Principal Network Analysis

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Jonathan B. Mei

M.Eng., Electrical Engineering, Massachusetts Institute of Technology
B.S., Electrical Engineering, Massachusetts Institute of Technology

Carnegie Mellon University
Pittsburgh, PA

May, 2018

Abstract

Many applications collect a large number of time series, for example, temperature continuously monitored by weather stations across the US or neural activity recorded by an array of electrical probes. These data are often referred to as *un*structured. A first task in their analytics is often to derive a low dimensional representation – a graph or discrete manifold – that describes the *inter*relations among the time series and their *intra*relations across time.

In general, the underlying graphs can be directed and weighted, possibly capturing the strengths of causal relations, not just the binary existence of reciprocal correlations. Furthermore, the processes generating the data may be non-linear and observed in the presence of unmodeled phenomena or unmeasured agents in a complex networked system. Finally, the networks describing the processes may themselves vary through time.

In many scenarios, there may be good reasons to believe that the graphs are only able to vary as linear combinations of a set of "principal graphs" that are fundamental to the system. We would then be able to characterize each principal network individually to make sense of the ensemble and analyze the behaviors of the interacting entities.

This thesis acts as a roadmap of computationally tractable approaches for learning graphs that provide structure to data. It culminates in a framework that addresses these challenges when estimating time-varying graphs from collections of time series. Analyses are carried out to justify the various models proposed along the way and to characterize their performance. Experiments are performed on synthetic and real datasets to highlight their effectiveness and to illustrate their limitations.

Acknowledgments

I could not have gotten to this point without the unwavering love and support from my parents Jingjing and Renwei and my brother Vinny. Thank you for listening, passing on advice when I need it, and being there when it matters.

I would not be here if not for the wisdom and guidance of my committee. Especially to my advisor Professor José Moura, thank you for providing just the right balance of gentle nudges forward, autonomy to pursue my own research interests, a nice environment in which to do so, encouragement in the face of adversity, intellectual discussions on almost any topic, sprinkled in with valuable life lessons. I am also grateful to Professor Jelena Kovačević, Professor Rob Kass, and Professor Pradeep Ravikumar for serving on my committee. Thank you for the profound insights and suggestions to help me make this thesis the best I could.

I should not have made it this far without a strong group of friends and colleagues to lean on. Fortunately I did not have to. Big thanks to my groupmates past and present, Nick O'Donoghue, Augusto Santos, Joel Harley, Kyle Anderson, Subhro Das, June Zhang, Joya Deri, Stephen Kruzick, Evgeny Toropov, Lynna Gui, Shanghang Zhang, Yuan Chen, Satwik Kottur, Mark Cheung, Raj Das, John Shi, and Jian Du, for showing me the ropes and joining in with my shenanigans. Praise be to Porter B party bandits, Anit Sahu, Vishwa Saragadam, Chia-Yin Tsai, Jian Wang, John Costanzo, Yi Hua, Brian Swenson, Hadi Amini, Javad Mohammadi, Siheng Chen, Yaoqing Yang, Vince Monardo, Harlin Lee, Nikos Arechiga, and Rohan Chabuskwar, who brought light to the basement. And thanks to Devon Rosner, Alexis Hakimi, Ahmed Kirmani, Andrea Colaço, Ben Shih, Ray Carley, Clay Long, Mats Forssell, Serim Park, Stephen Siena, Sean Yen, Paola Buitrago, Hyesoo Yang, Anson Wang, Mai Sawada, Vova Katson, Stas Mamontov, and Steven and Hikari Aday, for helping keep me sane.

Finally, thanks to David and LaVerne Owen-Barakat for their generous support of my studies through their graduate fellowship, which opened up many opportunities for me personally and professionally. Also, large parts of this work were made possible by NSF grants CCF 1011903 and CCF 1513936.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the main problem addressed by this thesis: learning meaningful network structures from time series.

1.1 Uncovering Graphs from Data

There is an explosion of data, generated, measured, and stored at very fast rates in many disciplines, from finance and social media to geology and biology. Much of this data takes the form of simultaneous, long running time series. Examples include protein-to-protein interactions in organisms, patient records in health care, customer consumptions in (power, water, natural gas) utility companies, cell phone usage for wireless service providers, companies' financial data, and social interactions among individuals in a population. The internet-of-things (IoT) is an imminent source of ever increasing large collections of time series. This data is often referred to as *un*structured.

Graph signals

Networks or graphs are becoming prevalent as models to describe the relationships among the nodes and the data they produce. These low-dimensional graph data represen-

tations are used for further analytics, for example, to compute statistics, make inferences, perform signal processing tasks [1, 2, 3], or quantify how topology influences diffusion in networks of nodes [4]. These methods all use the structure of the *known* graphs to extract knowledge and meaning from the observed data supported on the graphs.

Uncovering Graphs

However, in many problems the graph structure itself may be *unknown*, and a first issue is to infer from the data the unknown relations between the nodes. The diversity and *un*structured nature of the data challenges our ability to derive models from first principles; alternatively, because data is abundant, it is of great significance to develop methodologies that, in collaboration with domain experts, assist extracting low-dimensional representations that structure the data.

Early work in estimating low-dimensional graph-like structure includes dimensionality reduction approaches such as [5, 6]. These methods work on a static snapshot of data and do not incorporate the notion of time. Other methods [7, 8] consider multiple independent snapshots of data, again not incorporating time. We consider the data to be linked through time, so that the graphs in the corresponding model should represent a dynamic process.

Encapsulating the structure of a graph among entities in time series can be achieved with sparse optimization methods, in which estimated nonzero values correspond to actual edges in the graph, while zeros correspond to absence of edges in the graph. Vector autoregression [9, 10] yields graphs related to Granger causality structure [11] that lets the edges of the graph be further interpreted as corresponding to the existence of relationships between the nodes at the endpoints.

Latent variables

While these sparse methods can be powerful for establishing these interpretable relationships, it is often the case that not all relevant variables to the system can be measured.

This leads to observing relationships among the measured entities that may in fact be caused by common dynamics induced by unobserved variables. One way to address the effects of hidden variables is via sparse plus low-rank optimization [12]. There are of course other non-optimization methods to address these issues as well. However, not just for the beauty of the mathematics but given the types of guarantees available in the field of structured optimization and that we can develop for our problems (which we will later discuss in further detail), we focus on these optimization methods as the basis for our algorithms and analysis.

Time-varying graphs: Principal networks

While we may learn a single network structure for a time series data set, it may be the case that this structure itself is also dynamic over the course of the non-stationary time series. There have been several attempts at extending the methods for estimating single networks to this non-stationary setting [13, 14]. However, in many cases, there may be good reason to believe that the graphs are well described by a time varying linear combination of a set of "principal graphs" that are fundamental to the system. This description would additionally provide a more compact summary of the process. We would then ideally be able to characterize each principal network individually to make sense of the ensemble and analyze the behaviors of the interacting entities.

1.2 Summary of Thesis

This thesis attempts to address these various challenges of learning meaningful networks that describe unstructured time series data. We will expand on these challenges and review existing literature with greater detail within each chapter. We first estimate graph structure from stationary time series. We start with the linear setting with Gaussian noise

and have in mind applications to Graph Signal Processing in chapter 2. Next, we expand towards non-linear processes and address the presence of unobserved variables in chapter 3. Next, we extend the ideas taken from these stationary methods to non-stationary settings to estimate time-varying graphs in chapter 4. Finally, we conclude in chapter 5.

Chapter 2

Causal Graph Processes

This chapter focuses on estimating the network structure capturing the dependencies among time series in the form of a possibly directed, weighted adjacency matrix $\mathbf{A}$. Much of the current work on estimating network structure largely associates it with assuming that the process supported by the graph is Markov along the edges [7, 15] or aims to recover causality [9] in the sense popularized by Granger [11]. This chapter instead associates the graph with causal network effects, drawing inspiration from the Discrete Signal Processing on Graphs (DSP_G) framework [3, 16].

We provide a brief overview of the concepts and notations underlying DSP_G and then introduce related prior work in section 2.1 and our new network process in section 2.2. Next, we present algorithms to infer the network structure from data generated by such processes in section 2.3. We provide analysis on the convergence of our algorithms and the performance of the models for prediction in section 2.4. Finally, we show simulation results in section 2.5 and conclude the chapter in section 2.6.

2.1 Relation to Prior Work

We briefly review DSP_G and then describe previous models and methods used to estimate graph structure. Section 2.1.1 considers sparse Gaussian graphical model selection, Section 2.1.2 sparse vector autoregression, and Section 2.1.3 other graph signal processing approaches.

DSP_G review

Consider a graph $G = (\mathbf{V}, \mathbf{A})$ where the vertex set $\mathbf{V} = \{v_0, \ldots, v_{N-1}\}$ and $\mathbf{A}$ is the weighted adjacency matrix of the graph. A graph signal is defined as a map, $\mathbf{x} : \mathbf{V} \to \mathbb{C}$, $v_n \mapsto x_n$, that we can write as $\mathbf{x} = \begin{pmatrix} x_0 & x_1 & \ldots & x_{N-1} \end{pmatrix}^T \in \mathbb{C}^N$.

We adopt the adjacency matrix $\mathbf{A}$ as a shift [16], and assuming shift invariance and that the minimal polynomial $m_\mathbf{A}(x)$ of $\mathbf{A}$ equals its characteristic polynomial, graph filters in DSP_G are matrix polynomials of the form $h(\mathbf{A}) = h_0\mathbf{I} + h_1\mathbf{A} + \ldots + h_{L_h}\mathbf{A}^{L_h}$.

2.1.1 Sparse Graphical Model Selection

Sparse inverse covariance estimation [8, 15, 17] combines the Markov property with the assumptions of Gaussianity and symmetry using Graphical Lasso [8]. Let the data matrix representing all the K observations is given,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_K \end{pmatrix} \in \mathbb{R}^{N \times K}, \tag{2.1}$$

where $\mathbf{x}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, $\{\mathbf{x}_k\}$ are i.i.d., and an estimate for $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ is desired. The regularized likelihood function is maximized, leading to

$$\widehat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\operatorname{argmin}} \operatorname{tr}(\mathbf{S\Theta}) - \log|\mathbf{\Theta}| + \lambda \|\mathbf{\Theta}\|_1, \tag{2.2}$$

where $\mathbf{S} = \frac{1}{K}\mathbf{X}\mathbf{X}^T$ is the sample covariance matrix and $\|\mathbf{\Theta}\|_1 = \sum_{i,j} |\mathbf{\Theta}_{ij}|$.

Given time series data generated from a sparse graph process, the inverse covariance matrix $\mathbf{\Theta}$ can actually reflect higher order effects and be significantly less sparse than the graph underlying the process. For example, if a process is described by the dynamic equation with sparse state evolution matrix $\mathbf{A}$ and $\|\mathbf{A}\| \leq 1$,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k,$$

where $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\mathbf{w})$ is a random noise process that is generated independently from $\mathbf{w}_j$ for all $i \neq j$, then

$$\mathbf{\Sigma} = \mathbb{E}\left[\mathbf{x}_k \mathbf{x}_k^T\right] = \sum_{i=0}^{\infty} \mathbf{A}^i \mathbf{\Sigma}_\mathbf{w} \left(\mathbf{A}^T\right)^i \Longrightarrow \mathbf{\Theta} = \left(\sum_{i=0}^{\infty} \mathbf{A}^i \mathbf{\Sigma}_\mathbf{w} \left(\mathbf{A}^T\right)^i\right)^{-1}.$$

Even though the process can be described by sparse matrix $\mathbf{A}$, the true value of $\mathbf{\Theta}$ represents powers of $\mathbf{A}$ and need not be as sparse as $\mathbf{A}$. In addition, $\mathbf{A}$ may not be symmetric, i.e., the underlying graph may be directed, while $\mathbf{\Theta}$ is symmetric, and the corresponding graph is undirected.

2.1.2 Sparse Vector Autoregressive Estimation

For time series data, instead of estimating the inverse covariance structure, sparse vector autoregressive (SVAR) estimation [9, 18, 19] recovers matrix coefficients for multivariate processes. This SVAR model assumes the time series at each node are conditionally independent from each other according to a Markov Random Field (MRF) with adjacency structure given by $\mathbf{A}' \in \{0,1\}^{N \times N}$. This $\mathbf{A}'$ is related to Granger causality [20, 21] as shown in [9]. This problem assumes given a data matrix of the form in equation (2.1) that is generated by the dynamic equation with sparse evolution matrices $\{\mathbf{A}_m\}$,

$$\mathbf{x}_k = \sum_{m=1}^{M} \mathbf{A}_m \mathbf{x}_{k-m} + \mathbf{w}_k,$$

where $\mathbf{w}_j$ is a random noise process independent from $\mathbf{w}_j$, $i \neq j$, and $\mathbf{A}_m$ all have the same sparse structure, $\mathbf{A}'_{ij} = 0 \implies [\mathbf{A}_m]_{ij} = 0 \; \forall m$, where $[\mathbf{A}]_{ij}$ is element i, j in matrix $\mathbf{A}$. Then SVAR solves,

$$\{\widehat{\mathbf{A}}_m\} = \underset{\{\mathbf{A}_m\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=M+1}^{K} \left\| \mathbf{x}_k - \sum_{m=1}^{M} \mathbf{A}_m \mathbf{x}_{k-m} \right\|_2^2 + \lambda \sum_{i,j} \|\mathbf{a}_{ij}\|_2, \tag{2.3}$$

where $\mathbf{a}_{ij} = \begin{pmatrix} [\mathbf{A}_1]_{ij} & \dots & [\mathbf{A}_M]_{ij} \end{pmatrix}^T$ and $\|\mathbf{a}_{ij}\|_2$ promotes sparsity of the (i, j)-th entry of each $\mathbf{A}_m$ simultaneously and thus also sparsity of $\mathbf{A}'$. This optimization can be solved using Group Lasso [8]. SVAR estimates multiple weighted graphs $\mathbf{A}_m$ that can be used to estimate the sparsity structure $\mathbf{A}'$.

In contrast, our model is defined by a single weighted $\mathbf{A}$, a potentially more parsimonious model than (2.3). The corresponding time filter coefficients (introduced in section 2.2) are modeled as graph filters. Using this single adjacency matrix $\mathbf{A}$ and graph filters to describe the process enables principled analysis using the toolbox provided by the DSP_G framework.

2.1.3 Graph Signal Processing using the Laplacian

Frameworks for signal processing on graphs using the weighted (and/or possibly normalized) Laplacian matrix rather than the weighted adjacency matrix have been proposed [22, 23, 24]. For example, with a known graph Laplacian matrix, the polynomial coefficients for optimal graph filters can be learned and used to describe and compress signals [25].

The symmetric Laplacian is positive semidefinite with all real nonnegative eigenvalues and a real orthonormal eigenvector matrix. Most Laplacian-based Graph Signal Processing methods assume the Laplacian to be symmetric and implicitly take advantage of these properties in various ways.

Recently, methods for estimating symmetric Laplacians have been proposed [26, 27, 28]. These methods estimate Laplacians for independent graph signals with an interpretation similar to the (inverse) covariance matrix in section 2.1.1, and do not take into account the temporal structure and dependencies of the data. In addition, these methods all depend implicitly on the symmetry of the Laplacian, which yields an orthonormal eigenbasis. Furthermore, they depend on the conic geometry of the space of symmetric positive semidefinite matrices, which allows the utilization of convex optimization.

Symmetric Laplacians correspond to undirected graphs, having nonnegative real eigenvalues. This may be restrictive in applications, since it assumes symmetric relations among time series data. Asymmetric Laplacians are also now being studied, but they are restricted to having zero row (or column) sums, which is often undesirable.

In contrast, the directed adjacency matrix $\mathbf{A}$ that we assume may have positive as well as negative weights on edges and can have complex eigenvalues. We add that knowing the adjacency matrix does allow us to compute the Laplacian. In this chapter, we adopt the adjacency matrix as the basic building block.

2.2 Causal Graph Processes

Consider $x_{n,k}$, a discrete time series on node v_n in graph $G = (\mathbf{V}, \mathbf{A})$, where n indexes the nodes of the graph and k indexes the time samples. Let N be the total number of nodes and K be the total number of time samples, and

$$\mathbf{x}_k = \begin{pmatrix} x_{0,k} & x_{1,k} & \dots & x_{N-1,k} \end{pmatrix}^T \in \mathbb{C}^N$$

represents the graph signal at time sample k.

We consider a Causal Graph Process (CGP) [29, 30, 31, 32] to be a discrete time series

$\mathbf{x}_k$ on a graph $G = (\mathbf{V}, \mathbf{A})$ of the following form,

$$\begin{aligned}\mathbf{x}_k =& \mathbf{w}_k + \sum_{i=1}^{M} P_i(\mathbf{A}, \mathbf{c})\mathbf{x}_{k-i} = \mathbf{w}_k + \sum_{i=1}^{M} \left(\sum_{j=0}^{i} c_{ij}\mathbf{A}^j \right) \mathbf{x}_{k-i} \\ =& \mathbf{w}_k + (c_{10}\mathbf{I} + c_{11}\mathbf{A})\mathbf{x}_{k-1} + \left(c_{20}\mathbf{I} + c_{21}\mathbf{A} + c_{22}\mathbf{A}^2\right)\mathbf{x}_{k-2} + \dots \\ &+ \left(c_{M0}\mathbf{I} + \dots + c_{MM}\mathbf{A}^M\right)\mathbf{x}_{k-M},\end{aligned} \tag{2.4}$$

where $P_i(\mathbf{A}, \mathbf{c})$ is a matrix polynomial in $\mathbf{A}$, $\mathbf{w}_k$ is statistical noise, c_{ij} are scalar polynomial coefficients, and

$$\mathbf{c} = \begin{pmatrix} c_{10} & c_{11} & \dots & c_{ij} & \dots & c_{MM} \end{pmatrix}^T$$

is a vector collecting all the c_{ij}'s.

Note that the CGP model does *not* assume Markovianity in the nodes and edges of the graph adjacency matrix. Instead, the CGP is an autoregressive process (Markov) in the time series as in (2.4) whose coefficients $P_i(\mathbf{A}, \mathbf{c})$ are graph filters; thus, the CGP can incorporate the influence of many more (sometimes all) other nodes in a single step. Matrix polynomial $P_i(\mathbf{A}, \mathbf{c})$ is at most of order $\min(i, N_{\mathbf{A}})$, reflecting that $\mathbf{x}_k$ cannot be influenced by more than i-th order network effects from i time steps ago and in addition is limited (mathematically) by $N_{\mathbf{A}}$, the degree of the minimum polynomial of $\mathbf{A}$. Typically, we take the model order $M \ll N_{\mathbf{A}}$, and for the remainder of the chapter assume this holds for sake of notational clarity.

This model captures the intuition that activity on the network travels at some fixed speed (one graph shift per sampling period), and thus the activity at the current time instant at a given network node cannot be affected by network effects of order higher than that speed allows. In this way, the CGP model can be seen as generalizing the spatial dimension of the light cone [33] to be on a discrete manifold rather than only on a lattice corresponding to uniformly sampled space.

The current parameterization of the CGP model in (2.4) raises issues with identifiability. To address them, we assume that $P_1(\mathbf{A}, \mathbf{c}) \neq \alpha\mathbf{I}$ for $\alpha \in \mathbb{R}$. Then, without loss of generality, we can let $c_{10} = 0$ and $c_{11} = 1$ so that $P_1(\mathbf{A}, \mathbf{c}) = \mathbf{A}$. To verify this, consider the full parameterization using $(\mathbf{A}', \mathbf{c}')$. We show that we can instead use the reduced parameterization $(\mathbf{A}, \mathbf{c})$ with $P_1(\mathbf{A}, \mathbf{c}) = \mathbf{A}$ to represent the same process. First, we start with

$$P_1(\mathbf{A}', \mathbf{c}') = c'_{10}\mathbf{I} + c'_{11}\mathbf{A}' = \mathbf{A} = P_1(\mathbf{A}, \mathbf{c}) \Longrightarrow \mathbf{A}' = (c'_{11})^{-1}(\mathbf{A} - c'_{10}\mathbf{I}). \tag{2.5}$$

We can invert c_{11}, since by assumption $c'_{11} \neq 0$. Then consider the i-th polynomial,

$$\begin{aligned} P_i(\mathbf{A}', \mathbf{c}') &= \sum_{j=0}^{i} c'_{ij}(\mathbf{A}')^j = \sum_{j=0}^{i} c'_{ij}(c'_{11})^{-j}(\mathbf{A} - c'_{10}\mathbf{I})^j \\ &= \sum_{j=0}^{i} c'_{ij}(c'_{11})^{-j} \sum_{k=0}^{j} \binom{j}{k} (-c'_{10})^{j-k}\mathbf{A}^k \\ &= \sum_{k=0}^{i} \sum_{j=k}^{i} c'_{ij}(c'_{11})^{-j} \binom{j}{k} (-c'_{10})^{j-k}\mathbf{A}^k \\ &= \sum_{k=0}^{i} c_{ik}\mathbf{A}^k = P_i(\mathbf{A}, \mathbf{c}), \end{aligned} \tag{2.6}$$

when we define

$$c_{ik} = \sum_{j=k}^{i} c'_{ij}(c'_{11})^{-j} \binom{j}{k} (-c'_{10})^{j-k}.$$

In the remainder of this chapter, we assume that $P_1(\mathbf{A}, \mathbf{c}) \neq \alpha\mathbf{I}$ and use the reduced parameterization with $c_{10} = 0$ and $c_{11} = 1$ so that $\mathbf{c} \in \mathbb{R}^n$ where $n = (M-1)(M+4)/2$ to ensure that $\mathbf{A}$ and $\mathbf{c}$ are uniquely specified without ambiguity.

2.3 Estimating Adjacency Matrices

Given a time series $\mathbf{x}(t)$ on graph $G = (\mathbf{V}, \mathbf{A})$ with *unknown* $\mathbf{A}$, we wish to estimate the adjacency matrix $\mathbf{A}$. We assume the data follows the CGP model (2.4). A first approach to its estimation can be formulated as the following optimization problem,

$$(\mathbf{A}, \mathbf{c}) = \underset{\mathbf{A},\mathbf{c}}{\operatorname{argmin}} \ \frac{1}{2} \sum_{k=M}^{K-1} \left\| \mathbf{x}_k - \sum_{i=1}^{M} P_i(\mathbf{A}, \mathbf{c}) \mathbf{x}_{k-i} \right\|_2^2 + \lambda_1 \|\mathrm{vec}(\mathbf{A})\|_1 + \lambda_2 \|\mathbf{c}\|_1, \tag{2.7}$$

where $\mathrm{vec}(\mathbf{A})$ stacks the columns of the matrix $\mathbf{A}$.

In equation (2.7), the first term in the right hand side models $\mathbf{x}_k$ by the CGP model (2.4) in section 2.2, the regularizing term $\lambda_1 \|\mathrm{vec}(\mathbf{A})\|_1$ promotes sparsity of the estimated adjacency matrix, and the term $\lambda_2 \|\mathbf{c}\|_1$ promotes sparsity in the matrix polynomial coefficients. Regularizing $\mathbf{c}$ corresponds to performing autoregressive model order selection. If the true model has $P_i(\mathbf{A}, \mathbf{c}) = \mathbf{0}$ $\forall j \leq i \leq M$ for some $0 < j < M$, then regularization of $\mathbf{c}$ encourages the corresponding values to be $\mathbf{0}$. The matrix polynomial in the first term makes this problem nonconvex. That is, using a convex optimization based approach to solve (2.7) directly may result in finding a solution $(\widehat{\mathbf{A}}, \widehat{\mathbf{c}})$, minimizing the objective function locally, that is not near the true globally minimizing $(\mathbf{A}, \mathbf{c})$.

Instead, we break this estimation down into three separate, more tractable steps:

1. Solve for $\mathbf{R}_i = P_i(\mathbf{A}, \mathbf{c})$
2. Recover the structure of $\mathbf{A}$
3. Estimate c_{ij}

2.3.1 Solving for $P_i(\mathbf{A}, \mathbf{c})$

As previously stated, the graph filters $P_i(\mathbf{A}, \mathbf{c})$ are polynomials of A and are thus shift-invariant and must mutually commute. Then their commutator

$$[P_i(\mathbf{A}, \mathbf{c}),\ P_j(\mathbf{A}, \mathbf{c})] = P_i(\mathbf{A}, \mathbf{c})P_j(\mathbf{A}, \mathbf{c}) - P_j(\mathbf{A}, \mathbf{c})P_i(\mathbf{A}, \mathbf{c}) = 0, \quad \forall i, j.$$

Let $\mathbf{R}_i = P_i(\mathbf{A}, \mathbf{c})$, $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_M)$, and $\widehat{\mathbf{R}}_i$ be the estimate of $P_i(\mathbf{A}, \mathbf{c})$. This leads to the optimization problem,

$$\widehat{\mathbf{R}} = \underset{\mathbf{R}}{\operatorname{argmin}}\ \frac{1}{2}\sum_{k=M}^{K-1}\left\|\mathbf{x}_k - \sum_{i=1}^{M}\mathbf{R}_i\mathbf{x}_{k-i}\right\|_2^2 + \lambda_1\left\|\operatorname{vec}(\mathbf{R}_1)\right\|_1 + \lambda_3\sum_{i\neq j}\left\|[\mathbf{R}_i, \mathbf{R}_j]\right\|_F^2. \tag{2.8}$$

While this is still a non-convex problem, it is multi-convex. That is, when $\mathbf{R}_{-i} = \{\mathbf{R}_j : j \neq i\}$ (all $\mathbf{R}_j$ except for $\mathbf{R}_i$) are held constant, the optimization is convex in $\mathbf{R}_i$. This naturally leads to block coordinate descent as a solution,

$$\widehat{\mathbf{R}}_i = \underset{\mathbf{R}_i}{\operatorname{argmin}}\ \frac{1}{2}\sum_{k=M}^{K-1}\left\|\mathbf{x}_k - \sum_{j=1}^{M}\mathbf{R}_j\mathbf{x}_{k-j}\right\|_2^2 + \lambda_1\|\operatorname{vec}(\mathbf{R}_1)\|_1 + \lambda_3\sum_{j\neq i}\left\|[\mathbf{R}_i, \mathbf{R}_j]\right\|_F^2. \tag{2.9}$$

Each of these sub-problems for estimating $\mathbf{R}_i$ in a single sweep of estimating $\mathbf{R}$ is formulated as an ℓ_1-regularized least-squares problem that can be solved using standard gradient-based methods [34]. We compute a rough estimate of the computational cost for solving (2.9). Assume that finding the gradient is the most costly step in the optimization, and naively estimate matrix-matrix products to take $O(N^3)$ operations; then the optimization has worst-case complexity $O(M^2N^3 + KMN^2)$ incurred from minimizing over M separate blocks. The cost of the i-th problem is dominated in each iteration by $K-M$ matrix-vector products $\mathbf{R}_i\mathbf{x}_{k-i}$ with worst case total complexity $O((K-M)N^2)$ and by matrix-matrix products $\mathbf{R}_i\mathbf{R}_j$ for $j \neq i$ with worst case complexity $O((M-1)N^3)$. We now improve these

cost estimates by noting that, due to the sparsity in the $\mathbf{R}_i$ matrices, the factor of $O(MN^3)$ resulting from matrix multiplications can be reduced to $O(MS_N^2)$ when implemented using sparse matrix multiplications, where S_N is the sparsity of the $\mathbf{R}_i$ (i.e., $\max_i \|\mathbf{R}_i\|_0 = S_N$). In addition, the sparse matrix-vector products can also be reduced to $O(MS_N)$. Then, the total complexity is better estimated to be $O(MS_N^2)$, which scales more amenably for large data applications.

2.3.2 Recovering A

After obtaining estimates $\widehat{\mathbf{R}}_i$, we find an estimate for $\mathbf{A}$. One approach is to take $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1$. This appears to ignore the information from the remaining $\widehat{\mathbf{R}}_i$. However, this information is taken into account during the iterations when solving for $\widehat{\mathbf{R}}$, especially if we begin one new sweep to estimate $\mathbf{R}_1$ using (2.9) with $i = 1$. A second approach is also possible, explicitly using all the $\widehat{\mathbf{R}}_i$ together to find $\mathbf{A}$,

$$\widehat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \ \left\|\widehat{\mathbf{R}}_1 - \mathbf{A}\right\|_2^2 + \lambda_1 \|\operatorname{vec}(\mathbf{A})\|_1 + \lambda_3 \sum_{i=2}^{M} \left\| \left[\mathbf{A}, \widehat{\mathbf{R}}_i\right] \right\|_F^2 . \tag{2.10}$$

This can be seen as similar to running one additional step further in the block coordinate descent to find $\widehat{\mathbf{R}}_1$, except that this approach does not explicitly use the data. This has worst-case complexity $O(MN^3)$ dominated by matrix-matrix products $\mathbf{A}\mathbf{R}_i$ for $i \neq 1$. However, when matrices $\mathbf{A}$ and $\mathbf{R}_i$ are sparse, we again have reduced complexity of $O(MS_N^2)$.

2.3.3 Estimating c

We can estimate c_{ij} in one of two ways: we can estimate $\mathbf{c}$ either from $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{R}}_i$ or from $\widehat{\mathbf{A}}$ and the data $\mathbf{X}$.

To estimate c_{ij} from $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{R}}$, let

$$\begin{aligned}\mathbf{Q}_i &= \begin{pmatrix}\text{vec}(\mathbf{I}) & \text{vec}(\widehat{\mathbf{A}}) & \dots & \text{vec}(\widehat{\mathbf{A}}^i)\end{pmatrix} \\ \mathbf{c}_i &= \begin{pmatrix}c_{i0} & c_{i1} & \dots & c_{ii}\end{pmatrix}^T .\end{aligned}$$

We set up the optimization,

$$\widehat{\mathbf{c}}_i = \underset{\mathbf{c}_i}{\text{argmin}} \frac{1}{2} \left\| \text{vec}\left(\widehat{\mathbf{R}}_i\right) - \mathbf{Q}_i \mathbf{c}_i \right\|_2^2 + \lambda_2 \|\mathbf{c}_i\|_1. \tag{2.11}$$

Alternatively, to estimate c_{ij} from $\widehat{\mathbf{A}}$ and the data $\mathbf{X}$, we can use the optimization,

$$\widehat{\mathbf{c}} = \underset{\mathbf{c}}{\text{argmin}} \ \frac{1}{2} \left\| \mathbf{Y}(\widehat{\mathbf{A}}) - \mathbf{B}(\widehat{\mathbf{A}})\mathbf{c} \right\|_F^2 + \lambda_2 \|\mathbf{c}\|_1 \tag{2.12}$$

where

$$\begin{aligned}\mathbf{Y}(\widehat{\mathbf{A}}) &= \text{vec}\left(\mathbf{X}_M - \widehat{\mathbf{A}}\mathbf{X}_{M-1}\right) \\ \mathbf{B}(\widehat{\mathbf{A}}) &= \begin{pmatrix}\text{vec}\left(\mathbf{X}_{M-2}\right) & \dots & \text{vec}\left(\widehat{\mathbf{A}}^j \mathbf{X}_{M-i}\right) & \dots & \text{vec}\left(\widehat{\mathbf{A}}^M \mathbf{X}_0\right)\end{pmatrix} \\ \mathbf{X}_m &= \begin{pmatrix}\mathbf{x}_m & \mathbf{x}_{m+1} & \dots & \mathbf{x}_{m+K-M-1}\end{pmatrix},\end{aligned}$$

where in $\mathbf{B}(\widehat{\mathbf{A}})$ the i and j indices increment in correspondence to the indexing of c_{ij} in $\mathbf{c}$. That is, first set $i = 2$; then j is incremented from $0, \dots, i$, and i is incremented, and this repeats until $(i, j) = (M, M)$. Then (2.12) can also be solved using standard ℓ_1-regularized least squares methods with worst-case complexity $O(M^4)$ per iteration, dominated by computing (dense) matrix-vector products with $\mathbf{B}^\top \mathbf{B} \in \mathbb{R}^{n \times n}$ and $\mathbf{B}^\top \mathbf{Y} \in \mathbb{R}^n$, with $n = (M-1)(M+4)/2 = O(M^2)$. While this may seem daunting, we note that M, the lag order of the autoregressive process, is usually much lower than the total number of time samples K, so that the optimization to find $\mathbf{c}$ is unlikely to be the bottleneck in the overall model estimation.

2.3.4 Base Estimation Algorithm

The methods discussed so far can be interpreted as assuming that 1) the process is a linear autoregressive process driven by white Gaussian noise and 2) the elements in parameters $\mathbf{A}$ and $\mathbf{c}$ a priori follow zero-mean Laplace distributions. Under these assumptions, the objective function in (2.7) approximately corresponds to the log posterior density and its optimization to an approximate maximum a posteriori (MAP) estimate.

This framework can be extended to estimate more general autoregressive processes, such as those with a non-Gaussian noise model and certain forms of nonlinear dependence of the current state on past values of the state. In this case, we can formulate the general optimization as

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{c}}) = \underset{\mathbf{A},\mathbf{c}}{\operatorname{argmin}} \, f_1\Big(\operatorname{vec}(\mathbf{X}_M), \operatorname{vec}\Big(\sum_{i=1}^{M} P_i(\mathbf{A}, \mathbf{c})\mathbf{X}_{M-i}\Big)\Big) + g_1\big(\mathbf{A}\big) + g_2(\mathbf{c}), \tag{2.13}$$

where matrices $\mathbf{X}_m$ are defined under (2.12), $f_1(\cdot, \cdot)$ is a loss function that corresponds to a log-likelihood function dictated by the noise model, and $g_1(\cdot)$ and $g_2(\cdot)$ are regularization functions (usually convex norms) that correspond to log-prior distributions imposed on the parameters and are dictated by modeling assumptions. Again, the matrix polynomials $P_i(\mathbf{A}, \mathbf{c})$ introduce nonconvexity, so similarly as before, we can separate the estimation into three steps to reduce complexity. This leads to analogous formulations for the optimization problems (2.7)-(2.12) that we omit for brevity and clarity. In the remainder of this chapter, we refer to the specific formulations given in (2.7)-(2.12).

Algorithm 1 summarizes the 3-step algorithm outlined in this section for obtaining estimates $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{c}}$ for the adjacency matrix and filter coefficients; it is a more efficient and well-behaved alternative to directly using (2.13).

We call this 3-step procedure the base algorithm. In Algorithm 1, superscripts denote the iteration number, $\widehat{\mathbf{R}}_{<i}^{(t)}$ denotes $\{\widehat{\mathbf{R}}_j^{(t)} : j < i\}$ and likewise $\widehat{\mathbf{R}}_{>i}^{(t)}$ denotes $\{\widehat{\mathbf{R}}_j^{(t)} : j > i\}$.

Algorithm 1 Base estimation algorithm

Initialize, $t = 0$, $\widehat{\mathbf{R}}^{(t)} = \mathbf{0}$

while $\widehat{\mathbf{R}}^{(t)}$ not converged **do**

  for $i = 1 : M$ **do**

    Find $\widehat{\mathbf{R}}_i^{(t)}$ with fixed $\widehat{\mathbf{R}}_{<i}^{(t)}$, $\widehat{\mathbf{R}}_{>i}^{(t-1)}$ using (2.9).

  end for

  $t \leftarrow t + 1$

end while

Set $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1^{(t)}$ or estimate $\widehat{\mathbf{A}}$ from $\widehat{\mathbf{R}}^{(t)}$ using (2.10).

Solve for $\widehat{\mathbf{c}}$ from $\mathbf{X}$, $\widehat{\mathbf{A}}$ using (2.11) or from $\mathbf{X}$, $\widehat{\mathbf{R}}$ using (2.12).

2.3.5 Simplified Estimation Algorithm

The base algorithm can still be moderately expensive to evaluate computationally when scaling to larger problems and is difficult to analyze theoretically, mainly due to the nonconvexity of the commutativity-enforcing term. For further ease of computation and analysis, we consider a simplified version of the base algorithm in which the commutativity term of the optimization problem (2.9) is removed,

$$\widehat{\mathbf{R}} = \underset{\mathbf{R}}{\operatorname{argmin}} \; f_1\Big(\operatorname{vec}(\mathbf{X}_M), \operatorname{vec}\Big(\sum_{i=1}^{M} \mathbf{R}_i \mathbf{X}_{M-i}\Big)\Big) + \sum_{i=1}^{M} g_{1i}\big(\mathbf{R}\big). \tag{2.14}$$

Note that we have recombined the optimization to be joint over all the $\mathbf{R}_i$, since this is convex without the commutativity term. This can be followed by the same steps to find $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{c}}$ as in the base algorithm, taking $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1$ or as the solution to (2.10), and then solving (2.11) or (2.12) for $\widehat{\mathbf{c}}$. We call this the Simplified Algorithm, described in Algorithm 2. Using standard solvers for the estimation in (2.14) with ℓ_2 and ℓ_1 norms as seen before [34], the worst-case cost is $O(M(K-M)N^2)$, dominated by computing the matrix-vector product $\mathbf{R}_i \mathbf{x}_{k-i}$ for $i = 1, \ldots, M$ and for $k = M, \ldots, K$. Again, by

implementing sparse matrix-vector products, we can reduce the complexity to $O(M(K - M)S_{MN})$ in the best case.

Algorithm 2 Simplified estimation algorithm

1: Initialize $\widehat{\mathbf{R}} = \mathbf{0}$

2: Estimate $\widehat{\mathbf{R}}$ using (2.14)

3: Set $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1$ or estimate $\widehat{\mathbf{A}}$ from $\widehat{\mathbf{R}}$ using (2.10).

4: Solve for $\widehat{\mathbf{c}}$ from $\mathbf{X}$, $\widehat{\mathbf{A}}$ using (2.11) or from $\mathbf{X}$, $\widehat{\mathbf{R}}$ using (2.12).

2.3.6 Extended Estimation Algorithm

As an extension of the base algorithm, we can also choose the estimated matrix $\widehat{\mathbf{A}}$ and filter coefficients $\widehat{\mathbf{c}}$ to initialize the direct approach of using (2.13). Starting from these initial estimates, we may find better local minima than with initializations at $\mathbf{A} = \mathbf{0}$ and $\mathbf{c} = \mathbf{0}$ or at random estimates. We call this procedure the extended algorithm, summarized in Algorithm 3.

Algorithm 3 Extended estimation algorithm

1: Initialize $\mathbf{A}^{(0)} = \mathbf{0}$ and $\mathbf{c}^{(0)} = \mathbf{0}$.

2: Estimate $\mathbf{A}^{(0)}$, $\mathbf{c}^{(0)}$ using basic algorithm.

3: Find local minimum $\widehat{\mathbf{A}}$, $\widehat{\mathbf{c}}$ using initialization $\mathbf{A}^{(0)}$, $\mathbf{c}^{(0)}$ from nonconvex problem (2.13) using convex methods to find a local optimum.

2.4 Convergence of Estimation

In this section, we discuss the convergence of the basic, extended, and simplified algorithms described above. Convergence of the Base Algorithm, Algorithm 1, and of the Extended

Algorithm, Algorithm 3 follow by direct application of the results available in literature, as discussed in Sections 2.4.1 and 2.4.2. Performance of the Simplified Algorithm, Algorithm 2, is proven in Section 2.4.3.

2.4.1 Base Estimation Algorithm

In estimating $\mathbf{A}$ and $\mathbf{c}$, as illustrated in equations (2.10) and (2.12), the forms of the optimization problems are well studied when choosing ℓ_2 and ℓ_1 norms as loss and regularization functions [10, 35]. However, using these same norms, step 2 (the while loop) of the Base Algorithm is a nonconvex optimization. Hence, we would like to ensure that step 2 converges.

