_t)_{jk}$ to measure the relative contribu-

tion of each community to each $i, j$ edge. In principle, the edge decomposition can be used to assign nodes to communities as discussed in the last section. However, this approach can be unsatisfactory due to unstable community assignments. As alternative method is to assign communities in terms of $U_t$, which ensures the stability of the community structure through time. Specifically, measuring the contribution of node $i$ to each community with the relative magnitude of the $i$th element of each dimension of $U_t$, e.g., $(U_t)_{ik} / \sum_{k=1}^{K} (U_t)_{ik}$.

I can follow similar steps as in the last section to derive a gradient descent estimation algorithm. First, to enforce the non-negativity constraints, consider the Lagrangian

$$\begin{aligned} L &= \sum_{t=1}^{T} ||A_t - U_t V_t^T||_F^2 \\ &+ \lambda_t \sum_{t=1}^{T} \sum_{\tilde{t}=t-\frac{W}{2}}^{t+\frac{W}{2}} ||U_t - U_{\tilde{t}}||_F^2 \\ &+ \lambda_s \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{K} |V_t(i,j)| \\ &+ \sum_{t=1}^{T} Tr(\Phi_t U_t^T) + \sum_{t=1}^{T} Tr(\Psi_t V_t^T), \end{aligned} \tag{4.10}$$

where $\Phi_t, \Psi_t$ are Lagrange multipliers.

The following KKT optimality conditions are obtained by setting $\frac{\partial L}{\partial U_t} = \frac{\partial L}{\partial V_t} = 0$.

$$\begin{aligned} \Phi_t &= -2A_t V_t + 2U_t V_t^T V_t - 2\lambda_t \sum_{\tilde{t}=t-\frac{W}{2}}^{t-1} U_{\tilde{t}} \\ &- 2\lambda_t \sum_{\tilde{t}=t+1}^{t+\frac{W}{2}} U_{\tilde{t}} + 2W\lambda_t U_t \end{aligned} \tag{4.11}$$

$$\Psi_t = -2A_t^T U_t + 2V_t U_t^T U_t + 2\lambda_s. \tag{4.12}$$

Then, the KKT complimentary slackness conditions yield

$$0 = (-2A_tV_t + 2U_tV_t^TV_t - 2\lambda_t \sum_{\tilde{t}=t-\frac{W}{2}}^{t-1} U_{\tilde{t}})_{ij}(U_t)_{ij} \tag{4.13}$$
$$+ (-2\lambda_t \sum_{\tilde{t}=t+1}^{t+\frac{W}{2}} U_{\tilde{t}} + 2W\lambda_t U_t)_{ij}(U_t)_{ij}$$
$$0 = (-2A_t^TU_t + 2V_tU_t^TU_t + 2\lambda_s)_{ij}(V_t)_{ij}, \tag{4.14}$$

which after some algebra leads to the algorithm provided in Algo. IV.2. The theoretical properties are also the same as in the previous section. Most notably, the estimates of $U_t$ and $V_t$ will improve at each iteration with respect to Eq. 4.9.

**Algorithm IV.2** NMF with temporal and sparsity penalties

1: Set constants $\lambda_t, \lambda_s, W$.
2: Initialize $\{U_t\}, \{V_t\}$ as dense, positive random matrices.
3: **repeat**
4: **for** t=1..T **do**
5: Set
$$(U_t)_{ij} \leftarrow (U_t)_{ij} \frac{(A_tV_t + \lambda_t \sum_{\tilde{t}=t-\frac{W}{2}}^{t-1} U_{\tilde{t}} + \lambda_t \sum_{\tilde{t}=t+1}^{t+\frac{W}{2}} U_{\tilde{t}})_{ij}}{(U_tV_t^TV_t + W\lambda_tU_t)_{ij}}.$$
6: Set
$$(V_t)_{ij} \leftarrow (V_t)_{ij} \frac{(A_t^TU_t)_{ij}}{(V_tU_t^TU_t)_{ij} + \lambda_s}.$$
7: **end for**
8: **until** Convergence

### 4.3.1 Parameter Selection

I briefly discuss the important practical matter of choosing $K$, the inner rank of the matrix factorization.

For the goal of clustering, the rank should be equal to the number of underlying groups. The rank can be ascertained by examining the accuracy of the reconstruction as a function of rank. However, this tends to rely on subjective judgments and overfit the given data. Cross validation based approaches are theoretically preferable and

follow the same intuition.

The idea behind cross validation is to use random subsets of the data from each data slice to fit the model, and another subset from each data slice to assess accuracy. Different values of $K$ are then cycled over and the one that corresponds to the lowest test error is chosen.

Due to the data structure, I employ two-dimensional cross validation. Two-dimensional refers to the selection of *submatrices* for our training and test data. Special care is taken to ensure that the same rows and columns are held out of every data slice, and the dimensions of the training and test sets are identical.

The hold out pattern divides the rows into $k$ groups, the columns into $l$ groups, then uses the corresponding $kl$ submatrices to fit and test the model. In each submatrix, the given row and column group identifies a held out submatrix that is used as test data, while the remaining cells are used for training. The algorithm is shown in Algo. IV.3. The notation in the algorithm uses $\mathcal{I}_l$ and $\mathcal{I}_J$ as index sets to identify submatrices in the each data matrix.

One can then cycle over different values of $K$ to choose the one that minimizes average test error. Fig. 4.5 shows that this procedure correctly identifies 3 communities for the toy example. Consistency results are developed in *Perry and Owen* (2009) to provide theoretical foundations for this approach.

In principle the cross validation procedure can be used to select the penalties $\lambda_t, \lambda_s$ and the time window $W$. However, considering the scale of many modern network datasets, this would require too much computing time. Instead I typically choose the penalties by hand to emphasize readability and interpretability of the results, keeping in mind that if either penalty is set too large then the estimation results in degenerate solutions. For instance, the algorithm suffers from numerical instabilities when $\lambda_s$ is too large, since all $V_t$ elements are zero. If $\lambda_t$ is set to an extremely large number, then $U_t$ will be approximately constant for all time periods, so the effective model is

**Algorithm IV.3** Cross-validation for choosing the number of communities (rank)

1: Form row holdout set: $\mathcal{I}_l \subset \{1, .., n\}$

2: Form column holdout set: $\mathcal{I}_J = \subset \{1, .., n\}$

3: Set

$$(\tilde{U}_t, \tilde{V}_t) = \underset{U_t, V_t \geq 0}{\arg\min} \sum_t ||(A_t)_{-\mathcal{I}_l, -\mathcal{I}_J} - U_t V_t^T||_F^2$$

4: Set

$$\breve{U}_t = \underset{U_t \geq 0}{\arg\min} \sum_t ||(A_t)_{\mathcal{I}_l, -\mathcal{I}_J} - U_t \tilde{V}_t^T||_F^2$$

5: Set

$$\breve{V}_t = \underset{V_t \geq 0}{\arg\min} \sum_t ||(A_t)_{-\mathcal{I}_l, \mathcal{I}_J} - \tilde{U}_t V_t^T||_F^2$$

6: Set

$$(\hat{A}_t)_{\mathcal{I}_l, \mathcal{I}_J} = \breve{U}_t \breve{V}_t^T$$

7: Compute Test error

$$\text{Test Error} = \sum_t ||(A_t)_{\mathcal{I}_l, \mathcal{I}_J} - (\hat{A}_t)_{\mathcal{I}_l, \mathcal{I}_J}||_F^2$$

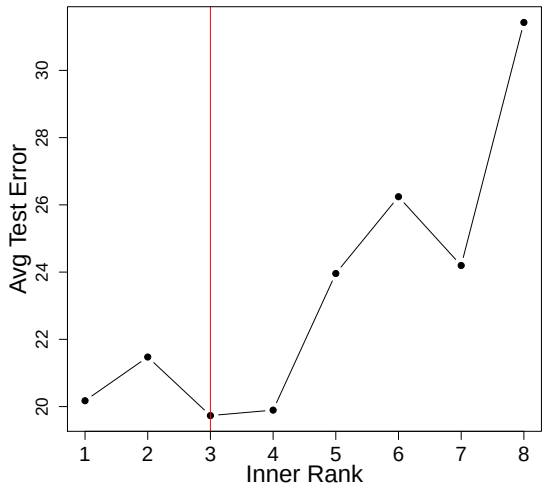



Figure 4.5: Cross validation indicates 3 communities (rank 3) features the lowest average test error for the toy example.

$A_t \approx UV_t^T$, e.g., the community structure is fixed for all observations.