When using block coordinate descent for general functions (i.e., repeatedly choosing one block of coordinate directions in which to optimize while holding all other blocks constant), neither the solution nor the objective function values are guaranteed to converge to a global or even a local minimum. However, borrowing results from [36], under some mild assumptions, using block coordinate descent to estimate $\mathbf{R}_i$ will converge. In fact, in equation (2.13), if we assume the objective function to be continuous and to have convex, compact sublevel sets in each coordinate block $\mathbf{R}_i$ (for example, if the functions for f_1, g_1, and g_2 are the ℓ_2, ℓ_1, and ℓ_1 norms respectively as in equation (2.7)), then the block coordinate descent will converge.

2.4.2 Extended Estimation Algorithm

Now we discuss the convergence of the extended estimation algorithm described in Algorithm 3 of section 2.3.6 assuming that in step 1 of Algorithm 3, the Base Algorithm has converged to an initial point $(\mathbf{A}^{(0)}, \mathbf{c}^{(0)})$. We assume that the objective function $F = f_1 + g_1 + g_2$ in equation (2.13) has compact sublevel sets and is bounded below.

Then an iterative (sub-)gradient method with appropriately chosen step sizes that produces updates of $(\mathbf{A}^{(t+1)}, \mathbf{c}^{(t+1)})$ such that $F(\mathbf{A}^{(t+1)}, \mathbf{c}^{(t+1)}) \leq F(\mathbf{A}^{(t)}, \mathbf{c}^{(t)})$, may converge to a local optimum (which is not necessarily the global optimum). If the functions f_1, g_1, and g_2 are the ℓ_2, ℓ_1, and ℓ_1 norms as in equation (2.7), these conditions are satisfied and the algorithm converges to a local optimum.

2.4.3 Simplified Estimation Algorithm

Here, we consider the convergence of the Simplified Algorithm, Algorithm 2. We state in this section formally the underlying assumptions and the results; we also outline the main arguments underlying the proofs. The proofs themselves are detailed in Appendices A.1-A.3. We consider the theoretical performance guarantees provided by Algorithm 2 of the simplified estimate $(\widehat{\mathbf{A}}, \widehat{\mathbf{c}})$ when using ℓ_1 regularized least squares to estimate $\widehat{\mathbf{R}}$,

$$\widehat{\mathbf{R}} = \underset{\mathbf{R}}{\operatorname{argmin}} \ \frac{1}{\sigma_u} \sum_{k=M}^{K-1} \left\| \mathbf{x}_k - \sum_{i=1}^{M} \mathbf{R}_i \mathbf{x}_{k-i} \right\|_2^2 + \lambda_1 \|\mathrm{vec}(\mathbf{R})\|_1, \tag{2.15}$$

which is a special case of (2.14), where $\sigma_u = \|\mathbb{E}[\mathbf{w}_k \mathbf{w}_k^\top\|$, and then using $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1$ and the optimization (2.12) to find $\widehat{\mathbf{c}}$. Dividing by σ_u makes the estimation unitless, although in practice we might incorporate its value into the λ_1 parameter.

Our error metric of interest will be:

$$\epsilon = \mathbb{E}\left[\frac{1}{N} \left\| \mathbf{x}_k - f(\widehat{\mathbf{A}}, \widehat{\mathbf{c}}, \mathbf{X}'_{k-1}) \right\|_2^2 \right] - \mathbb{E}\left[\frac{1}{N} \left\| \mathbf{x}_k - f(\mathbf{A}, \mathbf{c}, \mathbf{X}'_{k-1}) \right\|_2^2 \right], \tag{2.16}$$

where

$$\mathbf{X}'_{k-1} = \begin{pmatrix} \mathbf{x}_{k-1}^\top & \mathbf{x}_{k-2}^\top & \dots & \mathbf{x}_{k-M}^\top \end{pmatrix}^\top \in \mathbb{R}^{MN}$$

and

$$f(\mathbf{A}, \mathbf{c}, \mathbf{X}'_{k-1}) = \sum_{i=1}^{M} P_i(\mathbf{A}, \mathbf{c}) \mathbf{x}_{k-i}.$$

This error ϵ is the average excess prediction risk, the difference between the error of estimating x_k by the estimated CGP (first term in (2.16)) and the variance of the noise $\mathbf{w}_k$ in the CGP given in (2.4) (second term in (2.16)). The expectations in (2.16) are taken over a new sample

$$\mathbf{z}_k = \left(\mathbf{x}_k^\top \ \mathbf{X}_{k-1}'^\top\right)^\top \in \mathbb{R}^{(M+1)N} \tag{2.17}$$

drawn independently of the samples used to estimate $\left(\widehat{\mathbf{A}}, \widehat{\mathbf{c}}\right)$. We now state the assumptions underlying the main results.

Assumptions

First, we list the assumptions we make about the true process in order to derive our performance guarantees:

(A1) The CGP model class (2.4) is accurate:

$$\mathbb{E}[\mathbf{x}_k \,|\, \mathbf{X}_{k-1}'] = f(\mathbf{A}, \mathbf{c}, \mathbf{X}_{k-1}').$$

(A2) The noise process is uncorrelated with the CGP and its own past values: $\mathbb{E}\left[\mathbf{x}_j \mathbf{w}_k^\top\right] = \mathbf{0}$ and $\mathbb{E}\left[\mathbf{w}_j \mathbf{w}_k^\top\right] = \mathbf{0}$ for $j < k$. Further, the noise sequence $\{\mathbf{w}_k\}$ is i.i.d. multivariate Gaussian with distribution $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_w})$, and there exists σ_ℓ such that $0 < \sigma_\ell \leq \left\|\mathbf{\Sigma_w}^{-1}\right\|_2^{-1}$, the singular values of $\mathbf{\Sigma_w}$ are strictly bounded away from 0. Then we can represent the condition number of $\mathbf{\Sigma_w}$ as σ_u/σ_ℓ, where σ_u is given in (2.15).

(A3) The CGP is stationary and is already in steady state when we begin sampling. Under this assumption, the marginal distributions and expectations are $\mathbb{E}[\mathbf{x}_k] = \mathbf{0}$, $\mathbf{\Sigma}_0 \triangleq \mathbb{E}\left[\mathbf{x}_k \, \mathbf{x}_k^\top\right]$, and $\mathbf{\Sigma} \triangleq \mathbb{E}\left[\mathbf{z}_k \mathbf{z}_k^\top\right]$ where $\mathbf{z}_k$ is defined as in (2.17).

(A4) The stationary correlation matrices of the process $\mathbf{x}_k$ are absolutely summable in

induced norm:

$$\sum_{i=-\infty}^{\infty} \left\| \mathbb{E}\left[\mathbf{x}_k \mathbf{x}_{k-i}^\top\right] \right\|_2 = G < \infty.$$

This is a slightly stronger condition than stability (see proof of Lemma 2 in appendix A.1).

(A5) The true adjacency matrix and filter coefficients are sparse and bounded:

$$\left\| \begin{pmatrix} \mathbf{A} & P_2(\mathbf{A}, \mathbf{c}) & \dots & P_M(\mathbf{A}, \mathbf{c}) \end{pmatrix} \right\|_0 \leq S_{MN} \ll MN^2$$

and $\max_{1 \leq i \leq M}(\|\mathbf{A}^i\|_2) \leq L$, $\max_{1 \leq i \leq M}(\|\mathbf{A}^i\|_0) \leq t_N \ll N^2$, and $\|c\|_0 \leq s_M$ and $\|\mathbf{c}\|_1 \leq \rho$, where the matrix ℓ_0-"norm" $\|\mathbf{A}\|_0$ is the total number of nonzero entries in matrix $\mathbf{A}$. The quantities S_{MN} and s_M may grow with M and N, and t_N may grow with N.

(A6) The following holds for the bounds L and ρ in **(A5)**:

$$(1 + L)(1 + \rho) = Q \leq 2.$$

This is also a slightly stronger condition than stability (see proof of Lemma 2 in Appendix A.1).

(A7) The sample size K is large enough relative to the "stability" of the process, the log of the network size, and the sparsity. Also the autoregressive model order M is low relative to the length of the sample size K:

$$T = K - M \geq C\omega^2 \, S_{MN} \, (\log M + \log N)$$

$$M \lesssim o(\log K)$$

for some universal constant $C > 0$ and $\omega = \frac{\sigma_u}{\sigma_\ell} \frac{Q^2}{\mu_{\min}(\widetilde{\mathcal{A}})}$, where ω and $\mu_{\min}(\widetilde{\mathcal{A}})$ are related to measures of "stability" of the process [10], and their explicit forms are given in appendix A.1 in equations (A.1) and (A.3), and Q is given in **(A6)**.

(A8) The matrix $\mathbb{E}[\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})]$ is invertible, or alternatively, its minimum singular value

is strictly positive:

$$\left\| \left(\mathbb{E}[\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})] \right)^{-1} \right\|_2^{-1} \geq \kappa_{\mathbf{B}} NT > 0.$$

We point out that assumptions **(A1)**-**(A3)**, **(A7)**, and **(A8)** correspond to fairly standard assumptions in stationary time series analysis and studying performance of parameter estimation. **(A7)** assumes enough samples to do meaningful estimation, which is standard in high-dimensional estimation. In this assumption, the minimum number of samples is linked to the properties of the process. Assumption **(A8)** makes sure that $\mathbf{c}$ is identifiable when given the true value of $\mathbf{A}$.

As noted **(A4)** and **(A6)** are slightly stronger than "stability." This is because these assumptions are not necessarily implied by stationarity. We note that **(A5)** is perhaps the most restrictive assumption, as it imposes explicit sparsity conditions on the polynomials $P_i(\mathbf{A}, \mathbf{c})$ and vector $\mathbf{c}$. In network science terms, this assumption roughly corresponds to graphs with relatively longer node-to-node paths, so that higher order powers of $\mathbf{A}$ are not all dense. However, this assumption has the same flavor as the explicit sparsity assumptions made in other sparse estimation work. It would be interesting to relax this assumption to recent notions of approximate sparsity rather than exact sparsity, and that could be the direction of future work.

Theoretical Performance

Here, we present the main guarantee and a brief sketch for its proof, providing several lemmas of intermediate results used in the main result. The full proof is presented in Appendices A.1-A.3.

Theorem 1 (Main Result)**.** *For any* $0 < \beta < \nu < 1/2$*, and some universal constant* d_1*,*

*assumptions **(A1)-(A8)** are sufficient for the error ϵ in* (2.16) *to satisfy*

$$\epsilon \leq \left(\delta_{\mathbf{A}}\left(1+(\rho+\delta_{\mathbf{c}})\widehat{L}_M(\delta_{\mathbf{A}})\right)+(1+L)\delta_{\mathbf{c}}\right)^2 tr(\mathbf{\Sigma}_0)/N \tag{2.18}$$

with probability at least $1-\varepsilon_{\mathbf{A}}-\varepsilon_{\mathbf{c}}$, where

$$\begin{aligned}
\widehat{L}_M(\delta) &= \max_{1\leq i\leq M}\frac{(L+\delta)^i-L^i}{\delta}\\
\varepsilon_{\mathbf{A}} &\sim d_1\exp\{-O(K)\}\\
\varepsilon_{\mathbf{c}} &\sim 2\exp\{-O(K^{1-2\nu})\}\\
\delta_{\mathbf{A}} &= O\left(\sqrt{\log N/K}\right)\\
\delta_{\mathbf{c}} &= O\left(\sqrt{\log N/K^{2(\nu-\beta)}}\right).
\end{aligned} \tag{2.19}$$

The exact expressions for $\epsilon_{\mathbf{A}}$, $\epsilon_{\mathbf{c}}$, $\delta_{\mathbf{A}}$, and $\delta_{\mathbf{c}}$ can be found in the appendices. This theorem states that, as the number of nodes N and the number of time observations K grow, with high probability, the average excess prediction risk of the simplified estimate is not too large. The dependence of δ on K and L are through $T=K-M$ and Q, respectively, where L is defined in **(A5)**, and Q is defined in **(A6)**. The AR order M and network size N may also grow, as long as they grow slowly enough with respect to K. Note that, for large K, we have $\delta \ll L$, and the factor $\widehat{L}_M(\delta)=O(ML^{M-1})$. The full proof of Theorem 1 is in Appendix A.3, but we provide a brief overview here.

Before we outline the proof of the theorem, we first present two intermediate results. These two results use sparse vector autoregression estimation results to show that $\|\widehat{\mathbf{A}}-\mathbf{A}\|_2$ and $\|\widehat{\mathbf{c}}-\mathbf{c}\|_1$ are small with high probability. Then, with small errors in estimating $\mathbf{A}$ and $\mathbf{c}$, we can demonstrate that the error ϵ is also small.

Lemma 2. *Assume **(A1)** that $\mathbf{x}_k$ is generated according to the CGP model with $\mathbf{A}$ satisfying **(A5)**. Suppose **(A7)** that the sample size K is large enough. Let $d_i>0$ with $i=1,...,3$ be universal constants, and let $g(Q)=d_3\left(1+\frac{1+Q^2}{(2-Q)^2}\right)$ and $\ell_{MNK}=\sqrt{(\log M+\log N)/K}$.*

With $\varepsilon_{\mathbf{A}} = d_1 \exp\{-d_2 K/\omega^2\}$, $\lambda_1 = 4g(Q)\ell_{MNK}$, *and* $\widehat{\mathbf{R}} = (\widehat{\mathbf{R}}_1, \ldots, \widehat{\mathbf{R}}_M)$ *the solution to* (2.15)*, with probability at least* $1 - \varepsilon_{\mathbf{A}}$, $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1$ *satisfies the inequalities*

$$\|\widehat{\mathbf{A}} - \mathbf{A}\|_1 \leq 256 S_{MN} \ell_{MNK} Q^2 g(Q) \sigma_u / \sigma_\ell$$

$$\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \|\widehat{\mathbf{A}} - \mathbf{A}\|_F \leq \delta_{\mathbf{A}} \stackrel{\Delta}{=} 64\sqrt{S_{MN}} \ell_{MNK} Q^2 g(Q) \sigma_u / \sigma_\ell$$

where $\omega = \frac{\sigma_u}{\sigma_\ell} \frac{Q^2}{\mu_{\min}(\widetilde{\mathcal{A}})}$ *as defined in* ***(A7)****.*

This lemma states that, with the appropriate choice of λ_1, the assumptions **(A1)-(A8)** are sufficient to allow good estimation of $\mathbf{A}$ with high probability. That is, for each increase of sample size K, we choose the optimal value of the regularization parameter λ_1, and this choice yields consistency with high probability.

Lemma 3. *Suppose that assumptions* ***(A1)-(A8)*** *hold. Then for any* $0 < \beta < \nu < 1/2$*, and sparsity* $s_M \leq d_4 K / \log n$ *where* $d_4 > 0$ *is a universal constant, and* $\lambda_2 \geq q_1 = \Theta(N\sqrt{\log N} K^{1-\beta})$*, the solution to* (2.12) *satisfies*

$$P\left(\|\mathbf{c} - \widehat{\mathbf{c}}\|_1 \geq \delta_{\mathbf{c}}\right) \leq \varepsilon_{\mathbf{c}}$$

where

$$\delta_{\mathbf{c}} = O\left(\sqrt{\log N / K^{2(\nu-\beta)}}\right)$$

$$\varepsilon_{\mathbf{c}} \sim 2\exp\{-O(K^{1-2\nu})\}.$$

In a similar vein as Lemma 2, this lemma states that, for appropriate choice of λ_2, the assumptions **(A1)-(A8)** are sufficient to yield a good estimate of $\mathbf{c}$ with high probability.

Proof Overview for Theorem 1. Lemma 2 shows that we can achieve good performance in estimating $\mathbf{A}$ with high probability. With considerable effort and several additional insights, we show that good performance in estimating $\mathbf{A}$, translates into good estimation of $\mathbf{c}$ with high probability in Lemma 3.

Then, small errors $\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}$ and $\|\widehat{\mathbf{c}} - \mathbf{c}\|_1 \leq \delta_{\mathbf{c}}$ naturally lead to small prediction errors ϵ. Concretely, we proceed with some algebra, showing that

$$\begin{aligned}\epsilon &= \frac{1}{N}\mathbb{E}\left[\left\|f\left(\widehat{\mathbf{A}}, \widehat{\mathbf{c}}, \mathbf{X}'_{k-1}\right) - f\left(\mathbf{A}, \mathbf{c}, \mathbf{X}'_{k-1}\right)\right\|_2^2\right] \\ &\leq \frac{1}{N}\mathbb{E}\left[\left(\|f(\widehat{\mathbf{A}}, \widehat{\mathbf{c}}, \mathbf{X}'_{k-1}) - f(\mathbf{A}, \widehat{\mathbf{c}}, \mathbf{X}'_{k-1})\|_2 + \|f(\mathbf{A}, \widehat{\mathbf{c}}, \mathbf{X}'_{k-1}) - f(\mathbf{A}, \mathbf{c}, \mathbf{X}'_{k-1})\|_2\right)^2\right].\end{aligned}$$

Given small estimation errors, we can bound these two norms separately. The first can be bounded using both Lemmas 2 and 3, and the second can be bounded with Lemma 3. Applying the union bound, we arrive at our result. Again, the full detailed proof is in the appendices A.1-A.3. □

2.5 Experiments

We test our algorithms on two types of datasets, a real temperature sensor network time series (with $N = 150$ and $K = 365$) and a synthetically generated time series (with varying N and K). With the temperature dataset, we compare the performance of the different algorithms (1, 2, and 3) for estimating the CGP model (2.4) against that of the sparse vector autoregresssive Markov random field model (2.3) labeled as MRF, where $f_1(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$ and $g_1(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1$; with the synthetic dataset, we compare the estimates of the model parameters to the ground truth graph used to generate the data for Algorithm 2, and we evaluate the performance of the prediction by averaging through Monte Carlo simulations to empirically test Theorem 1.

To solve the regularized least squares iterations for estimating the CGP matrices, we used a fast implementation of proximal quasi-Newton optimization for ℓ_1 regularized problems [37]. To estimate the MRF matrices, we used an accelerated proximal gradient descent algorithm [38] to estimate the SVAR matrix coefficients from equation (2.3) since the code used in [9] is not tested for large graphs.

2.5.1 Temperature Data

The temperature dataset is a collection of daily average temperature measurements taken over 365 days at 150 cities around the continental United States [39]. The time series $\mathbf{x}_i = (x_{i,1} \ \ldots \ x_{i,K})$, $K = 365$, $i = 0, \ldots, 149$, is detrended by a 4th order polynomial at each measurement station i to form $\widetilde{\mathbf{x}}_i$. The data matrix $\mathbf{X}$ is formed from stacking the detrended data $\widetilde{\mathbf{x}}_i$.

We compare the prediction errors of assuming the detrended data $\widetilde{\mathbf{x}}_i$ are generated by the CGP or the MRF models, or by an undirected distance graph as described in [3]. The distance or geometric graph model uses an adjacency matrix $\mathbf{A}^{\text{dist}}$ to model the process

$$\mathbf{x}_k = \mathbf{w}_k + \sum_{i=1}^{M} h_i(\mathbf{A}^{\text{dist}})\mathbf{x}_{k-i},$$

where $h_i(\mathbf{A}^{\text{dist}}) = \sum_{j=0}^{L_h} c_{ji}(\mathbf{A}^{\text{dist}})^j$ are polynomials of order L_h of the distance matrix with elements chosen as

$$\mathbf{A}^{\text{dist}}_{mn} = \frac{e^{-d^2_{mn}}}{\sqrt{\sum_{j\in\mathcal{N}_n^\alpha} e^{-d^2_{nj}} \sum_{\ell\in\mathcal{N}_m^\alpha} e^{-d^2_{m\ell}}}}$$

with $\mathcal{N}_n^\alpha$ representing the neighborhood of the α cities nearest to city n and d_{mn} being the geographical Euclidean distance between cities m and n. In this model, the number of time lags M is taken to be fixed, and the polynomial coefficients c_{ji} are to be estimated. In our experiments, we assumed $M = 2$.

We separated the data into two subsequences, one consisting of the even time indices and one consisting of the odd time indices. One set was used as training data and the other set was left as testing data. This way, the test and training data were both generated from almost the same process but with different daily fluctuations. In this experiment, we compute the prediction MSE as

$$MSE = \frac{1}{N(K-M)} \sum_{i=M+1}^{K} \left\|\mathbf{x}_i - \widehat{\mathbf{x}}_i\right\|^2.$$

Here, since we do not have the ground truth graph for the temperature data, the exper-

iments can be seen as corresponding to the task of prediction. The test error indicates how well the estimated graph and corresponding estimated model can predict the data at the following time instance from past observations. The models are estimated using Algorithms 1, 2, and 3 with

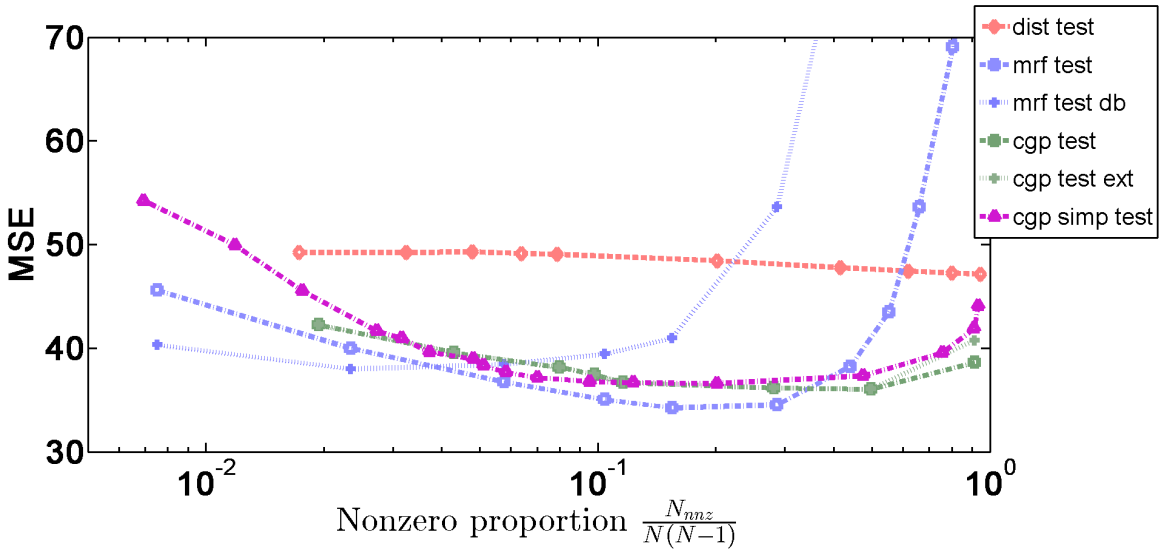
$$f_1(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

$$g_1(\mathbf{x}) = \lambda_1 \|\mathbf{x}\|_1.$$

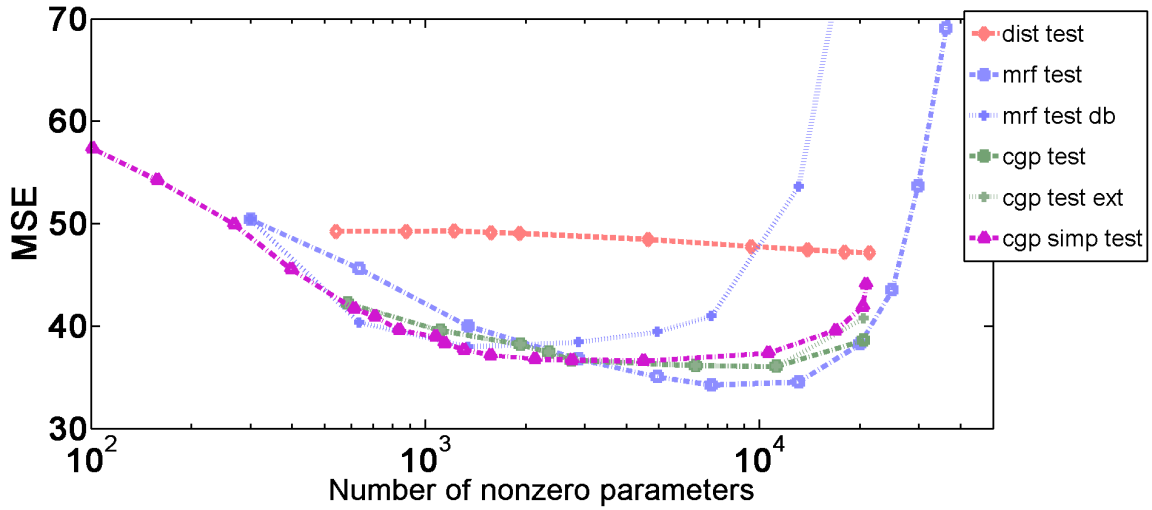
We repeated the training step for all values of $\lambda_1, \lambda_2, \lambda_3$ on a grid discretizing the interval $(0, 500]$. We used the values of the polynomial coefficient regularization parameter λ_2 and the commutativity regularization parameter λ_3 corresponding to the lowest training error to estimate the model on test data and determine the test error; the sparsity regularization parameter λ_1 directly affects the nonzero proportion and number of nonzero parameters, as expected.

Figure 2.1a shows the performance of the basic, extended, and simple Algorithms 1, 2, and 3, as well as the distance graph and MRF models, as a function of edge sparsity of the respective graphs. We see that directed graphs estimated from data (either CGP or MRF) perform better in testing than the undirected distance graphs that are derived independently of the data. In addition, for high sparsity or low number of nonzero entries in the estimated $\widehat{\mathbf{A}}$, ($p_{\text{nnz}} = \frac{N_{\text{nnz}}}{N(N-1)} < 0.3$, where p_{nnz} is the proportion of nonzero edges in the graph and N_{nnz} is the number of nonzero edges in the graph, not including self-edges), the performance of the CGP is competitive with the MRF model. At lower sparsity levels ($p_{\text{nnz}} > 0.3$), i.e., denser graphs, the MRF model performs better than the CGP model in minimizing training error (not shown), but not in testing.

Figure 2.1b shows the prediction performance of the same algorithms as in 2.1a as a function of the total number of nonzero parameters of the respective models. The same trends are present here as in the discussion above. Here, for the CGP model, the number



(a) Testing Errors vs Sparsity



(b) Testing Errors vs Nonzero Parameters

Figure 2.1: Compression and Prediction Error vs Nonzeros using order $M = 2$ model

of nonzero parameters is calculated as $N_{\text{param}}^{CGP} = N_{\text{nnz}}^{CGP} + N + M_{\text{nnz}}^{CGP}$, where N includes the diagonal entries and $M_{\text{nnz}}^{CGP} \leq M(M+1) - 3$ counts the nonzeros in $\mathbf{c}$. For the MRF model, the number of nonzero parameters is calculated as $N_{\text{param}}^{MRF} = M(N_{\text{nnz}}^{MRF} + N)$, where N includes the diagonal entries and the factor of M accounts for the fact that the nonzero entries of $\mathbf{A}^{(i)}$ can be different across i. We can see that for the same level of sparsity,

the MRF model has more nonzero parameters than the CGP model by approximately a factor of M. Here, the CGP has lower test error than MRF with fewer nonzero parameters

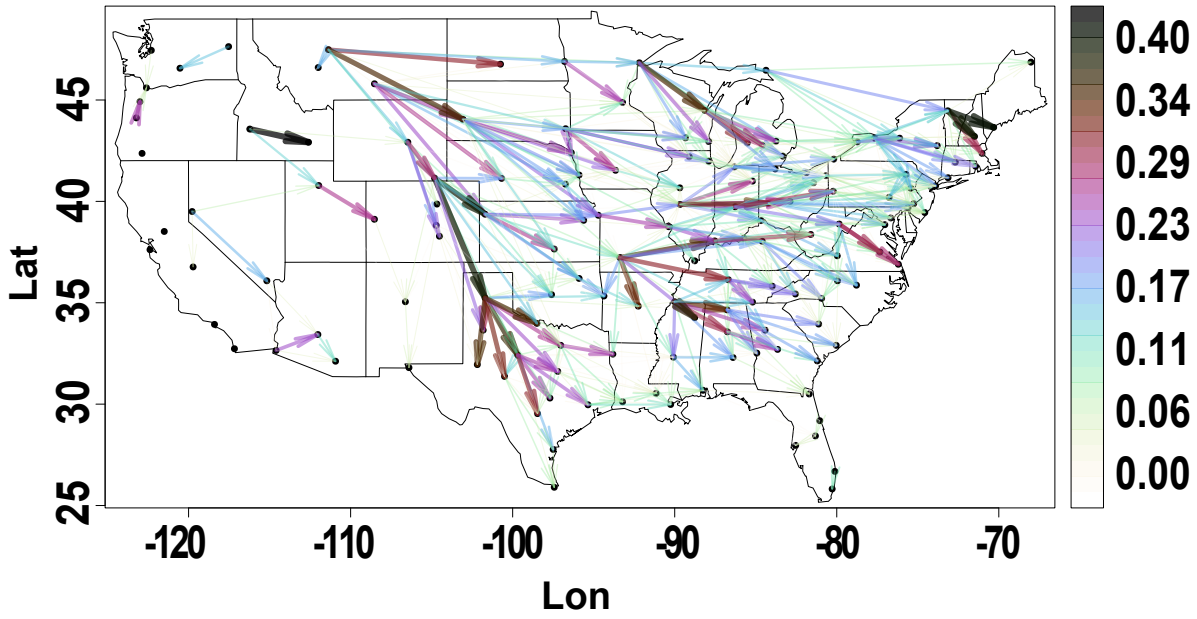


Figure 2.2: Estimated CGP temperature graph using order $M = 2$ model with sparsity $p_{\text{nnz}} = 0.05$

In figure 2.2, we visualize the temperature network estimated on the entire time series using the CGP model that has sparsity level $p_{\text{nnz}} = 0.05$. The x-axis corresponds to longitude while the y-axis corresponds to latitude. We see that the CGP model clearly picks out the predominant west-to-east direction of wind in the $x \geq -95$ portion of the country, as single points in this region are seen to predict multiple eastward points. It also shows the influence of the roughly north-northwest-to-south-southeast Rocky Mountain chain at $-110 \leq x \leq -100$. This CGP graph paints a picture of US weather patterns that is consistent with geographic and meteorological features. This consistency may be the most pleasing and surprising observation of this experiment. This also helps intuitively

explain why the distance graph, which is not derived from data, is not as good at predicting weather trends, since cities that are geometrically close may be geographically separated.

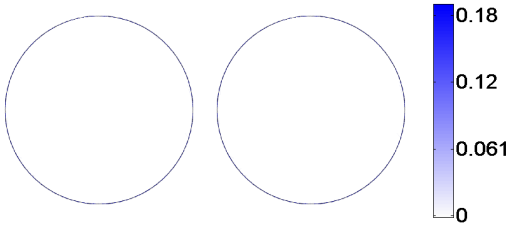
2.5.2 Synthetic Data

We test our simplified algorithm with

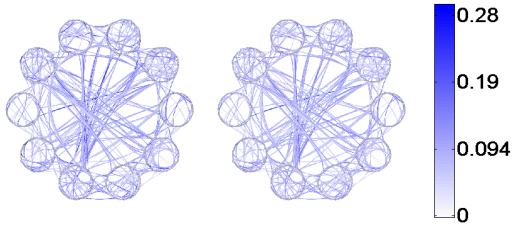
$$f_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$$

$$g_2(\mathbf{x}) = \|\mathbf{x}\|_1$$

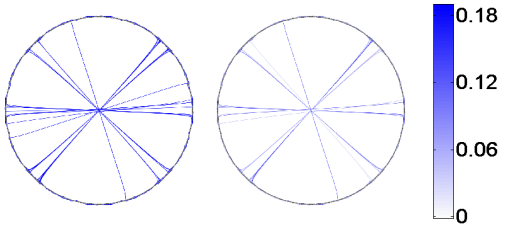
on larger synthetic datasets to empirically verify the theory developed in section 2.4.



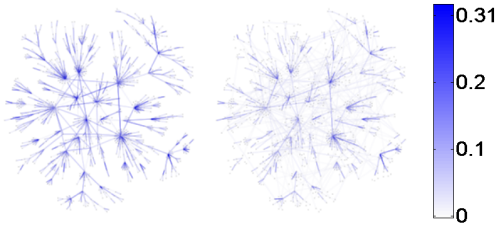
(a) True $\mathbf{A}$ (left) and Estimated $\widehat{\mathbf{A}}$ (right) for K-Regular graph



(b) True $\mathbf{A}$ (left) and Estimated $\widehat{\mathbf{A}}$ (right) for Stochastic Block-Model graph



(c) True $\mathbf{A}$ (left) and Estimated $\widehat{\mathbf{A}}$ (right) for Erdös-Renyi graph



(d) True $\mathbf{A}$ (left) and Estimated $\widehat{\mathbf{A}}$ (right) for Power Law graph

Figure 2.3: True (left) and estimated (right) edge weights (absolute values) for K-Regular (top left), Stochastic Block-Model (top right) Erdös-Renyi (bottom left), and Power Law (bottom right) graphs for N=1000

The random graphs corresponding to $\mathbf{A}$ were generated with 4 different topologies: K-

regular (KR), Stochastic Block-Model (SBM) [40], Erdös-Renyi (ER) [41], and Power Law (PL).

The KR graph was generated by taking a circle graph and connecting each node to itself using a weight of -1 and to its $\xi = 3$ neighbors (we use ξ for this quantity rather than K, since we have used K to denote a different quantity) to each side on the circle with weights drawn from a random uniform distribution $\mathcal{U}(0.5, 1)$. This resulted in a $(2\nu + 1)$-diagonal (in this case heptadiagonal) matrix. Finally the matrix was normalized by 1.5 times its largest eigenvalue.

The SBM graph was generated by creating 10 clusters with each of the 1000 nodes having uniform probability of belonging to a cluster. Edges between nodes were generated according to assigned intra- and inter- cluster probabilities summarized by a 10×10 matrix (intra-cluster probabilities were on the diagonal, and inter-cluster probabilities on the non-diagonal entries). This matrix was generated randomly and sparsely. Starting with $0.05\mathbf{I}$, we added an independent random quantity to each element, where the variables were uniformly distributed on $[0, 0.04)$. Then, entries below 0.025 were thresholded to 0. The edges generated were assigned weights from a Laplacian distribution with rate $\lambda_e = 2$. Finally, the matrix was normalized by 1.1 times its largest eigenvalue.

The ER graph was generated by taking edges from a standard normal $\mathcal{N}(0, 1)$ distribution and then thresholding edges to be between 1.6 and 1.8 in absolute value to yield an effective probability of an edge $p_{ER} \approx 0.04$. The edges were soft thresholded by 1.5 to be between 0.1 and 0.3 in magnitude. Finally, the matrix was normalized by 1.5 times its largest eigenvalue.

The PL graph was generated by starting with a 15 node ER graph with connection probability 0.8. New nodes were connected by two new edges to and from an existing node of weight drawn as $\mathcal{N}(0, 1)$ and then offset 0.25 away from 0. The connections were made

according to a modified preferential attachment scheme [42] in which the probability of the new node connecting to an existing node was proportional to the existing node's total weighted degree. The diagonal was set to $-1/2$. Lastly, the matrix was normalized by 1.5 times its largest eigenvalue.

Examples of these topologies with their weighted edges and representative estimates $\widehat{\mathbf{A}}$ can be seen in figure 2.3 (higher absolute weights are displayed in darker blue). Because of the large number of nodes, these topologies are difficult to visualize. So, *for display purposes only*, we chose parameters that lean towards fewer false alarms at the expense of having more missed edges—since more false alarms obstruct and obscure the layout. For KR, see figure 2.3a, we used a circular layout in which the true edges are along the perimeter and not through the interior. This is replicated in the estimated graph[1]. For the KR graph whose results are in figure 2.3a (false alarm rate (P_{FA}) 3×10^{-4} and miss rate (P_M) 0.16), we used a circular layout (as previously mentioned); for SBM, see figure 2.3b (displayed plot has $P_{FA} = 7 \times 10^{-3}$ and $P_M = 0.22$), and ER, see figure 2.3c (displayed plot with $P_{FA} = 4 \times 10^{-4}$ and $P_M = 0.58$), we used force-directed edge bundling [43] to group nearby edges into few thicker "strands" in addition to circular layouts; and for PL, see figure 2.3d (displayed plot with $P_{FA} = 4 \times 10^{-3}$ and $P_M = 0.69$), we used a Fruchterman-Reingold [44] node positioning. If we lower the value of the sparsity regularization parameter λ_1 on our grid, we can reduce the miss rate, while increasing the false alarm rate. Just for reference, we provide another pair of corresponding P_{FA} and P_M values: for KR, $P_{FA} = 0.01$ and $P_M = 0.07$; for SBM, $P_{FA} = 0.01$ and $P_M = 0.40$; for ER, $P_{FA} = 0.03$ and $P_M = 0.25$; for PL, $P_{FA} = 0.03$ and $P_M = 0.52$. Playing with λ_1 we can get other tradeoffs more suitable to specific applications. Just from these experiments, it seems that the SBM and

[1] In the estimated graph, there are some edges through the interior with much lower weights relative to the weights on the edges along the perimeter, which are not visible when visualized.

PL models are more difficult to estimate than the KR and ER models. This could be due to the average degree of SBM being high and the maximum degree of PL being high while the number of time observations K is held constant throughout these experiments, which would violate Assumption **(A5)** or **(A7)**. A more thorough understanding of these tradeoffs, such as what additional conditions need to be satisfied for relevant guarantees to hold, and what broad classes of graphs satisfy these conditions, is left as a topic of future investigation.

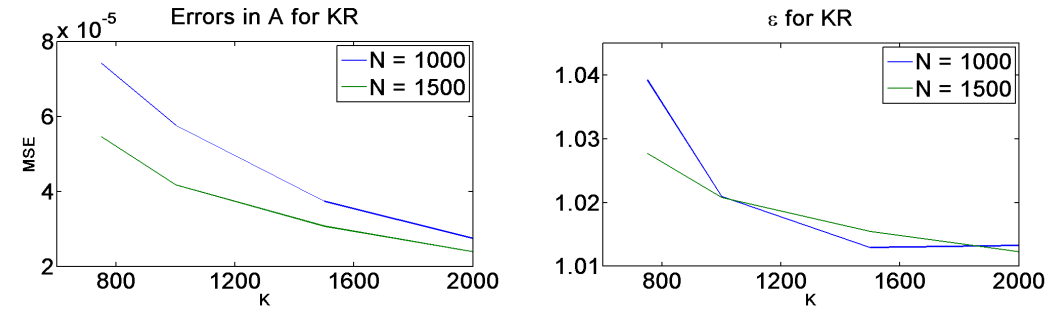
Once the $\mathbf{A}$ matrix was generated with $N \in \{1000, 1500\}$ nodes, for fixed $M = 3$, the coefficients c_{ij} for $2 \leq i \leq M$ and $0 \leq j \leq i$ were generated sparsely from a mixture of uniform distributions $2^{i+j}c_{ij} \sim \frac{1}{2}\mathcal{U}(-1, -0.45) + \frac{1}{2}\mathcal{U}(0.45, 1)$ and normalized by 1.5 to correspond to a stable system.