The parameter, $W$, controls the number of neighboring time steps to locally average. Larger values of $W$ mean that the model has more memory so it incorporates more time points for estimation. This risks missing sharper changes in the data and only detecting the most persistent patterns. On the other hand, small values of $W$ make the fitting more sensitive to sharp changes, but increase short term fluctuations due to smaller number of observations. I set $W = 2$ (looking one time period ahead and before) for all presented experiments. Larger values could be used in very noisy settings to further smooth results.

## 4.4 Applications

In this section I test the model on both synthetic and real-world examples. The synthetic networks allow us to validate the model's ability to highlight known community structure and node evolution, while the real examples exhibit the model's performance under practical conditions.

### 4.4.1 Synthetic Networks

#### 4.4.1.1 Catalano Communication Network

The first example utilizes the Catalano social network, which was part of the Visual Analytics Science and Technology (VAST) 2008 challenge (vas, 2008). The synthetic data consists of 400 unique cell phone IDs over a ten day period. Altogether, there are 9834 phone records with the following fields: calling phone identifier, receiving phone identifier, date, time of day, call duration, and cell tower closest to the call origin. The purpose of the challenge was to characterize the social structure over time for a fictitious, controversial socio-political movement. In particular, the challenge requires identifying five key individuals that organize activities and com-

Figure 4.6: The cell phone network from a day using a force directed layout algorithm in igraph. Node 200 is colored black.

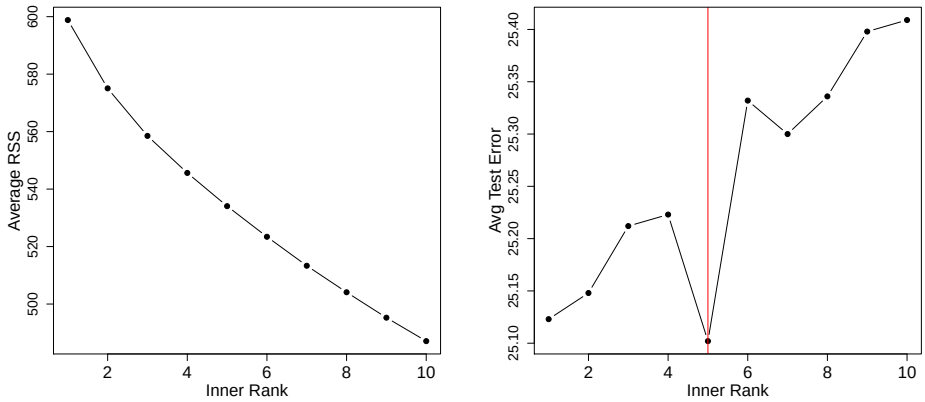



Figure 4.7: Choosing $K$ for the Catalano communications network. The left panel shows the average residual sum of squares, and the right panel shows the average test error obtained via cross validation ($5 \times 5$ fold) for different the approximation ranks. Cross validation indicates that 5 communities is optimal.

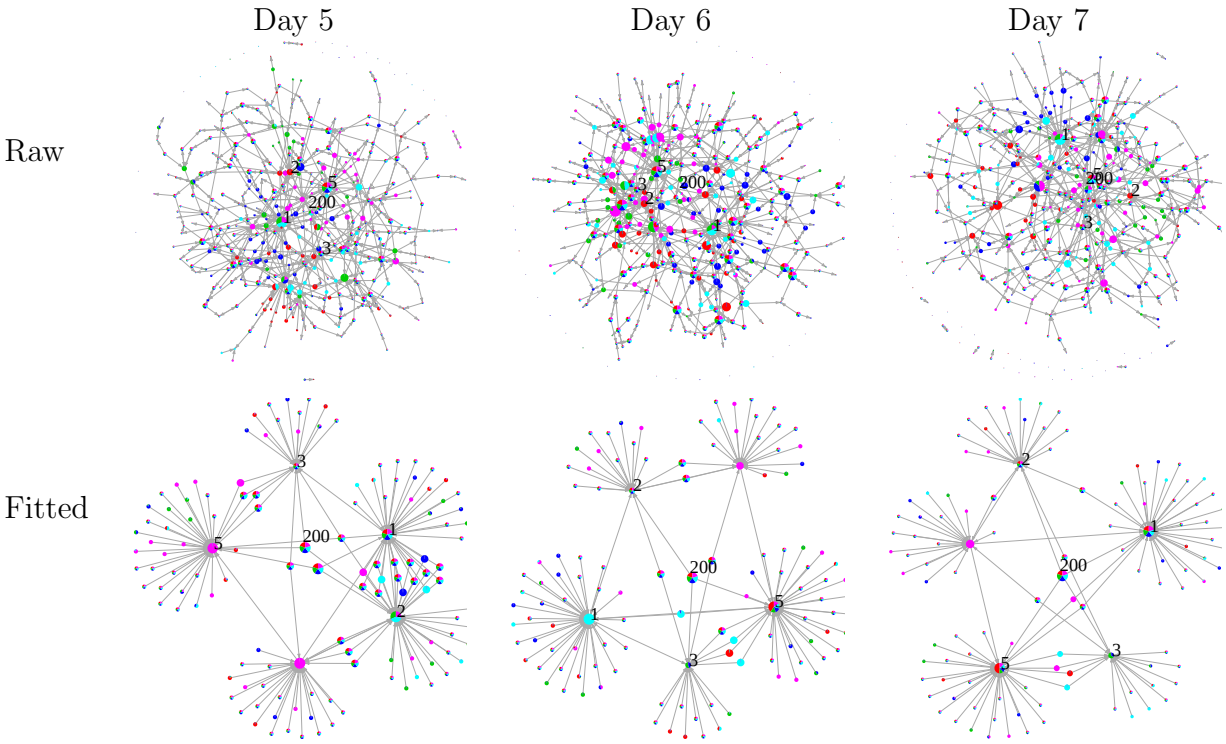



Figure 4.8: The raw (top row) and filtered Catalano networks (bottom row) colored by the $U_t$ community structure. A force directed layout in igraph was used to create this embedding. Nodes are colored by soft partitioning via the penalized NMF.

munications for the network; a hint was given to challenge participants that node 200 is one of the persons of interest.

I use the first seven days of data to illustrate our methodology, since there is a strong change in the connection patterns from day 8-10 for node 200 (see vas (2008); *Shaverdian et al.* (2009) and references therein). Directed networks are constructed daily by drawing an edge from the caller to the receiver. Fig. 4.6 shows an example of one day's network. The graph is too cluttered to visually identify leaders of the network or get a sense of the network structure.

I fit a sequence of rank 5 NMFs, as identified in Fig. 4.7 through cross validation, with a large temporal penalty to highlight the most persistent interactions. Fig. 4.8 shows two sets of graph drawings for three days (due to space limitations), with the nodes colored according to their community membership. The first row shows the graph constructed directly from the data, while the second row shows graph drawings of the *fitted* model $\hat{A} = U_t V_t^T$. The clustering results applied to the raw data are not interpretable, as the data is simply too cluttered. However, the persons of interest and the hierarchical structure of the communication network are clearly shown when considering the fitted networks. One can visually identify that node 200 consistently relays information to his neighbors (1,2,3,5), who disseminate information to their respective subordinates. Nodes higher up on the organizational hierarchy tend to belong to multiple communities, presumably since they disseminate information to different groups of subordinates.

Fig. 4.9 shows the results of applying Facetnet (*Lin et al.*, 2008), an alternative NMF methodology for dynamic overlapping community detection. Facetnet applies an underlying model with less flexibility resulting in poor reconstructions of the data, as seen in the fitted networks. I also collapse the data into a single network snapshot in order to apply static clustering algorithms. First, an edge is kept only if it was observed more than *Threshold* days. Then, spectral clustering and clique percolation