The data matrices $\mathbf{X}$ were formed by generating random initial samples (with 500 burn in samples to reach steady state) and zero-mean unit-covariance additive white Gaussian noise $\mathbf{w}_k$ computing $K \in \{750, 1000, 1500, 2000\}$ samples of $\mathbf{x}_k$ according to (2.4). Then 20 such Monte-Carlo data matrices were generated independently, and $(\widehat{\mathbf{A}}, \widehat{\mathbf{c}})$ was estimated using the simplified algorithm for varying values of λ_1 and ρ. The average error of the $\mathbf{A}$ matrix was computed as

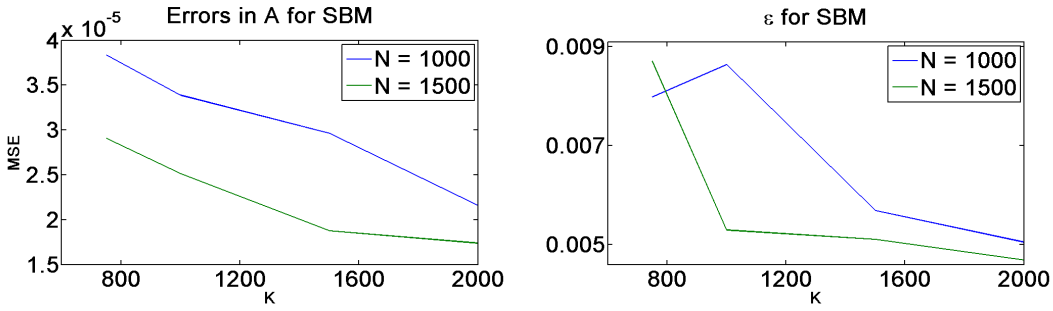
$$MSE = \frac{1}{N^2}\|\mathbf{A} - \widehat{\mathbf{A}}\|_F^2.$$

The empirical value of the error metric $\widehat{\epsilon}$ from (2.16) was also measured by generating another 20 independent sets of time series $\mathbf{z}_k$ (as in (2.17)) at steady state and using the estimates to predict $\mathbf{x}_k$ using $\mathbf{X}'_{k-1}$.

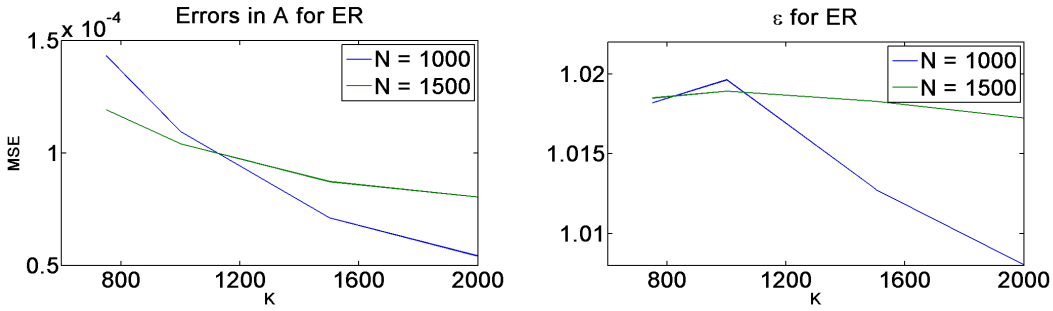
In figure 2.4, we show the errors in $\mathbf{A}$ and the empirical estimates of ϵ for different values of N and K. We performed the estimation across a grid of λ_3 and ρ and choose the lowest observed error to be $\widehat{\epsilon}$. We see several different behaviors. We observe that the average errors in $\mathbf{A}$ and empirical averages for ϵ for the KR graph decrease with both larger N and



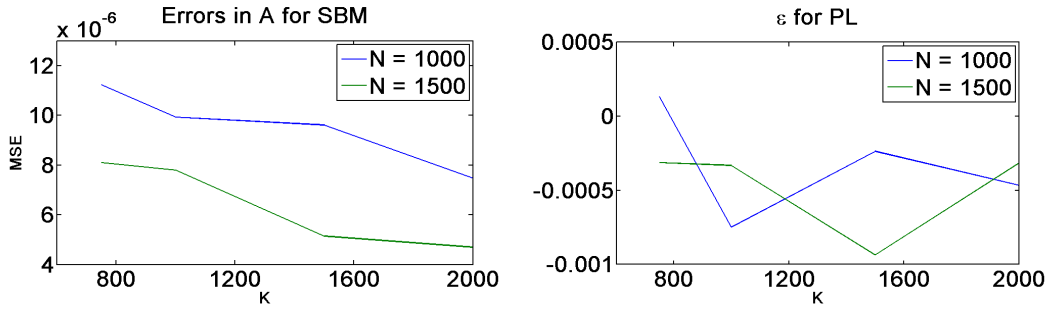
(a) Error in $\mathbf{A}$ (left) and ϵ (right) for KR graph



(b) Error in $\mathbf{A}$ (left) and ϵ (right) for SBM graph



(c) Error in $\mathbf{A}$ (left) and ϵ (right) for ER graph



(d) Error in $\mathbf{A}$ (left) and ϵ (right) for PL graph

Figure 2.4: Error in $\mathbf{A}$ (left) and ϵ (right) for K-Regular (top left), Stochastic Block-Model (top right), Erdös-Renyi (bottom left), and Power Law (bottom right) graphs for N=1000,1500

K. We observe similar behavior for the SBM graph. This suggests that graphs generated according to these topologies might satisfy the assumptions in section 2.4. On the other hand, the error in $\mathbf{A}$ for ER does not display a clear trend in N, although observing more samples still improves the estimate of $\mathbf{A}$ and the model prediction performance as expected. For the PL model, the error in estimating $\mathbf{A}$ decreased with increasing N and K, but the error ϵ fluctuated around 0 with no clear trend.

This varied behavior arises because the different structured and random graph topologies examined tend to exhibit certain network properties that do not all correspond directly to the assumptions in section 2.4. In particular, the K-regular graph by its construction does satisfy the assumptions. This is because taking the n-th power of the adjacency matrix of a ξ-regular graph of this form results in an $(n\xi)$-regular graph, which satisfies the sparsity **(A5)** with some constants that do not grow too fast with N and K. Thus, the results of the KR graph empirically conform to the predictions given by the theory in section 2.4. With the other topologies, the behavior of the sparsity constants is not as immediately clear, but the observed results suggest that some of the assumptions could be slightly loosened, or that other network statistics (e.g., diameter or maximum degree) could play a role in the performance.

2.6 Conclusions

This chapter presents a methodology to estimate the network structure (graph) capturing spatial (inter) and time (intra) dependencies among multiple time series. These data may arise in many different contexts. The data time dependencies are modeled by an auto-regressive (AR) process. The spatial dependencies are captured by describing the matrix coefficients of the AR process as graph polynomial filters [16]. The chapter presents three

algorithms to estimate the graph adjacency matrix and parameters of the graph polynomial filters. These algorithms optimize cost functions that combine a model following functional, e.g., an ℓ_2 error metric, with sparsity regularizers, e.g., ℓ_1 metric. The chapter carries out the convergence and performance analysis of these algorithms under a set of appropriate assumptions. Finally, the chapter illustrates the performance of these algorithms with a set of real data (temperature data collected by 150 weather stations covering the US) and with simulated data for four different types of networks. These experiments show the advantages and limitations of the approach.

Chapter 3

Single Index Latent Variable Models

How real is this relationship? This is a ubiquitous question that presents itself not only in judging interpersonal connections but also in evaluating correlations and causality throughout science and engineering. Two reasons for reaching incorrect conclusions based on observed relationships in collected data are chance and outside influences. For example, we can flip two coins that both show heads, or observe that today's temperature measurements on the west coast of the continental USA seem to correlate with tomorrow's on the east coast throughout the year. In the first case, we might not immediately conclude that coins are related, since the number of flips we observe is not very large relative to the possible variance of the process, and the apparent link we observed is up to chance. In the second case, we still may hesitate to use west coast weather to understand and predict east coast weather, since in reality both are closely following a seasonal trend.

Establishing interpretable relationships between entities while mitigating the effects of chance can be achieved via sparse optimization methods, such as regression (Lasso) [35] and inverse covariance estimation [8]. In addition, the extension to time series via vector autoregression [9, 10] yields interpretations related to Granger causality [11]. In each of these settings, estimated nonzero values correspond to actual relations, while zeros

correspond to absence of relations.

However, we are often unable to collect data to observe all relevant variables, and this leads to observing relationships that may be caused by common links with those unobserved variables. The hidden variables in this model are fairly general; they can possibly model underlying trends in the data, or the effects of a larger network on an observed subnetwork. For example, one year of daily temperature measurements across a country could be related through a graph based on geographical and meteorological features, but all exhibit the same significant trend due to the changing seasons. We have no single sensor that directly measures this trend. In the literature, a standard pipeline is to de-trend the data as a preprocessing step, and then estimate or use a graph to describe the variations of the data on top of the underlying trends [3, 16, 32].

Alternatively, attempts have been made to capture the effects of hidden variables via sparse plus low-rank optimization [17]. This has been extended to time series [12], and even to a non-linear setting via Generalized Linear Models (GLMs) [45]. What if the form of the non-linearity (link function) is not known? Regression using a GLM with an unknown link function is also known as a Single Index Model (SIM). Recent results have shown good performance when using SIMs for sparse regression [46].

So far, when choosing a model, current methods will impose a fixed form for the (non-)linearity, assume the absence of any underlying trends, perform separate pre-processing or partitioning in an attempt to remove or otherwise explicitly handle such trends, or take some combination of these steps. To address all of these issues, we propose a model with a non-linear function applied to a linear argument that captures the effects of latent variables, which manifest as unmodeled trends in the data. Thus, we introduce the Single Index Latent Variable (SILVar) model, which uses the SIM in a sparse plus low-rank optimization setting to enable general, interpretable multi-task regression in the presence

of unknown non-linearities and unobserved variables. That is, we propose the SILVar model not only to use for regression in difficult settings, but also as a tool for uncovering hidden relationships buried in data.

First, we establish notation and review prerequisites in Section 3.1. Next, we introduce the SILVar model and discuss several paradigms in which it can be applied in Section 3.2. Then, we detail the numerical procedure for learning the SILVar model in Section 3.3. Finally, we demonstrate the performance via experiments on synthetic and real data in Section 3.5.

3.1 Background

In this section, we introduce the background concepts and notation used throughout the remainder of the chapter.

3.1.1 Bregman Divergence and Convex Conjugates

For a given convex function ϕ, the Bregman Divergence [47] associated with ϕ between $\mathbf{y}$ and $\mathbf{x}$ is denoted

$$D_\phi(\mathbf{y}\|\mathbf{x}) = \phi(\mathbf{y}) - \phi(\mathbf{x}) - \nabla\phi(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}). \tag{3.1}$$

The Bregman Divergence is a non-negative (asymmetric) quasi-metric. Two familiar special cases of the Bregman Divergence are Euclidean Distance when $\phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ and Kullback-Liebler Divergence when $\phi(\mathbf{x}) = \sum_i x_i \log x_i$ in the case that $\mathbf{x}$ is a valid probability distribution (i.e., $\mathbf{x} \geq 0$ and $\sum_i x_i = 1$).

The convex conjugate ϕ_* of a function ϕ is given by

$$\phi_*(\mathbf{x}) \triangleq \sup_{\mathbf{y}} \ \mathbf{y}^\top\mathbf{x} - \phi(\mathbf{y}). \tag{3.2}$$

The convex conjugate arises naturally in optimization problems when deriving a dual form

for the original (primal) problem. For closed, convex, differentiable, 1-D function ϕ with invertible gradient, the following properties hold

$$\begin{aligned} \phi_*(x) &= x(\nabla\phi)^{-1}(x) - \phi((\nabla\phi)^{-1}(x)) \\ (\nabla\phi)^{-1} &= \nabla\phi_* \qquad\qquad (\phi_*)_* = \phi \end{aligned} \tag{3.3}$$

where $(\cdot)^{-1}$ denotes the inverse function, not the multiplicative inverse. In words, these properties give an alternate form of the conjugate in terms of gradients of the original function, state that the function inverse of the gradient is equal to the gradient of the conjugate, and state that the conjugate of the conjugate is the original function.

3.1.2 Generalized Linear and Single-Index Models

The Generalized Linear Model (GLM) can be described using several parameterizations. We adopt the one based on the Bregman Divergence [48]. For observations $y_i \in \mathbb{R}$ and $\mathbf{x}_i \in \mathbb{R}^p$, let $\mathbf{y} = (y_1 \dots y_n)^\top$, $\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_n)$. The model is parameterized by 1) a non-linear link function $g = (\nabla\phi)^{-1}$ where ϕ is a closed, convex, differentiable, invertible function; and 2) a vector $\mathbf{a} \in \mathbb{R}^p$. We have the model written as

$$\mathbb{E}[y_i|\mathbf{x}_i] = g\left(\mathbf{a}^\top \mathbf{x}_i\right), \tag{3.4}$$

(note that some references use g^{-1} as the link function where we use g) and the corresponding likelihood function written as

$$P(y_i|\mathbf{x}_i) = \exp\left\{-D_\phi\left(y_i \| g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right)\right\} \tag{3.5}$$

where the likelihood is expressed with respect to an appropriate base measure [49], which can be omitted for notational clarity.

Let $G = \phi_*$ and $g = \nabla G = (\nabla\phi)^{-1}$. Then, for data $\{\mathbf{x}_i, y_i\}$ with conditionally independent y_i given $\mathbf{x}_i$ (note that this is not necessarily assuming that $\mathbf{x}_i$ are independent),

learning the model $\mathbf{a}$ assuming g is known can be achieved via empirical risk minimization,

$$\begin{aligned}
\widehat{\mathbf{a}} &= \underset{\mathbf{a}}{\operatorname{argmax}} \prod_{i=1}^{n} \exp\left\{-D_\phi\left(y_i \| g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right)\right\} \\
&= \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=1}^{n} D_\phi\left(y_i \| g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right) \\
&= \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\phi\left(y_i\right) - \phi\left(g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right)\right. \\
&\qquad\qquad \left. -\nabla\phi\left(g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right)\left(y_i - g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right)\right] \\
&\overset{(a)}{=} \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[G_*\left(y_i\right) - y_i\left(\mathbf{a}^\top \mathbf{x}_i\right) - \phi\left(g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right)\right. \\
&\qquad\qquad \left. + \left(\mathbf{a}^\top \mathbf{x}_i\right) g\left(\mathbf{a}^\top \mathbf{x}_i\right)\right] \\
&\overset{(b)}{=} \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[G_*\left(y_i\right) + G\left(\mathbf{a}^\top \mathbf{x}_i\right) - y_i\left(\mathbf{a}^\top \mathbf{x}_i\right)\right] \\
&= \underset{\mathbf{a}}{\operatorname{argmin}}\, \widehat{F}_1\left(\mathbf{y}, \mathbf{X}, g, \mathbf{a}\right)
\end{aligned} \tag{3.6}$$

where equality (a) arises from the second property in (3.3), equality (b) arises from the first property, and we introduce

$$\widehat{F}_1(\mathbf{y}, \mathbf{X}, g, \mathbf{a}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \left[G_*(y_i) + G\left(\mathbf{a}^\top \mathbf{x}_i\right) - y_i\left(\mathbf{a}^\top \mathbf{x}_i\right)\right] \tag{3.7}$$

for notational compactness.

The Single Index Model (SIM) [50] takes the same form as the GLM. The crucial difference is in the estimation of the models. When learning a GLM, the link function g is known and the linear parameter $\mathbf{a}$ is estimated; however when learning a SIM, the link function g needs to be estimated along with the linear parameter $\mathbf{a}$.

Recently, it has been shown that, when the function g is restricted to be monotonic increasing and Lipschitz, learning SIMs becomes computationally tractable [49] with performance guarantees in high-dimensional settings [46]. Thus, with scalar u defining the set

$$\mathcal{C}^u = \{g : \forall y > x,\ 0 \leq g(y) - g(x) \leq u(y - x)\} \tag{3.8}$$

of monotonic increasing u-Lipschitz functions, this leads to the optimization problem,

$$\begin{aligned}(\widehat{g}, \widehat{\mathbf{a}}) &= \underset{g,\mathbf{a}}{\operatorname{argmin}}\, \widehat{F}_1(\mathbf{y}, \mathbf{X}, g, \mathbf{a}) \\ &\text{s.t. } g = \nabla G \in \mathcal{C}^1.\end{aligned} \tag{3.9}$$

3.1.3 Lipschitz Monotonic Regression and Estimating SIMs

The estimation of g with the objective function including terms G and G_* at first appears to be an intractably difficult calculus of variations problem. However, there is a marginalization technique that cleverly avoids directly estimating functional gradients with respect to G and G_* [49] and admits gradient-based optimization algorithms for learning. The marginalization utilizes Lipschitz monotonic regression (LMR) as a subproblem. Thus, before introducing this marginalization, we first review LMR.

LMR

Given ordered pairs $\{x_i, y_i\}$ and independent Gaussian w_i, consider the model

$$y_i = g(x_i) + w_i, \tag{3.10}$$

which intuitively treats $\{y_i\}$ as noisy observations of a function g indexed by x, sampled at points $\{x_i\}$. Let $\widehat{g}_i = g(x_i)$, an estimate of the function value, with $\widehat{\mathbf{g}} = (\widehat{g}_1 \dots \widehat{g}_n)^\top$, and $x_{[j]}$ denote the j^{th} element of the $\{x_i\}$ sorted in ascending order. Then LMR is described by the problem,

$$\begin{aligned}\widehat{\mathbf{g}} \triangleq \operatorname{LMR}(\mathbf{y}, \mathbf{x}) &= \underset{\mathbf{g}}{\operatorname{argmin}} \sum_{i=1}^{n} (g(x_i) - y_i)^2 \\ &\text{s.t. } 0 \le g\left(x_{[j+1]}\right) - g\left(x_{[j]}\right) \le x_{[j+1]} - x_{[j]} \quad \text{for } j = 1, \dots, n-1.\end{aligned} \tag{3.11}$$

While there may be in fact infinitely many possible (continuous) monotonic increasing Lipschitz functions g that pass through the points $\widehat{\mathbf{g}}$, the solution vector $\widehat{\mathbf{g}}$ is unique. We will introduce a simple yet effective algorithm for solving this problem later in Section 3.3.1.

Estimating SIMs

We now return to the objective function (3.9). Let $\widehat{\mathbf{g}} = \text{LMR}(\mathbf{y}, \mathbf{X}^\top \mathbf{a})$. Then the gradient w.r.t. $\mathbf{a}$ can be expressed in terms of an estimate of g without explicit computation of G or G_*,

$$\nabla_{\mathbf{a}} F_1 = \sum_{i=1}^{n} \left[(\widehat{g}_i - y_i) \, \mathbf{x}_i \right]. \tag{3.12}$$

This allows us to apply gradient or quasi-Newton methods to solve the minimization in $\mathbf{a}$, which is itself a convex problem since the original problem was jointly convex in g and $\mathbf{a}$.

3.2 Single Index Latent Variable Models

In this section, we build the Single Index Latent Variable (SILVar) model [51] from fundamental concepts.

3.2.1 Multitask Regression and Latent Variables

First, we extend the SIM to the multivariate case and then examine how latent variables can affect learning of the linear parameter. Let

$$\mathbf{y}_i = (y_{1i} \ \ldots \ y_{mi})^\top$$

$$\mathbf{g}(\mathbf{x}) = (g_1(x_1) \ \ldots \ g_m(x_m))^\top$$

$$\mathbf{A} = (\mathbf{a}_1 \ \ldots \ \mathbf{a}_m)^\top.$$

Consider the vectorization,

$$\mathbb{E}\left[y_{ji} | \mathbf{x}_i\right] = g_j\left(\mathbf{a}_j^\top \mathbf{x}_i\right) \Longrightarrow \mathbb{E}\left[\mathbf{y}_i | \mathbf{x}_i\right] = \mathbf{g}\left(\mathbf{A}\mathbf{x}_i\right). \tag{3.13}$$

For the remainder of this chapter, we make an assumption that all $g_j = g$ for notational simplicity, though the same analysis readily extends to the case where g_j are distinct.

Now, let us introduce a set of latent variables $\mathbf{z}_i \in \mathbb{R}^r$ with $r \ll p$ and the corresponding

linear parameter $\mathbf{B} = (\mathbf{b}_1 \dots \mathbf{b}_m)^\top \in \mathbb{R}^{m \times r}$ (note we can incorporate a linear offset by augmenting the variable $\mathbf{z} \leftarrow (\mathbf{z}^\top \; 1)^\top$ and adding the linear offset as a column of $\mathbf{B}$). This leads to the asymptotic maximum likelihood estimate,

$$\begin{aligned} \left(\overline{g}, \overline{\mathbf{A}}, \overline{\mathbf{B}}\right) = \; & \underset{g, \mathbf{A}, \mathbf{B}}{\operatorname{argmin}} \, F_2\left(\mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, g, \mathbf{A}, \mathbf{B}\right) \\ & \text{s.t. } g = \nabla G \in \mathcal{C}^1, \end{aligned} \tag{3.14}$$

where

$$F_2(\mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, g, \mathbf{A}, \mathbf{B}) \triangleq \mathbb{E}\left[\sum_{j=1}^{m} \left[G_*(y_{ji}) + G(\mathbf{a}_j^\top \mathbf{x}_i + \mathbf{b}_j^\top \mathbf{z}_i)\right] - \mathbf{y}_i^\top (\mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{z}_i)\right]. \tag{3.15}$$

Now consider the case in which the true distribution remains the same, but we only observe $\mathbf{x}_i$ and not $\mathbf{z}_i$,

$$\begin{aligned} (\widehat{g}, \widehat{\mathbf{A}}) = \; & \underset{g, \mathbf{A}}{\operatorname{argmin}} \, F_3(\mathbf{y}_i, \mathbf{x}_i, g, \mathbf{A}) \\ & \text{s.t. } g = \nabla G \in \mathcal{C}^1, \end{aligned} \tag{3.16}$$

where

$$F_3(\mathbf{y}_i, \mathbf{x}_i, g, \mathbf{A}) \triangleq \mathbb{E}\left[\sum_{j=1}^{m} \left[G_*(y_{ji}) + G(\mathbf{a}_j^\top \mathbf{x}_i)\right] - \mathbf{y}_i^\top (\mathbf{A}\mathbf{x}_i)\right]. \tag{3.17}$$

We now propose a relation between the two models in (3.14) and (3.16), which will finally lead to the SILVar model. Here we present the abridged theorem, and relegate the full expressions and derivation to Appendix B.1. To establish notation, let primes ($'$) denote derivatives, hats ($\widehat{\ }$) denote a parameter estimate with only observed variables, overbars ($\overline{\ }$) denote an underlying parameter estimate when we have access to both observed and latent variables, and we drop the subscripts from the random variables $\mathbf{x}$ and $\mathbf{z}$ to reduce clutter.

Theorem 4. *Assume that* $\widehat{g}'(0) \neq 0$ *and that* $|\widehat{g}''| \leq J$ *and* $|\overline{g}''| \leq J$ *for some* $J < \infty$. *Furthermore, assume in models* (3.15) *and* (3.16) *that* $\max_j \left(\|\widehat{\mathbf{a}}_j\|_1, \|\overline{\mathbf{a}}_j\|_1 + \|\overline{\mathbf{b}}_j\|_2\right) \leq k$,

$$\max\left(\mathbb{E}[\|\mathbf{x}\|_2 \|\mathbf{x}\|_\infty^2], \mathbb{E}[\|\mathbf{x}\|_2 \|\mathbf{x}\|_\infty \|\mathbf{z}\|_2], \mathbb{E}[\|\mathbf{x}\|_2 \|\mathbf{z}\|_2^2]\right) \leq s_{Nr},$$

where subscripts in s_{Nr} *indicate that the bounds may grow with the values in the subscript.*

Then, the parameters $\widehat{\mathbf{A}}$ *and* $\overline{\mathbf{A}}$ *from models* (3.15) *and* (3.16) *are related as*

$$\widehat{\mathbf{A}} = q(\overline{\mathbf{A}} + \mathbf{L}) + \mathbf{E},$$

where $q = \frac{\overline{g}'(0)}{\widehat{g}'(0)}$, $\boldsymbol{\mu}_{\mathbf{x}} = \mathbb{E}[\mathbf{x}_i]$,

$$\mathbf{L} = \left(\overline{\mathbf{B}}\mathbb{E}[\mathbf{z}\mathbf{x}^\top] + (\overline{\mathbf{g}}(\mathbf{0}) - \widehat{\mathbf{g}}(\mathbf{0}))\,\boldsymbol{\mu}_{\mathbf{x}}^\top\right)\left(\mathbb{E}[\mathbf{x}\mathbf{x}^\top]\right)^\dagger$$

$$\Rightarrow \mathit{rank}(\mathbf{L}) \leq r + 1,$$

and $\mathbf{E} = \widehat{\mathbf{A}} - q(\overline{\mathbf{A}} + \mathbf{L})$ *is bounded as*

$$\frac{1}{MN}\|\mathbf{E}\|_F \leq \frac{2J\sigma_\ell\sqrt{N}}{\widehat{g}'(0)M} s_{Nr} k^2,$$

where $\sigma_\ell = \left\|\left(\mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]\right)^\dagger\right\|_2$, *the largest singular value of the pseudo-inverse of the covariance.*

The proof is given in Appendix B.1. The assumptions require that the 2nd order Taylor series expansion is accurate around the point $\mathbf{0}$, that the model parameters $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are bounded, and that the distributions generating the data are not too spread out (beyond $\mathbf{0}$ where the Taylor series expansion is performed). These are all intuitively reasonable and unsurprising assumptions. Though the theorem poses a hard constraint on $|\widehat{g}''|$, we hypothesize that this is a rather strong condition that can be weakened to be in line with similar models.

The theorem determines certain scaling regimes under which the sparse plus low-rank approximation remains reasonable. For instance, consider the case where $M \sim N$ and the moments scale as $s_{Nr} \sim \sqrt{N}$, which is reasonable given their form (i.e., very loosely, $\|\mathbf{x}\|_\infty \sim 1$ and $\|\mathbf{x}\|_2 \sim \sqrt{N}$ and small latent power $\|\mathbf{z}\|_2 \sim 1$ relative to N). Then, to keep the error of constant order, the power of each row of the matrix would need to stay constant $k \sim 1$. If we see $\mathbf{A}$ as a graph adjacency matrix, then, in rough terms, this can correspond intuitively to the case in which the node in-degrees stay constant so that the local complexity remains the same even as the global complexity increases while the network grows. Again, we hypothesize that this overall bound can be tightened given more

assumptions (e.g., on the network topology). Thus, we propose the SILVar model,

$$\widehat{\mathbf{y}} = \widehat{\mathbf{g}}\left(\left(\widehat{\mathbf{A}} + \widehat{\mathbf{L}}\right)\mathbf{x}\right), \tag{3.18}$$

and learn it using the optimization problem,

$$\begin{aligned}(\widehat{g}, \widehat{\mathbf{A}}, \widehat{\mathbf{L}}) =& \underset{g,\mathbf{A},\mathbf{L}}{\operatorname{argmin}} \widehat{F}_3(\mathbf{Y}, \mathbf{X}, g, \mathbf{A}+\mathbf{L}) + h_1(\mathbf{A}) + h_2(\mathbf{L}) \\ & \text{s.t. } g = \nabla G \in \mathcal{C}^1,\end{aligned} \tag{3.19}$$

where

$$\widehat{F}_3(\mathbf{Y}, \mathbf{X}, g, \mathbf{A}) = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{m}\left[G_*(y_{ji}) + G\left(\mathbf{a}_j^\top \mathbf{x}_i\right)\right] - \mathbf{y}_i^\top(\mathbf{A}\mathbf{x}_i)\right], \tag{3.20}$$

the empirical version of F_3, and h_1 and h_2 are regularizers on $\mathbf{A}$ and $\mathbf{L}$ respectively. Two natural choices for h_2 would be $h_2(\mathbf{L}) = \lambda_2\|\mathbf{L}\|_*$ the nuclear norm and $h_2(\mathbf{L}) = \mathbb{I}\{\|\mathbf{L}\|_* \leq \lambda_2\}$ the indicator of the nuclear norm ball, both relating to the nuclear norm of $\mathbf{L}$, since $\mathbf{L}$ is approximately low rank due to the influence of a relatively small number of latent variables. We may choose different forms for h_1 depending on our assumptions about the structure of $\mathbf{A}$. For example, if $\mathbf{A}$ is assumed sparse, we may use $h_1(\mathbf{A}) = \lambda_1\|\mathrm{v}(\mathbf{A})\|_1$, the ℓ_1 norm applied element-wise to the vectorized $\mathbf{A}$ matrix. These examples are extensions to the "sparse and low-rank" models, which have been shown under certain geometric incoherence conditions to be identifiable [17]. In other words, if the sparse component is not too low-rank, and if the low-rank component is not too sparse, then $\mathbf{A}$ and $\mathbf{L}$ can be recovered uniquely.

3.2.2 Connection to Related Problems

In this section, we show how the SILVar model can be used in various problem settings commonly considered throughout the literature.

Generalized Robust PCA

Though we posed our initial problem as a regression problem, if we take our measurement vectors to be $\mathbf{x}_i = \mathbf{e}_i$ the canonical basis vectors (i.e., so that the overall data matrix $\mathbf{X} = \mathbf{I}$), then we arrive at

$$\widehat{\mathbf{Y}} = \widehat{\mathbf{g}}(\widehat{\mathbf{A}} + \widehat{\mathbf{L}}). \tag{3.21}$$

This is precisely the model behind Generalized Robust PCA [52], but with the twist of estimating the link function as well [53]. What is worth noting is that although we arrived at our model via a regression problem with latent variables, the model itself is also able to describe a problem that arises from very different assumptions on how the data is generated and structured.

We also note that the SILVar model can be modified to share a space with the Generalized Low-Rank (GLR) Models [54]. The GLR framework is able to describe arbitrary types of data with an appropriate choice of convex link function g determined *a priori*, while the SILVar model is restricted to a certain continuous class of convex link functions but aims to learn this function. The modification is simply a matrix factorization $\mathbf{L} = \mathbf{UV}$ (and "infinite" penalization on $\mathbf{A}$). The explicit factorization makes the problem non-convex but instead block convex, which still allows for alternating convex steps in $\mathbf{U}$ with fixed $\mathbf{V}$ (and vice versa) to tractably reach local minima under certain conditions. Nonetheless, due to the non-convexity, further discussion of this particular extension will be beyond the scope of this work and remains as an interesting direction for future investigation.

Extension to Autoregressive Models

We can apply the SILVar model to learn from time series as well. Consider a set of N time series each of length K, $\mathbf{X} \in \mathbb{R}^{N \times K}$. We assume the noise at each time step is independent (note that, with this assumption, the time series are still dependent across time), and take

in our previous formation, $\mathbf{y}_i \leftarrow \mathbf{x}_k$ and $\mathbf{x}_i \leftarrow \mathbf{x}_{k-1:k-M} = (\mathbf{x}_{k-1}^\top \dots \mathbf{x}_{k-M}^\top)^\top$ so that the model of order M takes the form,

$$\widehat{\mathbf{x}}_k = \mathbf{g}\left(\sum_{i=1}^{M} \left(\mathbf{A}^{(i)} + \mathbf{L}^{(i)}\right) \mathbf{x}_{k-i}\right), \tag{3.22}$$

and learn it using the optimization problem,

$$(\widehat{g}, \widehat{\mathbf{A}}, \widehat{\mathbf{L}}) = \underset{g,\mathbf{A},\mathbf{L}}{\operatorname{argmin}} \widehat{F}_4(\mathbf{X}, g, \mathbf{A}+\mathbf{L}) + h_1(\mathbf{A}) + h_2(\mathbf{L}) \tag{3.23}$$

$$\text{s.t. } g = \nabla G \in \mathcal{C}^1,$$

where $\mathbf{A} = \left(\mathbf{A}^{(1)} \dots \mathbf{A}^{(M)}\right)$ and $\mathbf{L} = \left(\mathbf{L}^{(1)} \dots \mathbf{L}^{(M)}\right)$ and

$$\widehat{F}_4(\mathbf{X}, g, \mathbf{A}) = \frac{1}{K-M} \sum_{k=M+1}^{K} \left[\sum_{j=1}^{N} \left[G_*(x_{jk}) + G\left(\sum_{i=1}^{M} \mathbf{a}_j^{(i)} \mathbf{x}_{k-i}\right)\right] - \mathbf{x}_k^\top \left(\sum_{i=1}^{M} \mathbf{A}^{(i)} \mathbf{x}_{k-i}\right)\right],$$

where $\mathbf{A}^{(i)} = \left(\mathbf{a}_1^{(i)} \dots \mathbf{a}_N^{(i)}\right)^\top$, similarly to before. Note that the analysis in the previous section follows naturally in this setting, so that here $\operatorname{rank}(\mathbf{L}_i) \leq r+1$. Then, the matrix $\mathbf{A}$ may be assumed to be group sparse, relating to generalized notions of Granger Causality [9, 20], and one possible corresponding choice of regularizer taking the form

$$h_1(\mathbf{A}) = \lambda_1 \sum_{i,j} \left\| \left(a_{ij}^{(1)} \dots a_{ij}^{(M)}\right) \right\|_2.$$

Another structural assumption could be that of the Causal Graph Process model

$$h_1(\mathbf{A}) = \lambda_1 \|\mathrm{v}(\mathbf{A}^{(1)})\|_1 + \lambda_2 \sum_{i \neq j} \|\mathbf{A}^{(i)}\mathbf{A}^{(j)} - \mathbf{A}^{(j)}\mathbf{A}^{(i)}\|_F^2$$

from the previous chapter (2.4). Since this particular regularization is again block convex, convex steps can still be taken in each $\mathbf{A}^{(i)}$ with all other blocks $\mathbf{A}^{(j)}$ for $j \neq i$ fixed, for a computationally tractable algorithm to reach a local minimum under certain conditions. However, further detailed discussion will remain outside the scope of this chapter.

The hidden variables in this time series model can even possibly model underlying trends in the data. For example, one year of daily temperature measurements across a country could be related through a graph based on geographical and meteorological features, but all exhibit the same significant trend due to the changing seasons. In previous work, a

standard pipeline is to detrend the data as a preprocessing step, and then estimate or use a graph to describe the variations of the data on top of the underlying trends [3, 16, 32]. Instead, the time series can also be modeled as a modified autoregressive process, depending on a low-rank trend $\mathbf{L}' = (\boldsymbol{\ell}'_1 \ \dots \ \boldsymbol{\ell}'_K) \in \mathbb{R}^{N\times K}$ and the variations of the process about that trend,

$$\widehat{\mathbf{x}}_k = g\left(\boldsymbol{\ell}'_k + \sum_{i=1}^{M} \mathbf{A}^{(i)}\left(\mathbf{x}_{k-i} - \boldsymbol{\ell}'_{k-i}\right)\right). \tag{3.24}$$

Substituting this into Equation (3.22) yields

$$\boldsymbol{\ell}'_k + \sum_{i=1}^{M} \mathbf{A}^{(i)}\left(\mathbf{x}_{k-i} - \boldsymbol{\ell}'_{k-i}\right) = \sum_{i=1}^{M}\left(\mathbf{A}^{(i)} + \mathbf{L}^{(i)}\right)\mathbf{x}_{k-i} \implies \sum_{i=1}^{M}\mathbf{L}^{(i)}\mathbf{x}_{k-i} = \boldsymbol{\ell}'_k - \sum_{i=1}^{M}\mathbf{A}^{(i)}\boldsymbol{\ell}'_{k-i} \tag{3.25}$$

Thus we estimate the trend using ridge regression for numerical stability purposes but without enforcing $\mathbf{L}'$ to be low rank. We can accomplish this via the simple optimization,

$$\widehat{\mathbf{L}}' = \underset{\mathbf{L}'}{\operatorname{argmin}} \sum_{k=M+1}^{K} \left\| \boldsymbol{\ell}'_k - \sum_{i=1}^{M}\mathbf{A}^{(i)}\boldsymbol{\ell}'_{k-i} - \sum_{i=1}^{M}\mathbf{L}^{(i)}\mathbf{x}_{k-i} \right\|_2^2 + \lambda\|\mathbf{L}'\|_F^2 \tag{3.26}$$

with λ being the regularization parameter. In this way, the extension of SILVar to autoregressive models can allow joint estimation of the effects of the trend and of these variations supported on the graph, as will be demonstrated via experiments in Section 3.5.

3.3 Efficiently Learning SILVar Models

In this section, we describe the formulation and corresponding algorithm for learning the SILVar model. Surprisingly, in the single-task setting, learning a SIM is jointly convex in g and $\mathbf{a}$ as demonstrated in [49]. The pseudo-likelihood functional $\widehat{F}_3$ used for learning the SILVar model in (3.23) is thus also jointly convex in g, $\mathbf{A}$, and $\mathbf{L}$ by a simple extension from the single-task regression setting.

Lemma 5 (Corollary of Theorem 2 of [49])**.** *The $\widehat{F}_3$ in the SILVar model learning problem* (3.18) *is jointly convex in g, $\mathbf{A}$, and $\mathbf{L}$.*

This convexity is enough to ensure that the learning can converge and be computation-

ally efficient. Before describing the full algorithm, one detail remains: the implementation of the LMR introduced in Section 3.1.3.

3.3.1 Lipschitz Monotonic Regression

To tackle LMR, we first introduce the related simpler problem of monotonic regression, which is solved by a well-known algorithm, the pooled adjacent violators (PAV) [55]. The monotonic regression problem is formulated as

$$\begin{aligned} \mathrm{PAV}(\mathbf{y}, \mathbf{x}) \triangleq \underset{\mathbf{g}}{\operatorname{argmin}} & \sum_{i=1}^{n} \left(g(x_i) - y_i\right)^2 \\ \text{s.t. } & 0 \leq g\left(x_{[j+1]}\right) - g\left(x_{[j]}\right) \quad \text{for } j = 1, \ldots, n-1. \end{aligned} \tag{3.27}$$

The PAV algorithm has a complexity of $O(n)$, which is due to a single complete sweep of the vector $\mathbf{y}$. The monotonic regression problem can also be seen as a standard ℓ_2 projection onto the convex set of monotonic functions.

We introduce a simple generalization to the monotonic regression,

$$\begin{aligned} \mathrm{GPAV}_{\mathbf{t}}(\mathbf{y}, \mathbf{x}) \triangleq \underset{\mathbf{g}}{\operatorname{argmin}} & \sum_{i=1}^{n} \left(g(x_i) - y_i\right)^2 \\ \text{s.t. } & t_{[j+1]} \leq g\left(x_{[j+1]}\right) - g\left(x_{[j]}\right) \quad \text{for } j = 1, \ldots, n-1. \end{aligned} \tag{3.28}$$

This modification allows weaker (if $t_i < 0$) or stronger (if $t_i > 0$) local monotonicity conditions to be placed on the estimated function g. Let $t_{[1]} = 0$ and $s_{[i]} = \sum_{j=1}^{i} t_{[j]}$, and in vectorized form,

$$\mathbf{s} \triangleq \mathrm{cusum}(\mathbf{t}). \tag{3.29}$$

Then, we can rewrite,

$$\begin{aligned} \mathrm{GPAV}_{\mathbf{t}}(\mathbf{y}, \mathbf{x}) = \underset{\mathbf{g}}{\operatorname{argmin}} & \sum_{i=1}^{n} \left(g(x_i) - s_i - (y_i - s_i)\right)^2 \\ \text{s.t. } & 0 \leq g\left(x_{[j+1]}\right) - s_{[j+1]} - \left(g\left(x_{[j]}\right) - s_{[j]}\right) \quad \text{for } j = 1, \ldots, n-1, \end{aligned} \tag{3.30}$$

which leads us to recognize that

$$\mathrm{GPAV}_{\mathbf{t}}(\mathbf{y}, \mathbf{x}) = \mathrm{PAV}(\mathbf{y} - \mathbf{s}, \mathbf{x}) + \mathbf{s}. \tag{3.31}$$

Returning to LMR (3.11), we pose it as the projection onto the intersection of 2 convex sets, the monotonic functions and the functions with upper bounded differences, for which each projection can be solved efficiently using GPAV (for the second set, we can use GPAV on the negated function and negate the result). Thus to perform this optimization efficiently utilizing PAV as a subroutine, we can use an accelerated Dykstra's Algorithm [56], which is a general method to find the projection onto the non-empty intersection of any number of convex sets. The accelerated algorithm can has a geometric convergence rate for $O(\log n)$ passes, with each pass utilizing PAV with $O(n)$ cost, for a total cost of $O(n \log n)$. We make a brief sidenote that this is better than the $O(n^2)$ in [57] and simpler than the $O(n \log n)$ in [58] achieved using a complicated tree-like data structure, and has no learning rate parameter to tune compared to an ADMM-based implementation that would also yield $O(n \log n)$. We provide the steps of LMR in Algorithm 4 as a straightforward application of the the accelerated Dykstra's algorithm, but leave the details of the derivation of the acceleration to [56]. In addition, the numerical stability parameter is included to handle small denominators, but in practice we did not observe the demoninator falling below $\varepsilon = 10^{-9}$ in our simulations before convergence.

While the GPAV allows us to quickly perform Lipschitz monotonic regression with our particular choice of $\mathbf{t}$, it should be noted that using other choices for $\mathbf{t}$ can easily generalize the problem to bi-Hölder regression as well, to find functions in the set

$$\mathcal{C}_{\ell,\alpha}^{u,\beta} = \{g : \forall y > x,\ \ell|y-x|^{\alpha} \leq g(y) - g(x) \leq u|y-x|^{\beta}\}.$$

This set may prove interesting for further analysis of the estimator and is left as a topic for future investigation.

Algorithm 4 Lipschitz monotone regression (LMR)

Let $t_1 = 0$ and compute $t_{[i+1]} = x_{[i+1]} - x_{[i]}$, $\mathbf{s} = \text{cusum}(\mathbf{t})$. Set numerical stability parameter $0 < \varepsilon < 1$, $error = \infty$, and tolerance $0 < \delta < 1$.

Initialize $\mathbf{g}_\ell^{(0)} = \text{PAV}(\mathbf{y}, \mathbf{x})$, $\mathbf{g}_u^{(0)} = -\text{PAV}(-\mathbf{y} + \mathbf{s}, \mathbf{x}) + \mathbf{s}$, $\mathbf{v}_0 = \mathbf{0}$, $\mathbf{v}_1 = \mathbf{g}_u - \mathbf{g}_\ell$, $\mathbf{w} = \mathbf{g}_u$, $k = 0$

while $error \geq \delta$ **do**

  $\mathbf{g}_\ell^{(k+1)} \leftarrow \text{PAV}\left(\mathbf{g}_u^{(k)} - \mathbf{v}_0, \mathbf{x}\right)$

  $\mathbf{g}_u^{(k+1)} \leftarrow -\text{PAV}\left(-(\mathbf{g}_\ell^{(k+1)} - \mathbf{v}_1) + \mathbf{s}, \mathbf{x}\right) + \mathbf{s}$

  $\mathbf{v}_0 \leftarrow \mathbf{g}_\ell^{(k+1)} - (\mathbf{g}_u^{(k+1)} - \mathbf{v}_0)$

  $\mathbf{v}_1 \leftarrow \mathbf{g}_u^{(k+1)} - (\mathbf{g}_\ell^{(k+1)} - \mathbf{v}_1)$

  if $\left\|\|\mathbf{g}_\ell^{(k)} - \mathbf{g}_u^{(k)}\|_2^2 + (\mathbf{g}_\ell^{(k+1)} - \mathbf{g}_u^{(k+1)})^\top(\mathbf{g}_\ell^{(k)} - \mathbf{g}_u^{(k)})\right| \geq \varepsilon$ **then**

    $\alpha^{(k+1)} \leftarrow \frac{\|\mathbf{g}_\ell^{(k)} - \mathbf{g}_u^{(k)}\|_2^2}{\|\mathbf{g}_\ell^{(k)} - \mathbf{g}_u^{(k)}\|_2^2 + (\mathbf{g}_\ell^{(k+1)} - \mathbf{g}_u^{(k+1)})^\top(\mathbf{g}_\ell^{(k)} - \mathbf{g}_u^{(k)})}$

    $\mathbf{z} \leftarrow \mathbf{g}_u^{(k)} + \alpha^{(k+1)}(\mathbf{g}_u^{(k+1)} - \mathbf{g}_u^{(k)})$

  else

    $\mathbf{z} \leftarrow \frac{1}{2}(\mathbf{g}_\ell^{(k+1)} + \mathbf{g}_u^{(k+1)})$

  end if

  $error \leftarrow \|\mathbf{z} - \mathbf{w}\|_2$

  $\mathbf{w} \leftarrow \mathbf{z}$

  $k \leftarrow k + 1$

end while

return z

3.3.2 Optimization Algorithms

With LMR in hand, we can outline the algorithm for solving the convex problem (3.19). This procedure can be performed for general h_1 and h_2, but proximal mappings are efficient to compute for many common regularizers. Thus, we describe the basic algorithm using gradient-based proximal methods (e.g., accelerated gradient [38], and quasi-Newton [59]), which require the ability to compute a gradient and the proximal mapping.

In addition, we can compute the objective function value for purposes of backtracking [60], evaluating algorithm convergence, or computing error (e.g., for validation or testing purposes). While our optimization outputs an estimate of g, the objective function depends on the value of G and its conjugate G_*. Though we cannot necessarily determine G or G_* uniquely since $G + c$ and G both yield g as a gradient for any $c \in \mathbb{R}$, the value of $G(x) + G_*(y)$ is unique for a fixed g. To see this, consider $\widetilde{G} = G + c$ for some constant c. Then,

$$\begin{aligned} \widetilde{G}(x) + \widetilde{G}_*(y) &= G(x) + c + \max_z [zy - \widetilde{G}(z)] \\ &= G(x) + c + \max_z [zy - G(z) - c] \\ &= G(x) + \max_z [zy - G(z)] \\ &= G(x) + G_*(y) \end{aligned} \tag{3.32}$$

This allows us to compute the objective function by performing the cumulative numerical integral of $\widehat{\mathbf{g}}$ on points $\mathrm{v}(\boldsymbol{\Theta})$ (e.g., `G=cumtrapz(theta,ghat)` in Matlab). Then, a discrete convex conjugation (also known as the discrete Legendre transform (DLT) [61]) computes G_*.

We did not notice significant differences in performance accuracy due to our implementation using quasi-Newton methods and backtracking compared to an accelerated proximal gradient method, possibly because both algorithms were run until convergence. Thus, we show only one set of results. However, we note that the runtime was improved by imple-

menting the quasi-Newton and backtracking method.

Algorithm 5 Single Index Latent Variable (SILVar) Learning

1: Initialize $\widehat{\mathbf{A}} = \mathbf{0}$, $\widehat{\mathbf{L}} = \mathbf{0}$

2: **while** not converged **do** Proximal Methods

3: Computing gradients:

$$\boldsymbol{\Theta} \leftarrow (\widehat{\mathbf{A}} + \widehat{\mathbf{L}})\mathbf{X}$$

$$\widehat{\mathbf{g}} \leftarrow \mathrm{LMR}(\mathrm{v}(\mathbf{Y}), \mathrm{v}(\boldsymbol{\Theta}))$$

$$\nabla_{\mathbf{A}} F_3 = \nabla_{\mathbf{L}} F_3 = \sum_{i \in \mathcal{I}} (\widehat{\mathbf{g}}(\boldsymbol{\theta}_i) - \mathbf{y}_i)\, \mathbf{x}_i^\top$$

4: Optionally compute function value:

$$\widehat{\mathbf{G}} \leftarrow \texttt{cumtrapz}(\mathrm{v}(\boldsymbol{\Theta}), \widehat{\mathbf{g}})$$

$$\widehat{\mathbf{G}}_* \leftarrow \mathrm{DLT}(\mathrm{v}(\boldsymbol{\Theta}), \widehat{\mathbf{G}}, \mathrm{v}(\mathbf{Y}))$$

$$\widehat{F}_3 = \sum_{ij} \widehat{\mathbf{G}}_*(y_{ij}) + \widehat{\mathbf{G}}(\theta_{ij}) - y_{ij}\theta_{ij}$$

5: **end while**

6: **return** $(\widehat{\mathbf{g}}, \mathbf{A}, \mathbf{L})$

Algorithm 5 describes the learning procedure and details the main function and gradient computations while assuming a proximal operator is given. The computation of the gradient and the update vector depends on the particular variation of proximal method utilized; with stochastic gradients, the set $\mathcal{I} \subset \{1, \ldots, n\}$ could be pseudorandomly generated at each iteration, while with standard gradients, $\mathcal{I} = \{1, \ldots, n\}$. The `cumtrapz` procedure takes as input the coordinates of points describing the function $\widehat{\mathbf{g}}$. The DLT procedure takes as its first two inputs the coordinates of points describing the function $\widehat{\mathbf{G}}$, and as its third input the points at which to evaluate the convex conjugate $\widehat{\mathbf{G}}_*$.

Here, an observant reader may notice the subtle point that the function G_* may only be

defined inside a finite or semi-infinite interval. This can occur if the function g is bounded below and/or above, so that G has a minimum/maximum slope, and G_* is infinite for any values below that minimum or above that maximum slope of G (to see this, one may refer back to the definition of conjugation (3.2)). Fortunately, this does not invalidate our method. It is straightforward to see that $\widehat{g}(x) = x$ is always a feasible solution and that $\widehat{G}_*(y) = y^2/2$ is defined for all y; thus starting from this solution, with appropriately chosen step sizes, gradient-based algorithms will avoid the solutions that make the objective function infinite. Furthermore, even if new data $\mathbf{y}_j$ falls outside the valid domain for the learned $\widehat{G}_*$ and we incur an "infinite" loss using the model, evaluating $\widehat{\mathbf{g}}\left(\left(\widehat{\mathbf{A}} + \widehat{\mathbf{B}}\right)\mathbf{x}_j\right)$ is still well defined. This problem is not unique to SIM's, as assuming a fixed link function g in a GLM can also incur infinite loss if new data does not conform to modeling assumptions implicit to the link function (e.g., the log-likelihood for a negative $\mathbf{y}$ under a non-negative distribution), and making a prediction using the learned GLM is still well-defined. Practically, the loss for SILVar computed using the DLT may be large, but will not be infinite [61].

3.4 Performance

We now provide conditions under which the optimization recovers the parameters of the model.

First let us establish a few additional notations needed. Let $(\widetilde{g}, \widetilde{\mathbf{A}}, \widetilde{\mathbf{L}})$ be the best Lipschitz monotonic function, sparse matrix, and low-rank matrix that models the true data generation. Let $\widetilde{\mathbf{A}}$ be sparse on the index set S of size $|S| = s_{\mathbf{A}}$ and

$$\widetilde{\mathbf{L}} = \mathbf{U}\Lambda\mathbf{V}^\top$$

be the SVD of $\widetilde{L}$, where

$$\mathbf{U} \in \mathbb{R}^{M \times r_{\mathbf{L}}}$$

$$\Lambda \in \mathbb{R}^{r_{\mathbf{L}} \times r_{\mathbf{L}}}$$

$$\mathbf{V} \in \mathbb{R}^{N \times r_{\mathbf{L}}}$$

where $r_{\mathbf{L}} = \text{rank}(\widetilde{\mathbf{L}})$. Then, for a generic matrix $\mathbf{A}$, let $\mathbf{A}_S$ be equal to $\mathbf{A}$ on S and 0 on S^c, so that

$$\widetilde{\mathbf{A}}_S = \widetilde{\mathbf{A}}$$

$$\mathbf{A}_S + \mathbf{A}_{S^c} = \mathbf{A},$$

and for a generic matrix $\mathbf{L}$, let

$$\mathbf{L}_R = \mathbf{L} - (\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\mathbf{L}(\mathbf{I} - \mathbf{V}\mathbf{V}^\top)$$

$$\mathbf{L}_{R^c} = \mathbf{L} - \mathbf{L}_R$$

so that $\widetilde{\mathbf{L}}_R = \widetilde{\mathbf{L}}$ and trivially $\mathbf{L}_{R^c} + \mathbf{L}_R = \mathbf{L}$. Consider the set of approximately S-sparse and R-low-rank matrices

$$\mathcal{B}_\gamma(S, R) = \{(\mathbf{\Phi}, \mathbf{\Psi}) : \gamma\|\mathbf{\Phi}_{S^c}\|_1 + \|\mathbf{\Psi}_{R^c}\|_* \leq 3(\gamma\|\mathbf{\Phi}_S\|_1 + \|\mathbf{\Psi}_R\|_*)\}.$$

Intuitively, this is the set of matrices for which the energy of $\mathbf{\Phi}$ is on the same sparsity set as $\widetilde{\mathbf{A}}$ and similarly the energy of $\mathbf{\Psi}$ is in the same low-rank space as $\widetilde{\mathbf{L}}$. Finally, let the marginalization of the loss functional w.r.t. g be

$$\widehat{m}(\mathbf{Y}, \mathbf{X}, \mathbf{A}) = \min_{g \in \mathcal{C}^1} \widehat{F}_3(\mathbf{Y}, \mathbf{X}, g, \mathbf{A}),$$

the value of the marginalized function at the true matrix parameters be

$$\widehat{\widetilde{g}} = \min_{g \in \mathcal{C}^1} \widehat{F}_3(\mathbf{Y}, \mathbf{X}, g, \widetilde{\mathbf{A}} + \widetilde{\mathbf{L}}),$$

the Hessian of the marginalized functional w.r.t. the matrix parameter be

$$\widetilde{\mathcal{I}} = \nabla^2_{\text{v}(\mathbf{A})} \widehat{m}(\mathbf{Y}, \mathbf{X}, \widetilde{\mathbf{A}} + \widetilde{\mathbf{L}}),$$

and denote the quadratic form in shorthand

$$\|\mathbf{v}\|_{\widetilde{\mathcal{I}}} = \mathbf{v}^\top \widetilde{\mathcal{I}} \mathbf{v}.$$

Now, consider the following assumptions:

1. There exists some $\alpha > 0$ such that

$$\|\mathbf{\Phi} + \mathbf{\Psi}\|_{\widetilde{\mathcal{I}}} \geq \alpha \|\mathbf{\Phi} + \mathbf{\Psi}\|_F^2, \quad \forall (\mathbf{\Phi}, \mathbf{\Psi}) \in \mathcal{B}_\gamma(S, R).$$

   This is essentially a Restricted Strong Convexity condition standard for structured recovery [62].

2. Let $\widehat{\widetilde{\mathbf{\Gamma}}} \in \mathbb{R}^{M \times K}$ be the matrix with columns given by

$$\widehat{\widetilde{\mathbf{\Gamma}}}_k = \widehat{\widetilde{g}}((\widetilde{\mathbf{A}} + \widetilde{\mathbf{L}})\mathbf{x}_k) - \mathbf{y}_k, \quad k = 1, \ldots, K.$$

   Then for some $\lambda > 0$ and $\gamma > 0$,

$$\frac{1}{K} \left\| \widehat{\widetilde{\mathbf{\Gamma}}} \mathbf{X}^\top \right\|_\infty \leq \lambda\gamma/2$$
$$\frac{1}{K} \left\| \widehat{\widetilde{\mathbf{\Gamma}}} \mathbf{X}^\top \right\|_2 \leq \lambda/2$$

   where $\|\mathbf{A}\|_\infty$ is the largest element of $\mathbf{A}$ in magnitude, and $\|\mathbf{A}\|_2$ is the largest singular value of $\mathbf{A}$.

   This states that the error is not too powerful and is not too correlated with the regression variables, and it is also similar to standard conditions for structured recovery.

3. Let $\tau = \gamma \sqrt{\frac{s_\mathbf{A}}{r_\mathbf{L}}}$ and $\mu = \frac{1}{32 r_\mathbf{L}(1+\tau^2)}$, then

$$\frac{\max(\|\mathbf{\Phi}\|_2, \|\mathbf{\Psi}\|_\infty/\gamma)}{\gamma\|\mathbf{\Phi}\|_1 + \|\mathbf{\Psi}\|_*} \leq \mu, \quad \forall (\mathbf{\Phi}, \mathbf{\Psi}) \in \mathcal{B}_\gamma(S, R).$$

   This is a condition on the incoherence between the sparsity set S and low rank subspace R (see [17, 52, 63, 64] for other similar variations on rank-sparsity incoherence considered previously). Basically, this states that S should be such that S-sparse matrices are not approximately low-rank, while R should be such that R-low-rank matrices are not approximately sparse.

Then we have the following result,

Theorem 6. *Under Assumptions 1-3, the solution to the optimization problem* (3.23) *satisfies*

$$\|\widehat{\mathbf{A}} - \widetilde{\mathbf{A}}\|_F \leq \frac{3\lambda\gamma\sqrt{s_{\mathbf{A}}}}{\alpha}\left(2+\sqrt{\frac{2}{2+\tau^2}}\right) \leq \frac{9\lambda\gamma\sqrt{s_{\mathbf{A}}}}{\alpha} \tag{3.33}$$

$$\|\widehat{\mathbf{L}} - \widetilde{\mathbf{L}}\|_F \leq \frac{3\lambda\sqrt{r_{\mathbf{L}}}}{\alpha}\left(2+\sqrt{\frac{2\tau^2}{1+2\tau^2}}\right) \leq \frac{9\lambda\sqrt{r_{\mathbf{L}}}}{\alpha}. \tag{3.34}$$

We give the proof for this theorem in Appendix B.2. The theorem relates the performance of the optimization to conditions on the problem parameters. Of course, in practice these conditions are difficult to check, since they require knowledge of ground truth. Even given ground truth, verifying Assumption 3 requires solving another separate non-convex optimization problem. Thus, future directions for work could include investigation into which kinds of sparse matrix (or network) models for $\mathbf{A}$ and low-rank models $\mathbf{L}$ produce favorable scaling regimes in the problem parameters with high probability.

3.5 Experiments

We study the performance of the algorithm via simulations on synthetically generated data as well as real data. In these experiments, we show the different regression settings, as discussed in Section 3.2.2, under which the SILVar model can be applied.

3.5.1 Synthetic Data

The synthetic data corresponds to a multi-task regression setting. The data was generated by first creating random sparse matrix $\mathbf{A}_f = (\mathbf{A}\ \mathbf{B})$ where $\mathbf{B} \in \mathbb{R}^{m\times H}$ and $H = \lfloor hp \rfloor$ the number of hidden variables, and h is the proportion of hidden variables. The elements of $\mathbf{A}_f$ were first generated as i.i.d. Normal variables, and then a sparse mask was applied

to $\mathbf{A}$ choose $\lfloor m \log_{10}(p) \rfloor$ non-zeros, and $\mathbf{B}$ was scaled by a factor of $\frac{1}{3\sqrt{H}}$. Next, the data $\mathbf{X}_f = (\mathbf{X}^\top \ \mathbf{Z}^\top)^\top$ were generated with each vector drawn i.i.d. from a multivariate Gaussian with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}_f \in \mathbb{R}^{(p+H)\times(p+H)}$. This was achieved by creating $\mathbf{\Sigma}_f^{1/2}$ by thresholding a matrix with diagonal entries of 1 and off-diagonal entries drawn from an i.i.d. uniform distribution in the interval $[-0.5, 0.5]$ to be greater than 0.35 in magnitude. Then $\mathbf{\Sigma}_f^{1/2}$ was pre-multiplied with a matrix of i.i.d. Normal variables. Finally, we generated

$$\mathbf{Y} = g(\mathbf{A}_f \mathbf{X}_f) + \mathbf{W} \tag{3.35}$$

for 2 different link functions

$$\begin{aligned} g_1(x) &= \log(1 + e^{c_1 x}) \\ g_2(x) &= \frac{2}{1 + e^{-c_2 x}} - 1, \end{aligned} \tag{3.36}$$

and $\mathbf{W}$ is added i.i.d. Gaussian noise. Then, the task is to learn $(\widehat{g}, \widehat{\mathbf{A}}, \widehat{\mathbf{L}})$ the SILVar model (3.18) from $(\mathbf{Y}, \mathbf{X})$ without access to $\mathbf{Z}$. As comparison, we use an Oracle GLM model, in which the true g is given but the parameters $(\widehat{\mathbf{A}}_o, \widehat{\mathbf{L}}_o)$ still need to be learned. We note that while there is a slight mismatch between the true and assumed noise distributions, the task is nonetheless difficult, yet our estimation using the SILVar model can still exhibit good performance with respect to the Oracle.

The experiments were carried out across a range of problem sizes. The dimension of $\mathbf{y}_i$ was fixed at $m = 25$. The dimension of the observed $\mathbf{x}_i$ was set to $p \in \{25, 50\}$, and the proportion of the dimension of hidden $\mathbf{z}_i$ was set to $h \in \{0.1, 0.2\}$. The number of data samples was varied in $k \in \{25, 50, 100, 150, 200\}$. Validation was performed using a set of samples generated independently but using the same $\mathbf{A}_f$ and g, and using a grid of $(\lambda_S, \lambda_L) \in \left\{10^{i/4} \middle| i \in \{-8, -7, \ldots, 7, 8\}\right\}^2$. The whole process of data generation and model learning is repeated 20 times for each experimental condition. The average ℓ_1 errors between the true $\mathbf{A}$ and the best estimated $\widehat{\mathbf{A}}$ (determined via validation) are shown in

Figure 3.1.

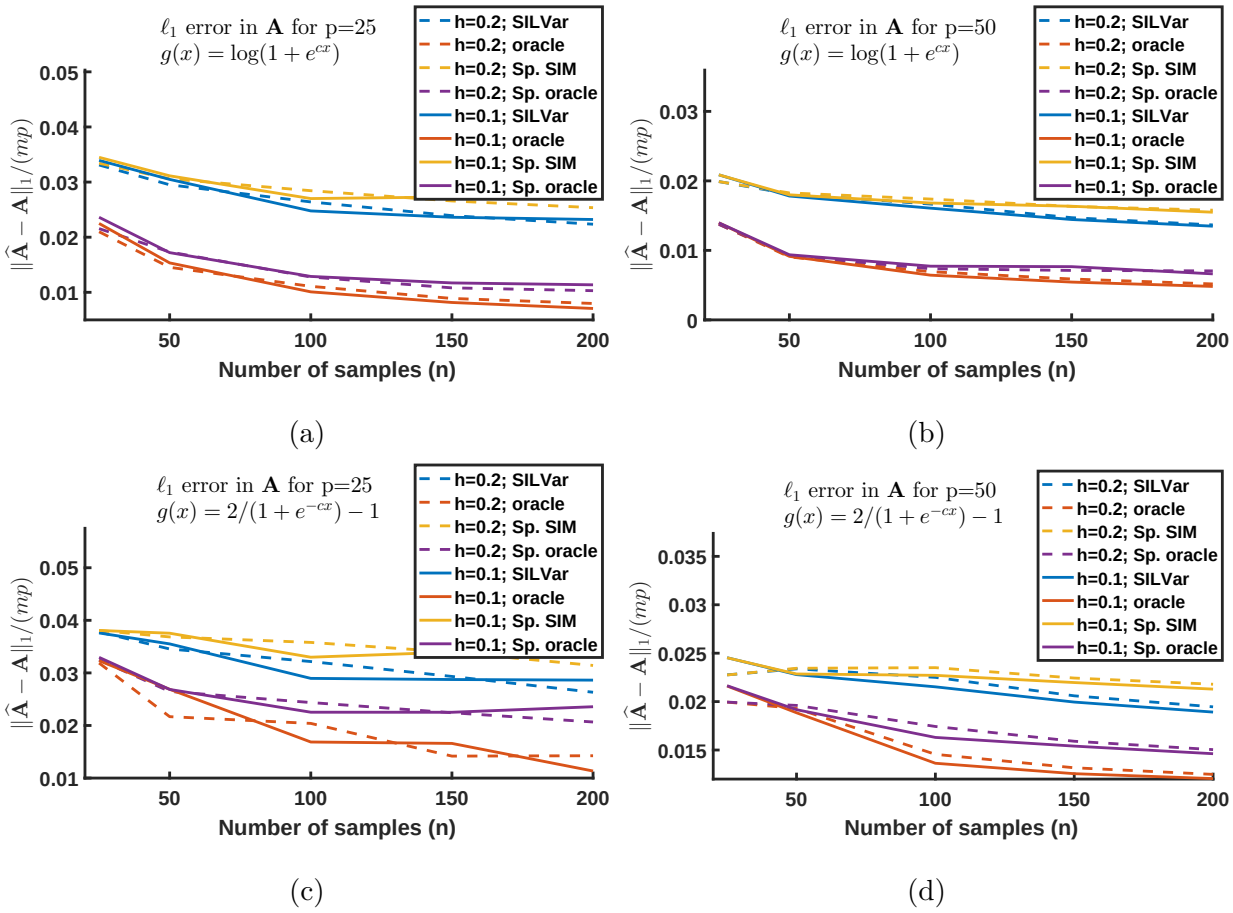


Figure 3.1: Different errors for estimating $\mathbf{A}$ in Toy Data

The first row is for the function g_1 and the second row is for the function g_2 for the various experimental conditions. The empirical results show that the SILVar model and learning algorithm lose out slightly on performance for estimating the link function g but overall still match the Oracle fairly well in most cases. We also see that SILVar performs better than just the sparse SIM [46], but that the purely sparse Oracle GLM can still provide an improvement over SILVar for not knowing the link function. We also see several other intuitive behaviors for both the SILVar and Oracle models: as we increase the amount of data n, the performance increases. Note that due to the scaling of weights $\mathbf{B}$ by the factor

of $1/\sqrt{H}$ to keep the average power $\|\mathbf{B}\|_F$ roughly constant, increasing the proportion of hidden variables (from $h = 0.1$ to $h = 0.2$) did not directly lead to significant performance decreases. We also note that g_2 seems to be somehow more difficult to estimate than g_1, and additionally that the performance of the SILVar model w.r.t. the Oracle model is worse with g_2 than with g_1. This could have something to do with the 2 saturations in g_2 as opposed to the 1 saturation in g_1, corresponding in an intuitive sense to a more non-linear setting that contains more information needing to be captured while estimating $\widehat{g}$.

3.5.2 Temperature Data

In this setting, we wish to learn the graph capturing relations between the weather patterns at different cities. The data is a real world multivariate time series consisting of daily temperature measurements (in °F) for 365 consecutive days during the year 2011 taken at each of 150 different cities across the continental USA.

Previously, the analysis on this dataset has been performed by first fitting with a 4th order polynomial and then estimating a sparse graph from an autoregressive model using a known link function $g(x) = x$ assuming Gaussian noise [32].

Here, we fit the time series using a 2nd order AR SILVar model (3.22) with regularizers for group sparsity

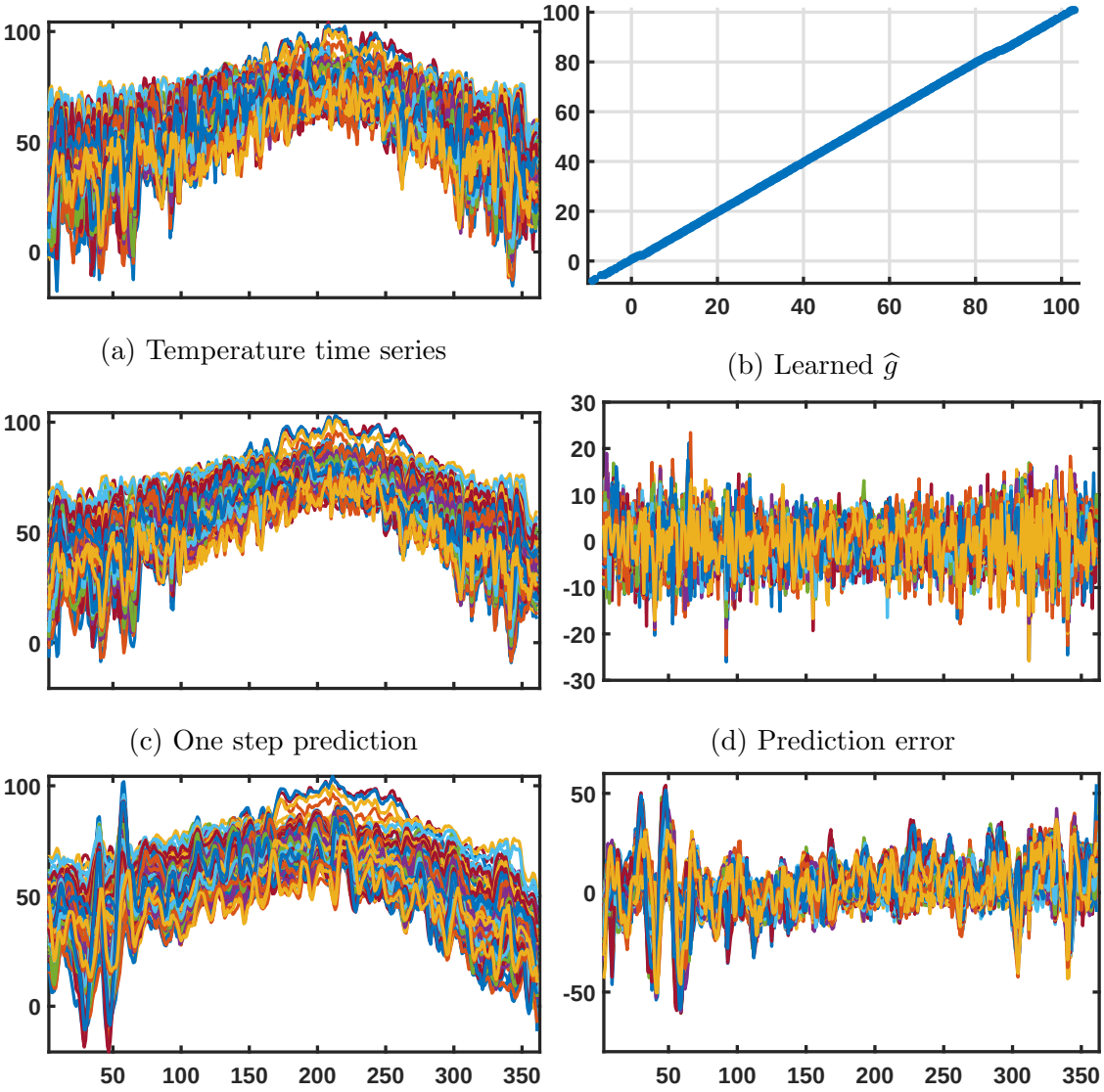
$$h_1(\mathbf{A}) = \lambda_1 \sum_{i,j} \left\| \left(a_{ij}^{(1)} \dots a_{ij}^{(M)} \right) \right\|_2 \tag{3.37}$$

where $a_{ij}^{(m)}$ is the ij entry of matrix $\mathbf{A}^{(m)}$, and nuclear norm

$$h_2(\mathbf{L}) = \lambda_2 \sum_{i=1}^{M} \left\| \mathbf{L}^{(i)} \right\|_* . \tag{3.38}$$

We also estimate the underlying trend using (3.26).

In Figure 3.2, we plot the original time series, estimated trends, the estimated network, and residuals. The plots are all from time steps 3 to 363, since those are the indices for the



(a) Temperature time series

(b) Learned $\widehat{g}$

(c) One step prediction

(d) Prediction error

(e) Estimated trend

(f) De-trended time series

Figure 3.2: Time series, link function, trends, and prediction errors computed using the learned SILVar model

trends that we can reliably estimate using the simple procedure (3.26). In Figure 3.2a, we show the original time series, which clearly exhibit a seasonal trend winter–summer–winter. Figure 3.2b shows the estimated link function $\widehat{g}$, which turns out to be linear, though a priori we might not intuitively expect a process as complicated as weather to behave this

way when sampled so sparsely. Figures 3.2c and 3.2d show the full prediction

$$\widehat{\mathbf{x}}_k = \sum_{i=1}^{M} (\widehat{\mathbf{A}}^{(i)} + \widehat{\mathbf{L}}^{(i)}) \mathbf{x}_{k-i} \tag{3.39}$$

and the residuals

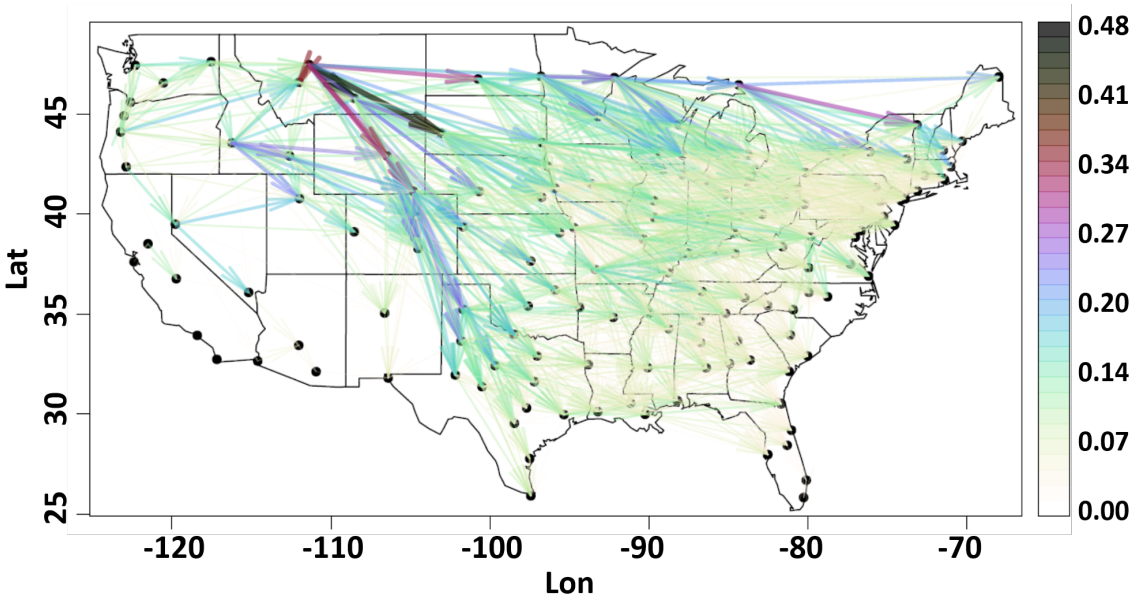
$$\mathbf{x}_k - \widehat{\mathbf{x}}_k, \tag{3.40}$$

respectively. Note the much lower vertical scale in Figure 3.2d. Figures 3.2e and 3.2f show the trend $\widehat{\mathbf{L}}'$ learned using (3.26) and the time series with the estimated trend removed, respectively. The trend estimation procedure captures the basic shape of the seasonal effects on the temperature. Several of the faster fluctuations in the beginning of the year are captured as well, suggesting that they were caused by some larger scale phenomena. Indeed there were several notable storm systems that affected the entire USA in the beginning of the year in short succession [65, 66].

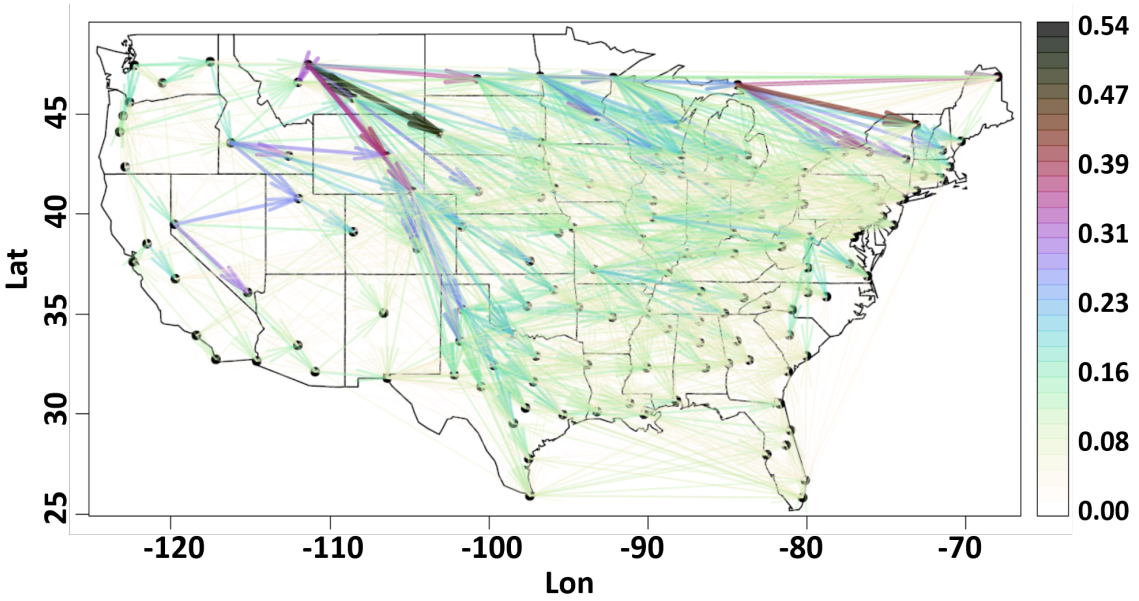
Figure 3.3 compares two networks $\widehat{\mathbf{A}}'$ estimated using SILVar and using just sparse SIM without accounting for the low-rank trends, both with the same sparsity level of 12% non-zeros for display purposes, and where

$$\widehat{a}'_{ij} = \left\| \left(\widehat{a}_{ij}^{(1)} \dots \widehat{a}_{ij}^{(M)} \right) \right\|_2 .$$

Figure 3.3a shows the network $\widehat{\mathbf{A}}'$ that is estimated using SILVar. The connections imply predictive dependencies between the temperatures in cities connected by the graph. It is intuitively pleasing that the patterns discovered match well previously established results based on first de-trending the data and then separately estimating a network [32]. That is, we see the effect of the Rocky Mountain chain around $-110°$ to $-105°$ longitude and the overall west-to-east direction of the weather patterns, matching the prevailing winds. In contrast to that of SILVar, the graph estimated by the sparse SIM shown in Figure 3.3b on the other hand has many additional connections with no basis in actual weather patterns. Two particularly unsatisfying cities are: sunny Los Angeles, California at $(-118, 34)$, with its multiple connections to snowy northern cities including Fargo, North Dakota at



(a) Weather graph learned using SILVar



(b) Weather graph learned using Sp. SIM (without low-rank)

Figure 3.3: Learned weather stations graphs

$(-97, 47)$; and Caribou, Maine at $(-68, 47)$, with its multiple connections going far westward against prevailing winds including to Helena, Montana at $(-112, 47)$. These do not show in the graph estimated by SILVar and shown in Figure 3.3a.

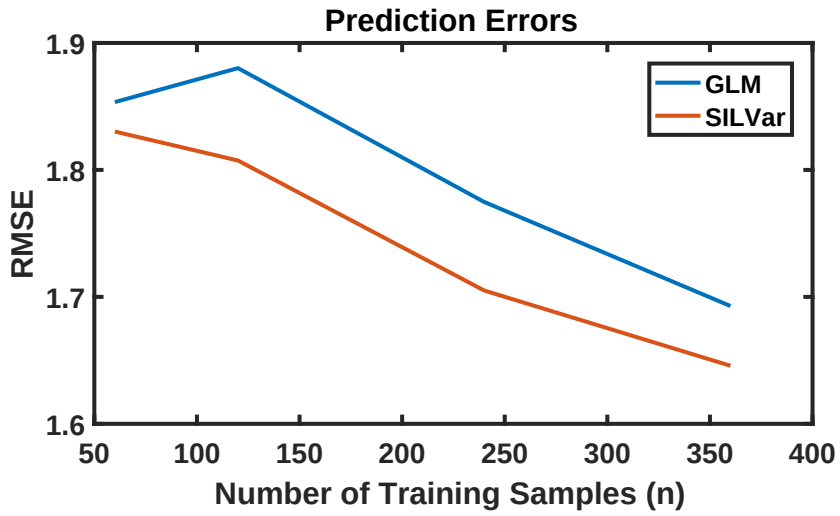
3.5.3 Bike Traffic Data

The bike traffic data was obtained from HealthyRide Pittsburgh [67]. The dataset contains the timestamps and station locations of departure and arrival (among other information) for each of 127,559 trips taken between 50 stations within the city from May 31, 2015 to September 30, 2016, a total of 489 days.