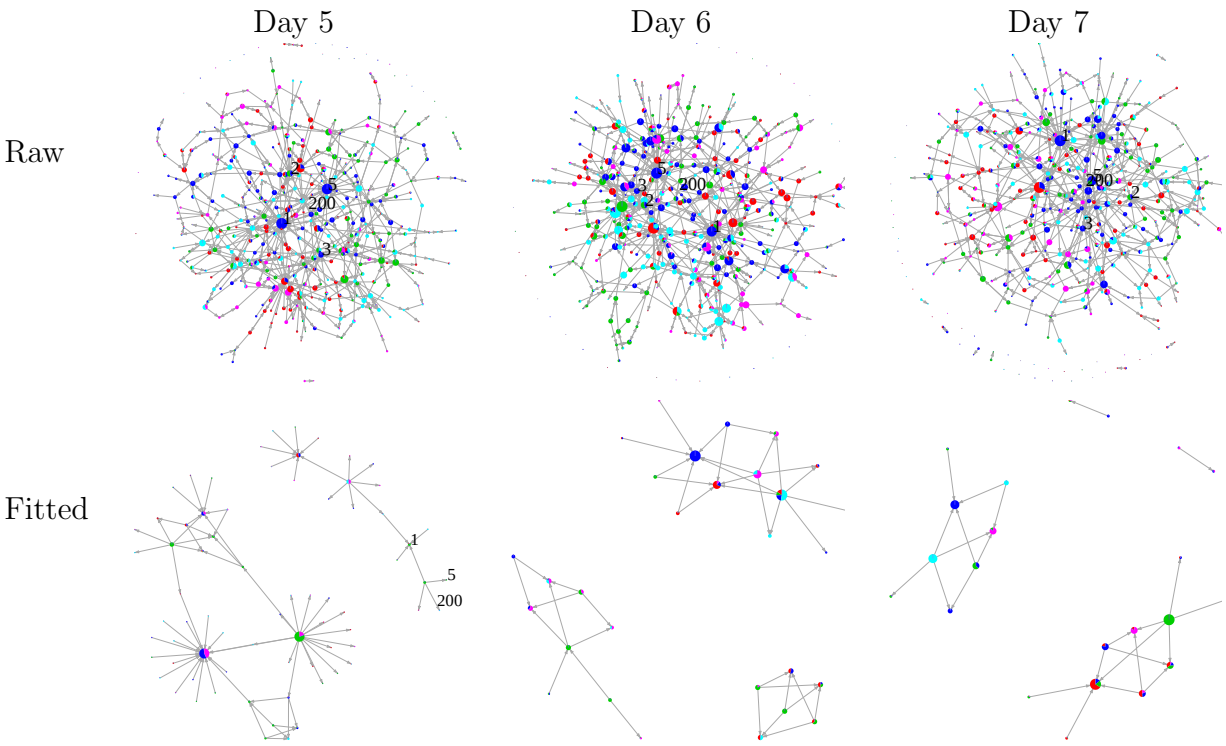



Figure 4.9: Results of applying the Facetnet factorization *Lin et al.* (2008) with a prior weight of $\lambda = 0.8$. The raw (top row) and filtered Catalano networks (bottom row) colored by the Facetnet factorization.

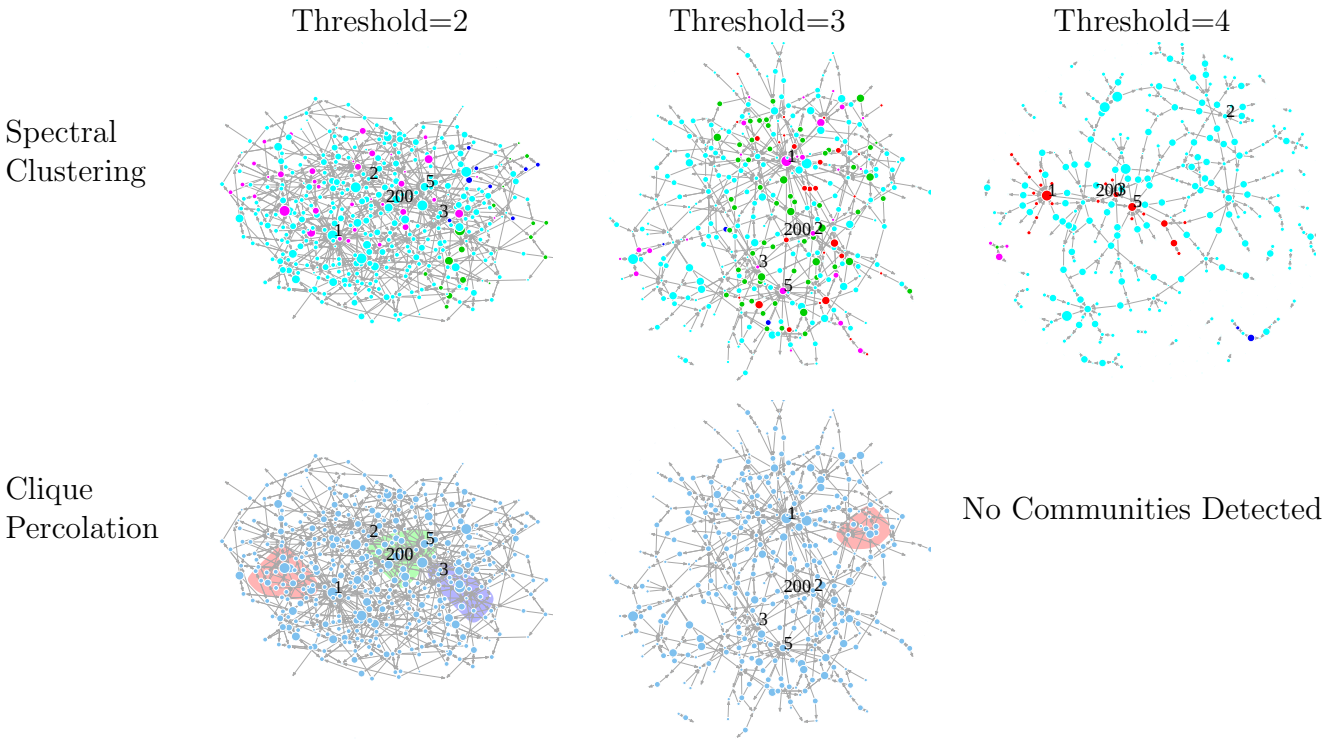



Figure 4.10: The first and second rows apply static clustering methods to the collapsed data (averaging over time). All alternative methods struggle to identify the key individuals or hierarchical organizational structure.

are applied to the resultant network snapshot. All alternative methods struggle, as the data is too 'hairball' like. On the other hand, the fitted penalized NMF model provides a unified framework to filter the network and visualize community structure. VAST never officially released correct answers for the challenge. However, our analysis closely matches winning entries (*Shen and Ma*, 2008; *Ye et al.*, 2009; *Shaverdian et al.*, 2009). Treating the conclusions of the entries as ground truth, I have provided a simple workflow that uncovers patterns in the data that are not directly obtainable with traditional methods.

#### 4.4.1.2 Preferential Attachment Process

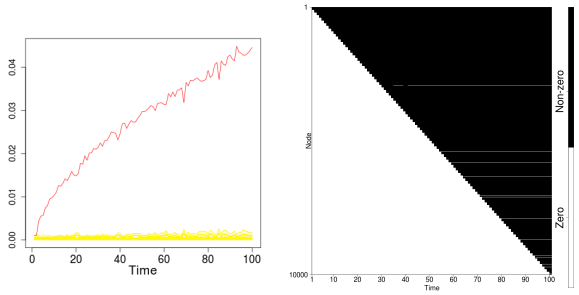
In this simulation, nodes attach according to a preferential attachment model (*Newman et al.*, 2006; *Barabsi and Albert*, 1999) until 10000 nodes have 'attached' to the embedding. I observe this growing process at 100 uniformly spaced time points. Thus, at each time point 100 new nodes attach to the graph. I use source code from a networks MATLAB toolbox (*Bounova*, 2011) that generates preferential attachment graphs according to the standard model.

In the preferential attachment model, $\Pi(i)$, which represents the probability that a new node connects to node $i$, depends on node $i$'s degree. Specifically,

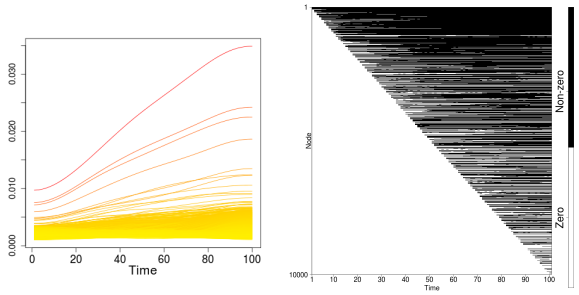
$$\Pi(i) \propto d_i \tag{4.15}$$

where $d_i$ is the degree of the $ith$ node. This generating framework leads to networks whose asymptotic degree distribution follows a power-law distribution with parameter $\gamma = 3$. Graphs with heavy-tailed degree distributions are commonly observed in a variety of areas, such as the Internet, protein interactions, citation networks, among others (*Clauset et al.*, 2009).