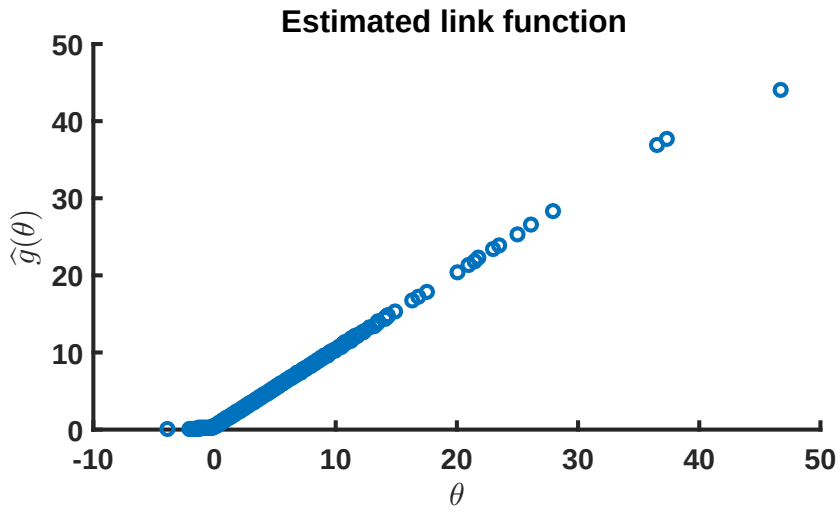
We consider the task of using the total number of rides departing from and arriving in each location at 6:00AM-11:00AM to predict the number of rides departing from each location during the peak period of 11:00AM-2:00PM for each day. This corresponds to $\mathbf{Y} \in \mathbb{N}_0^{50\times 489}$ and $\mathbf{X} \in \mathbb{N}_0^{100\times 489}$, where $\mathbb{N}_0$ is the set of non-negative integers, and $\mathbf{A}, \mathbf{L} \in \mathbb{R}^{50\times 100}$. We estimate the SILVar model (3.18) and compare its performance against a sparse plus low-rank GLM model with an underlying Poisson distribution and fixed link function $g_{\text{GLM}}(x) = \log(1 + e^x)$. We use $n \in \{60, 120, 240, 360\}$ training samples and compute errors on validation and test sets of size 48 each, and learn the model on a grid of $(\lambda_S, \lambda_L) \in \left\{10^{i/4} \middle| i \in \{-8, -7, \ldots, 11, 12\}\right\}^2$. We repeat this 10 times for each setting, using an independent set of training samples each time. We compute testing errors in these cases for the optimal (λ_S, λ_L) with lowest validation errors for both SILVar and GLM models.

We also demonstrate that the low-rank component of the estimated SILVar model captures something intrinsic to the data. Naturally, we expect people's behavior and thus traffic to be different on business days and on non-business days. A standard pre-processing step would be to segment the data along this line and learn two different models. However,

as we use the full dataset to learn one single model, we hypothesize that the learned low-rank $\widehat{\mathbf{L}}$ captures some aspects of this underlying behavior.



(a)



(b)

Figure 3.4: (a) Root mean squared errors (RMSEs) from SILVar and Oracle models; (b) Link function learned using SILVar model

Figure 3.4a shows the test Root Mean Squared Errors (RMSEs) for both SILVar and GLM models for varying training sample sizes, averaged across the 10 trials. We see that the SILVar model outperforms the GLM model by learning the link function in addition

to the sparse and low-rank regression matrices. Figure 3.4b shows an example of the link function learned by the SILVar model with $n = 360$ training samples. Note that the learned SILVar link function performs non-negative clipping of the output, which is consistent with the count-valued nature of the data.

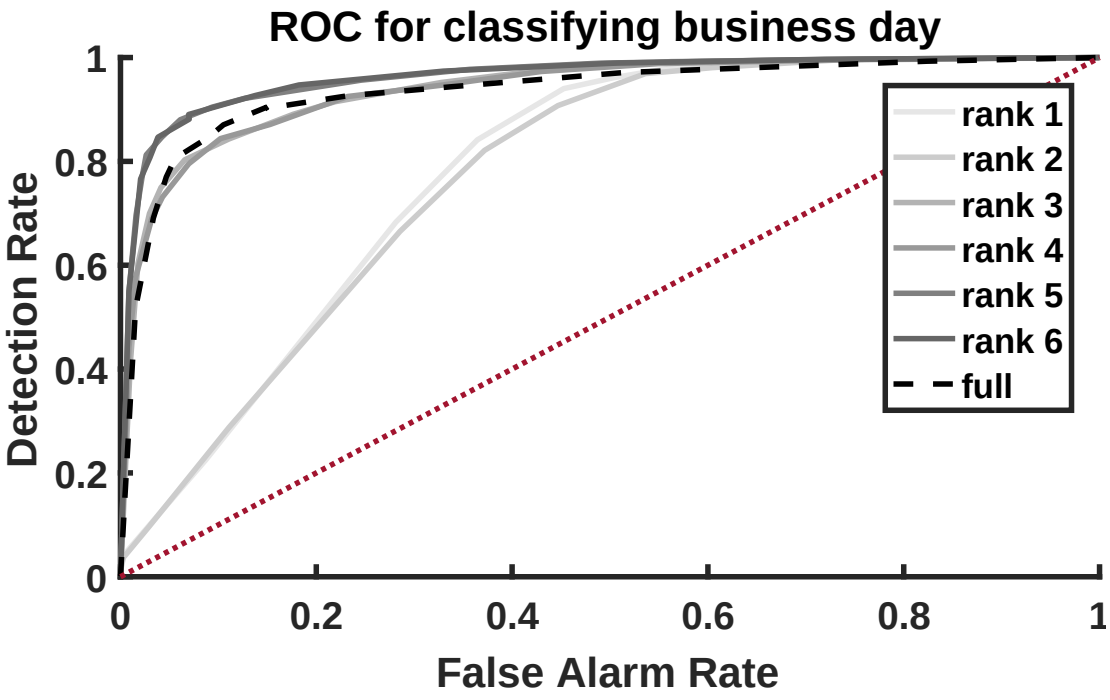


Figure 3.5: Receiver operating characteristics (ROCs) for classifying each day as a business day or non-business day, using the low-rank embedding provided by $\widehat{\mathbf{L}}$ learned from the SILVar model and using the full data

We also test the hypothesis that the learned $\widehat{\mathbf{L}}$ contains information about whether a day is business or non-business through its relatively low-dimensional effects on the data. We perform the singular value decomposition (SVD) on the optimally learned $\widehat{\mathbf{L}} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top$ for $n = 360$ and project the data onto the r top singular components $\widetilde{\mathbf{X}}_r = \widehat{\mathbf{\Sigma}}_r\widehat{\mathbf{V}}_r^\top\mathbf{X}$. We then use $\widetilde{\mathbf{X}}_r$ to train a linear support vector machine (SVM) to classify each day as either a business day or a non-business day, and compare the performance of this lower dimensional feature to that of using the full vector $\mathbf{X}$ to train a linear SVM. If our hypothesis is true

then the performance of the classifier trained on $\widetilde{\mathbf{X}}_r$ should be competitive with that of the classifier trained on $\mathbf{X}$. We use 50 training samples of $\widetilde{\mathbf{X}}_r$ and of $\mathbf{X}$ and test on the remainder of the data. We repeat this 50 times by drawing a new batch of 50 samples each time. We then vary the proportion of business to non-business days in the training sample to trace out a receiver operating characteristic (ROC).

In Figure 3.5, we see the results of training linear SVM on $\widetilde{\mathbf{X}}_r$ for $r \in \{1, ..., 6\}$ and on the full data for classifying business and non-business days. We see that using only the first two singular vectors, the performance is fairly poor. However, by simply taking 3 or 4 singular vectors, the classification performance almost matches that of the full data. Surprisingly, using the top 5 or 6 singular vectors achieves performance greater than that of the full data. This suggests that the projection may even play the role of a de-noising filter in some sense. Since we understand that behavior should correlate well with the day being business or non-business, this competitive performance of the classification using the lower dimensional features strongly suggests that the low-rank $\widehat{\mathbf{L}}$ indeed captures the effects of latent behavioral factors on the data.

Finally, in Figure 3.6, we plot the (i, i) entries of the optimal $\widehat{\mathbf{A}}$ at $n = 360$. This corresponds to locations for which incoming bike rides at 6:00AM-11:00AM are good predictors of outgoing bike rides at 11:00AM-2:00PM, beyond the effect of latent factors such as day of the week. We may intuitively expect this to correlate with locations that have restaurants open for lunch service, so that people would be likely to ride in for lunch or ride out after lunch. This is confirmed by observing that these stations are in Downtown (-80,40.44), the Strip District (-79.975, 40.45), Lawrenceville (-79.96, 40.47), and Oakland (-79.96, 40.44), known locations of many restaurants in Pittsburgh. It is especially interesting to note that Oakland, sandwiched between the University of Pittsburgh and Carnegie Mellon University, is included. Even though the target demographic is largely

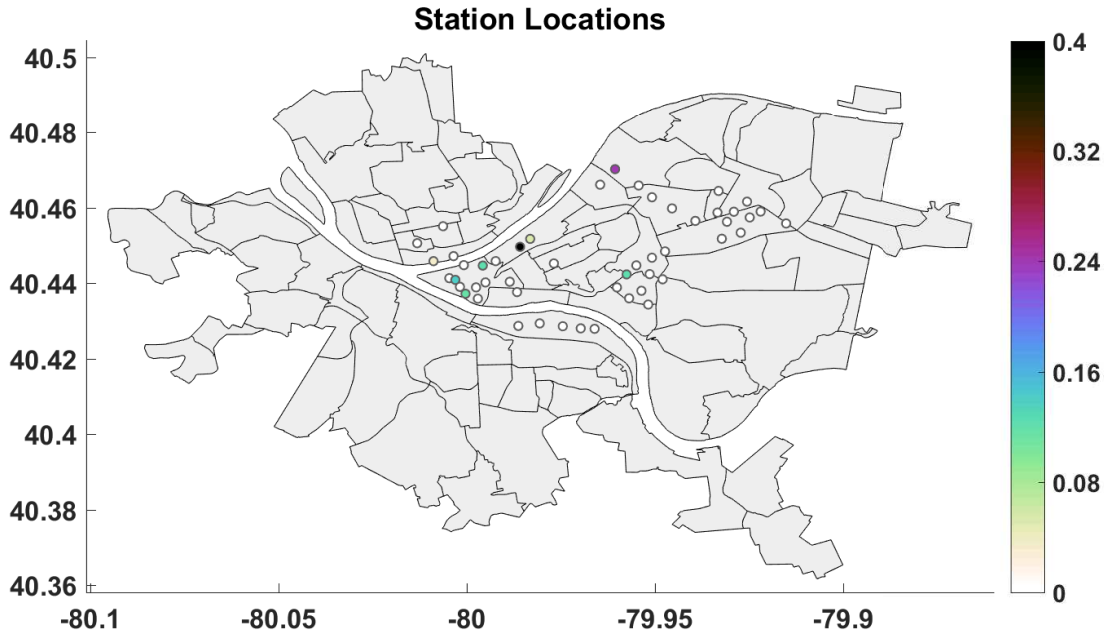


Figure 3.6: Intensities of the self-loop at each station

within walking distance, there is a high density of restaurants open for lunch, which may explain its non-zero coefficient. The remainder of the locations with non-zero coefficients a_{ii} are also near high densities of lunch spots, while the other locations with coefficients a_{ii} of zero are largely either near more residential areas or near restaurants more known for dinner or nightlife rather than lunch, such as Shadyside ($x \geq -79.95$) and Southside ($y \leq 40.43$)).

3.6 Conclusion

Data exhibit complex dependencies, and it is often a challenge to deal with non-linearities and unmodeled effects when attempting to uncover meaningful relationships among various interacting entities that generate the data. We introduce the SILVar model for performing semi-parametric sparse regression and estimating sparse graphs from data under the

presence of non-linearities and latent factors or trends. The SILVar model estimates a non-linear link function g as well as structured regression matrices $\mathbf{A}$ and $\mathbf{L}$ in a sparse and low-rank fashion. We justify the form of the model and relate it to existing methods for general PCA, multi-task regression, and vector autoregression. We provide computationally tractable algorithms for learning the SILVar model and demonstrate its performance against existing regression models and methods on both simulated and real data sets, namely 2011 US weather sensor network data and 2015-2016 Pittsburgh bike traffic data. We see from the simulated data that the SILVar model matches the performance of an Oracle GLM that knows the true link function and only needs to estimate $\mathbf{A}$ and $\mathbf{L}$; we show empirically on the temperature data that the learned $\mathbf{L}$ can capture the effects of underlying trends in time series while $\mathbf{A}$ represents a graph consistent with US weather patterns; and we see that, in the bike data, SILVar outperforms a GLM with a fixed link function, the learned $\mathbf{L}$ encodes latent behavioral aspects of the data, and $\mathbf{A}$ discovers notable locations consistent with the restaurant landscape of Pittsburgh.

Chapter 4

Time Varying Graphs and Principal Network Analysis

Until this chapter, we have considered stationary models in which the parameters of the models – the networks – stay constant through time. Now, we wish to extend these models to the non-stationary case for estimating graph structures that themselves vary in time. We wish to capture significant changepoints of the graphs that otherwise vary smoothly between these points if they exist. Furthermore, we desire an interpretable basis that characterizes the variation of these graphs.

4.1 Notation and Related Works

In this section, we will describe some possible modifications to the previously considered models to capture non-stationarity, several existing methods for time varying graphs in these settings, and the challenges that can arise.

We continue to consider multivariate generalized linear models, so that different assumptions about the nature of the non-linearity and non-stationarity lead to various esti-

mators and algorithms.

To reduce notational clutter when convenient, define

$$\mathcal{A} = (v(\mathbf{A}_1) \ \ldots \ v(\mathbf{A}_K))^\top \in \mathbb{R}^{K \times MN}, \tag{4.1}$$

which collects matrices $\mathbf{A}_k \in \mathbb{R}^{M \times N}$ through time index k, and $v(\mathbf{A})$ denotes the vectorization of a matrix $\mathbf{A}$. Also, we use the various shorthand forms

$$\boldsymbol{\theta}_{kj} = \boldsymbol{\theta}_j(\mathbf{A}_k) = \boldsymbol{\theta}(\mathbf{A}_k, \mathbf{x}_j) = \mathbf{A}_k \mathbf{x}_j \tag{4.2}$$

so that the model is allowed to change at each time step k, and the interpretation for $\boldsymbol{\theta}_{kj}$ is the linear part of the model prediction made by using the network from time k on data from time j. The actual model itself only makes sense at $k = j$, but we will see shortly why we wish to have the notational flexibility to describe these offset $k \neq j$ quantities.

Returning to our model, we have the familiar

$$\mathbb{E}[\mathbf{y}_k | \mathbf{x}_k] = g\left(\mathbf{A}_k \mathbf{x}_k\right) = g\left(\boldsymbol{\theta}_{kk}\right), \tag{4.3}$$

with a possibly non-linear link function g and an associated loss functional at time k of the general offset form

$$f_j(\mathbf{A}_k) = \frac{1}{N}\left[\mathbf{1}^\top \Big(G_*(\mathbf{y}_j) + G(\boldsymbol{\theta}_{kj}) \Big) - \boldsymbol{\theta}_{kj}^\top \mathbf{y}_j \right] \tag{4.4}$$

where $g = \nabla G$. As currently specified, the model is overdetermined. In the following, we review assumptions from prior art that restrict the model space and make the model estimation statistically tractable.

4.1.1 Related Works

The main assumptions take the form that the model does not change "too quickly" (or "too often") through time. This is characterized with respect to some metric for signal smoothness. This assumption tends to take on two primary variants: 1) bounded total variation (TV); or 2) Lipschitz temporal gradient. The behavior of these variants can lead

to strikingly different qualitative interpretations. Low total variation can be viewed as switching between networks at discrete changepoints, whereas Lipschitz gradient can be viewed as a slowly evolving network.

One method for enforcing smoothness on the learned model is to impose a hidden Markov model in a nonparametric Bayesian setting [68]. This model is flexible in imposing low TV by allowing the number of switched networks to be learned simultaneously with the networks themselves. However, it requires Markov Chain Monte Carlo (MCMC) sampling to learn, which provides a more detailed picture about the final learned model than a point estimate but for which convergence of the sampler is in general known to be difficult to determine [69]. For this reason, theory for the resulting estimate is harder to establish. It additionally does not address the presence of latent variables, although it is possible to incorporate into the framework. Instead, we put our emphasis on optimization based approaches and deal with the effects of latent variables.

Optimization based methods also come in several flavors. One approach to impose Lipschitz gradient is to simply use a kernel estimator [13] to obtain locally stationary solutions [70] using our offset loss functionals from (4.4),

$$\widehat{\mathbf{A}}_k = \underset{\mathbf{A}_k}{\operatorname{argmin}} \sum_{j=1}^{K} w_{kj} f_j(\mathbf{A}_k) + h(\mathbf{A}_k), \tag{4.5}$$

where w_{kj} is the kernel weight of a symmetric kernel with some bandwidth η centered at k evaluated at j, and $h(\mathbf{A})$ is a regularizer on the sparsity of the network whose form we omit for simplicity. This is embarrassingly parallel, and allows estimating the network at a single time point without needing to compute those at other time points if deemed irrelevant. However, the smoothness of the kernel makes it more challenging to interpret for detecting discrete changepoints as the low TV setting allows.

Alternatively, an approach for enforcing low total variation [14] is to regularize the

difference between neighboring time points

$$\widehat{\mathcal{A}} = \underset{\mathcal{A}}{\operatorname{argmin}} \sum_{k=1}^{K} f_k(\mathbf{A}_k) + \sum_{k=2}^{K} \|\mathbf{A}_k - \mathbf{A}_{k-1}\|_F. \tag{4.6}$$

This group sparse regularizer encourages the difference between networks at adjacent time points to either be all zero or non-zero. This minimization is over all networks $\mathbf{A}_1, \ldots, \mathbf{A}_K$ through $\mathcal{A}$ defined in (4.1). It couples the problem across time points, so that the solution is a joint estimator and is no longer so simply parallelized. Furthermore, while this model can capture significant changepoints, it is less able to describe smoother variation.

4.1.2 Local Linear Regression and Changepoints

The main conceptual workhorse for our approach is still kernel regression. However, we consider a locally linear regression instead of a vanilla kernel regression. This has the benefit of a lower bias near boundaries of the dataset, so it is natural to adopt for our purposes of changepoint detection. Changepoints can be thought of as a boundary that appears in the middle of the data; alternatively, the boundaries can be thought of as changepoints at the edges of the data domain.

The local linear regression estimator is defined similarly to the kernel estimator. For a clean signal x_k, noise signal v_k, and noisy observations $y_k = x_k + v_k$, the local linear regression estimator is

$$(\widehat{a}_k, \widehat{b}_k) = \underset{a_k, b_k}{\operatorname{argmin}} \sum_{j=1}^{K} w_{kj} \|y_j - (a_k + (j-k)b_k)\|_2^2, \tag{4.7}$$

where $\widehat{a}_k$ is an estimate for the value of the clean signal x_k at time k, while $\widehat{b}_k$ represents an estimate for the slope of the clean signal x_k at time k. This estimator, as compared to kernel regression, has a lower bias near the edges of the data [71].

We will be borrowing a clever use of the local linear regression [72, 73] that estimates smooth 1D signals with discrete discontinuities. They formulate several related problems

with varying kernels,

$$(\widehat{a}_k^{(i)}, \widehat{b}_k^{(i)}) = \underset{a_k, b_k}{\operatorname{argmin}} \sum_{j=1}^{K} w_{kj}^{(i)} \|y_j - (a_k + (j-k)b_k)\|_2^2, \tag{4.8}$$

for $i \in \{\ell, c, r\}$ denoting left, center, and right, respectively, for which the difference between the estimators is in the kernel weights. Specifically, $w_{kj}^{(c)}$ is still a symmetric kernel centered at k evaluated at j, but

$$\begin{aligned} w_{kj}^{(r)} &= \begin{cases} w_{kj}^{(c)} & j \geq k \\ 0 & j < k \end{cases} \\ w_{kj}^{(\ell)} &= w_{jk}^{(r)}. \end{aligned} \tag{4.9}$$

The intuition behind this suite of estimators is that, qualitatively, each estimator will perform best in certain settings depending on the presence and location of changepoints. Consider a single changepoint at location k'. For some small enough κ relative to the kernel bandwidth η, if $k \in [k' - \kappa, k')$, then both the right and center estimators will smooth data with two different local signal values from across the changepoint, while the left estimator will only use data with one local signal value; hence, we would expect the left estimator to be the best. Similarly, we expect the right estimator to perform best when $k \in (k', k' + \kappa]$. Finally, when there is no changepoint, we would expect the center estimator to be the best since it uses more relevant data and the underlying signal is actually smooth, matching the model assumption.

Given this intuition, how do we quantitatively decide which estimator of the three to use? At each time point, we can compute the empirical residuals of each estimator for $i \in \{\ell, c, r\}$,

$$\widehat{\epsilon}_k^{(i)} \triangleq \sum_{j=1}^{K} w_{kj}^{(i)} \|y_j - (a_k + (j-k)b_k)\|_2^2. \tag{4.10}$$

Letting $\gamma_c = \gamma \geq 0$ and $\gamma_\ell = \gamma_r = 1$, we take the index and estimator

$$\begin{aligned} I_k &= \underset{i \in \{\ell, c, r\}}{\operatorname{argmin}} \gamma_i \widehat{\epsilon}_k^{(i)} \\ \widehat{a}_k &= \widehat{a}_k^{(I_k)}. \end{aligned} \tag{4.11}$$

The exact value of γ depends on assumptions about various problem settings, including those on the noise variance, minimum magnitude of the changepoints, and kernel width and shape, just to name a few. Some practical considerations for choices of γ are discussed below equation (4.20) in section 4.2.1.

4.1.3 Matrix Decompositions

Consider the problem of expressing a given matrix as the product of two matrices

$$\mathcal{A} = \mathbf{C}\mathcal{B}^\top \tag{4.12}$$

where $\mathcal{A} \in \mathbb{R}^{M \times N}$, $\mathbf{C} \in \mathbb{R}^{M \times R}$, and $\mathcal{B} \in \mathbb{R}^{N \times R}$ for some $R > 0$. In general, if $\operatorname{rank}(\mathcal{A}) \leq R$ for some positive integer R, then there can be infinite solutions for the matrices $\mathbf{C}$ and $\mathcal{B}$. Particular structure may be imposed on $\mathbf{C}$ and/or $\mathcal{B}$ to obtain desired properties or interpretations. These two matrices can be determined by any number of matrix factorization algorithms, such as a straightforward R-rank singular value decomposition in which

$$\begin{aligned} \mathcal{A} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \\ \mathbf{C} &= \mathbf{U}\mathbf{\Sigma}^\beta \qquad \mathcal{B} = \mathbf{V}\mathbf{\Sigma}^{1-\beta} \end{aligned} \tag{4.13}$$

for some choice of $\beta \in [0, 1]$, typically.

This solution can be seen as a common starting point that can give rise to other solutions, since the existence of the SVD is guaranteed while its computation is stable [74], and it yields one nice structure, with $\mathbf{U}$ and $\mathbf{V}$ defining orthonormal bases. However, since we expect in our application the networks to have different types of structure both temporally (smooth) and spatially (sparse), we would naturally expect any reasonably interpretable decomposition to exhibit qualitatively similar behavior. The SVD does not

necessarily give us these behaviors. Thus, we consider matrix factorization techniques that can give rise to a temporally smooth $\mathbf{U}$ and a spatially sparse $\mathbf{V}$ (where we associate time with $\mathbf{U}$ and network structure with $\mathbf{V}$ indirectly via our arbitrary choice of how to stack $\mathcal{A}$ as in (4.1)).

There are many algorithms for performing matrix factorization, such as probabilistic frameworks that utilize sampling [75]. Again, we focus on optimization based frameworks. Many of these tend to center around variational methods [76] or ADMM [77, 78]. Ultimately, we choose to use inertial Proximal Alternating Linear Minimization (iPALM) [79]. The iPALM method has similarities in form with ADMM, as they both utilize the separability of the objective function and computationally fast proximal operators. However, iPALM guarantees that certain types of nonconvex optimization problems, including regularized matrix factorization, converge to local minima via an adaptive choice of step size. We provide more details on implementation in section 4.3.

4.2 Smooth Regression with Changepoints

We propose an optimization based approach that learns time varying graph structure jointly based on the full time series data and attempt to bridge the gap between a smooth solution and the estimation of changepoints, as previously discussed. Our method offers the parallelizability of the previous smooth approach while being able to capturing both smooth variation *and* significant jumps.

4.2.1 Formulation and computation

To reduce notational clutter, we omit the low-rank component, but the formulations follow exactly analogously from Chapter 3 (i.e., we can consider the additive low-rank contribution

from latent variables as $\mathcal{A} \leftarrow \mathcal{A}+\mathcal{L}$). We formulate three related locally linear regression based optimization problems,

$$\begin{aligned}(\widehat{\mathcal{A}}^{(i)}, \widehat{\mathcal{A}}'^{(i)}) &= \underset{\mathcal{A},\mathcal{A}}{\operatorname{argmin}} \; F^{(i)}(\mathcal{A}, \mathcal{A}') \\ &\triangleq \underset{\mathcal{A},\mathcal{A}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{j=1}^{K} w_{kj}^{(i)} f_j(\mathbf{A}_k + (j-k)\mathbf{A}'_k) + \lambda \mathrm{h}(\mathcal{A})\end{aligned} \tag{4.14}$$

for $i \in \ell, c, r$, and where $h(\mathcal{A})$ is a regularizer on the structure of $\mathcal{A}$, the $\mathbf{N} \times N$ matrix $\mathcal{A}'$ corresponds to the instantaneous time derivative of $\mathcal{A}$ and is essentially a nuisance parameter in our setting, and the superscripts of ℓ, c, r denote left, center, and right, respectively. For $h_1(\mathcal{A})$, we can consider for example a group sparsity that corresponds to Granger causality if $\mathcal{A}$ is autoregressive (AR) coefficient matrices, similar to before,

$$\mathrm{h}(\mathcal{A}) = \sum_{k=M+1}^{K} \sum_{i,j} \|\mathbf{a}_{kij}\|_2 \tag{4.15}$$

where $\mathbf{a}_{kij} = \Big([\mathbf{A}_{k,1}]_{ij} \; \ldots \; [\mathbf{A}_{k,M}]_{ij} \Big)$, the vectors collecting the ij elements of the respective AR coefficient matrices across all lag orders. Again, as before, we can consider alternate forms of prior assumptions, including the commutativity corresponding to graph filters in the Causal Graph Process model from chapter 2. However, since it is a non-convex penalty, further discussion will remain outside the scope of this chapter but is noted as an interesting direction for future work.

With each of these three estimators, we can see that they remain still trivially parallelizable across time points k,

$$\begin{aligned}(\widehat{\mathbf{A}}_k^{(i)}, \widehat{\mathbf{A}}_k'^{(i)}) &= \underset{\mathbf{A}_k,\mathbf{A}'_k}{\operatorname{argmin}} \; F_k^{(i)}(\mathbf{A}_k, \mathbf{A}'_k) \\ &\triangleq \underset{\mathbf{A}_k,\mathbf{A}'_k}{\operatorname{argmin}} \sum_{j=1}^{K} w_{kj}^{(i)} f_j(\mathbf{A}_k + (j-k)\mathbf{A}'_k) + \lambda \mathrm{h}_1(\mathbf{A}_k)\end{aligned} \tag{4.16}$$

where $\mathrm{h}_1(\mathbf{A}_k)$ simply acts on one of the $\mathbf{A}_k$.

Then following the intuition from section 4.1.2, at each time point, we can compute the

empirical residuals of each estimator for $i \in \{\ell, c, r\}$,

$$\widehat{\epsilon}_k^{(i)} \triangleq \sum_{j=1}^{K} w_{kj}^{(i)} f_j \left(\widehat{\mathbf{A}}_k^{(i)} + (j-k)\widehat{\mathbf{A}}_k^{\prime(i)} \right) \tag{4.17}$$

In general, we can also use our favorite loss functions between the estimates and the true values,

$$\widetilde{\epsilon}_k^{(i)} \triangleq \sum_{j=1}^{K} w_{kj}^{(i)} D \left(g \left(\boldsymbol{\theta}_j \left(\widehat{\mathbf{A}}_k^{(i)} + (j-k)\widehat{\mathbf{A}}_k^{\prime(i)} \right) \right) \,\middle\|\, \mathbf{y}_j \right). \tag{4.18}$$

Letting $\gamma_c = \gamma \geq 0$ and $\gamma_\ell = \gamma_r = 1$, we take the estimate

$$\begin{aligned} I_k &= \underset{i \in \{\ell, c, r\}}{\operatorname{argmin}} \gamma_i \widehat{\epsilon}_k^{(i)} \\ \widehat{\mathbf{A}}_k &= \widehat{\mathbf{A}}_k^{(I_k)}, \end{aligned} \tag{4.19}$$

where again γ is determined by the problem setting, and in practice could be chosen via some validation procedure. Finally, the changepoints can be detected by considering the set of indices for which

$$\widehat{\mathcal{J}} = \left\{ k : (I_k = \ell) \bigcap (I_{k+1} = r) \right\}. \tag{4.20}$$

This tends to produce smooth behavior except near the changepoints, where the one-sided estimators are used. As an extra post-processing step, we may compute the central estimate on the boundaries of the contiguous segments to ensure smoothness in these regions.

For $\gamma = 0$, we always choose the central estimator and declare no changepoints, while for $\gamma \to \infty$ we never choose the central estimator. This range of γ values draws out the ROC for the changepoint detector. In fact, it is shown in [72, 73] that, when $\lambda = 0$, even $\gamma > 1$ implies (theoretically) never using the central estimator in the interior of the interval, and thus they advise against this. However, we argue that, if a quick implementation is desired, we could implicitly choose $\gamma = \infty$ by forgoing the computation of the central estimator altogether in the first pass. We would instead compare the crossover points for the errors of the left and right estimators to estimate changepoints.

Then, as a smooth solution is desired, we can run the central estimator on the contiguous

segments as in the optional post-processing step described previously. This gives us one fewer parameter and lower computation at the cost of some robustness, as this requires both left and right estimators to be estimated reasonably well. Using this modification, we would detect more false changepoints that might otherwise be classified as being best fit by the central estimator under a regime for which $\gamma \leq 1$.

This intuition leads to a speedup of the original detection scheme for generic γ as well. We can first find all the potential changepoints as determined by these crossovers. Then *only* at the two time points on either side of the crossover, compute the central estimators to verify that the left and right estimators are indeed better fit than the central (i.e., that $\mathbf{I}_k \neq 0$ for either of these points).

Finally, we point out that formulating this problem as a generic regression allows us to consider graphs as either directed or undirected, as established previously (Chapter 3, Section 3.2.2). To estimate directed graphs on a time series, we may for example use an autoregressive model

$$\begin{Bmatrix} \mathbf{y}_k \\ \mathbf{x}_k \\ \mathbf{A}_k \end{Bmatrix} \leftarrow \begin{Bmatrix} \mathbf{x}_k \\ (\mathbf{x}_{k-1}^\top \ \ldots \ \mathbf{x}_{k-M}^\top)^\top \\ (\mathbf{A}_{k,1} \ \ldots \ \mathbf{A}_{k,M}) \end{Bmatrix} \tag{4.21}$$

and to estimate undirected graphs, for example,

$$\begin{Bmatrix} \mathbf{y}_k \\ \mathbf{x}_k \\ \mathbf{A}_k \end{Bmatrix} \leftarrow \begin{Bmatrix} \mathbf{x}_k \\ \mathbf{x}_k \\ (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \odot \mathbf{A}_k \end{Bmatrix} \tag{4.22}$$

where in row i we regress $[\mathbf{x}_k]_{\setminus i}$ (all indices of $\mathbf{x}_k$ except for the ith) on $[\mathbf{x}_k]_i$ (the ith index of $\mathbf{x}_k$) by enforcing $\mathbf{A}_k$ to have 0's on the diagonal. In general, we can also enforce any arbitrary nonzero structure $\mathbf{J} \in \{0,1\}^{M \times N}$ on $\mathbf{A}_k$ if we have corresponding prior structural information on the graph matrix.

4.2.2 Estimation Procedure

Since our optimization problems (4.14) are convex, to produce our estimates, we can use any number of convex algorithms to solve the formulation. For concreteness, we provide an outline for one such approach using proximal gradient methods.

The gradient computations required are,

$$\begin{aligned}\frac{\partial F^{(i)}}{\partial \mathbf{A}_k} &\triangleq \mathbf{E}_k = \mathbf{J} \odot \sum_{j=1}^{K} w_{kj}^{(i)} \left(g(\boldsymbol{\theta}_{kj}) - \mathbf{y}_j\right) \mathbf{x}_j^\top \\ \frac{\partial F^{(i)}}{\partial \mathbf{A}_k'} &\triangleq \mathbf{H}_k = \mathbf{J} \odot \sum_{j=1}^{K} (j-k) w_{kj}^{(i)} \left(g(\boldsymbol{\theta}_{kj}) - \mathbf{y}_j\right) \mathbf{x}_j^\top\end{aligned} \tag{4.23}$$

where $\mathbf{J}$ is the binary mask enforcing the 0 structure. We can let $\mathcal{E}$ and $\mathcal{H}$ collect and stack $\{\mathbf{E}_k\}$ and $\{\mathbf{H}_k\}$ (respectively) the same way as $\mathcal{A}$ collects and stacks $\{\mathbf{A}_k\}$,

$$\begin{aligned}\mathcal{E} &= \left(v(\mathbf{E}_1) \ \ldots \ v(\mathbf{E}_K)\right)^\top \\ \mathcal{H} &= \left(v(\mathbf{H}_1) \ \ldots \ v(\mathbf{H}_K)\right)^\top.\end{aligned} \tag{4.24}$$

Finally, we need the proximal operators for the regularizers,

$$\text{prox}_1(\mathbf{A}_k, t) = \underset{\mathbf{Y}}{\text{argmin}} \frac{1}{2}\|\mathbf{A}_k - \mathbf{Y}\|_F^2 + t\text{h}_1(\mathbf{Y}) \tag{4.25}$$

For $\text{h}_1(\mathbf{A}_k) = \sum_{i,j} \|\mathbf{a}_{kij}\|_2$, we have

$$\begin{aligned}&\mathbf{S}_k = \text{prox}_1(\mathbf{A}_k, t) \\ \Longrightarrow &\mathbf{s}_{kij} = \begin{cases} \left(\frac{\|\mathbf{a}_{kij}\|_2 - t}{\|\mathbf{a}_{kij}\|_2}\right) \mathbf{a}_{kij} & \|\mathbf{a}_{kij}\|_2 > t \\ 0 & \|\mathbf{a}_{kij}\|_2 \leq t \end{cases},\end{aligned} \tag{4.26}$$

where the indexing of $\mathbf{s}_{kij}$ within $\mathbf{S}_k$ is the same as that of $\mathbf{a}_{kij}$ within $\mathbf{A}_k$, which corresponds to a form of group soft thresholding that can be performed quickly.

To put it all together, algorithm 6 describes the full proximal gradient implementation.

4.3 Principal Network Analysis

While our method for estimating time-varying networks is fairly general, in many cases we might expect that the graphs vary in time as a linear combination of some small set of so called "Principal Networks." Thus, we can decompose the estimated set of time varying graphs as

$$\mathcal{A} = \mathbf{C}\mathcal{B}^\top, \tag{4.31}$$

where $\mathbf{C} \in \mathbb{R}^{K-M\times R}$ collects the time-varying weights and $\mathcal{B} \in \mathbb{R}^{MN^2\times R}$ are the vectorized Principal Networks. The matrices $\mathbf{C}$ and $\mathcal{B}$ can be determined by any number of matrix factorization algorithms. We use an inertial version of Proximal Alternating Linearized Minimization (PALM, or iPALM for the inertial version) because of its theoretical guarantee of convergence to stationary points without requiring a tuning parameter [79, 80].

We perform matrix factorization on the output from algorithm 6 directly. To reduce notational clutter, we omit the low-rank component, but the formulations follow exactly analogously from Chapter 3 (i.e., we can consider the additive low-rank contribution from latent variables as $\mathcal{A}\leftarrow\mathcal{A}+\mathcal{L}$). Before we pose the optimization problem, we introduce the regularization to be used. For some $R > 0$, let

$$\widetilde{\mathbf{A}}_k = \widetilde{\mathbf{A}}(\mathcal{B}, \mathbf{c}_k) = \sum_{r=1}^{R} c_k^{(r)} \mathbf{B}^{(r)}, \tag{4.32}$$

where

$$\begin{aligned}
\mathbf{c}_k &= (c_k^{(1)} \ \dots \ c_k^{(R)})^\top \in \mathbb{R}^{R\times 1} \\
\mathbf{C} &= (\mathbf{c}_1 \ \dots \ \mathbf{c}_K)^\top \in \mathbb{R}^{K\times R} \\
\mathbf{B}^{(r)} &\in \mathbb{R}^{M\times N} \\
\mathcal{B} &= \left(v(\mathbf{B}^{(1)}) \ \dots \ v(\mathbf{B}^{(R)})\right) \in \mathbb{R}^{MN\times R} \\
\widetilde{\mathcal{A}} &= \mathbf{C}\mathcal{B}^\top \in \mathbb{R}^{K\times MN}.
\end{aligned} \tag{4.33}$$

Let our sparsity regularizer be $h_1(\mathcal{B})$. For concreteness, in the autoregressive (AR)

setting we may choose a group regularization corresponding to Granger Causality,

$$h_1(\mathcal{B}) = \sum_{r=1}^{R} \|\mathbf{b}_{rij}\|_2 \tag{4.34}$$

where similarly to before $\mathbf{b}_{rij} = \left(\left[\mathbf{B}_1^{(r)}\right]_{ij} \ \dots \ \left[\mathbf{B}_M^{(r)}\right]_{ij} \right)$, the vectors collecting the ij elements of the respective AR coefficient matrices across all lag orders.

Also, let our low rank regularizer be

$$h_*(\widetilde{\mathcal{A}}) = \frac{1}{2}\left(\|\mathbf{C}\|_F^2 + \|\mathcal{B}\|_F^2\right). \tag{4.35}$$

Now, we give the matrix factorization optimization problem,

$$\underset{\mathcal{B},\mathbf{C}}{\operatorname{argmin}} \ F_{\text{mf}}(\mathcal{B}, \mathbf{C}) \triangleq \frac{1}{2}\|\widehat{\mathcal{A}} - \mathbf{C}\mathcal{B}^\top\|_F^2 + \lambda_* h_*(\widetilde{\mathcal{A}}) + \lambda_1 h_1(\mathcal{B}). \tag{4.36}$$

The Principal Network Analysis (PNA) yields sparse $\mathcal{B}$ that can be viewed as our principal networks, which fundamentally underlie the process and are present at any given time point in some linear combination with each other, with temporally smoothly varying weights $\mathbf{C}$ potentially with changepoints as a result of $\mathcal{A}$ being estimated in the same way. We could alternatively regularize $\mathbf{C}$ to be smooth (e.g., via a grouped sparse version of Total Generalized Variation [81]) while not regularizing $\mathcal{B}$ or even to regularize both. However, since we perform this factorization on the estimated $\widehat{\mathcal{A}}$, which is already sparse and smooth, it is somewhat redundant and only increases the computational complexity.