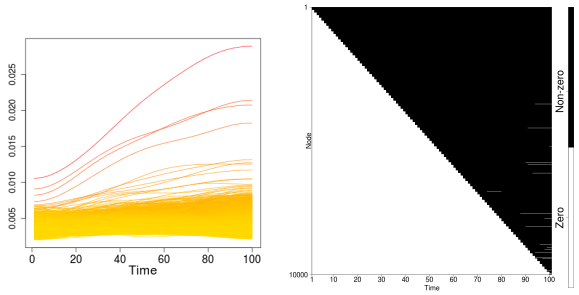
In practice, an analyst would not know that the data comes from a preferential




(a) No penalties




(b) $\lambda_t = 50, \lambda_s = 5$




(c) $\lambda_t = 100, \lambda_s = 5$

Figure 4.11: Fitted values for $U$ and $V$ over time for the preferential attachment simulation. The left column shows a time plot of $U_t$ over different parameter values. Each line corresponds to a node on the graph. The right column identifies the nonzero elements of $V_t$. Each row corresponds to a node on the graph and time varies along the horizontal axis.

attachment process. In which case, an exploratory analysis may include inspecting the network sequence on a set of standard metrics (degree, transitivity, centrality, etc.), graph drawings, as well as community detection approaches. A sequence of one-dimensional ($K = 1$) penalized NMFs can serve as the basis for a complimentary exploratory tool that helps uncover different connectivity patterns and evolution in the data. In particular, due to the smoothness penalty, time plots in $U_t$ for each node become useful for uncovering the number and types of node evolutions in the data. Similarly, heatmaps or displays of the sparsity pattern of $V_t$ are useful to identify when nodes/groups become significantly active.

Since preferential attachment networks have been extensively studied, I show only the NMF-based displays. Fig. 4.11 shows important (hub) nodes that distinct trajectories that indicate their increasing importance to the network over time. The $V_t$ sparsity features a pseudo-upper triangular form. This corresponds to the node attachment order and reflects that nodes permanently attach after connecting to the network. Such displays can be created quickly and can help the process of identifying interesting nodes, formulating research questions, and so on.

Also shown in Fig. 4.11 is that penalization is important to the usefulness and interpretability of the displays. For instance, without a temporal penalty, the time plots emphasize only the highest degree node. With appropriate penalties, an analyst can visually identify the different hub nodes.

### 4.4.2 Real Networks

#### 4.4.2.1 arXiv Citations

I investigate the citation network data analyzed in the previous chapter (Section 3.5.2), provided as part of the 2003 KDD Cup (*Gehrke et al.*, 2003).

To review the data, the graphs are from the e-print service arXiv for the 'high energy physics theory' section. It covers papers in the period from October 1993 to

December 2002, and is organized into monthly networks. In particular, if paper $i$ cites paper $j$, then the graph contains a directed edge from $i$ to $j$. Any citations to or from papers outside the dataset are not included. Following convention, edges are aggregated, that is, the citation graph for a given month will contain all citations from the beginning of the data up to, and including, the current month. Altogether, there are 22750 nodes (papers) with 176602 edges (references) over 112 months.

Section 3.5.2 contains an analysis of the data using existing tools, including network drawings, network statistic time-series, and a likelihood-based mixture model for extracting communities.

To visualize how nodes in the network evolved, Fig. 4.12 displays results from the matrix factorization model using a sequence of one-dimensional approximations ($K = 1$). The adjacency matrix is constructed so that $U_t$ scores nodes by their importance to the average *incoming* connections, and $(U_t)_{1j}$ measures the time-varying authority of paper $j$. $V_t$ yields similar scores based on outgoing connections. As observed with the preferential attachment experiment, the paper trajectories are smoothed effectively and important dynamics are highlighted by employing penalties. Specifically, there are two important periods in the data. The first period covers 1996-1999, and featured papers mostly on an extension of string theory called M-theory. M-theory was first proposed in 1995 and led to new research in theoretical physics. A number of scientists, including Witten, Sen and Polchinski, were important to the historical development of the theory, and as seen in Tables 4.1 and Table 4.2, our NMF approach identifies these important authors and their works. From 1999-2000 the rate of citations to these papers tended to decrease, while focus shifted to other topics and subfields that M-theory gave rise to. These citation patterns are reflected in the bold and dashed trajectories in Fig. 4.12. The displays of $V_t$ sparsity show that papers do not appear uniformly throughout time. Instead as other network statistics show, papers 'attach' at a faster rate around year 2000.

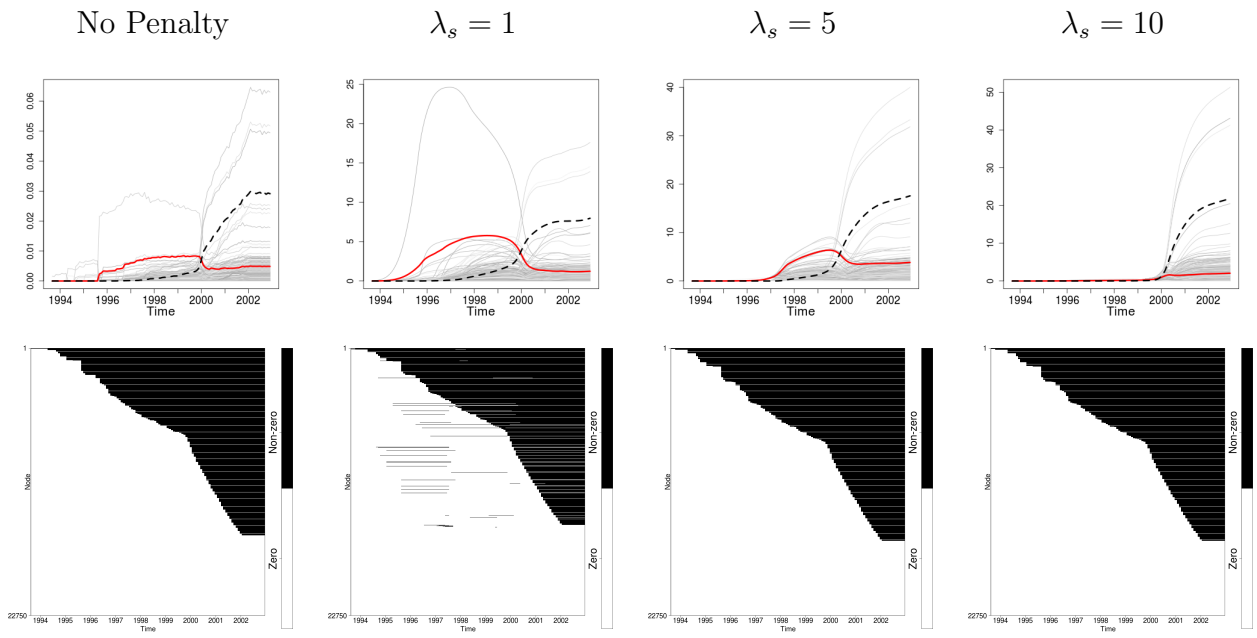



Figure 4.12: Fitted values for $U_t$ and $V_t$ for the arXiv data with $\lambda_t = 5$. Each light gray line corresponds to a paper (node) on the graph. The bold lines show the average of the 10 papers with highest average $\hat{U}$ from 1996-1999, and 2000 onwards (dashed). Each row in the heatmaps corresponds to a paper and time varies along the horizontal axis.

I provide comparisons in the last chapter with the alternative methodologies utilized in *Leicht et al.* (2007) to investigate dynamic citation network from the US Supreme Court. First, I applied the leading eigenvector modularity-based method for community discovery (*Newman*, 2006a) to the fully formed citation network ($t = 112$). The second alternative methodology is a mixture model in *Leicht et al.* (2007) to extract groups of papers according to their common temporal citation profiles.

The left panel of Fig. 3.9 shows the degree of each paper over time, colored by the leading eigenvector community assignments. The optimal number of groups is over two hundred. There are four large groups of papers, with the other groups containing only a handful of papers. This approach does not utilize the temporal profile of each paper, and as a consequence the groups are interpretable from a static connectivity point of view only.

Fig. 3.10 shows reasonable time-profiles from the mixture model. One group grows slowly from the beginning of the observational period, while the other group experiences rapid growth starting around the year 2000. These results compliment the NMF-based Fig. 4.12, and together provide a robust methodology to identify important papers, as well as characterize the data in terms of the number and types of different nodes/groups in the data.

#### 4.4.2.2 Global Trade Flows

In this example, I analyze a subset of the data from Chapter II (Section 2.5), consisting of annual bilateral trade flows between 164 countries from 1980-1997 (*Feenstra et al.*, 2004). Thus, I observe a dynamic, weighted graph at 18 time points, where each directional edge denotes the total value of exports from one country to another. Again since trade flows can differ in size by orders of magnitude, I work with trade values that are expressed in log dollars.