To solve our nonconvex problem (4.36) using iPALM, we require a gradient computation on the smooth parts of the loss function and proximal operator computations for the non-smooth parts. Let

$$H(\mathcal{B}, \mathbf{C}) = \frac{1}{2}\|\widehat{\mathcal{A}} - \mathbf{C}\mathcal{B}^\top\|_F^2 + \lambda_* h_*(\widetilde{\mathcal{A}}). \tag{4.37}$$

The gradient computations required are

$$\begin{aligned} \frac{\partial H}{\partial \mathcal{B}} &= \mathcal{Q}^\top \mathbf{C} + \lambda_* \mathcal{B} \\ \frac{\partial H}{\partial \mathbf{C}} &= \mathcal{Q}\mathcal{B} + \lambda_* \mathbf{C}, \end{aligned} \tag{4.38}$$

where $\mathcal{Q} = \mathbf{C}\mathcal{B}^\top - \widehat{\mathcal{A}}$. To apply iPALM, we also need to compute an upper bound for the

following Lipschitz constants,

$$\begin{aligned} L_{\mathcal{B}}(\mathbf{C}) &\triangleq \frac{\left\| \frac{\partial H}{\partial \mathcal{B}}(\mathcal{U}, \mathbf{C}) - \frac{\partial H}{\partial \mathcal{B}}(\mathcal{V}, \mathbf{C}) \right\|_F}{\|\mathcal{U} - \mathcal{V}\|_F} \\ L_{\mathbf{c}}(\mathcal{B}) &\triangleq \frac{\left\| \frac{\partial H}{\partial \mathbf{C}}(\mathcal{B}, \mathbf{U}) - \frac{\partial H}{\partial \mathbf{C}}(\mathcal{B}, \mathbf{V}) \right\|_F}{\|\mathbf{U} - \mathbf{V}\|_2}. \end{aligned} \tag{4.39}$$

Thus we consider

$$\begin{aligned} \left\| \frac{\partial H}{\partial \mathcal{B}}(\mathcal{U}, \mathbf{C}) - \frac{\partial H}{\partial \mathcal{B}}(\mathcal{V}, \mathbf{C}) \right\|_F &= \|(\mathcal{U} - \mathcal{V})\mathbf{C}^\top \mathbf{C} + \lambda_*(\mathcal{U} - \mathcal{V})\|_F \\ &\leq \left(\|\mathbf{C}^\top \mathbf{C}\|_F + \lambda_* \right) \|\mathcal{U} - \mathcal{V}\|_F \\ \implies L_{\mathcal{B}}(\mathbf{C}) \leq \overline{L}_{\mathcal{B}}(\mathbf{C}) &\triangleq \|\mathbf{C}\|_F^2 + \lambda_*. \end{aligned} \tag{4.40}$$

Similarly,

$$\begin{aligned} \left\| \frac{\partial F}{\partial \mathbf{C}}(\mathcal{B}, \mathbf{U}) - \frac{\partial F}{\partial \mathbf{C}}(\mathcal{B}, \mathbf{V}) \right\|_F &= \|(\mathbf{U} - \mathbf{V})\mathcal{B}^\top \mathcal{B} + \lambda_*(\mathbf{U} - \mathbf{V})\|_F \\ \implies L_{\mathbf{C}}(\mathcal{B}) \leq \overline{L}_{\mathbf{C}}(\mathcal{B}) &\triangleq \|\mathcal{B}\|_F^2 + \lambda_*. \end{aligned} \tag{4.41}$$

These upper bounds on the Lipschitz constants adaptively determine the step sizes in each coordinate block at each iteration, and thus they are slightly loose but more straightforward to compute than some other possible tighter bounds. Thus iPALM requires a bit more derivation up front to apply, but does not need tuning for step sizes, as compared to ADMM, which does not require the Lipschitz computation, but has a step size or learning rate parameter that can require properly tuning or setting in more challenging non-convex problems [77, 78].

Finally, we need the proximal operators for the regularizers,

$$\begin{aligned} \text{prox}_1(\mathcal{B}, t) &= \underset{\mathcal{Y}}{\text{argmin}} \frac{1}{2}\|\mathcal{B} - \mathcal{Y}\|_F^2 + t\text{h}_1(\mathcal{Y}) \\ \text{prox}_s(\mathbf{C}, t) &= \mathbf{C}, \end{aligned} \tag{4.42}$$

which can be solved quickly using soft thresholding. Putting everything together, we have algorithm 7.

We note that a natural direction for an interesting line of future work would be to combine these two steps of first estimating a time varying graph and then performing matrix factorization into a single joint formulation that directly learns the factors, or

iterates between the two steps as subproblems. This could bridge the gap between a full temporally coupled solution and its computation, since the factorized form would have fewer total parameters to estimate.

4.4 Experiments

We test the time varying graph estimation on simulated and real data sets. We perform principal network analysis on the results of a US Senate voting record data set to yield the principal networks that we show has interpretable meaning in the context of the contemporary political climate.

4.4.1 Simulated Data

We generated networks of size $N \in \{25, 50\}$ of length $K \in \{100, 250\}$. We generated $\mathbf{A}$ randomly as Erdös-Rényi in structure and $\mathcal{N}(0, 1)$ in weight with a diagonal uniformly generated in $[1, 2)$. We generate the time-varying weights as piecewise constants with magnitude generated uniformly at random in $[2, 4)$ plus a sinusoid of magnitude also generated uniformly at random in $[1/4, 1/2)$, in which changepoints are generated uniformly at random in $\left[\frac{K-M}{R+1}(i - \frac{1}{8}), \frac{K-M}{R+1}(i + \frac{1}{8})\right)$ for $i = 1, \ldots, R$ where R is the number of changepoints. This is a fairly difficult setting conceptually, as the non-zero structure of the graphs is actually constant throughout the duration of the time series as all of the true underlying graphs are present, but the magnitudes change, so that the changepoints are jumps in magnitude rather than structural changes in existence of edges. In figure 4.1, we see that the near the detected changepoint, using the same sparsity regularization parameter λ, the networks estimated with the changepoints are somewhat better able to capture the weights on either side of the changepoint, while the central estimator seems to average too strongly across

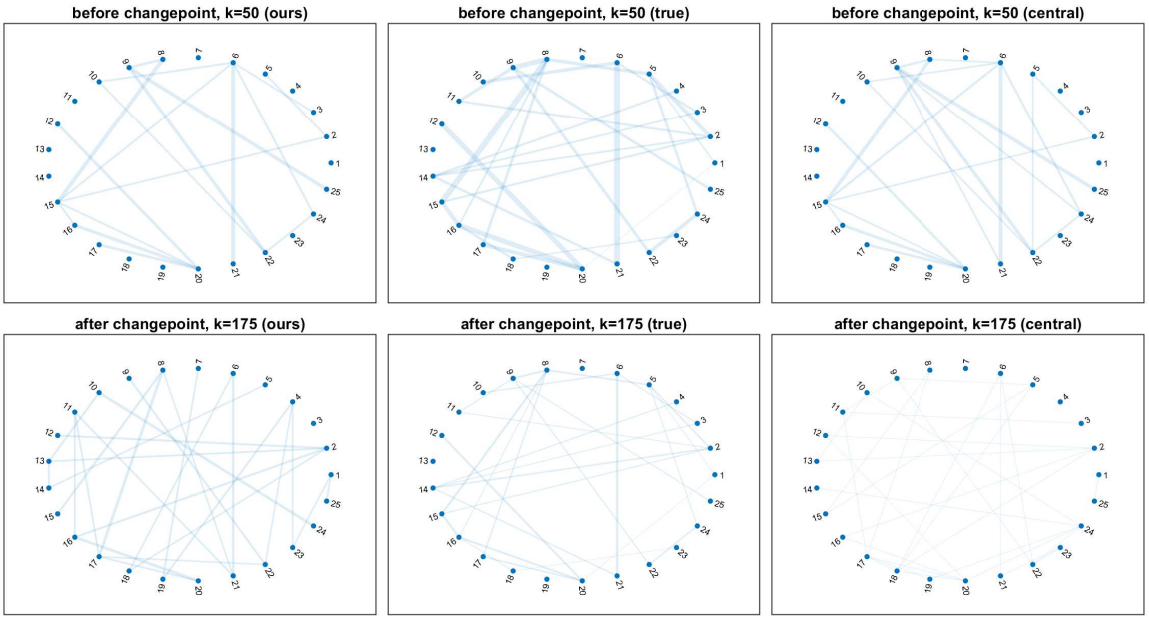


Figure 4.1: Graphs of the top 50 edges

the changepoint, resulting in weaker edges. We show in figure 4.2 the performance of our

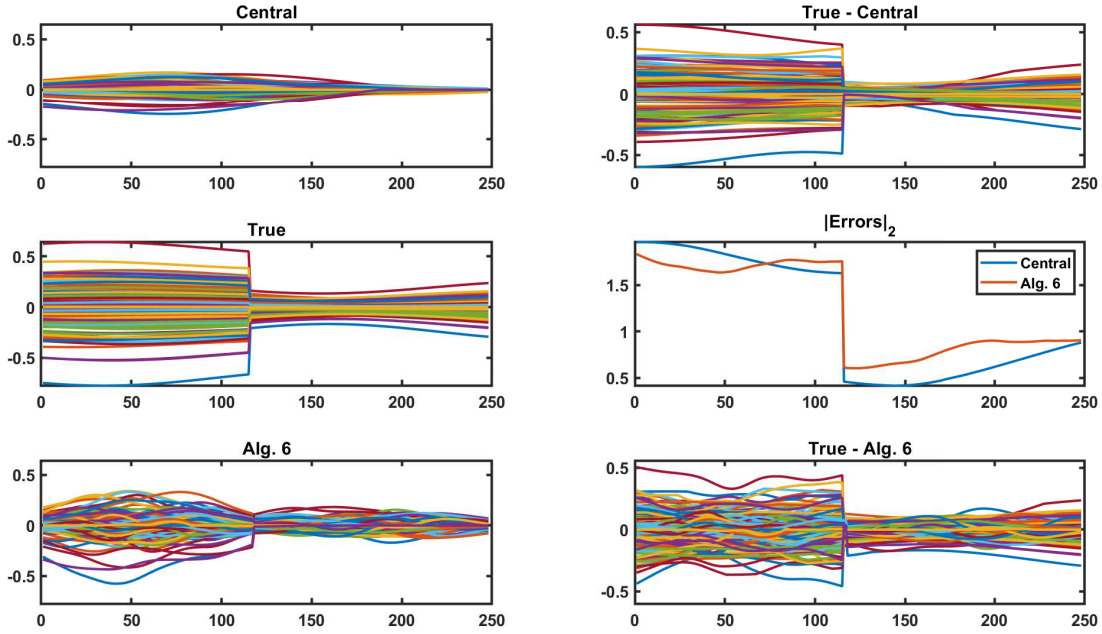


Figure 4.2: Time series of elements of graph adjacency matrix and corresponding errors

estimation method as compared to the central estimator for $\lambda = 0.1$ and $w_{kj}^{(c)} \propto e^{-\frac{(j-k)^2}{200}}$. In figure 4.2 each line represents the time series of the weights corresponding to one edge. We see that using our method, the changepoint is estimated well, and the overall error is less than that of the normal central kernel estimate. We note that the estimated magnitudes of the edge weights for the graph are shrunken as a result of the regularization, and also that there is still some bias towards 0 at the boundaries. This suggests an extension of the method would be to let the regularization parameter λ vary through the interval rather than stay constant. A proper time profile of the λ value would be of interest.

4.4.2 US Senate Voting Data

We also compiled $K = 500$ votes from the United States senate roll call records of sessions 111-112, corresponding to a period from 2010-2011 [82, 83]. There are 2 senators from each of 50 states, so that there are at most $N = 100$ senators voting on each item. However, instead of directly tracking a growing and shrinking network composed of individual senators, we track the seat that the senator occupies, so that the time series of votes generated by senators continue being produced by their replacements. We treat yes votes as $+1$, no votes as -1, and abstentions/vacancies as 0 so that the time series is $\mathbf{X} \in \{-1, 0, +1\}^{100 \times 500}$ (see figure 4.3).

Since a significant portion of votes is nearly unanimous and/or procedural as according to senate rules, as opposed to actually debated and/or legislative and thus truly informative, it is unclear that modeling the effect that temporally neighboring votes have on each other is more meaningful than recovering the reciprocal relations (at least using the frameworks presented in this chapter). For some examples of such unanimous or procedural votes, see table 4.1 [83]. Thus, we apply the undirected graph estimation model from equation (4.22) with the low-rank component included to track the network of relations

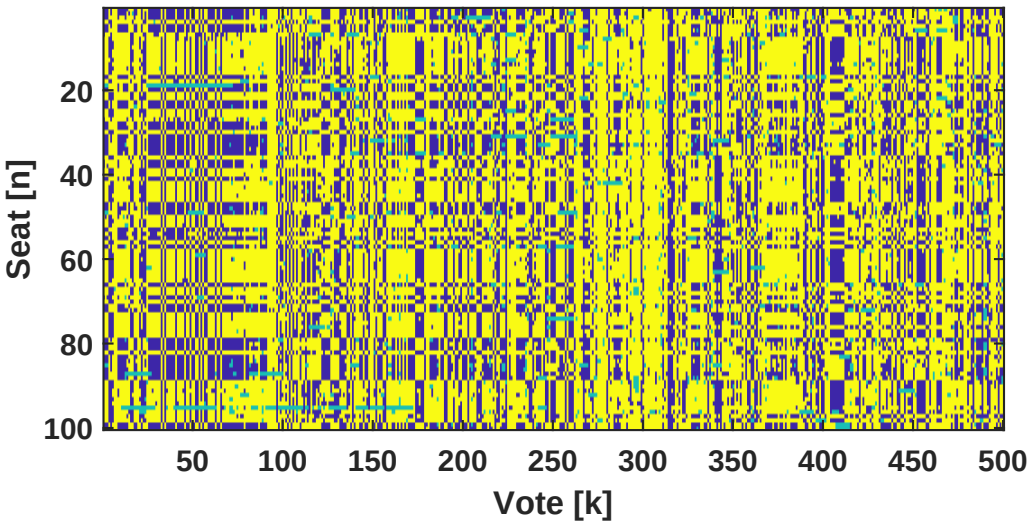


Figure 4.3: Time series of votes by seat, yellow (+1) for Yea, blue (-1) for Nay, and green (0) for abstention/vacant seat.

Table 4.1: Examples of unanimous/procedural votes

Yea - Nay	Outcome	Purpose
97 - 0	Agreed to	A resolution honoring the victims and heroes of the shooting... in Tucson, Arizona.
96 - 1	Agreed to	... to provide penalties for aiming laser pointers at airplanes...
100 - 0	Confirmed	Confirmation Robert S. Mueller, III, of California, to be Director of the Federal Bureau of Investigation
0 - 97	Rejected	... budget request for the United States Government for fiscal year 2012, and... budgetary levels for fiscal years 2013 through 2021.

among the senate.

We use a slightly different sparsity regularization to account for the symmetric structure of the adjacency matrix to be estimated,

$$\mathrm{h}(\mathcal{A}) = \sum_k \sum_{i<j} \|([\mathbf{A}_k]_{ij}\ [\mathbf{A}_k]_{ji})\|_2 \,.$$

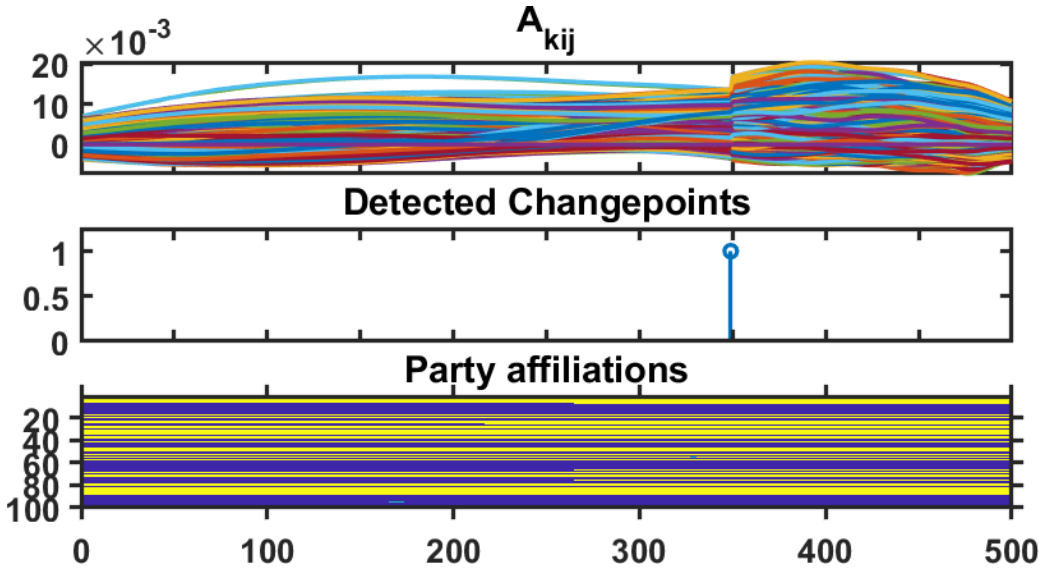


Figure 4.4: Top: time series of entries of estimated graph adjacency. Each color is one time element of the matrix. Middle: Changepoints detected by algorithm. Bottom: Actual party affiliations for each seat, yellow for Republican, blue for Democrat.

We see at the top of figure 4.4 the $N^2 = 10^4$ individual elements of the estimated adjacency matrix, one color for each ij entry plotted as a function of time k. In the middle, we see that the algorithm detected a changepoint at timepoint 349. In the bottom of the figure, we see evidence that there was actually the start of a new session (transition from 111-112) at timepoint 264 (Jan. 2011) in which 5 new senators were elected of the opposite political party to the previous senator they replaced (Republicans replacing Democrats). This happened as the so-called "Tea Party," which started as a vocal grassroots movement and was active in organizing national demonstrations, had gained major traction in the

US Republican Party [84]. However, there is a significant lag in the detection of this changepoint. We conjecture that this is due to the large segment of largely uninformative unanimous votes in the beginning of the new session (see figure 4.3) that were predicted equally well (or poorly) under either left or right model.

In figures 4.5 and 4.6, we compare estimated networks using our method and the central kernel estimate from $k = 175$ and $k = 425$, corresponding to two points respectively before and after the start of the new session as well as the detected changepoint. The top layouts are achieved by using a planar force-directed embedding based on the edges estimated in $\widehat{\mathbf{A}}_k$ (in particular, we use the Fruchterman-Reingold [44] layout as implemented in MATLAB ® version 2018a). Intuitively, shorter average path lengths between nodes correspond to larger forces and lower distances in the planar embedding. The bottom layouts are computed using the top layouts as initialization points. Neighboring points vote more similarly, signifying ideological closeness. We additionally label seats according to the ground truth party of the senator occupying them at that time, with Republicans in red, Democrats in blue, and the Independent in yellow. In figure 4.5, we can see that the graph clearly shows the polarization between the two parties. Interestingly, the Independent is shown as ideologically closer to the democrats but on the closest edge to the Republicans. Indeed, the Independent is Senator Sanders, whom we know to hold views that are espoused by both parties, but in 2016 ultimately ran in the Democratic primary race to try to become the party's presidential candidate. In addition, we point out two seats, 7 (Arkansas) and 27 (Indiana) who were Democratic prior to the election, but switched to Republican afterwards. These senators were on the boundary of the Democratic cluster to begin with, which is consistent with the fact that both states historically lean Republican as measured by the previous 8 presidential elections (2008 was in fact an exception for Indiana, while 1992 and 1996 were exceptions for Arkansas as the Democratic candidate

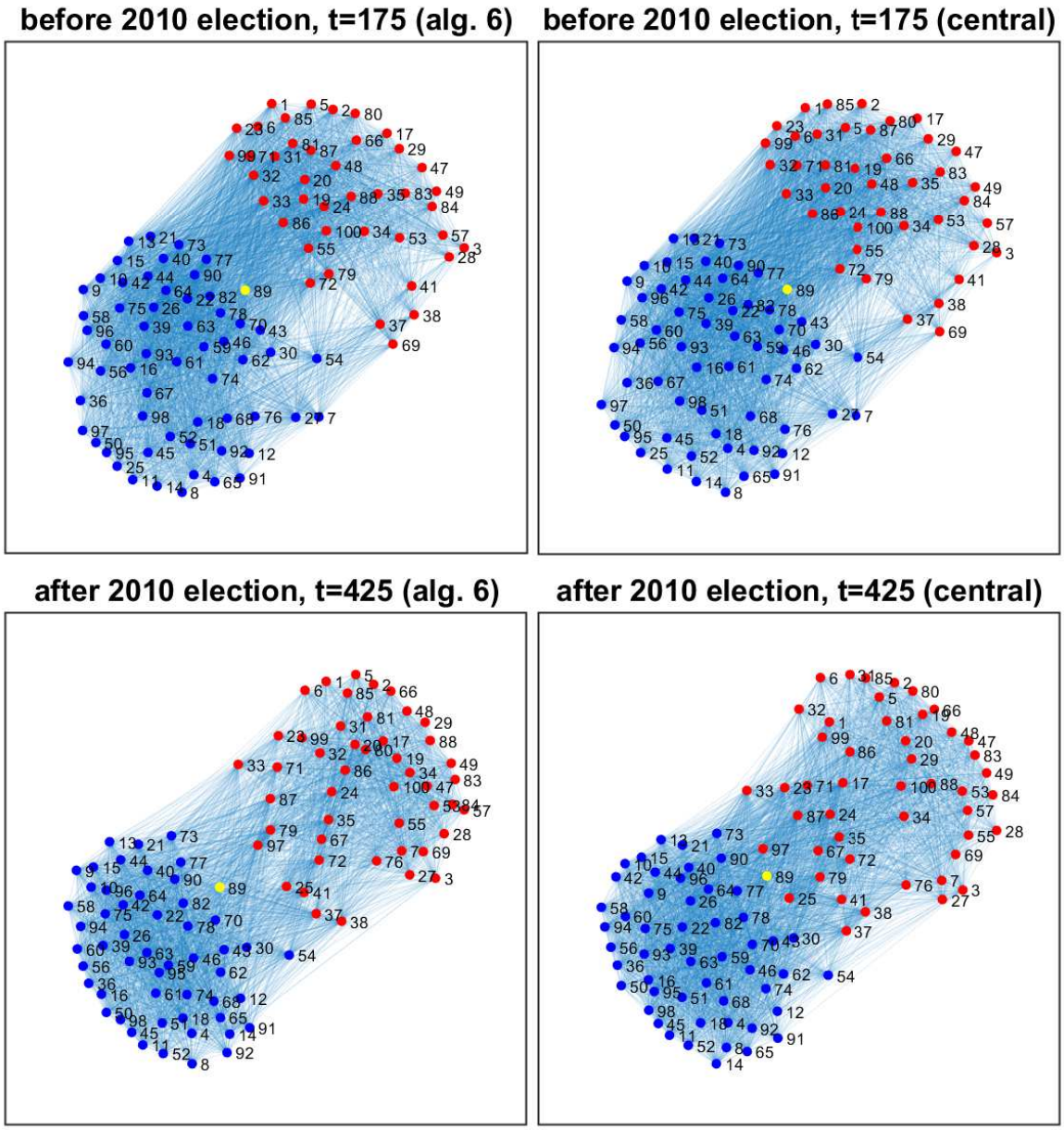


Figure 4.5: Top: Estimated adjacency graph from middle of first session; Bottom: Estimated adjacency graph from middle of second session; Left: Our method; Right: Central estimator

was from Arkansas) [85, 86]. Also, we note that the shapes of the clusters change, with the Democratic clique flattening while the Republican clique stretches. We must be careful in

drawing any conclusions from this phenomenon, but this could weakly support (i.e., not provide any evidence to disprove) the conclusions drawn by political researchers claiming that official recognition of the "Tea Party Movement" and its unity resulted in the Republican Party moving ideologically further away from the previous center of the political space [87, 88]. Finally, we see that the two methods estimate very similar graphs, showing that our method performs a similar estimation to the smooth central kernel estimator in regimes away from changepoints.

In figure 4.6, we visualize graphs estimated on either side of the detected changepoint at $k = 349$, this time much closer to the changepoint. We see that the central kernel shows less change between the two timepoints, while there is a noticeable difference across the changepoint in our method. In the top row, there is also some movement relative to the timepoints visualized in figure 4.5, possibly corresponding to posturing ahead of the election. Continuing our previous interpretations, this could be the existing Republicans trying to maintain their seats over "Tea Party" primary challengers [87]. We clarify one subtlety on this theory: though the visualized timepoint is actually *after* the beginning of the new session, the kernel bandwidth is such that the votes from *before* in the old session still influence the estimated graph in both methods, as the visualized timepoint still occurs before the *detected* changepoint. Finally, in figure 4.7 we visualize the result of PNA applied to this dataset, again using a force-directed layout. This time, note that we do not label the nodes by color since these networks are now across all time, so several seats could be either red or blue depending on the time. We find that the first two Principal Networks (PN) capture the main overall polarized structure of the two parties. PN 3, on the other hand, seems to highlight the most salient and influential seats in the senate. In this case, we see visually that the time series for PN 3 has the largest time derivatives, so that the network corresponds to the edges that had the largest changes throughout the dataset. Again,

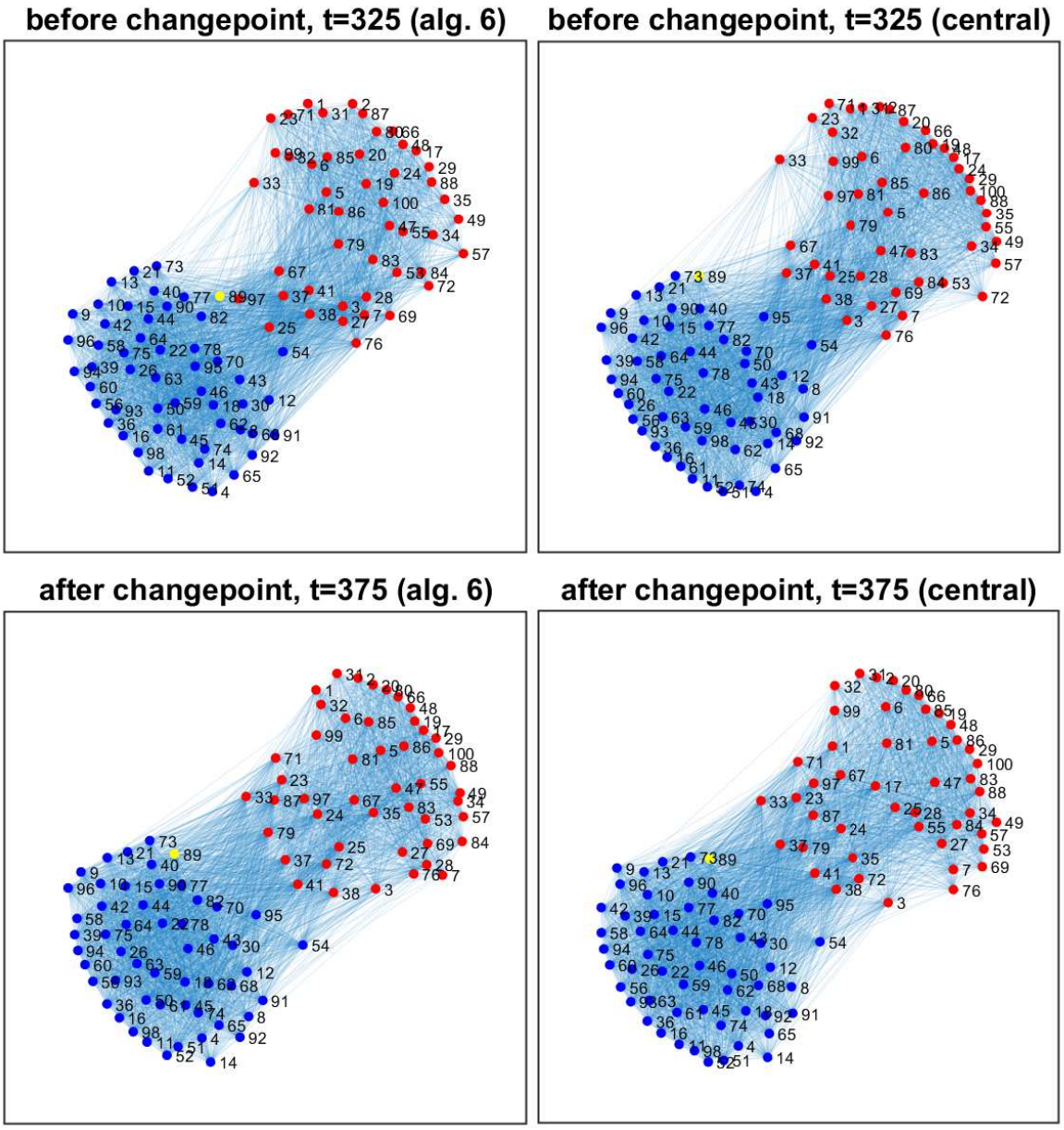


Figure 4.6: Top: Estimated adjacency graph from the first session near the detected changepoint; Bottom: Estimated adjacency graph from the second session near the detected changepoint; Left: Our method; Right: Central estimator

Independent senator Sanders (89) is seen towards the interior of the network, indicating the existence of many edges with many seats with all manner of ideologies. We see that

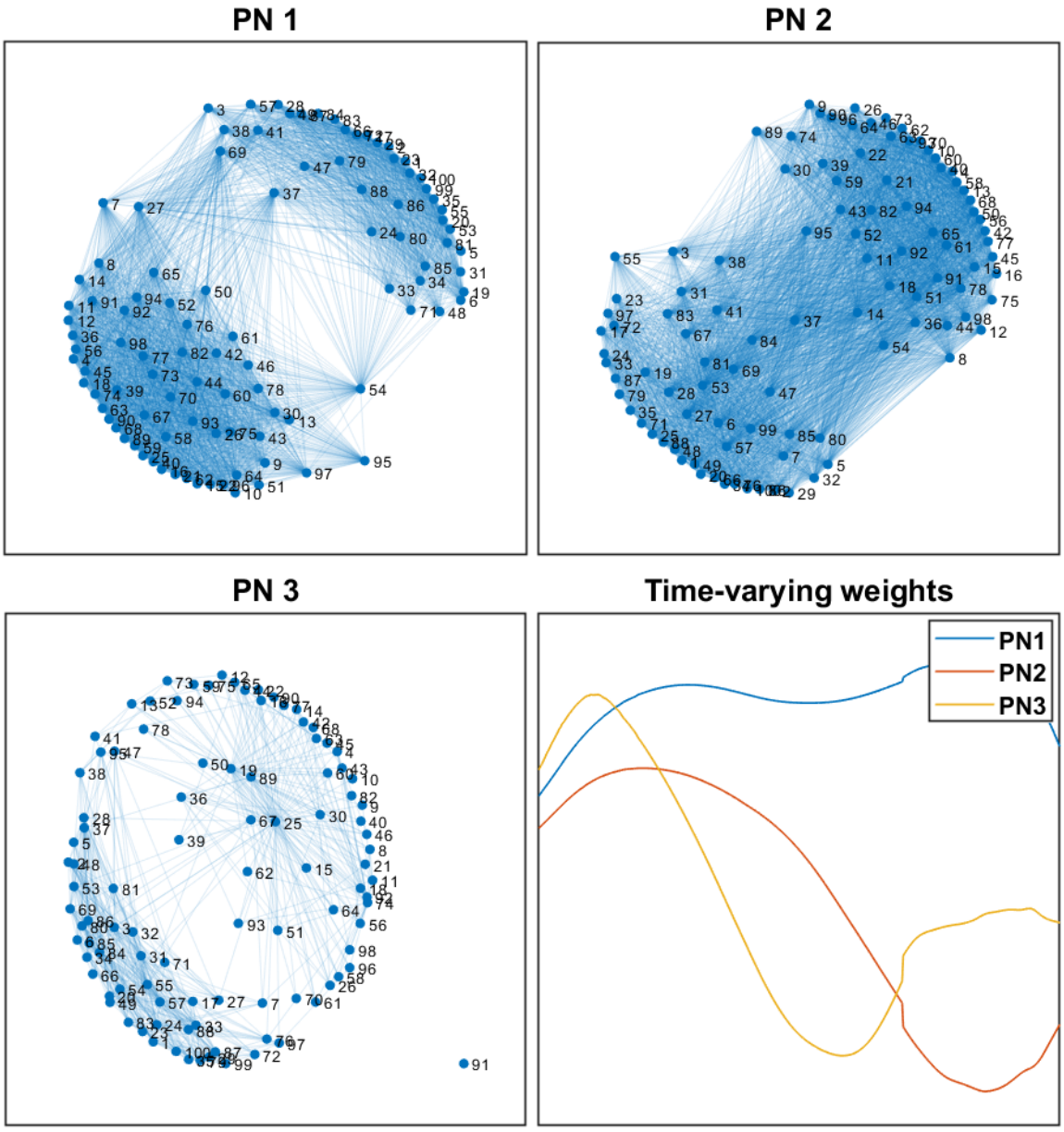


Figure 4.7: Principal Networks of the senator voting record; Bottom-right: relative time-varying weights for each network

seat 25 is also towards the center of this network. Interestingly, this is Democratic senator Burris from Illinois, who was specially appointed by the governor as a result of then-senator Obama's resignation after his election to the presidency in 2008, who retired prior to the

beginning of the new session to be replaced by Republican senator Kirk. This change in ideologies occurring before the single detected changepoint was clearly not large enough in magnitude to result in another changepoint, but rather it led to the seat seeming to share ideologies across both parties and a strong presence in PN 3 as a seat with many large enough edges changing smoothly.

4.5 Conclusion

We have presented a framework for learning the fundamental graphs that underlie a system described by smooth graphs with potential changepoints. This framework includes a time-varying graph estimation method that is easily parallelized, as a middle ground to existing methods for estimating time-varying graphs. The framework additionally uses a provably convergent matrix factorization to find the fundamental graphs. We demonstrate the use of this framework on simulated data as well as real US senate voting records for identifying interesting entities within the network and the times at which salient events occurred.

Algorithm 6 TV Graphs using proximal gradient descent

Require: Threshold factor $\gamma > 0$, regularization parameter $\lambda > 0$, max. iterations $t_{\max}$.

1: Initialize $(\mathcal{A}^{(i)}, \mathcal{A}'^{(i)}) = (\mathbf{0}, \mathbf{0})$ or randomly, set iteration $t = 0$, initial step size s_0.

2: **while** not converged and $t < t_{\max}$ **do** (in parallel over k)

3: Compute gradients using equation (4.23):

$$\mathbf{E}_k \leftarrow \frac{\partial F^{(i)}}{\partial \mathbf{A}_k} \qquad \mathbf{H}_k \leftarrow \frac{\partial F^{(i)}}{\partial \mathbf{A'}_k}. \tag{4.27}$$

4: Compute proximal step

$$\begin{aligned} \mathbf{A}_k &\leftarrow \text{prox}_1(\mathbf{A}_k - s_t \mathbf{E}_k, s_t \lambda) \\ \mathbf{A}'_k &\leftarrow \mathbf{A}'_k - s_t \mathbf{H}_k. \end{aligned} \tag{4.28}$$

5: $t \leftarrow t + 1$

6: **end while**

7: **for** $k \in \{M+1, \ldots, K\}$ **do** (in parallel)

8: Compute empirical errors and select indices

$$\begin{aligned} \widetilde{\epsilon}_k^{(i)} &= \sum_{j=1}^{K} w_{kj}^{(i)} D\left(g\left(\boldsymbol{\theta}_j\left(\widehat{\mathbf{A}}_k^{(i)} + (j-k)\widehat{\mathbf{A}}_k^{\prime(i)}\right)\right) \,\middle\|\, \mathbf{y}_j\right) \\ I_k &= \underset{i \in \{\ell, c, r\}}{\text{argmin}}\, \gamma_i \widetilde{\epsilon}_k^{(i)} \\ \widehat{\mathbf{A}}_k &= \widehat{\mathbf{A}}_k^{(I_k)}, \end{aligned} \tag{4.29}$$

9: **end for**

10: Compute changepoints

$$\widehat{\mathcal{J}} = \left\{k : (I_k = \ell) \bigcap (I_{k+1} = r)\right\}. \tag{4.30}$$

11: (Optional for smoothness) Compute central estimate near boundaries

12: **return** $\widehat{\mathcal{A}}, \widehat{\mathcal{J}}$

Algorithm 7 PNA using iPALM with two coordinate blocks

1: Let $t = -1$. Set $\epsilon = \infty$, and tolerance $\delta > 0$. Overloading superscripts to denote the iteration, initialize $\mathbf{C}^{-1} = \mathbf{C}^0 = rand(K - M \times R)$ and $\mathcal{B}^{-1} = \mathcal{B}^0 = rand(MN^2 \times R)$ or with R-rank SVD.

2: **while** $\epsilon \geq \delta$ and $t < t_{\max}$ **do**

3: $t \leftarrow t + 1$

4: Set inertial coefficient $\zeta = \frac{t}{t+3}$

5: Update $\mathcal{B}$ with fixed $\mathbf{C}$:

Use inertia: $\mathcal{Y} = \mathcal{B}^t + \zeta(\mathcal{B}^t - \mathcal{B}^{t-1})$

Compute gradient and Lipschitz constant for $\mathcal{B}$ from (4.38) and (4.41):

$$\mathcal{G}_b = \partial_{\mathcal{B}} F(\mathcal{Y}) \qquad \overline{L}_b = \overline{L}_{\mathcal{B}}(\mathbf{C}^t) \tag{4.43}$$

Apply Proximal operator from (4.42):

$$\mathcal{B}^{t+1} = prox_1(\mathcal{Y} - \mathcal{G}_b/\overline{L}_b, \lambda_{1,b}/\overline{L}_b)$$

6: Update $\mathbf{C}$ with fixed $\mathcal{B}$:

Use inertia: $\mathbf{Z} = \mathbf{C}^t + \zeta(\mathbf{C}^t - \mathbf{C}^{t-1})$

Compute gradient and Lipschitz constant for $\mathbf{c}$ from (4.38) and (4.40):

$$\mathbf{G}_C = \partial_{\mathbf{C}} F(\mathbf{z}) \qquad \overline{L}_c = \overline{L}_{\mathbf{C}}(\mathcal{B}^{t+1}) \tag{4.44}$$

Apply Proximal operator from (4.42):

$$\mathbf{C}^{t+1} = prox_s(\mathbf{Z} - \mathbf{G}_c/\overline{L}_c, \lambda_{s,c}/\overline{L}_c) = \mathbf{Z} - \mathbf{G}_c/\overline{L}_c$$

7: Compute error $\epsilon \leftarrow \left\| \begin{pmatrix} \mathcal{B}^{t+1} \\ \mathbf{C}^{t+1} \end{pmatrix} - \begin{pmatrix} \mathcal{B}^t \\ \mathbf{C}^t \end{pmatrix} \right\|_F \Big/ \left\| \begin{pmatrix} \mathcal{B}^t \\ \mathbf{C}^t \end{pmatrix} \right\|_F$

8: **end while**

9: **return** $(\mathcal{B}^t, \mathbf{C}^t)$

Chapter 5

Conclusion

This thesis addressed several challenges associated with learning meaningful networks that describe unstructured time series data. Techniques for estimating graphs from data should produce an interpretable result even in the presence of noise. In addition, we are often not able to directly measure or control for every possible variable that might have an influence on our observed data. Finally, the graph itself may change through time as the process evolves on top of it. We provided a thorough collection of problem formulations, accompanying algorithms, and theoretical analysis to tackle these various issues. Our approach involved utilizing sparse and low-rank optimization methods to produce interpretable results in presence of noise and latent variables; non-parametric regression to describe nonlinearities; and kernel estimation and matrix factorization to capture non-stationary behavior of the networks in time.

In chapter 2, we started with applications to Graph Signal Processing in a linear setting under Gaussian noise. We detailed computationally friendly algorithms for learning these networks. We evaluated the performance of the algorithms theoretically and empirically. We analyzed estimation error and conducted experiments on simulated data and real US temperature sensor network data.

In chapter 3, we built on this by then considering non-linear processes and addressing the presence of unobserved variables. We established bounds on the modeling error incurred by the effects of the nonlinearity on the influence of the hidden variables. Then we outlined the tractable algorithm for learning the graph, the effects of hidden variables, and the non-linearity. We provide conditions under which the estimation of the graph structure performs well. Then we wrapped up the chapter by using the algorithm to estimate networks on simulated data, to revisit the US temperature dataset, and to find insights about a bicycle sharing network in our beloved Pittsburgh, Pennsylvania.

Finally, we arrived at Principal Network Analysis (PNA) in chapter 4. We extended the ideas taken from these stationary methods to non-stationary settings to estimate time-varying graphs in chapter 4. We propose a scheme for simultaneously detecting change-points in the graphs while allowing the intermediate intervals to be smooth. We also describe a method for decomposing the time series of graphs into a collection of "principal networks" that may be somehow foundational to the system while we observe it and are present with differing weights throughout the course of the process. We demonstrate the method on US Senate voting records and find that the results from performing PNA identified a salient election and transition, highlighted important senate seats, and provided a picture for the overall dynamics of the country's contemporary political climate consistent with some prevailing theories.

The major contributions of this thesis are:

1. Providing a comprehensive optimization-based approach for estimating graphs from time series data;
2. Effectively handling both directed and undirected networks by framing the problem generically such that we may apply structural constraints to the graph adjacency matrix to be learned;

3. Inferring the effects of trends and latent variables on the observed data and separating out its presence from that of the desired graph;
4. Applying the developed methods to learn time-varying networks, with a particular focus on the case in which we suspect the networks vary as a combination of a set of "Principal Networks" that are fundamental to the system.

One direction for future analysis would be to relax the strict sparsity assumption for Causal Graph Process estimation. Instead, one could consider matrices that are approximately sparse, with weights that fall off in magnitude.

Another idea would be to characterize how graph topology affects the assumptions made in establishing the performance of the CGP estimator of chapter 2. This could be either statistically as random ensembles or via summary statistics on an instantiated graph. Though we briefly explored this question empirically, it would be enlightening to have a quantitative description of the behavior we observed.

A next step in the analysis for SILVar presented in chapter 3 would be to provide further conditions on the data generation that could guarantee estimation performance. This could include showing that the assumptions are satisfied either with high probability or exactly by: considering noise from a certain family of random variables; enforcing a deterministic bound on the range of the data; or imposing some limits on the abilities or knowledge of an attacker in an adversarial setting.

One natural extension for Principal Network Analysis as described in chapter 4 would be to combine the matrix factorization with the time-varying graph estimation, perhaps in an interative fashion. This extension would correspond to one possible point in the tradeoff space between a fully-coupled solution and the embarrassingly parallel solution.

An additional interesting area of exploration would be providing performance guarantees on the time-varying estimation. This includes results for accuracy in learning both the

graph and the locations of the changepoints. It also includes convergence results for the combined formulation described in the previous paragraph, since the full iterative problem does not automatically inherit convergent behavior from its subproblems.

A promising future application for this work is in analysis of neuroscience data. In fact, it is a collection of binary neural probe array spiking time series that inspired the directions for chapters 3 and 4 (specifically the nonlinearities in the binary data and the time-varying nature of the tasks in the experiments from which the data was generated). Learning the time-varying functional connectivity between the neurons observed by probe arrays could inform neuroscientists of how the information flows during certain tasks and yield insights into how the brain regions operate and interact. Some preliminary experiments of our methods on such data gave results consistent with basic sanity checks. However, a thorough analysis still requires both more care in the application of the method and domain expertise guiding the interpretation of the results.

Appendix A

Proofs for Chapter 2 on Causal Graph Processes

A.1 Proof of Lemma 2

Lemma 2. *Assume **(A1)** that $\mathbf{x}_k$ is generated according to the CGP model with $\mathbf{A}$ satisfying **(A5)**. Suppose **(A7)** that the sample size K is large enough. Let $d_i > 0$ with $i = 1, ..., 3$ be universal constants, and let $g(Q) = d_3\left(1 + \frac{1+Q^2}{(2-Q)^2}\right)$ and $\ell_{MNK} = \sqrt{(\log M + \log N)/K}$. With $\varepsilon_{\mathbf{A}} = d_1 \exp\{-d_2 K/\omega^2\}$, $\lambda_1 = 4g(Q)\ell_{MNK}$, and $\widehat{\mathbf{R}} = (\widehat{\mathbf{R}}_1, \ldots, \widehat{\mathbf{R}}_M)$ the solution to (2.15), with probability at least $1 - \varepsilon_{\mathbf{A}}$, $\widehat{\mathbf{A}} = \widehat{\mathbf{R}}_1$ satisfies the inequalities*

$$\|\widehat{\mathbf{A}} - \mathbf{A}\|_1 \leq 256 S_{MN} \ell_{MNK} Q^2 g(Q) \sigma_u / \sigma_\ell$$

$$\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \|\widehat{\mathbf{A}} - \mathbf{A}\|_F \leq \delta_{\mathbf{A}} \stackrel{\Delta}{=} 64\sqrt{S_{MN}} \ell_{MNK} Q^2 g(Q) \sigma_u / \sigma_\ell$$

*where $\omega = \frac{\sigma_u}{\sigma_\ell} \frac{Q^2}{\mu_{\min}(\widetilde{\mathcal{A}})}$ as defined in **(A7)**.*

First we define several quantities that correspond to the "stability" of the system and are used throughout the proof. Let $\mathcal{A}(z) = \mathbf{I} - \sum_{i=1}^{M} P_i(\mathbf{A}, \mathbf{c}) z^i$ and $\widetilde{\mathcal{A}}(z) = \mathbf{I} - z\widetilde{\mathbf{A}}$ where $\widetilde{\mathbf{A}}$

is the system matrix for the stacked state in companion form. That is,

$$\widetilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & P_2(\mathbf{A},\mathbf{c}) & \dots & \mathbf{P}_{M-1}(\mathbf{A},\mathbf{c}) & P_M(\mathbf{A},\mathbf{c}) \\ \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} \end{pmatrix}, \tag{A.1}$$

$\widetilde{\mathbf{x}}_k = \widetilde{\mathbf{A}}\widetilde{\mathbf{x}}_{k-1} + \widetilde{\mathbf{w}}_k$ where $\widetilde{\mathbf{x}}_k = (\mathbf{x}_k^\top \ \dots \ \mathbf{x}_{k-M}^\top)^\top$ and $\widetilde{\mathbf{w}}_k = (\mathbf{w}_k^\top \ \mathbf{0}^\top \ \dots \ \mathbf{0}^\top)^\top$. Then define

$$\mu_{\max}(\mathcal{A}) = \max_{|z|=1} \|\mathcal{A}(z)\|_2$$

$$\mu_{\min}(\mathcal{A}) = \min_{|z|=1} \|(\mathcal{A}(z))^{-1}\|_2^{-1}.$$

Now we proceed with the proof.

Proof of Lemma 2. To use Propositions 4.1-4.3 from [10], we bound the quantities $\mu_{\min}(\mathcal{A})$ and $\mu_{\max}(\mathcal{A})$. Letting $\mathcal{B}(z) = \mathbf{I} - \mathcal{A}(z)$,

$$\begin{aligned} \sqrt{\mu_{\max}(\mathcal{A})} &\leq 1 + \max_{|z|=1} |z| \, \|\mathbf{A}\|_2 + \sum_{i=2}^{M} \max_{|z|=1} |z^i| \, \|P_i(\mathbf{A},\mathbf{c})\|_2 \\ &\leq 1 + L + \sum_{i=2}^{M} \sum_{j=0}^{i} |c_{ij}| \, \|\mathbf{A}^j\|_2 \\ &\leq 1 + L + L \sum_{i=2}^{M} \sum_{j=1}^{i} |c_{ij}| + \sum_{i=2}^{M} |c_{i0}| \\ &\leq (1 + L + L\rho + \rho) = Q \end{aligned} \tag{A.2}$$

by assumption **(A5)**, and similarly

$$\begin{aligned} \sqrt{\mu_{\min}(\mathcal{A})} &= \min_{|z|=1} \left\| (\mathbf{I} - \mathcal{B}(z))^{-1} \right\|_2^{-1} \\ &\overset{(a)}{\geq} 1 - \max_{|z|=1} \|(\mathcal{B}(z))\|_2 \geq 2 - Q, \end{aligned}$$

where the inequality marked (a) is due to assumption **(A5)** and the rest follows from similar logic as (A.2). Propositions 4.2-4.3 hold with high probability when $T \geq d_0 S_{MN} \widetilde{\omega}^2 (\log M +$

$\log N$), where

$$\omega \geq \widetilde{\omega} = \frac{\|\mathbf{\Sigma_w}\|}{\left\|\mathbf{\Sigma_w^{-1}}\right\|^{-1}} \frac{\mu_{\min}(\mathcal{A})}{\mu_{\min}(\widetilde{\mathcal{A}})}. \tag{A.3}$$

Thus, we take $\omega = \widetilde{\omega}$, and when assumption **(A7)** holds, this condition is satisfied as well. Finally, we substitute these upper and lower bounds for $\mu_{\min}(\mathcal{A})$ and $\mu_{\max}(\mathcal{A})$ into the statements of Proposition 4.1 to yield the statement of Lemma 2. □

A.2 Proof of Lemma 3

Lemma 3. *Suppose that assumptions **(A1)-(A8)** hold. Then for any $0 < \beta < \nu < 1/2$, and $s_M \leq d_4 K / \log n$ and $\lambda_2 \geq q_1$, the solution to (2.12) satisfies*

$$P\left(\|\mathbf{c} - \widehat{\mathbf{c}}\|_1 \geq \delta_{\mathbf{c}}\right) \leq \varepsilon_{\mathbf{c}}$$

where

$$\delta_{\mathbf{c}} = 64 s_M \lambda_2 / \alpha_1$$

and

$$\begin{aligned}\varepsilon_{\mathbf{c}} =& \varepsilon_{Re} + \varepsilon_{\mathbf{A}} + \exp\{-d_4 K^{1-2\beta}\} \\ &+ \varepsilon_{De} + (6M+1)\exp\{-d_4 T^{1-2\beta}\},\end{aligned}$$

where ε_{Re} and ε_{De} are defined in Propositions 3.C and 3.D found in subsections C-D, respectively, $\varepsilon_{\mathbf{A}}$ is defined in Lemma 2, and α_1 and q_1 are given as

$$\begin{aligned}\alpha_1 &= \kappa_{\mathbf{B}} NT/(2K^{\nu}) + \delta_{Re} K(tr(\mathbf{\Sigma}_0) + G\sqrt{N}) \\ q_1 &= 2LT^{1-\beta}\sqrt{N t_N}\pi\sigma_u g(Q) + \delta_{\mathbf{A}} \widehat{L}_M(\delta_{\mathbf{A}})\sqrt{N} u,\end{aligned}$$

where

$$\begin{aligned}u =& 2T^{1-\beta}\sqrt{t_N}\pi\sigma_u g(Q) \\ &+ (L + \delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}}))M\rho K^{1-\beta}(tr(\mathbf{\Sigma}_0) + G\sqrt{N})\end{aligned}$$

and $\delta_{Re} = \delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}}) n \left(2(1+L) + \delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})\right)$.

We see that the interpretation of this lemma is similar to that of Lemma 2, so that

taking $\lambda_2 = q_1$ is sufficient to achieve the performance in the lemma. We know that $\delta_{\mathbf{A}} = \Theta(\sqrt{\log N/K})$. Then with this choice of λ_2, we have $\alpha_1 = \Theta(NK^{1-\nu} + G\sqrt{N}K\delta_{Re}) = \Theta(NK + G\sqrt{N}K\delta_{\mathbf{A}}) = \Theta(NK^{1-\nu} + \sqrt{(N\log N)K}) = \Theta(NK^{1-\nu})$ and $u = \Theta(\sqrt{N}K^{1-\beta})$ (from the second term) so that $q_1 = \Theta(\sqrt{N}K^{1-\beta} + \sqrt{(N\log N)/K}u) = \Theta(N\sqrt{\log N}K^{1-\beta})$; thus $\delta_{\mathbf{c}} = O(q_1/\alpha_1) = O\left(\sqrt{\log N/K^{2(\nu-\beta)}}\right)$. Also, note that

$$\begin{aligned}
\varepsilon_{\mathbf{c}} \sim & 2\exp\{-O(K^{1-2\nu})\} + d_1\exp\{-O(K)\} \\
& + (12M+2)\exp\{-O(T^{1-2\beta})\} \\
\sim & 2\exp\{-O(K^{1-2\nu})\} + (12M+2)\exp\{-O(T^{1-2\beta})\} \\
\sim & 2\exp\{-O(K^{1-2\nu})\},
\end{aligned}$$

since $K^{1-2\nu}$ is the slowest growing exponent.

To prove this lemma, we need two intermediate results showing that 1) a restricted eigenvalue ($\mathrm{Re}(\alpha,\tau)$) condition holds with high probability; and 2) a deviation ($\mathrm{De}(q)$) condition holds with high probability. These two conditions, which will be described shortly, describe the geometric properties of the objective function in the sparse optimization problem. Together, these conditions imply the desired result. These conditions are commonly encountered in sparse estimation, and the proof follows closely [10].

A.2.1 Eigenvalue and Deviation Conditions

We begin with the statements of the two conditions. Let $n = (M-1)(M+4)/2$, the length of $\mathbf{c}$.

The $\mathrm{Re}(\alpha,\tau)$ condition is satisfied for $\Gamma \in \mathbb{R}^{n\times n}$ if for all vectors $\theta \in \mathbb{R}^n$

$$\theta^\top\Gamma\theta \geq \alpha\|\theta\|_2^2 - \tau\|\theta\|_1^2. \tag{A.4}$$

In sparse estimation, this condition is used to show curvature of the objective function is large enough near the true value of the parameter in sparse directions [89]. In other

words, large perturbations in the objective function correspond to small perturbations in the parameter estimate.

The De(q) condition is satisfied for $(\Gamma, \gamma) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$ if for the true value of the parameter $\mathbf{c}$,

$$\|\gamma - \Gamma\mathbf{c}\|_\infty \leq q. \tag{A.5}$$

In sparse estimation, this condition intuitively states that the gradients of objective function are small near the true parameter value [89]. In other words, the estimated parameter value that minimizes the objective function is near the true parameter value.

A.2.2 Discussion

For our problem, we wish to show that Re(α_1, τ) and De(q_1) are satisfied for $(\widehat{\Gamma}, \widehat{\gamma}) = \left(\mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\widehat{\mathbf{A}}), \mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{Y}(\widehat{\mathbf{A}})\right)$ with high probability. If these two conditions hold for these matrices, then the estimate $\widehat{\mathbf{c}}$ will satisfy $\|\widehat{\mathbf{c}} - \mathbf{c}\|_1 \leq 64 s_M \lambda_2/\alpha_1$ for $\lambda_2 \geq q_1$.

Usually, in performance analysis of sparse estimation, these two restricted eigenvalue and deviation conditions would be shown using the true data (i.e., in our problem, that Re(α_0, τ) and De(q_0) are satisfied for $(\Gamma, \gamma) = \left(\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A}), \mathbf{B}(\mathbf{A})^\top \mathbf{Y}(\mathbf{A})\right)$). Matrices $\mathbf{B}(\mathbf{A})$ and $\mathbf{Y}(\mathbf{A})$ have entries that can be described by a multivariate Gaussian variable, so showing the two conditions on (Γ, γ) is relatively straightforward. However in our two-stage estimation, we can only use the estimated parameter $\widehat{\mathbf{A}}$ instead of the true parameter $\mathbf{A}$. Thus, the challenge in showing that these two conditions are satisfied comes from only being able to use the matrices $\mathbf{B}(\widehat{\mathbf{A}})$ and $\mathbf{Y}(\widehat{\mathbf{A}})$ that are complicated random functions of the true parameters and data, and are no longer described by a multivariate Gaussian variable.

Our approach is to show that if the two conditions are satisfied on the matrices $\mathbf{B}(\mathbf{A})$ and $\mathbf{Y}(\mathbf{A})$, then the two conditions will also be satisfied on the matrices $\mathbf{B}(\widehat{\mathbf{A}})$ and $\mathbf{Y}(\widehat{\mathbf{A}})$

that are actually available at the second stage. Thus, we rely on two additional intermediate results, which we prove first, showing that these two conditions are indeed satisfied on the matrices $\mathbf{B}(\mathbf{A})$ and $\mathbf{Y}(\mathbf{A})$ with high probability. We additionally point out that although the problem of estimating $\mathbf{c}$ is overdetermined, it may still be ill-conditioned due to the possibly highly correlated form of the data matrices $\mathbf{B}(\mathbf{A})$ and $\mathbf{Y}(\mathbf{A})$. Since it is not immediately obvious that the performance would be good, it is important to still demonstrate that these conditions hold.

To summarize the above discussion, for our problem we first show that $\mathrm{Re}(\alpha_0, \tau)$ and $\mathrm{De}(q_0)$ are satisfied for $(\Gamma, \gamma) = \big(\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A}), \mathbf{B}(\mathbf{A})^\top \mathbf{Y}(\mathbf{A})\big)$, and then show that this implies that $\mathrm{Re}(\alpha_1, \tau)$ and $\mathrm{De}(q_1)$ are satisfied for $(\widehat{\Gamma}, \widehat{\gamma}) = \Big(\mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\widehat{\mathbf{A}}), \mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{Y}(\widehat{\mathbf{A}})\Big)$. We show $\mathrm{Re}(\alpha_1,\tau)$ in Proposition 3.C and $\mathrm{De}(q_1)$ in Proposition 3.D.

A.2.3 Proof of Supplementary Proposition 1

Proposition 3.C. *Suppose that assumptions **(A1)-(A8)** hold. Then for any $\theta \in \mathbb{R}^n$ and any $0 < \nu < 1/2$,*

$$P\left(\|\mathbf{B}(\mathbf{A})\theta\|_2^2 \leq \alpha_0 \|\theta\|_2^2 - \tau \|\theta\|_1^2\right) \leq \varepsilon_{Re}$$

where $\tau = d_5 \alpha_0 (1+L)^4 n^2 G^2 K^\nu / (2(\log n)\kappa_{\mathbf{B}}^2 N T^2)$, $\alpha_0 = \kappa_{\mathbf{B}} NT/2$, and $\varepsilon_{Re} = 2\exp\{-d_4 \kappa_{\mathbf{B}}^2 T/((1+L)^4 n^2 G^2 K^{2\nu})\}$.

That is, $Re(\alpha_0, \tau)$ holds for estimating $\mathbf{c}$ using $\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})$ with probability at least $1 - \varepsilon_{Re}$.

Proof. This proof follows loosely the proof of Proposition 4.2 in [10], first bounding the quadratic form of $\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})$ using the Hanson-Wright concentration inequality followed by a discretization argument.

Let $\mathcal{D}(\mathbf{A}, \theta) = \Big(\mathbf{D}_0(\mathbf{A},\theta)^\top \;\; \mathbf{D}_1(\mathbf{A},\theta)^\top \;\; \ldots \;\; \mathbf{D}_{K-M-1}(\mathbf{A},\theta)^\top \Big)^\top$ where the block rows of $\mathcal{D}(\mathbf{A}, \theta)$ are

$\mathbf{D}_i(\mathbf{A},\theta) = \begin{pmatrix} \mathbf{0}_{N\times Ni} & P_2(\mathbf{A},\theta) & \dots & P_M(\mathbf{A},\theta) & \mathbf{0}_{N\times N(K-M-2-i)} \end{pmatrix}$. Then consider any vector $\|\theta\| \leq 1$,

$$\begin{aligned} \|\mathbf{B}(\mathbf{A})\theta\|_2^2 &= \sum_{k=M+1}^{K} \left\| \sum_{i=2}^{M} P_i(\mathbf{A},\theta)\mathbf{x}_{k-i} \right\|_2^2 \\ &= \|\mathcal{D}(\mathbf{A},\theta)\mathbf{x}_{1:K-2}\|_2^2, \end{aligned} \tag{A.6}$$

where $\mathbf{x}_{1:K-2} = \begin{pmatrix} \mathbf{x}_1^\top & \dots & \mathbf{x}_{K-2}^\top \end{pmatrix}^\top$. By an extension of Gershgorin's Circle Theorem to block matrices [90],

$$\begin{aligned} \|\mathcal{D}(\mathbf{A},\theta)\|_2 &\leq \sum_{i=2}^{M} \|P_i(\mathbf{A},\theta)\|_2 \leq \sum_{i=2}^{M} \left\| \sum_{j=0}^{i} \theta_{ij}\mathbf{A}^j \right\|_2 \\ &\leq \sum_{i=2}^{M}\sum_{j=0}^{i} |\theta_{ij}| \|\mathbf{A}^j\|_2 \leq (1+L) \sum_{i=2}^{M}\sum_{j=0}^{i} |\theta_{ij}| \\ &= (1+L)\|\theta\|_1 \leq (1+L)\sqrt{n}\|\theta\|_2 \\ &\leq (1+L)\sqrt{n}. \end{aligned}$$

Now,

$$\begin{aligned} \|\mathbf{B}(\mathbf{A})\theta\|_2^2 - \mathbb{E}\left[\|\mathbf{B}(\mathbf{A})\theta\|_2^2\right] &= \|\mathcal{D}(\mathbf{A},\theta)\mathbf{x}_{1:K-2}\|_2^2 - \mathbb{E}\left[\|\mathcal{D}(\mathbf{A},\theta)\mathbf{x}_{1:K-2}\|_2^2\right] \\ &\overset{(a)}{\leq} \|\mathcal{D}(\mathbf{A},\theta)\|_2^2 \|\mathbf{\Sigma}_0\| NK\eta \leq (1+L)^2 nGNK\eta \end{aligned} \tag{A.7}$$

with probability at least $1 - 2\exp\{-d_4 NK \min(\eta,\eta^2)\}$ for some universal constant d_4, where (a) follows from the Hanson-Wright inequality [91]. Then from the discretization argument of Lemma F.2 [10], for an integer $s \geq 1$ (to be specified later) and set $\mathcal{K}_{2s} = \{\theta \in \mathbb{R}^n : \|\theta\| \leq 1, \|\theta\|_0 \leq 2s\}$,

$$\begin{aligned} &P\left(\sup_{\theta\in\mathcal{K}_{2s}} \|\mathbf{B}(\mathbf{A})\theta\|_2^2 - \mathbb{E}\left[\|\mathbf{B}(\mathbf{A})\theta\|_2^2\right] \geq (1+L)^2 nNKG\eta \right) \\ &\leq 2\exp\{-d_4 NK\min(\eta,\eta^2) + 2s\min(\log n, \log(21en/2s))\}. \end{aligned}$$

Finally, from Lemmas 12 and 13 in [89] and taking $s = \lceil d_4 NK\eta^2/(4\log n)\rceil$ and $\eta = \kappa_{\mathbf{B}} NT/[54(1+L)^2 nGNK^\nu] \leq 1$ with $0 < \nu < 1/2$ so that $\eta^2 \leq \eta$ where $\kappa_{\mathbf{B}} NT$ is the minimum singular value of $\mathbb{E}[\mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})]$ as defined in **(A8)**,

we have $\mathbf{B}(\mathbf{A})^\top\mathbf{B}(\mathbf{A})$ satisfies $\text{Re}(\kappa_{\mathbf{B}}NT/(2K^\nu), \kappa_{\mathbf{B}}NT/(2sK^\nu))$ with probability at least $1 - 2\exp\{-d_4\kappa_{\mathbf{B}}^2 T/((1+L)^4 n^2 G^2 K^{2\nu})\}$. □

A.2.4 Proof of Supplementary Proposition 2

Proposition 3.D. *Suppose that assumptions **(A1)-(A8)** hold. Then for any* $0 < \beta < 1/2$,

$$P(\|\mathbf{B}(\mathbf{A})^\top\mathbf{e}\|_\infty \geq q_0) \leq \varepsilon_{De},$$

where $q_0 = 2LT^{1-\beta}\sqrt{Nt_N}\pi\sigma_u g(Q)$ *and* $\varepsilon_{De} = 6M\exp\{-d_4 T^{1-2\beta}\}$.

That is, De(q_0) *holds for estimating* $\mathbf{c}$ *using* $\mathbf{Y}(\mathbf{A})\mathbf{B}(\mathbf{A})$ *with probability at least* $1 - \varepsilon_{De}$.

Proof. The proof is a straightforward application of Proposition 2.4 from [10] and the union bound (as in the proof of Proposition 4.3). Letting $\mathbf{e} = \mathbf{Y}(\mathbf{A}) - \mathbf{B}(\mathbf{A})\mathbf{c}$ and $T = K - M$,

$$\begin{aligned}
\left\|\mathbf{B}(\mathbf{A})^\top\mathbf{e}\right\|_\infty &= \max_{\substack{1\leq i\leq M \\ 1\leq j\leq i}} \left|\left(\sum_{k=M+1}^{K} \mathbf{x}_{k-i}^\top (\mathbf{A}^j)^\top \mathbf{w}_k\right)\right| \\
&\leq \max_{\substack{1\leq i\leq M \\ 1\leq j\leq i}} \left|\text{tr}\left((\mathbf{A}^j)^\top \mathbf{w}_{M+1:K}\mathbf{x}_{M-i+1:K-i}^\top\right)\right| \\
&\leq \max_{\substack{1\leq i\leq M \\ 1\leq j\leq i}} T\|\mathbf{A}^j\|_1 \left\|\mathbf{w}_{M+1:K}\mathbf{x}_{M-i+1:K-i}^\top/T\right\|_\infty \\
&\overset{(a)}{\leq} T\max_{1\leq j\leq M} \sqrt{t_N}\|\mathbf{A}^j\|_F \left(\pi\sigma_u\left(1 + \frac{1+\mu_{max}(\mathcal{A})}{\mu_{min}(\mathcal{A})}\right)\eta\right) \\
&\overset{(b)}{\leq} 2\sqrt{t_N}T\max_{1\leq j\leq M} \sqrt{N}\|\mathbf{A}^j\|_2 \left(\pi\sigma_u g(Q)\eta\right) \\
&\leq 2LT\sqrt{Nt_N}\pi\sigma_u g(Q)\eta
\end{aligned}$$

with probability at least $1 - 6M\exp\{-d_4 T\ \min(\eta, \eta^2)\}$, where t_N is the maximum sparsity of $\mathbf{A}^j$ as defined in assumption **(A5)**, (a) is implied by Proposition 2.4 from [10] and (b) is implied by the analysis in Lemma 2. To finish the proof, we take $\eta = T^{-\beta}$ for any $0 < \beta < 1/2$, noting that for this choice, $\eta^2 < \eta$. □

A.2.5 Proof of Lemma 3

Armed with these two results, we resume with our lemma.

Proof. From Lemma 2, we have that $P(\|\widehat{\mathbf{A}}-\mathbf{A}\|_2 \geq \delta_{\mathbf{A}}) \leq \varepsilon_{\mathbf{A}}$. First, we consider the Re condition. Let $\Delta_\alpha = \alpha_1 - \alpha_0$ and $\widetilde{\mathcal{B}} = \mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})$. Then,

$$
\begin{aligned}
&P\left(\|\mathbf{B}(\widehat{\mathbf{A}})\theta\|_2^2 \leq \alpha_1\|\theta\|_2^2 - \tau\|\theta\|_1^2\right) \\
&= P\left(\|\mathbf{B}(\widehat{\mathbf{A}})\theta\|^2 \leq (\alpha_0+\Delta_\alpha)\|\theta\|^2 - \tau\|\theta\|_1^2 \bigcap \|\widetilde{\mathcal{B}}\|_2 < \Delta_\alpha\right) \\
&\quad + P\left(\|\mathbf{B}(\widehat{\mathbf{A}})\theta\|_2^2 \leq \alpha_1\|\theta\|_2^2 - \tau\|\theta\|_1^2 \bigcap \|\widetilde{\mathcal{B}}\|_2 \geq \Delta_\alpha\right) \\
&\leq P\left(\|\mathbf{B}(\widehat{\mathbf{A}})\theta\|_2^2 \leq \alpha_0\|\theta\|_2^2 + \theta^\top\widetilde{\mathcal{B}}\theta - \tau\|\theta\|_1^2 \bigcap \|\widetilde{\mathcal{B}}\|_2 < \Delta_\alpha\right) + P\left(\|\widetilde{\mathcal{B}}\|_2 \geq \Delta_\alpha\right) \\
&\leq P\left(\|\mathbf{B}(\mathbf{A})\theta\|_2^2 \leq \alpha_0\|\theta\|_2^2 - \tau\|\theta\|_1^2\right) + P\left(\|\widetilde{\mathcal{B}}\|_2 \geq \Delta \bigcap \|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}\right) + P\left(\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \geq \delta_{\mathbf{A}}\right) \\
&\leq \varepsilon_{Re} + \varepsilon_{\mathbf{A}} + P\left(\|\widetilde{\mathcal{B}}\|_2 \geq \Delta_\alpha \bigcap \|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}\right).
\end{aligned}
$$

We bound the last probability by manipulating the first term,

$$
\begin{aligned}
\|\widetilde{\mathcal{B}}\|_2 &= \|\mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})\|_2 \\
&\leq \|\mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\mathbf{A})\|_2 + \|\mathbf{B}(\widehat{\mathbf{A}})^\top \mathbf{B}(\mathbf{A}) - \mathbf{B}(\mathbf{A})^\top \mathbf{B}(\mathbf{A})\|_2 \qquad \text{(A.8)} \\
&\leq \|\mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\mathbf{A})\|_2 \left(2\|\mathbf{B}(\mathbf{A})\|_2 + \|\mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\mathbf{A})\|_2\right).
\end{aligned}
$$

Under the constraint $\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}$, for any vector $\|\theta\|_2 \leq 1$ following from the same logic as (A.7),

$$
\begin{aligned}
\|(\mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\mathbf{A}))\theta\|_2^2 &= \|(\mathcal{D}(\widehat{\mathbf{A}}, \theta) - \mathcal{D}(\mathbf{A}, \theta))\mathbf{x}[0\!:\!K-3]\|_2^2 \\
&\leq \left(\delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})\right)^2 n\|\mathbf{X}\|_F^2
\end{aligned}
$$

and

$$
\|\mathbf{B}(\mathbf{A})\|_2^2 \leq (1+L)^2 n\|\mathbf{X}\|_F^2. \qquad \text{(A.9)}
$$

Thus, letting

$$
\delta_{Re} \triangleq \delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})n\left(2(1+L) + \delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})\right),
$$

we have

$$
\|\widetilde{\mathcal{B}}\|_2 \leq \delta_{Re}\|\mathbf{X}\|_F^2 \Longrightarrow P\left(\|\widetilde{\mathcal{B}}\|_2 \geq \Delta_\alpha \bigcap \|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}\right) \leq P(\|\mathbf{X}\|_F^2 \geq \Delta_\alpha/\delta_{Re}) \leq \varepsilon_{\mathcal{B}}, \qquad \text{(A.10)}
$$

where by the Hanson-Wright inequality [91], taking $\Delta_\alpha = \delta_{Re} K^{1-\beta}(\mathrm{tr}(\mathbf{\Sigma}_0) + G\sqrt{N})$, we have $\varepsilon_{\mathcal{B}} = \exp\{-d_4 K^{1-2\beta}\}$. Finally,

$$P\left(\|\mathbf{B}(\widehat{\mathbf{A}})\theta\|_2^2 \leq \alpha_1\|\theta\|_2^2 - \tau\|\theta\|_1^2\right) \leq \varepsilon_{Re} + \varepsilon_{\mathbf{A}} + \varepsilon_{\mathcal{B}}. \tag{A.11}$$

Second, letting $\mathbf{e} = \mathbf{Y}(\mathbf{A}) - \mathbf{B}(\mathbf{A})\mathbf{c}$ and $\widehat{\mathbf{e}} = \mathbf{Y}(\widehat{\mathbf{A}}) - \mathbf{B}(\widehat{\mathbf{A}})\mathbf{c}$ and $\Delta_q = q_1 - q_0$, we consider the De condition.

$$\begin{aligned} P\left(\left\|\mathbf{B}(\widehat{\mathbf{A}})^\top\widehat{\mathbf{e}}\right\|_\infty \geq q_1\right) &\leq P\left(\left\|\mathbf{B}(\mathbf{A})^\top\mathbf{e}\right\|_\infty \geq q_0\right) + P\left(\left\|\mathbf{B}(\widehat{\mathbf{A}})^\top\widehat{\mathbf{e}} - \mathbf{B}(\mathbf{A})^\top\mathbf{e}\right\|_\infty \geq \Delta_q\right) \\ &\leq \varepsilon_{De} + P\left(\left\|(\mathbf{B}(\widehat{\mathbf{A}})^\top - \mathbf{B}(\mathbf{A})^\top)\mathbf{e}\right\|_\infty \geq \Delta_{q_1}\right) + P\left(\left\|\mathbf{B}(\widehat{\mathbf{A}})^\top(\widehat{\mathbf{e}} - \mathbf{e})\right\|_\infty \geq \Delta_{q_2}\right), \end{aligned}$$

where $\Delta_q = \Delta_{q_1} + \Delta_{q_2}$. We examine the term

$$\begin{aligned} \left\|\left(\mathbf{B}(\widehat{\mathbf{A}}) - \mathbf{B}(\mathbf{A})\right)^\top\mathbf{e}\right\|_\infty &= \max_{\substack{1\leq i\leq M \\ 1\leq j\leq i}} \left|\left(\sum_{k=i}^{T-1+i} \mathbf{x}_{k-i}^\top(\widehat{\mathbf{A}}^j - \mathbf{A}^j)^\top\mathbf{w}_k\right)\right| \\ &= \max_{\substack{1\leq i\leq M \\ 1\leq j\leq i}} \left|\mathrm{tr}\left((\widehat{\mathbf{A}}^j - \mathbf{A}^j)^\top\mathbf{w}_{M+1:K}\mathbf{x}_{M-i+1:K-i}^\top\right)\right| \\ &\leq \delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})\sqrt{Nt_N}\max_{1\leq i\leq M}\left\|\mathbf{w}_{M+1:K}\mathbf{x}_{M-i+1:K-i}^\top\right\|_\infty \\ &\overset{(a)}{\leq} 2\delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})T^{1-\beta}\sqrt{Nt_N}\pi\sigma_u g(Q) \end{aligned} \tag{A.12}$$

with probability at least $1 - 6M\exp\{-d_4 T^{1-2\beta}\}$, where t_N is defined in **(A6)**, and in (a) we again invoke Proposition 2.4 from [10] similarly as in Proposition 3.D.

To finish, we see that

$$\widehat{\mathbf{w}}_k - \mathbf{w}_k = \sum_{m=1}^{M}\sum_{\ell=1}^{m} c_{m\ell}(\widehat{\mathbf{A}}^\ell - \mathbf{A}^\ell)\mathbf{x}_{k-m}$$

and that $\|\widehat{\mathbf{A}}^j\|_2 \leq \|\mathbf{A}^j + (\widehat{\mathbf{A}}^j - \mathbf{A}^j)\|_2 \leq L + \delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}})$.

These imply,

$$\begin{aligned}\left\|\mathbf{B}(\widehat{\mathbf{A}})^\top(\widehat{\mathbf{e}}-\mathbf{e})\right\|_\infty \leq & \max_{\substack{1\leq i\leq M\\ 1\leq j\leq i}}\left|\left(\sum_{k=i}^{T-1+i}\mathbf{x}_{k-i}^\top(\widehat{\mathbf{A}}^j)^\top(\widehat{\mathbf{w}}_k-\mathbf{w}_k)\right)\right| \\ \overset{(a)}{\leq} & (L+\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}}))\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}})\sqrt{N}\times\max_{1\leq k\leq M}\sum_{m=1}^{M}\left\|\left(\sum_{k=i}^{T-1+i}\|\mathbf{c}_{m\cdot}\|\mathbf{x}_{k-i}^\top\mathbf{x}_{k-m}\right)\right\|_F \\ \leq & (L+\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}}))\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}})\sqrt{N}M\rho\|\mathbf{X}\|_F^2 \\ \overset{(b)}{\leq} & (L+\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}}))\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}})\sqrt{N}M\rho K^{1-\beta}(\mathrm{tr}(\mathbf{\Sigma}_0)+G\sqrt{N})\end{aligned}$$

with probability at least $1-\exp\{-d_4K^{1-2\beta}\}$ where (a) follows from similar logic as (A.12) and (b) from similar logic to (A.10).

Finally, we arrive at

$$\Delta_q=\delta_{\mathbf{A}}\widehat{L}_M(\delta_{\mathbf{A}})\sqrt{N}u,$$

where

$u=2T\sqrt{t_N}\pi\sigma_u g(Q)+(L+\delta_{\mathbf{A}}\widehat{L}(\delta_{\mathbf{A}}))M\rho K(\mathrm{tr}(\mathbf{\Sigma}_0)+G\sqrt{N})$ and $P\left(\left\|\mathbf{B}(\widehat{\mathbf{A}})^\top\widehat{\mathbf{e}}\right\|_\infty\geq q_1\right)\leq\varepsilon_{De}+(6M+1)\exp\{-d_4T\}$ since $T<K$. □

A.3 Proof of Theorem 1

Finally, with the two lemmas in hand, we return to the proof of the main theorem, which we restate for convenience.

Theorem 1. *For any $0<\beta<\nu<1/2$, and some universal constant d_1, assumptions **(A1)-(A8)** are sufficient for the error ϵ in (2.16) to satisfy*

$$\epsilon\leq\left(\delta_{\mathbf{A}}\left(1+(\rho+\delta_{\mathbf{c}})\widehat{L}_M(\delta_{\mathbf{A}})\right)+(1+L)\delta_{\mathbf{c}}\right)^2 tr(\mathbf{\Sigma}_0)/N \tag{A.13}$$

with probability at least $1-\varepsilon_{\mathbf{A}}-\varepsilon_{\mathbf{c}}$, where

$$\widehat{L}_M(\delta)=\max_{1\leq i\leq M}\frac{(L+\delta)^i-L^i}{\delta}$$

$$
\begin{aligned}
\varepsilon_{\mathbf{A}} &\sim d_1 \exp\{-O(K)\} \\
\varepsilon_{\mathbf{c}} &\sim 2 \exp\{-O(K^{1-2\nu})\} \\
\delta_{\mathbf{A}} &= O\left(\sqrt{\log N/K}\right) \\
\delta_{\mathbf{c}} &= O\left(\sqrt{\log N/K^{2(\nu-\beta)}}\right).
\end{aligned}
\tag{A.14}
$$

Proof of Theorem 1. First, applying the union bound to results from Lemmas 2 and 3, we see that

$$
P\left(\left(\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}\right) \bigcap \left(\|\widehat{\mathbf{c}} - \mathbf{c}\|_1 \leq \delta_{\mathbf{c}}\right)\right) \geq 1 - \varepsilon_{\mathbf{A}} - \varepsilon_{\mathbf{c}}.
$$

Thus, we proceed using $\|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \leq \delta_{\mathbf{A}}$ and $\|\widehat{\mathbf{c}} - \mathbf{c}\|_1 \leq \delta_{\mathbf{c}}$. Then, $\|\widehat{\mathbf{A}}\|_2 \leq \|\mathbf{A} + (\widehat{\mathbf{A}} - \mathbf{A})\|_2 \leq L + \delta_{\mathbf{A}}$. Similarly, $\|\widehat{\mathbf{c}}\|_1 \leq \rho + \delta_{\mathbf{c}}$. Then,

$$
\begin{aligned}
\max_{1 \leq i \leq M} \|\widehat{\mathbf{A}}^i - \mathbf{A}^i\|_2 &\leq \|\widehat{\mathbf{A}} - \mathbf{A}\|_2 \max_{0 \leq i \leq M-1} \left\| \sum_{j=0}^{i} \widehat{\mathbf{A}}^j \mathbf{A}^{i-j} \right\|_2 \\
&\leq \delta_{\mathbf{A}} \max_{0 \leq i \leq M-1} \sum_{j=0}^{i} \left\| \widehat{\mathbf{A}} \right\|_2^j \|\mathbf{A}\|_2^{i-j} \\
&\leq \delta_{\mathbf{A}} \max_{0 \leq i \leq M-1} \sum_{j=0}^{i} (L + \delta_{\mathbf{A}})^j L^{i-j} \\
&\leq \delta_{\mathbf{A}} \widehat{L}_M(\delta_{\mathbf{A}}).
\end{aligned}
\tag{A.15}
$$

Note that $\widehat{L}_M(\delta) = O(ML^{M-1})$ when $\delta \to 0$.