I fit a sequence of rank 6 NMFs, as identified in Fig. 4.13 through cross validation,

| Title | Authors | In-Degree | Out-Degree | # citations (Google) |
|---|---|---|---|---|
| Heterotic and Type I String Dynamics from Eleven Dimensions | Horava and Witten | 783 | 18 | 2334 |
| Five-branes And $M$ -Theory On An Orbifold | Witten | 169 | 15 | 251 |
| Type IIB Superstrings, BPS Monopoles, And Three-Dimensional Gauge Dynamics | Hanany and Witten | 437 | 20 | 844 |
| D-Branes and Topological Field Theories | Bershadsky, et al. | 271 | 15 | 463 |
| Lectures on Superstring and M Theory Dualities | Schwarz | 247 | 68 | 534 |
| D-Strings on D-Manifolds | Bershadsky et al. | 172 | 22 | 247 |
| String Theory Dynamics In Various Dimensions | Witten | 263 | 0 | 2263 |
| Branes, Fluxes and Duality in M(atrix)-Theory M(atrix)-Theory | Ganor, et al. | 184 | 16 | 243 |
| Dirichlet-Branes and Ramond-Ramond Charges | Polchinski | 370 | 0 | 2592 |
| Matrix Description of M-theory on $T^5$ and $T^5/Z_2$ | Seiberg | 208 | 30 | 353 |

Table 4.1:
The top 10 papers with highest average $\hat{U}$ from 1996-1999. # Citations counts all references to the work, including by papers outside of the data. These counts obtained via Google.

| Title | Authors | In-Degree | Out-Degree | # citations (Google) |
|---|---|---|---|---|
| The Large N Limit of Superconformal Field Theories and Supergravity | Maldacena | 1059 | 2 | 10697 |
| Anti De Sitter Space And Holography | Witten | 766 | 2 | 6956 |
| Gauge Theory Correlators from Non-Critical String Theory | Gubser et al. | 708 | 0 | 6004 |
| String Theory and Noncommutative Geometry | Seiberg and Witten | 796 | 12 | 3833 |
| Large N Field Theories, String Theory and Gravity | Aharony et a. | 446 | 74 | 3354 |
| An Alternative to Compactification | Randall and Sundrum | 733 | 0 | 5693 |
| Noncommutative Geometry and Matrix Theory: Compactification on Tori | Connes et al. | 512 | 3 | 1810 |
| M Theory as a Matrix Model: a Conjecture | Banks et al. | 414 | 0 | 2460 |
| D-branes and the Noncommutative Torus | Douglas and Hull | 296 | 2 | 866 |
| Dirichlet-Branes and Ramond-Ramond Charges | Polchinski | 370 | 0 | 2592 |

Table 4.2: The top 10 papers with highest average $\hat{U}$ from 2000 onwards.

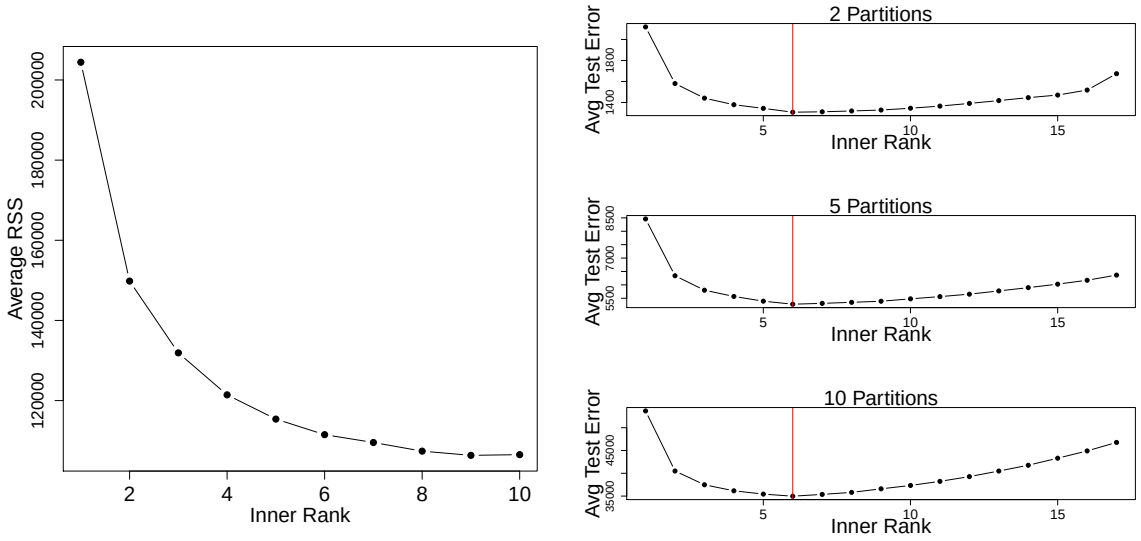



Figure 4.13: Choosing $K$ for World Trade Data. The left panel shows the average residual sum of squares. The right panel shows the average test error obtained via cross validation for different number of partitions. Cross validation consistently indicates 6 communities ($K = 6$) as optimal.

and display the network based on fitted trade flows ($\hat{A}_t = U_t V_t^T$) in Fig. 4.14. Only three years (1980, 1990, 1997) are shown due to space constraints.

All countries belong to more than one community, which reflects the interconnected nature of the global economy. However, there are countries, primarily from African and Central America, that are dominated by a single community or belong to only a subset of the six communities. For instance, in 1997, Ecuador, Venezuela and Panama only connect with the USA and hence, belong mostly to the green community. These countries tend to have monolithic economies.

There are also interesting findings that correspond with historical events. For instance, in 1980 there is a strong community (circled in the figure) consisting of countries aligned with the former USSR, which acted as a hub. However by 1990, this community has dissolved, and is reflected in the edge and node colorings of these countries (more diversified trading relationships). In 1990, countries in Asia that experienced persistent and rapid economic growth in the 1990's (*Stiglitz*, 1996; *Nelson*

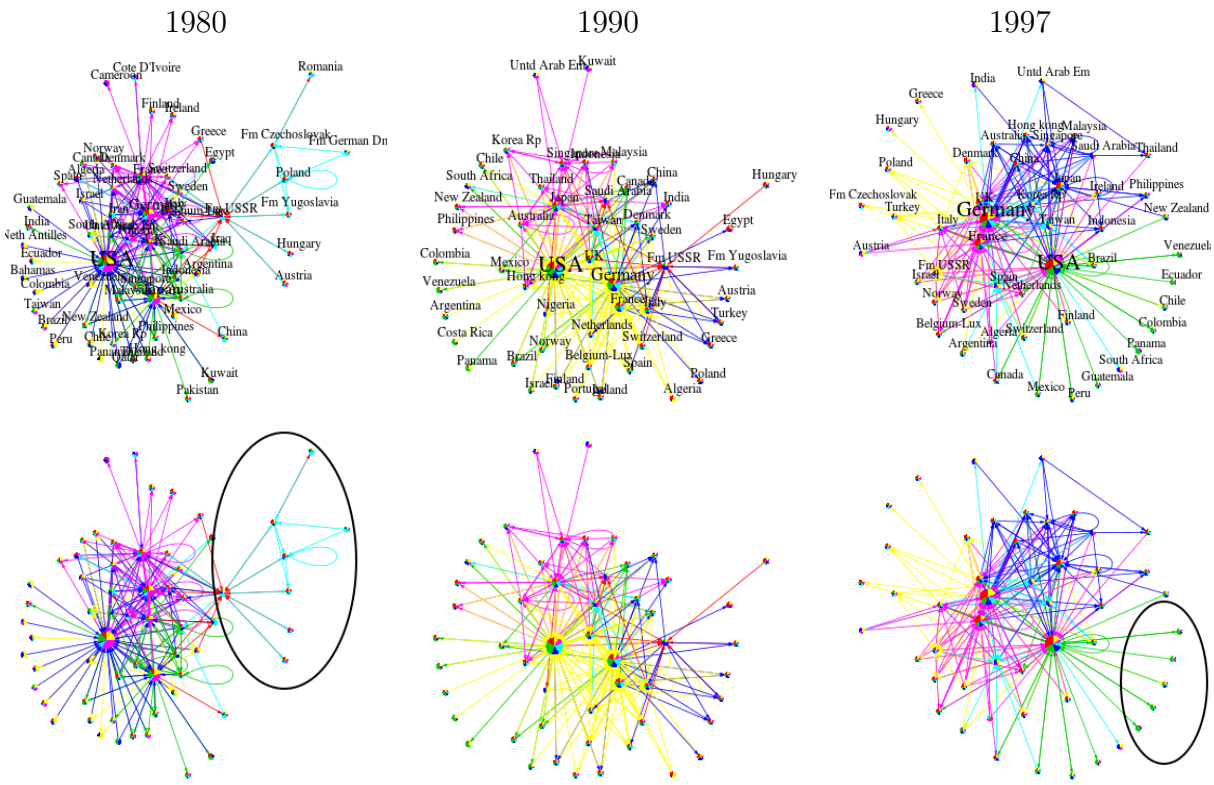



Figure 4.14: World trade networks over time, where countries are colored corresponding to their membership in 6 communities. Edges are colored by the community with largest relative contribution. The bottom row shows the same network drawing without labels.

*and Pack*, 1998) move closer to the center of the trading network with membership in all communities.

## 4.5 Discussion

The main idea behind the approach presented in this paper is to abstract the network sequence to a sequence of lower dimensional spaces using matrix factorizations for visual exploration, community detection and structural discovery. Next, I highlight some of the strengths and weaknesses of this approach.

Table 4.3: Average runtimes for the penalized NMF with temporal and sparsity penalties. The computational time scales approximately linearly with the number of time points and nodes.

| Data | Nodes | Time Points | Runtime (seconds) |
|---|---|---|---|
| Catalano | 400 | 7 | 0.29 |
| World Trade | 164 | 18 | 0.51 |
| Preferential Attachment | 10000 | 100 | 39.45 |
| arXiv Citations | 22750 | 112 | 60.64 |

### 4.5.1 Strengths

An important benefit is the versatility and scalability of matrix factorization model. Table 4.3 shows runtimes for all experiments. The computational cost is low enough to use in combination with other analysis and visualization tools. Moreover, the penalized NMF approach is compatible with both binary and weighted networks.

Using the model as a basis for an exploratory visual tool can help users uncover different connectivity patterns and evolution in the data. The estimates of $U_t$ and $V_t$ can be used for community discovery or a ranking of nodes based on their importance to connectivity for subsequent analysis. Displays of the factorizations can provide a sense of the data complexity, namely the types and number of node evolutions.

### 4.5.2 Weaknesses

The optimal choice of tuning parameters $(\lambda_t, \lambda_s)$ is dependent on perception and how the edge weights are scaled. This can limit the benefits of the proposed approach when given multiple datasets.

Time plots and heatmaps to visualize each factor yield limited information about global topology. For example, one can see from Figs. 4.11 and 4.12 that there are dominant nodes, but in principle, there could be many topologies that feature dominant nodes. One cannot say for sure without additional analysis that the networks follow a particular connectivity model. Thus, combining the matrix factorization

model in this article with existing analysis and visualization tools can provide a more comprehensive analysis of the data.

### 4.5.3 Future Work

An important area of exploration would be to systematically compare penalized versions of NMF and SVD. In this chapter I chose to focus on NMF, since I find the corresponding displays preferable in terms of interpretability. This is generally consistent with existing literature on matrix factorization. However, SVD of graph related matrices have deep connections to classical spectral layout and problems in community detection. There may be classes of graph topologies and particular visualization goals under which SVD is preferable.

There could also be other types and combinations of penalties that are useful in visualization and detection of graph structure. For instance, depending on the precise meaning of a directional edge, one may desire both smoothness and sparsity for $U_t$, $V_t$ or both factors. Nonetheless, variants on the penalty structure will result in models that require roughly the same computational costs. Thus, this chapter provides evidence that penalized matrix factorization models are promising for structural and functional discovery in dynamic networks.

# CHAPTER V

# Concluding Remarks and Future Work

In this thesis, I consider several approaches to the challenging tasks of visualization and pattern discovery in three-way data. Difficulties that arise in this area commonly result from the complexity, high dimension and heterogeneity of the data. Each chapter in the thesis introduced methods to help overcome such difficulties for the tasks of clustering, data integration, visualization and data representation.

A commonality between all three chapters is the local smoothness of the underlying patterns, which is encouraged in the estimation through kernel smoothing and regularization. Numerical work in the dissertation shows that smoothing improves statistical power and informativeness of displays when there are strong nonlinearities underlying the data and/or large noise levels.

As with other exploratory and visualization tools, the different models are sensitive to the scaling of the data. For instance, if investigating annual world trade values that are expressed in nominal dollars instead of log nominal dollars, then results are dominated by the United States, because that country has by far the largest variance. Just as in principal component analysis and many other multivariate methods, the analyst should make a decision on standardizing observations based on what aspect of the data is of interest. Similarly, the results can change in response to the algorithm and model parameterization. Yet, the main results in the numerical work are

consistently found in repeated analyses of the data and show significant overlap between the different models, thus supporting the notion that the results are not noise artifacts. For instance, many common articles are identified as important using the data integration model of Chapter III and the functional community detection model of Chapter IV. With world trade data, Chapters II and IV discover similar changes to grouping structure resulting from historical events, like the fall of the former Soviet Union.

Thoughout this dissertation, the data is assumed to be structured $\{X_m \in \mathbb{R}^{n\times p}\}_{m=1}^{M}$. A more general problem would be to consider $\{X_m \in \mathbb{R}^{n_m\times p}\}_{m=1}^{M}$. This data structure can be found, for example, in multicorpus document-term data, where $n_m$ denotes the number of documents in each of the $M$ different corpora (e.g., Wall Street Journal vs. The Financial Times vs. Bloomberg), with $p$ terms appearing in all documents.

Minimizing an objective function, such as

$$\min_{\{U_m\geq 0, V_m\geq 0\}} \quad \sum_{m=1}^{M} ||X_m - U_m V_m^T||_F^2 + \lambda_1 \sum_{m,\tilde{m}=1}^{M} ||V_m - V_{\tilde{m}}||_F^2 + \lambda_2 \sum_{m=1}^{M}\sum_{j=1}^{K} ||V_m(,j)||_1, \tag{5.1}$$

would allow us to uncover and visualize the perspectives within the different corpora. In considering the more general data structure, there are also potential applications in discovering weighted communities within biological networks observed over different experimental conditions (see *Li et al.* (2011) for further discussion).

Finally, it is important to note that is increasingly common to measure different types of data. For instance, it is a challenging and important problem in many modern applications to combine information from network and traditional node (sample) $\times$ variable data matrices. The creation of sophisticated tools for representation and integration of such data has the potential to reveal the nature of interactions among components and hence, improve decision making in complex environments.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


(2008), *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2008, Columbus, Ohio, USA, 19-24 October 2008*, IEEE.

Airoldi, E. M., C. Huttenhower, D. Gresham, C. Lu, A. A. Caudy, M. J. Dunham, J. R. Broach, D. Botstein, and O. G. Troyanskaya (2009), Predicting cellular growth from gene expression signatures, *PLoS Comput Biol*, *5*(1), e1000,257.

Archambault, D., H. Purchase, and B. Pinaud (2011), Animation, small multiples, and the effect of mental map preservation in dynamic graphs, *Visualization and Computer Graphics, IEEE Transactions on*, *17*(4), 539 –552, doi:10.1109/TVCG.2010.78.

Asur, S., S. Parthasarathy, and D. Ucar (2009), An event-based framework for characterizing the evolutionary behavior of interaction graphs, *ACM Trans. Knowl. Discov. Data*, *3*(4), 16:1–16:36, doi:10.1145/1631162.1631164.

Ball, B., B. Karrer, and M. E. J. Newman (2011), Efficient and principled method for detecting communities in networks, *Phys. Rev. E*, *84*, 036,103, doi:10.1103/PhysRevE.84.036103.

Barabsi, A.-L., and R. Albert (1999), Emergence of scaling in random networks, *Science*, *286*(5439), 509–512, doi:10.1126/science.286.5439.509.

Bartels, R. H., and G. W. Stewart (1972), Solution of the matrix equation AX + XB = C [F4], *Commun. ACM*, *15*, 820–826, doi:http://doi.acm.org/10.1145/361573.361582.

Bernanke, B. S., and K. Rogoff (2001), *NBER Macroeconomics Annual 2000, Volume 15*, MIT PRess.

Berry, M. W., M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons (2006), Algorithms and applications for approximate nonnegative matrix factorization, in *Computational Statistics and Data Analysis*, pp. 155–173.

Bounova, G. (2011), Matlab tools for network analysis, *Tech. rep.*, Massachusetts Institute of Technology.

Boyd, S., and L. Vandenberghe (2004), *Convex optimization*, xiv+716 pp., Cambridge University Press, Cambridge.

Brandes, U., D. Fleischer, and T. Puppe (2006), Dynamic spectral layout of small worlds, 10.1007/11618058_3.

Buja, A., D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen (2008), Data visualization with multidimensional scaling, *Journal of Computational and Graphical Statistics*, *17*(2), 444–472, doi:10.1198/106186008X318440.

Cai, D., X. He, J. Han, and T. Huang (2011), Graph regularized nonnegative matrix factorization for data representation, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *33*(8), 1548 –1560, doi:10.1109/TPAMI.2010.231.

Chen, Z., and A. Cichocki (2005), Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints, in *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep.*

Cheng, Y., and G. M. Church (2000), Biclustering of expression data, in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103, AAAI Press.

Chu, M., F. Diele, R. Plemmons, and S. Ragni (2004), Optimality, computation, and interpretation of nonnegative matrix factorizations, *SIAM JOURNAL ON MATRIX ANALYSIS*, pp. 4–8030.

Chung, F. R. K. (1997), *Spectral Graph Theory*, Amer. Math. Soc.

Cichocki, A., and R. Zdunek (2007), Regularized alternating least squares algorithms for non-negative matrix/tensor factorization, in *Proceedings of the 4th international symposium on Neural Networks: Advances in Neural Networks, Part III*, ISNN '07, pp. 793–802, Springer-Verlag, Berlin, Heidelberg, doi:http://dx.doi.org/10.1007/978-3-540-72395-0_97.

Cichocki, A., R. Zdunek, S. Choi, R. Plemmons, and S. ichi Amari (2007), Novel multi-layer nonnegative tensor factorization with sparsity constraints, in *IN: PROC. 8-TH INTERNATIONAL CONFERENCE ON ADAPTIVE AND NATURAL COMPUTING ALGORITHMS*, pp. 271–280.

Cichocki, A., R. Zdunek, A. Phan, and S. Amari (2009), *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, John Wiley & Sons.

Clauset, A., C. Shalizi, and M. Newman (2009), Power-law distributions in empirical data, *SIAM Review*, *51*(4), 661–703, doi:10.1137/070710111.

de Leeuw, J., K. Hornik, and P. Mair (2009), Isotone optimization in r: Pool-adjacent-violators algorithm (pava) and active set methods, *Journal of Statistical Software*, *32*(5), 1–24.

de Solla Price, D. J. (1965), Networks of scientific papers, *Science*, *149*(3683), 510–515, doi:10.1126/science.149.3683.510.

Devarajan, K. (2008), Nonnegative matrix factorization: An analytical and interpretive tool in computational biology, *PLoS Comput Biol*, *4*(7), e1000,029, doi: 10.1371/journal.pcbi.1000029.

Ding, C., X. He, and H. D. Simon (2005), On the equivalence of nonnegative matrix factorization and spectral clustering, in *Proc. SIAM Data Mining Conf*, pp. 606–610.

Ding, C., T. Li, and W. Peng (2008), On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing, *Comput. Stat. Data Anal.*, *52*(8), 3913–3927, doi:10.1016/j.csda.2008.01.011.

Eng, K., S. Keles, and W. G. (2008), A linear mixed effects clustering model for multi-species time course gene expression data, *Tech. rep.*

Fath, B. D., S. E. Jrgensen, B. C. Patten, and M. Strakraba (2004), Ecosystem growth and development, *Biosystems*, *77*(1-3), 213 – 228, doi:10.1016/j.biosystems.2004.06.001.

Feenstra, R. C., R. E. Lipsey, H. Deng, A. C. Ma, and H. Mo (2004), World trade flows: 1962:2000, *NBER Working Paper no. 11040*.

Fienberg, S. E. (2012), A brief history of statistical models for network analysis and open challenges, *Journal of Computational and Graphical Statistics*, *21*(4), 825–839, doi:10.1080/10618600.2012.738106.

Fortunato, S., and M. Barthlemy (2007), Resolution limit in community detection, *Proceedings of the National Academy of Sciences*, *104*(1), 36–41, doi:10.1073/pnas.0605965104.

Frishman, Y., and A. Tal (2008), Online dynamic graph drawing, *Visualization and Computer Graphics, IEEE Transactions on*, *14*(4), 727 –740, doi:10.1109/TVCG.2008.11.

Gehrke, J., P. Ginsparg, and J. M. Kleinberg (2003), Overview of the 2003 kdd cup, in *SIGKDD Explorations*, vol. 5, pp. 149 –151.

Ghani, S., N. Elmqvist, and J.-S. Yi (2012), Perception of animated node-link diagrams for dynamic graphs, *Computer Graphics Forum (Proc. EuroViz 2012)*, *31*(3), 1205–1214.

Girvan, M., and M. E. J. Newman (2002a), Community structure in social and biological networks, *Proceedings of the National Academy of Sciences of USA*, *99*, 7821–7826.

Girvan, M., and M. E. J. Newman (2002b), Community structure in social and biological networks, *Proceedings of the National Academy of Sciences*, *99*(12), 7821–7826, doi:10.1073/pnas.122653799.

Goldenberg, A., A. X. Zheng, S. E. Fienberg, and E. M. Airoldi (2009), A survey of statistical network models, *ArXiv e-prints*.

Golub, G., S. Nash, and C. Van Loan (1979), A hessenberg-schur method for the problem ax + xb= c, *Automatic Control, IEEE Transactions on*, *24*(6), 909 – 913, doi:10.1109/TAC.1979.1102170.

Gong, L., C. Teng, A. Livne, C. Brunetti, and L. A. Adamic (2011), Coevolution of network structure and content, *CoRR*, *http://arxiv.org/abs/1107.5543*.

Guo, J., G. James, E. Levina, G. Michailidis, and J. Zhu (2010), Principal component analysis with sparse fused loadings, *Journal of Computational and Graphical Statistics*, *19*(4), 930–946, doi:10.1198/jcgs.2010.08127.

Hastie, T., R. Tibshirani, and J. H. Friedman (2001), *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*, 533 pp., New York: Springer-Verlag.

Hazan, T., S. Polak, and A. Shashua (2005), Sparse image coding using a 3d non-negative tensor factorization, in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 50 – 57 Vol. 1, doi:10.1109/ICCV.2005.228.

Hoff, P. D., A. E. Raftery, and M. S. Handcock (2002), Latent space approaches to social network analysis, *Journal of the American Statistical Association*, *97*(460), 1090–1098, doi:10.1198/016214502388618906.

Hoyer, P. O. (2002), Non-negative sparse coding, in *In Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pp. 557–565.

Hoyer, P. O. (2004), Non-negative matrix factorization with sparseness constraints, *J. Mach. Learn. Res.*, *5*, 1457–1469.

Kim, H., and H. Park (2008), NONNEGATIVE MATRIX FACTORIZATION BASED ON ALTERNATING NONNEGATIVITY CONSTRAINED LEAST SQUARES AND ACTIVE SET METHOD, *SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS*, *30*(2), 713–730, doi:{10.1137/07069239X}.

Kleinberg, J. M. (1999), Authoritative sources in a hyperlinked environment, *J. ACM*, *46*(5), 604–632, doi:10.1145/324133.324140.

Kolar, M., L. Song, A. Ahmed, and E. P. Xing (2010), Estimating time-varying networks, *Annals of Applied Statistics*, *4*(2), 94–123, doi:10.1214/09-AOAS308.

Koren, Y. (2005), Drawing graphs by eigenvectors: theory and practice, *Computers & Mathematics with Applications*, *49*(1112), 1867 – 1888, doi:10.1016/j.camwa.2004.08.015.

Lazzeroni, L., and A. Owen (2000), Plaid models for gene expression data, *Statistica Sinica*, *12*, 61–86.

Lee, D. D., and H. S. Seung (1999), Learning the parts of objects by non-negative matrix factorization, *Nature*, *401*, 788–791.

Lee, D. D., and H. S. Seung (2001), Algorithms for non-negative matrix factorization, *Advances in neural information processing systems*, pp. 556–562.

Leicht, E. A., G. Clarkson, K. Shedden, and M. E. Newman (2007), Large-scale structure of time evolving citation networks, *The European Physical Journal B*, *59*, 75–83, doi:10.1140/epjb/e2007-00271-7.

Leskovec, J., J. Kleinberg, and C. Faloutsos (2005), Graphs over time: densification laws, shrinking diameters and possible explanations, in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pp. 177–187, ACM, New York, NY, USA, doi:10.1145/1081870.1081893.

Li, W., C.-C. Liu, T. Zhang, H. Li, M. S. Waterman, and X. J. Zhou (2011), Integrative analysis of many weighted co-expression networks using tensor computation, *PLoS Comput Biol*, *7*(6), e1001,106, doi:10.1371/journal.pcbi.1001106.

Lin, C.-J. (2007), On the convergence of multiplicative update algorithms for nonnegative matrix factorization, *Neural Networks, IEEE Transactions on*, *18*(6), 1589 –1596, doi:10.1109/TNN.2007.895831.

Lin, Y.-R., Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng (2008), Facetnet: a framework for analyzing communities and their evolutions in dynamic networks, in *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pp. 685–694, ACM, New York, NY, USA, doi:10.1145/1367497.1367590.

Luan, Y., and H. Li (2003), Clustering of time-course gene expression data using a mixed-effects model with b-splines, *Bioinformatics*, *19*(4), 474–482, doi:10.1093/bioinformatics/btg014.

Ma, P., C. I. Castillo-Davis, W. Zhong, and J. S. Liu (2006), A data-driven clustering method for time course gene expression data, *Nucleic Acids Research*, *34*(4), 1261–1269, doi:10.1093/nar/gkl013.

Madeira, S. C., and A. L. Oliveira (2004), Biclustering algorithms for biological data analysis: a survey, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *1*, 24–45.

Meyer, C. D. (Ed.) (2000), *Matrix analysis and applied linear algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Mukerjee, H. (1988), Monotone nonparametric regression, *The Annals of Statistics*, *16*, 741–750.

Nelson, R. R., and H. Pack (1998), The asian miracle and modern growth theory, *Policy Research Working Paper Series 1881*, The World Bank.

Newman, M. (2010), *Networks: An Introduction*, Oxford University Press.

Newman, M., A. Barabási, and D. Watts (2006), *The Structure And Dynamics of Networks*, Princeton Studies in Complexity, Princeton University Press.

Newman, M. E. J. (2006a), Finding community structure in networks using the eigenvectors of matrices, *Phys. Rev. E*, *74*, 036,104, doi:10.1103/PhysRevE.74.036104.

Newman, M. E. J. (2006b), Modularity and community structure in networks, *Proceedings of the National Academy of Sciences*, *103*(23), 8577–8582, doi:10.1073/pnas.0601602103.

Osborne, C. (1991), Statistical calibration: A review, *International Statistical Review*, *59*, 309–336.

Paatero, P., and U. Tapper (1994), Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics*, *5*(2), 111–126, doi:10.1002/env.3170050203.

Palla, G., I. Derényi, I. Farkas, and T. Vicsek (2005), Uncovering the overlapping community structure of complex networks in nature and society, *Nature*, *435*, 814–818, doi:10.1038/nature03607.

Pan, J., K. Fang, and K. Fang (2002), *Growth Curve Models and Statistical Diagnostics*, Springer Series in Statistics, Springer.

Peng, R. (2008), A method for visualizing multivariate time series data, *Journal of Statistical Software, Code Snippets*, *25*(1), 1–17.

Perry, P. O., and A. B. Owen (2009), Bi-cross-validation of the svd and the non-negative matrix factorization., *Annals of Applied Statistics*, *3*(2), 564–594.

Psorakis, I., S. Roberts, M. Ebden, and B. Sheldon (2011), Overlapping community detection using bayesian non-negative matrix factorization, *Phys. Rev. E*, *83*, 066,114, doi:10.1103/PhysRevE.83.066114.

Qin, L.-X., and S. G. Self (2006), The clustering of regression models method with applications in gene expression data, *Biometrics*, *62*(2), 526–533, doi:10.1111/j.1541-0420.2005.00498.x.

Raginsky, M., R. Willett, C. Horn, J. Silva, and R. Marcia (2012), Sequential anomaly detection in the presence of noise and limited feedback, *Information Theory, IEEE Transactions on*, *58*(8), 5544–5562, doi:10.1109/TIT.2012.2201375.

Ramsay, J., and B. Silverman (2005), Modelling functional responses with multivariate covariates, in *Functional Data Analysis*, Spring Series in Statistics, pp. 223–245, Springer New York.

Rangel, C., J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani (2004), Modeling t-cell activation using gene expression profiling and state-space models, *Bioinformatics*, *20*(9), 1361–1372, doi:10.1093/bioinformatics/bth093.

Richard, E., P.-A. Savalle, and N. Vayatis (2012), Graph Prediction in a Low-Rank and Autoregressive Setting, *ArXiv e-prints*.

Robertson, T., F. T. Wright, and R. L. Dykstra (1988), *Order restricted statistical inference*, Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics, xx+521 pp., John Wiley & Sons Ltd., Chichester.

Rodriguez, F., and D. . Rodrik (2001), *Trade Policy and Economic Growth: A Skeptic's Guide to the Cross-National Evidence*, pp. 261–338, MIT Press.

Rohe, K., and B. Yu (2012), Co-clustering for Directed Graphs; the Stochastic Co-Blockmodel and a Spectral Algorithm, *ArXiv e-prints*.

Rohe, K., S. Chatterjee, and B. Yu (2011), Spectral clustering and the high-dimensional stochastic blockmodel., *Ann. Stat.*, *39*(4), 1878–1915, doi:10.1214/11-AOS887.

Rosenberger, W. F., and L. M. Haines (2002), Competing designs for phase i clinical trials: a review, *Stat. Med.*, *21*, 2757–2770.

Sarkar, P., and A. W. Moore (2005), Dynamic social network analysis using latent space models, *SIGKDD Explor. Newsl.*, *7*(2), 31–40, doi:10.1145/1117454.1117459.

Schafer, R., R. Opgen-Rhein, and K. Strimmer (2006), Reverse engineering genetic networks using the genenet package, *R News*, *6*(5), 50–53.

Shaverdian, A. A., H. Zhou, G. Michailidis, and H. V. Jagadish (2009), Algebraic visual analysis: the catalano phone call data set case study, in *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*, VAKD '09, pp. 74–82, ACM, New York, NY, USA, doi:10.1145/1562849.1562858.

Shen, Z., and K.-L. Ma (2008), Mobivis: A visualization system for exploring mobile data, in *Visualization Symposium, 2008. PacificVIS '08. IEEE Pacific*, pp. 175 –182, doi:10.1109/PACIFICVIS.2008.4475474.

Stiglitz, J. E. (1996), Some lessons from the east asian miracle, *The World Bank Research Observer*, *11*(2), 151–177, doi:10.1093/wbro/11.2.151.

Sun, J., C. Faloutsos, S. Papadimitriou, and P. S. Yu (2007), Graphscope: parameter-free mining of large time-evolving graphs, in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pp. 687–696, ACM, New York, NY, USA, doi:10.1145/1281192.1281266.

Turner, H., T. Bailey, and W. Krzanowski (2005a), Improved biclustering of microarray data demonstrated through systematic performance tests, *Computational Statistics & Data Analysis*, *48*(2), 235 – 254, doi:10.1016/j.csda.2004.02.003.

Turner, H., T. Bailey, W. Krzanowski, and C. Hemingway (2005b), Biclustering models for structured microarray data, *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, *2*(4), 316 –329, doi:10.1109/TCBB.2005.49.

von Landesberger, T., A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner (2011), Visual analysis of large graphs: State-of-the-art and future research challenges, *Computer Graphics Forum*, *30*(6), 1719–1749, doi: 10.1111/j.1467-8659.2011.01898.x.

von Landserber, T., A. Kuijper, T. Schreck, J. Kohlhammer, J.-J. van Wijk, and D.-W. Fellner (2010), Visual analysis of large graphs, *EuroGraphics state of the art reports*.

Wang, F., T. Li, X. Wang, S. Zhu, and C. Ding (2011), Community discovery using nonnegative matrix factorization, *Data Min. Knowl. Discov.*, *22*, 493–521, doi: http://dx.doi.org/10.1007/s10618-010-0181-y.

Wang, Y., and Y. Zhang (2012), Non-negative matrix factorization: a comprehensive review, *Knowledge and Data Engineering, IEEE Transactions on*, *PP*(99), 1, doi: 10.1109/TKDE.2012.51.

Welling, M., and M. Weber (2001), Positive tensor factorization, *Pattern Recogn. Lett.*, *22*(12), 1255–1261, doi:10.1016/S0167-8655(01)00070-8.

Witten, D. M., R. Tibshirani, and T. Hastie (2009), A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, *Biostatistics*, *10*(3), 515–534, doi:10.1093/biostatistics/kxp008.

Ye, Q., B. Wu, D. Hu, and B. Wang (2009), Exploring temporal egocentric networks in mobile call graphs, in *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on*, vol. 2, pp. 413 –417, doi:10.1109/FSKD.2009.617.

Yi, J. S., N. Elmqvist, and S. Lee (2010), Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations, *International Journal of Human-Computer Interaction*, *26*(11-12), 1031–1051.

Zou, H., T. Hastie, and R. Tibshirani (2006), Sparse principal component analysis, *Journal of Computational and Graphical Statistics*, *15*(2), 265–286, doi:10.1198/106186006X113430.