Next, dropping from the list of arguments for the function $f(\mathbf{A}, \mathbf{c}, \mathbf{X}'_{k-1})$ defined below

equation (2.16) the explicit dependence on $\mathbf{X}'_{k-1}$ for compactness of notation,

$$\begin{aligned}
\epsilon &= \mathbb{E}\Big[\|\mathbf{x}_k - f(\widehat{\mathbf{A}},\widehat{\mathbf{c}})\|_2^2\Big] - \mathbb{E}\big[\|\mathbf{x}_k - f(\mathbf{A},\mathbf{c})\|_2^2\big] \\
&= \mathbb{E}\left[\Big(f(\mathbf{A},\mathbf{c}) - f(\widehat{\mathbf{A}},\widehat{\mathbf{c}})\Big)^\top \Big(2\mathbf{x}_k - f(\mathbf{A},\mathbf{c}) - f(\widehat{\mathbf{A}},\widehat{\mathbf{c}})\Big)\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\Big(f(\mathbf{A},\mathbf{c}) - f(\widehat{\mathbf{A}},\widehat{\mathbf{c}})\Big)^\top \Big(2\mathbf{w}_k + f(\mathbf{A},\mathbf{c}) - f(\widehat{\mathbf{A}},\widehat{\mathbf{c}})\Big)\right] \\
&= \mathbb{E}\left[\|f(\widehat{\mathbf{A}},\widehat{\mathbf{c}}) - f(\mathbf{A},\mathbf{c})\|_2^2\right] \\
&= \mathbb{E}\left[\|f(\widehat{\mathbf{A}},\widehat{\mathbf{c}}) - f(\mathbf{A},\widehat{\mathbf{c}}) + f(\mathbf{A},\widehat{\mathbf{c}}) - f(\mathbf{A},\mathbf{c})\|_2^2\right] \\
&\leq \mathbb{E}\left[\Big(\|f(\widehat{\mathbf{A}},\widehat{\mathbf{c}}) - f(\mathbf{A},\widehat{\mathbf{c}})\|_2 + \|f(\mathbf{A},\widehat{\mathbf{c}}) - f(\mathbf{A},\mathbf{c})\|_2\Big)^2\right] \\
&\overset{(b)}{\leq} \Big(\|\mathbf{D}(\widehat{\mathbf{A}},\widehat{\mathbf{c}}) - \mathbf{D}(\mathbf{A},\widehat{\mathbf{c}})\|_2 + \|\mathbf{D}(\mathbf{A},\widehat{\mathbf{c}}) - \mathbf{D}(\mathbf{A},\mathbf{c})\|_2\Big)^2 \mathbb{E}\big[\|\mathbf{X}'_{k-1}\|_2^2\big] \\
&= \Big(\underbrace{\|\mathbf{D}(\widehat{\mathbf{A}},\widehat{\mathbf{c}}) - \mathbf{D}(\mathbf{A},\widehat{\mathbf{c}})\|_2}_{V_1} + \underbrace{\|\mathbf{D}(\mathbf{A},\widehat{\mathbf{c}} - \mathbf{c})\|_2}_{V_2}\Big)^2 M \mathrm{tr}(\mathbf{\Sigma}_0),
\end{aligned}$$

where $\mathbf{D}(\mathbf{A}',\mathbf{c}') = \begin{pmatrix} \mathbf{A}' & P_1(\mathbf{A}',\mathbf{c}') & \dots & P_M(\mathbf{A}',\mathbf{c}') \end{pmatrix}$, and where (a) is due to $\mathbf{x}_k - f(\mathbf{A},\mathbf{c}) = \mathbf{w}_k \perp \mathbf{x}_j \ \forall j < k$, and (b) is due to $\mathbf{D}(\mathbf{A}',\mathbf{c}')\mathbf{X}'_{k-1} = f(\mathbf{A}',\mathbf{c}')$. Now consider the term V_1,

$$\begin{aligned}
V_1 &\leq \sum_{i=1}^{M} \|P_i(\widehat{\mathbf{A}},\widehat{\mathbf{c}}) - P_i(\mathbf{A},\widehat{\mathbf{c}})\|_2 \leq \delta_{\mathbf{A}} + \sum_{i=2}^{M}\sum_{j=1}^{i} |\widehat{c}_{ij}| \|\widehat{\mathbf{A}}^j - \mathbf{A}^j\|_2 \\
&\leq \delta_{\mathbf{A}} + \delta_{\mathbf{A}} \widehat{L}_M(\delta_{\mathbf{A}}) \|\widehat{\mathbf{c}}\|_1 \leq \delta_{\mathbf{A}} \left(1 + (\rho + \delta_{\mathbf{c}}) \widehat{L}_M(\delta_{\mathbf{A}})\right).
\end{aligned}$$

Next we similarly bound V_2,

$$\begin{aligned}
V_2 &\leq \sum_{i=1}^{M} \|P_i(\mathbf{A},\widehat{\mathbf{c}} - \mathbf{c})\|_2 \leq \sum_{i=2}^{M}\sum_{j=0}^{i} |\widehat{c}_{ij} - c_{ij}| \|\mathbf{A}^j\|_2 \\
&\leq \sum_{i=2}^{M} |\widehat{c}_{i0} - c_{i0}| + \sum_{i=2}^{M}\sum_{j=1}^{i} |\widehat{c}_{ij} - c_{ij}| \|\mathbf{A}^j\|_2 \\
&\leq \|\widehat{\mathbf{c}} - \mathbf{c}\|_1 + L\|\widehat{\mathbf{c}} - \mathbf{c}\|_1 \overset{(a)}{\leq} (1 + L)\delta_{\mathbf{c}}.
\end{aligned}$$

where (a) follows from $\|\widehat{\mathbf{c}} - \mathbf{c}\|_1 \leq \delta_{\mathbf{c}}$ proven in lemma 3, and the result follows. □

Appendix B

Proofs for Chapter 3 on Single Index Latent Variable Models

B.1 Proof of Theorem 4

We first restate the theorem for convenience.

Theorem 4. *Assume that* $\widehat{g}'(0) \neq 0$ *and that* $|\widehat{g}''| \leq J$ *and* $|\overline{g}''| \leq J$ *for some* $J < \infty$. *Furthermore, assume in models* (3.15) *and* (3.16) *that* $\max_j \left(\|\widehat{\mathbf{a}}_j\|_1, \|\overline{\mathbf{a}}_j\|_1 + \|\overline{\mathbf{b}}_j\|_2 \right) \leq k$,

$$\max \left(\mathbb{E}[\|\mathbf{x}\|_2 \|\mathbf{x}\|_\infty^2], \mathbb{E}[\|\mathbf{x}\|_2 \|\mathbf{x}\|_\infty \|\mathbf{z}\|_2], \mathbb{E}[\|\mathbf{x}\|_2 \|\mathbf{z}\|_2^2] \right) \leq s_{Nr},$$

where subscripts in the parameters on the RHS indicate that the bounds may grow with the values in the subscript. Then, the parameters $\widehat{\mathbf{A}}$ *and* $\overline{\mathbf{A}}$ *from models* (3.15) *and* (3.16) *are related as*

$$\widehat{\mathbf{A}} = q(\overline{\mathbf{A}} + \mathbf{L}) + \mathbf{E},$$

where $q = \frac{\overline{g}'(0)}{\widehat{g}'(0)}$, $\boldsymbol{\mu}_{\mathbf{x}} = \mathbb{E}[\mathbf{x}_i]$,

$$\mathbf{L} = \left(\overline{\mathbf{B}} \mathbb{E}[\mathbf{z}\mathbf{x}^\top] + \left(\overline{\mathbf{g}}(\mathbf{0}) - \widehat{\mathbf{g}}(\mathbf{0}) \right) \boldsymbol{\mu}_{\mathbf{x}}^\top \right) \left(\mathbb{E}[\mathbf{x}\mathbf{x}^\top] \right)^\dagger \Longrightarrow \mathit{rank}(\mathbf{L}) \leq r + 1,$$

and $\mathbf{E} = \widehat{\mathbf{A}} - q(\overline{\mathbf{A}} + \mathbf{L})$ *is bounded as*

$$\frac{1}{MN}\|\mathbf{E}\|_F \leq \frac{2J\sigma_\ell\sqrt{N}}{\widehat{g}'(0)M} s_{Nr} k^2,$$

where $\sigma_\ell = \left\| \left(\mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]\right)^\dagger \right\|_2$*, the largest singular value of the pseudo-inverse of the covariance.*

Proof. We begin with the Taylor series expansion, using the Lagrange form of the remainder,

$$\mathbb{E}\left[\left(\widehat{\mathbf{g}}(\mathbf{0}) + \widehat{g}'(0)\widehat{\mathbf{A}}\mathbf{x} - \overline{\mathbf{g}}(\mathbf{0}) - \overline{g}'(0)(\overline{\mathbf{A}}\mathbf{x} + \overline{\mathbf{B}}\mathbf{z})\right)\mathbf{x}^\top\right] = \mathbb{E}\left[\left(\widehat{\mathbf{g}}''(\boldsymbol{\xi}) \odot (\widehat{\mathbf{A}}\mathbf{x}_i).\hat{}2 - \overline{\mathbf{g}}''(\boldsymbol{\eta}) \odot (\overline{\mathbf{A}}\mathbf{x}_i + \overline{\mathbf{B}}\mathbf{z}).\hat{}2\right)\mathbf{x}^\top\right], \tag{B.1}$$

where

$$|\xi_j| \leq |\widehat{\mathbf{a}}_j\mathbf{x}|, \qquad |\eta_j| \leq |\overline{\mathbf{a}}_j\mathbf{x} + \overline{\mathbf{b}}_j\mathbf{z}|,$$

$\overline{\mathbf{B}} = \left(\overline{\mathbf{b}}_1 \dots \overline{\mathbf{b}}_m\right)^\top$ similarly to before, $\mathbf{x} \odot \mathbf{y}$ denotes the element-wise (Hadamard) product, and $\mathbf{x}.\hat{}2 = \mathbf{x} \odot \mathbf{x}$ denotes element-wise squaring. First let us consider the quantity

$$\begin{aligned}(\overline{\mathbf{a}}_j\mathbf{x} + \overline{\mathbf{b}}_j\mathbf{z})^2 &= (\overline{\mathbf{a}}_j\mathbf{x})^2 + 2\overline{\mathbf{a}}_j\mathbf{x}\overline{\mathbf{b}}_j\mathbf{z} + (\overline{\mathbf{b}}_j\mathbf{z})^2 \\ &\leq (\|\overline{\mathbf{a}}_j\|_1\|\mathbf{x}\|_\infty + \|\overline{\mathbf{b}}_j\|_2\|\mathbf{z}\|_2)^2\end{aligned}$$

Then,

$$\begin{aligned}\|\left((\overline{\mathbf{A}}\mathbf{x} + \overline{\mathbf{B}}\mathbf{z}).\hat{}2\right)\mathbf{x}^\top\|_F &\leq \|(\overline{\mathbf{A}}\mathbf{x} + \overline{\mathbf{B}}\mathbf{z}).\hat{}2\|_2\|\mathbf{x}\|_2 \\ &\leq \|\mathbf{x}\|_2\sqrt{\sum_{j=1}^N (\overline{\mathbf{a}}_j\mathbf{x} + \overline{\mathbf{b}}_j\mathbf{z})^4} \\ &\overset{(a)}{\leq} \|\mathbf{x}\|_2 \sum_{j=1}^N (\overline{\mathbf{a}}_j\mathbf{x} + \overline{\mathbf{b}}_j\mathbf{z})^2 \\ &\leq \|\mathbf{x}\|_2 \sum_{j=1}^N (\|\overline{\mathbf{a}}_j\|_1\|\mathbf{x}\|_\infty + \|\overline{\mathbf{b}}_j\|_2\|\mathbf{z}\|_2)^2 \\ \implies \mathbb{E}[\|\left((\overline{\mathbf{A}}\mathbf{x} + \overline{\mathbf{B}}\mathbf{z}).\hat{}2\right)\mathbf{x}^\top\|_F] &\leq s_{Nr} \sum_{j=1}^N (\|\overline{\mathbf{a}}_j\|_1 + \|\overline{\mathbf{b}}_j\|_2)^2 \\ &\leq s_{Nr} N k^2\end{aligned}$$

where (a) follows from the fact that $\|\mathbf{x}\|_4 \leq \|\mathbf{x}\|_2 \; \forall \mathbf{x} \in \mathbb{R}^N$.

Similarly, we have

$$\begin{aligned}\implies \mathbb{E}[\|((\widehat{\mathbf{A}}\mathbf{x})^{\wedge}2)\mathbf{x}^\top\|_F] &\leq \|\mathbf{x}\|_2 \sum_{j=1}^{N} (\|\widehat{\mathbf{a}}_j\|_1 \|\mathbf{x}\|_\infty)^2 \\ &\leq s_{Nr} N k^2\end{aligned}$$

To finish off, substituting the definition of $\mathbf{E}$ and q into (B.1) we have

$$\begin{aligned}\|\mathbf{E}\|_F \leq & J\mathbb{E}\left[\left\|\left((\widehat{\mathbf{A}}\mathbf{x}_i)^{\wedge}2 + (\overline{\mathbf{A}}\mathbf{x}_i + \overline{\mathbf{B}}\mathbf{z}_i)^{\wedge}2\right)\mathbf{x}_i^\top\right\|_F\right] \left\|\left(\widehat{g}'(0)\mathbb{E}[\mathbf{x}_i\mathbf{x}_i^\top]\right)^\dagger\right\|_F \\ &\leq \frac{2J\sigma_\ell}{\widehat{g}'(0)} s_{Nr} k^2 N\sqrt{N}.\end{aligned}$$

□

B.2 Proof of Theorem 6

First, we present and prove some intermediate results. For convenience of notation, for the remainder of this section let $\mathbf{\Phi} = \widehat{\mathbf{A}} - \widetilde{\mathbf{A}}$ and $\mathbf{\Psi} = \widehat{\mathbf{L}} - \widetilde{\mathbf{L}}$.

B.2.1 Propositions

Proposition 4.1. *Under Assumption 2, the solution* $(\widehat{\mathbf{A}}, \widehat{\mathbf{L}})$ *to the optimization problem* (3.23) *satisfies*

$$\begin{aligned}\gamma\|\mathbf{\Phi}_{S^c}\|_1 + \|\mathbf{\Psi}_{R^c}\|_* &\leq 3(\gamma\|\mathbf{\Phi}_S\|_1 + \|\mathbf{\Psi}_R\|_*) \\ \gamma\|\mathbf{\Phi}\|_1 + \|\mathbf{\Psi}\|_* &\leq 4\sqrt{r_{\mathbf{L}}}(\tau\|\mathbf{\Phi}\|_F + \|\mathbf{\Psi}\|_F).\end{aligned}$$

Proof. We start with the convexity of the marginalized objective functional at $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{L}})$,

$$m(\widehat{\mathbf{A}}+\widehat{\mathbf{L}}) \geq m(\widetilde{\mathbf{A}}+\widetilde{\mathbf{L}}) + \operatorname{tr}\left(\left[\frac{1}{K}\widehat{\widetilde{\mathbf{\Gamma}}}\mathbf{X}^\top\right]^\top (\widehat{\mathbf{A}}+\widehat{\mathbf{L}}-(\mathbf{A}+\mathbf{L}))\right)$$

Then consider the optimality of the full objective functional at $(\widehat{g}, \widehat{\mathbf{A}}, \widehat{\mathbf{L}})$,

$$m(\widehat{\mathbf{A}} + \widehat{\mathbf{L}}) + \lambda(\gamma\|\widehat{\mathbf{A}}\|_1 + \|\widehat{\mathbf{L}}\|_*)$$

$$
\begin{aligned}
&= \widehat{F}_3(\mathbf{Y}, \mathbf{X}, \widehat{g}, \widehat{\mathbf{A}} + \widehat{\mathbf{L}}) + (\gamma\|\widehat{\mathbf{A}}\|_1 + \|\widehat{\mathbf{L}}\|_*) \\
&\leq \widehat{F}_3(\mathbf{Y}, \mathbf{X}, \widehat{\widetilde{g}}, \widetilde{\mathbf{A}} + \widetilde{\mathbf{L}}) + (\gamma\|\widetilde{\mathbf{A}}\|_1 + \|\widetilde{\mathbf{L}}\|_*) \\
&= m(\widetilde{\mathbf{A}} + \widetilde{\mathbf{L}}) + \lambda(\gamma\|\widetilde{\mathbf{A}}\|_1 + \|\widetilde{\mathbf{L}}\|_*) \\
\Longrightarrow & m(\widehat{\mathbf{A}}+\widehat{\mathbf{L}}) - m(\widetilde{\mathbf{A}} + \widetilde{\mathbf{L}}) \leq \lambda(\gamma\|\widetilde{\mathbf{A}}\|_1 + \|\widetilde{\mathbf{L}}\|_*) - \lambda(\gamma\|\widehat{\mathbf{A}}\|_1 + \|\widehat{\mathbf{L}}\|_*) \\
\Longrightarrow & \operatorname{tr}\left(\left[\frac{1}{K}\widehat{\widetilde{\mathbf{\Gamma}}}\mathbf{X}^\top\right]^\top (\widehat{\mathbf{A}}+\widehat{\mathbf{L}} - (\widetilde{\mathbf{A}}+\widetilde{\mathbf{L}}))\right) \\
&\leq \lambda(\gamma(\|\widetilde{\mathbf{A}}\|_1 - \|\widehat{\mathbf{A}}\|_1) + \|\widetilde{\mathbf{L}}\|_* - \|\widehat{\mathbf{L}}\|_*)
\end{aligned}
$$

where the last inequality utilizes convexity of the marginalized objective. Then using Assumption 2,

$$
\begin{aligned}
& -\frac{\lambda\gamma}{2}\|\mathbf{\Phi}\|_1 - \frac{\lambda}{2}\|\mathbf{\Psi}\|_* \leq \lambda\gamma(\|\widetilde{\mathbf{A}}\|_1 - \|\widehat{\mathbf{A}}\|_1) + \lambda(\|\widetilde{\mathbf{L}}\|_* - \|\widehat{\mathbf{L}}\|_*) \\
\Longrightarrow & 0 \leq \frac{\gamma}{2}\|\mathbf{\Phi}\|_1 + \gamma(\|\widetilde{\mathbf{A}}\|_1 - \|\widehat{\mathbf{A}}\|_1) + \frac{1}{2}\|\mathbf{\Psi}\|_* + (\|\mathbf{L}\|_* - \|\widehat{\mathbf{L}}\|_*) \\
\Longrightarrow & 0 \leq \frac{\gamma}{2}(\|\mathbf{\Phi}_S\|_1 + \|\mathbf{\Phi}_{S^c}\|_1) + \gamma(\|\mathbf{\Phi}_S\|_1 - \|\mathbf{\Phi}_{S^c}\|_1) + \frac{1}{2}(\|\mathbf{\Psi}_R\|_* + \|\mathbf{\Psi}_{R^c}\|_*) + (\|\mathbf{\Psi}_R\|_* - \|\mathbf{\Psi}_{R^c}\|_*) \\
\Longrightarrow & 0 \leq -(\gamma\|\mathbf{\Phi}_{S^c}\|_1 + \|\mathbf{\Psi}_{R^c}\|_*) + 3(\gamma\|\mathbf{\Phi}_S\|_1 + \|\mathbf{\Psi}_R\|_*)
\end{aligned}
$$

where the penultimate inequality arises from decomposability of the norm. Specifically,

$$
\begin{cases}
\left|[\widetilde{\mathbf{A}}]_s\right| - \left|[\widehat{\mathbf{A}}]_s\right| \leq \left|[\widetilde{\mathbf{A}}]_s - [\widehat{\mathbf{A}}]_s\right| = \left|[\mathbf{\Phi}]_s\right|, & s \in S \\
\left|[\widetilde{\mathbf{A}}]_s\right| - \left|[\widehat{\mathbf{A}}]_s\right| = -\left|[\widehat{\mathbf{A}}]_s\right| = -\left|[\mathbf{\Phi}]_s\right|, & s \notin S
\end{cases}
$$

Thus, we have the first inequality

$$
\gamma\|\mathbf{\Phi}_{S^c}\|_1 + \|\mathbf{\Psi}_{R^c}\|_* \leq 3(\gamma\|\mathbf{\Phi}_S\|_1 + \|\mathbf{\Psi}_R\|_*)
$$

Finally, for the second inequality,

$$
\begin{aligned}
\gamma\|\mathbf{\Phi}\|_1 + \|\mathbf{\Psi}\|_* &\leq 4(\gamma\|\mathbf{\Phi}_S\|_1 + \|\mathbf{\Psi}_R\|_*) \\
&\leq 4(\gamma\sqrt{s_{\mathbf{A}}}\|\mathbf{\Phi}_S\|_F + \sqrt{r_{\mathbf{L}}}\|\mathbf{\Psi}_R\|_F) \\
&\leq 4\sqrt{r_{\mathbf{L}}}(\tau\|\mathbf{\Phi}_S\|_F + \|\mathbf{\Psi}_R\|_F)
\end{aligned}
$$

$$\leq 4\sqrt{r_{\mathbf{L}}}(\tau\|\mathbf{\Phi}\|_F+\|\mathbf{\Psi}\|_F)$$

□

Proposition 4.2. *Under Assumptions 2 and 3, the solution of* $(\widehat{\mathbf{A}},\widehat{\mathbf{L}})$ *to the optimization problem* (3.23) *satisfies*

$$2|tr(\mathbf{\Phi}^\top\mathbf{\Psi})| \leq \mu(\gamma\|\mathbf{\Phi}\|_1+\|\mathbf{\Psi}\|_*)^2.$$

Proof. This follows directly from Proposition 4.1, Assumption 3, and applying Cauchy-Schwarz twice

$$\begin{aligned}2|\mathrm{tr}(\mathbf{\Phi}^\top\mathbf{\Psi})| &\leq \|\mathbf{\Psi}\|_\infty\|\mathbf{\Phi}\|_1+\|\mathbf{\Psi}\|_*\|\mathbf{\Phi}\|_2\\ &\leq \mu\gamma(\gamma\|\mathbf{\Phi}\|_1+\|\mathbf{\Psi}\|_*)\|\mathbf{\Phi}\|_1+\mu(\gamma\|\mathbf{\Phi}\|_1+\|\mathbf{\Psi}\|_*)\|\mathbf{\Psi}\|_*.\end{aligned}$$

□

B.2.2 Theorem 6

Now we restate the theorem again for convenience.

Theorem 6. *Under Assumptions 1-3, the solution to the optimization problem* (3.23) *satisfies*

$$\|\widehat{\mathbf{A}}-\widetilde{\mathbf{A}}\|_F \leq \frac{3\lambda\gamma\sqrt{s_{\mathbf{A}}}}{\alpha}\left(2+\sqrt{\frac{2}{2+\tau^2}}\right) \leq \frac{9\lambda\gamma\sqrt{s_{\mathbf{A}}}}{\alpha}$$
$$\|\widehat{\mathbf{L}}-\widetilde{\mathbf{L}}\|_F \leq \frac{3\lambda\sqrt{r_{\mathbf{L}}}}{\alpha}\left(2+\sqrt{\frac{2\tau^2}{1+2\tau^2}}\right) \leq \frac{9\lambda\sqrt{r_{\mathbf{L}}}}{\alpha}.$$

Proof. Starting with Proposition 4.1, since our solution is in this restricted set, we can use the stronger convexity condition implied by Assumption 1,

$$m(\widehat{\mathbf{A}}+\widehat{\mathbf{L}})\geq m(\widetilde{\mathbf{A}}+\widetilde{\mathbf{L}})+\mathrm{tr}\left(\left[\frac{1}{K}\widehat{\widetilde{\mathbf{\Gamma}}}\mathbf{X}^\top\right]^\top\left(\widehat{\mathbf{A}}+\widehat{\mathbf{L}}-(\mathbf{A}+\mathbf{L})\right)\right)+\alpha\|\mathbf{\Phi}+\mathbf{\Psi}\|_F^2$$

Revisiting the objective functional at optimality and skipping repetitive algebra (see

proof for Proposition 4.1),

$$\begin{aligned}\alpha\|\boldsymbol{\Phi}+\boldsymbol{\Psi}\|_F^2 &\leq \frac{\lambda}{2}\left(3(\gamma\|\boldsymbol{\Phi}_S\|_1+\|\boldsymbol{\Psi}_R\|_*)-(\gamma\|\boldsymbol{\Phi}_{S^c}\|_1+\|\boldsymbol{\Psi}_{R^c}\|_*)\right)\\ &\leq \frac{3\lambda}{2}(\gamma\|\boldsymbol{\Phi}\|_1+\|\boldsymbol{\Psi}\|_*)\\ &\leq 6\lambda\sqrt{r_{\mathbf{L}}}(\tau\|\boldsymbol{\Phi}\|_F+\|\boldsymbol{\Psi}\|_F)\end{aligned}$$

Now from Proposition 4.2, we have

$$\begin{aligned}\|\boldsymbol{\Phi}+\boldsymbol{\Psi}\|_F^2 &\geq \|\boldsymbol{\Phi}\|_F+\|\boldsymbol{\Psi}\|_F^2 - 2|\mathrm{tr}(\boldsymbol{\Phi}^\top\boldsymbol{\Psi})|\\ &\geq \|\boldsymbol{\Phi}\|_F+\|\boldsymbol{\Psi}\|_F^2 - \mu(\gamma\|\boldsymbol{\Phi}\|_1+\|\boldsymbol{\Psi}\|_*)^2\end{aligned}$$

Combining the previous two inequalities, we have

$$\begin{aligned}&\alpha(\|\boldsymbol{\Phi}\|_F^2+\|\boldsymbol{\Psi}\|_F^2)-\alpha\mu(\gamma\|\boldsymbol{\Phi}\|_1+\|\boldsymbol{\Psi}\|_*)^2\leq 6\lambda\sqrt{r_{\mathbf{L}}}(\tau\|\boldsymbol{\Phi}\|_F+\|\boldsymbol{\Psi}\|_F)\\ \Longrightarrow&\alpha(\|\boldsymbol{\Phi}\|_F^2+\|\boldsymbol{\Psi}\|_F^2)-16\alpha\mu r_{\mathbf{L}}(\tau\|\boldsymbol{\Phi}\|_F+\|\boldsymbol{\Psi}\|_F)^2\\ &\leq 6\lambda\sqrt{r_{\mathbf{L}}}(\tau\|\boldsymbol{\Phi}\|_F+\|\boldsymbol{\Psi}\|_F)\\ \Longrightarrow&\boldsymbol{\zeta}^\top\mathbf{Q}\boldsymbol{\zeta}\leq 0\end{aligned}$$

where

$$\boldsymbol{\zeta} = \begin{pmatrix}\|\boldsymbol{\Phi}\|_F & \|\boldsymbol{\Psi}\|_F & 1\end{pmatrix}^\top$$

$$\mathbf{Q} = \begin{pmatrix} \alpha\left(\frac{2+\tau^2}{2(1+\tau^2)}\right) & -\alpha\left(\frac{\tau}{2(1+\tau^2)}\right) & -3\lambda'\tau \\ -\alpha\left(\frac{\tau}{2(1+\tau^2)}\right) & \alpha\left(\frac{1+2\tau^2}{2(1+\tau^2)}\right) & -3\lambda' \\ -3\lambda'\tau & -3\lambda' & 0 \end{pmatrix},$$

which is a conic in standard form, $\lambda' = \lambda\sqrt{r_{\mathbf{L}}}$, and the entries of $\mathbf{Q}$ follow from taking μ given in Assumption 3.

Checking its discriminant,

$$\left(2+\tau^2\right)\left(1+2\tau^2\right)-\tau^2 > 0$$

so we have that the conic equation describes an ellipse, and there are bounds on the values of $\|\boldsymbol{\Phi}\|_F$ and $\|\boldsymbol{\Psi}\|_F$.

For these individual bounds, we consider the points at which the gradients of $\boldsymbol{\zeta}^\top\mathbf{Q}\boldsymbol{\zeta}$

vanish w.r.t. each of $\|\mathbf{\Phi}\|_F$ and $\|\mathbf{\Psi}\|_F$. For $\|\mathbf{\Phi}\|_F$,

$$\begin{aligned}&\partial_{\|\mathbf{\Phi}\|_F}\boldsymbol{\zeta}^\top\mathbf{Q}\boldsymbol{\zeta}=0\\ \Longrightarrow&\alpha\left(\frac{2+\tau^2}{2(1+\tau^2)}\right)\|\mathbf{\Phi}\|_F^*=3\lambda'\tau+\alpha\left(\frac{\tau}{2(1+\tau^2)}\right)\|\mathbf{\Psi}\|_F^*.\end{aligned}$$

Plugging this into the equation defined by $\boldsymbol{\zeta}^\top\mathbf{Q}\boldsymbol{\zeta}=0$ yields

$$\|\mathbf{\Phi}\|_F^*=\frac{3\lambda'\tau}{\alpha}\left(2\pm\sqrt{\frac{2}{2+\tau^2}}\right).$$

Since we are seeking an upper bound for $\|\mathbf{\Phi}\|_F$, it can be seen that we take the positive root,

$$\|\mathbf{\Phi}\|_F\leq\frac{3\lambda'\tau}{\alpha}\left(2+\sqrt{\frac{2}{2+\tau^2}}\right)\leq\frac{9\lambda'\tau}{\alpha}$$

Similarly, for $\|\mathbf{\Psi}\|_F$,

$$\begin{aligned}&\partial_{\|\mathbf{\Psi}\|_F}\boldsymbol{\zeta}^\top\mathbf{Q}\boldsymbol{\zeta}=0\\ \Longrightarrow&\alpha\left(\frac{2+\tau^2}{2(1+\tau^2)}\right)\|\mathbf{\Psi}\|_F^*=3\lambda'\tau+\alpha\left(\frac{\tau}{2(1+\tau^2)}\right)\|\mathbf{\Phi}\|_F^*.\end{aligned}$$

Finally, plugging this into $\boldsymbol{\zeta}^\top\mathbf{Q}\boldsymbol{\zeta}=0$ and solving for the upper bound yields

$$\|\mathbf{\Psi}\|_F\leq\frac{3\lambda'}{\alpha}\left(2+\sqrt{\frac{2\tau^2}{1+2\tau^2}}\right)\leq\frac{9\lambda'}{\alpha}$$

□

Bibliography

[1] M. O. Jackson, *Social and Economic Networks*. Princeton University Press, Nov. 2010. 1.1

[2] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 467–475. [Online]. Available: http://dl.acm.org/citation.cfm?id=2073796.2073849 1.1

[3] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013. 1.1, 2, 2.5.1, 3, 3.2.2

[4] M. Newman, *Networks: An Introduction*. Oxford University Press, Mar. 2010. 1.1

[5] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000. [Online]. Available: http://www.sciencemag.org/content/290/5500/2323 1.1

[6] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000. [Online]. Available: http://www.sciencemag.org/content/290/5500/2319 1.1

[7] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the Lasso," *Ann. Stat.*, vol. 34, no. 3, pp. 1436–1462, Jun. 2006. [Online]. Available: http://projecteuclid.org/Dienst/getRecord?id=euclid.aos/1152540754/ 1.1, 2

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical Lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–41, Jul. 2008. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3019769&tool=pmcentrez&rendertype=abstract 1.1, 2.1.1, 2.1.2, 3

[9] A. Bolstad, B. Van Veen, and R. Nowak, "Causal network inference via group sparse regularization," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628–2641, Jun. 2011. 1.1, 2, 2.1.2, 2.5, 3, 3.2.2

[10] S. Basu and G. Michailidis, "Regularized estimation in sparse high-dimensional time series models," *The Annals of Statistics*, vol. 43, no. 4, pp. 1535–1567, Aug. 2015. [Online]. Available: http://projecteuclid.org/euclid.aos/1434546214 1.1, 2.4.1, **(A7)**, 3, A.1, A.2, A.2.3, A.2.3, A.2.4, A.2.5

[11] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, Aug. 1969. [Online]. Available: http://www.jstor.org/stable/1912791 1.1, 2, 3

[12] A. Jalali and S. Sanghavi, "Learning the dependence graph of time series with latent factors," *arXiv:1106.1887 [cs]*, Jun. 2011, arXiv: 1106.1887. [Online]. Available: http://arxiv.org/abs/1106.1887 1.1, 3

[13] L. Song, M. Kolar, and E. P. Xing, "Time-varying dynamic Bayesian networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1732–1740. [Online]. Available: http://papers.nips.cc/paper/3716-time-varying-dynamic-bayesian-networks.pdf 1.1, 4.1.1

[14] M. Kolar and E. P. Xing, "Estimating networks with jumps," *Electronic Journal of Statistics*, vol. 6, pp. 2069–2106, 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4085697/ 1.1, 4.1.1

[15] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence," *Electronic Journal of Statistics*, vol. 5, pp. 935–980, 2011. [Online]. Available: http://projecteuclid.org/euclid.ejs/1316092865 2, 2.1.1

[16] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014. 2, 2.1, 2.6, 3, 3.2.2

[17] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Latent variable graphical model selection via convex optimization," *The Annals of Statistics*, vol. 40, no. 4, pp. 1935–1967, Aug. 2012. [Online]. Available: http://projecteuclid.org/euclid.aos/1351602527 2.1.1, 3, 3.2.1, 3

[18] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2189–2199, Aug. 2004. 2.1.2

[19] J. Songsiri and L. Vandenberghe, "Topology selection in graphical models of autoregressive processes," *J. Mach. Learn. Res.*, vol. 11, pp. 2671–2705, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953020 2.1.2

[20] C. W. J. Granger, "Causality, cointegration, and control," *J. of Econ. Dynamics and Control*, vol. 12, no. 23, pp. 551–559, Jun. 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165188988900553 2.1.2, 3.2.2

[21] ——, "Some recent development in a concept of causality," *Journal of Econometrics*, vol. 39, no. 12, pp. 199–211, Sep. 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304407688900450 2.1.2

[22] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013. 2.1.3

[23] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, Mar. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520310000552 2.1.3

[24] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon.*

Anal., vol. 21, no. 1, pp. 5–30, Jul. 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1063520306000546 2.1.3

[25] D. Thanou, D. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3849–3862, Aug. 2014. 2.1.3

[26] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016. 2.1.3

[27] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*, May 2016, pp. 920–929. [Online]. Available: http://proceedings.mlr.press/v51/kalofolias16.html 2.1.3

[28] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 6350–6354. 2.1.3

[29] J. Mei and J. M. F. Moura, "Signal processing on graphs: Performance of graph structure estimation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 6165–6169. 2.2

[30] ——, "Fitting graph models to big data," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, Nov. 2015, pp. 387–390. 2.2

[31] ——, "Signal processing on graphs: Estimating the structure of a graph," in *2015*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2015, pp. 5495–5499. 2.2

[32] ——, "Signal processing on graphs: causal modeling of unstructured data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017. 2.2, 3, 3.2.2, 3.5.2, 3.5.2

[33] G. Goerg and C. Shalizi, "Mixed LICORS: A nonparametric algorithm for predictive state reconstruction," in *Artificial Intelligence and Statistics*, Apr. 2013, pp. 289–297. [Online]. Available: http://proceedings.mlr.press/v31/goerg13a.html 2.2

[34] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007. 2.3.1, 2.3.5

[35] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178 2.4.1, 3

[36] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim Theory Appl*, pp. 475–494, 2001. 2.4.1

[37] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *Machine Learning: ECML 2007*. Springer, Sep. 2007, pp. 286–297. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-74958-5_28 2.5

[38] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Foundations of Computational Mathematics*, vol. 15, no. 3, pp. 715–732, Jul. 2013. [Online]. Available: http://link.springer.com/article/10.1007/s10208-013-9150-3 2.5, 3.3.2

[39] "National Climactic Data Center," ftp://ftp.ncdc.noaa.gov/pub/data/gsod, 2011. [Online]. Available: ftp://ftp.ncdc.noaa.gov/pub/data/gsod 2.5.1

[40] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 83, no. 1 Pt 2, p. 016107, Jan. 2011. 2.5.2

[41] P. Erdös and A. Rényi, "On the Evolution of Random Graphs," in *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 1960, pp. 17–61. 2.5.2

[42] A. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999. [Online]. Available: http://www.sciencemag.org/content/286/5439/509 2.5.2

[43] D. Holten and J. J. Van Wijk, "Force-Directed Edge Bundling for Graph Visualization," *Computer Graphics Forum*, vol. 28, no. 3, pp. 983–990, Jun. 2009. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2009.01450.x/abstract 2.5.2

[44] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and Experience*, vol. 21, no. 11, pp. 1129–1164, Nov. 1991. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/spe.

4380211102/abstract 2.5.2, 4.4.2

[45] M. T. Bahadori, Y. Liu, and E. P. Xing, "Fast structure learning in generalized stochastic processes with latent factors," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 284–292. [Online]. Available: http://doi.acm.org/10.1145/2487575.2487578 3

[46] R. Ganti, N. Rao, L. Balzano, R. Willett, and R. Nowak, "On learning high dimensional structured single index models," in *Thirty-First AAAI Conference on Artificial Intelligence*, Feb. 2017. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14480 3, 3.1.2, 3.5.1

[47] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, Jan. 1967. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0041555367900407 3.1.1

[48] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, no. Oct, pp. 1705–1749, 2005. [Online]. Available: http://www.jmlr.org/papers/v6/banerjee05b.html 3.1.2

[49] S. Acharyya and J. Ghosh, "Parameter estimation of Generalized Linear Models without assuming their link function," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 10–18. [Online]. Available: http://www.jmlr.org/proceedings/papers/v38/acharyya15.html 3.1.2,

3.1.2, 3.1.3, 3.3, 5

[50] H. Ichimura, "Semiparametric Least Squares (SLS) and weighted SLS estimation of Single-Index Models," *Journal of Econometrics*, vol. 58, no. 1, pp. 71–120, Jul. 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/030440769390114K 3.1.2

[51] J. Mei and J. M. F. Moura, "SILVar: Single index latent variable models," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2790–2803, Jun. 2018. 3.2

[52] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1970392.1970395 3.2.2, 3

[53] R. S. Ganti, L. Balzano, and R. Willett, "Matrix completion under monotonic Single Index Models," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1873–1881. [Online]. Available: http://papers.nips.cc/paper/5916-matrix-completion-under-monotonic-single-index-models.pdf 3.2.2

[54] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized low rank models," *Foundations and Trends in Machine Learning*, vol. 9, no. 1, pp. 1–118, Jun. 2016. [Online]. Available: http://ftp.nowpublishers.com/article/Details/MAL-055 3.2.2

[55] R. L. Dykstra, "An isotonic regression algorithm," *Journal of Statistical Planning and Inference*, vol. 5, no. 4, pp. 355–363, Jan. 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378375881900367 3.3.1

[56] W. López and M. Raydan, "An acceleration scheme for Dykstra's algorithm," *Computational Optimization and Applications*, vol. 63, no. 1, pp. 29–44, Jan. 2016. [Online]. Available: https://link.springer.com/article/10.1007/s10589-015-9768-y 3.3.1

[57] L. Yeganova and W. J. Wilbur, "Isotonic regression under Lipschitz constraint," *Journal of Optimization Theory and Applications*, vol. 141, no. 2, pp. 429–443, May 2009. [Online]. Available: https://link.springer.com/article/10.1007/s10957-008-9477-0 3.3.1

[58] S. M. Kakade, V. Kanade, O. Shamir, and A. Kalai, "Efficient learning of generalized linear and single index models with isotonic regression," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 927–935. [Online]. Available: http://papers.nips.cc/paper/4429-efficient-learning-of-generalized-linear-and-single-index-models-with-isotonic-regression.pdf 3.3.1

[59] M. Schmidt, E. V. D. Berg, M. P. Friedl, and K. Murphy, "Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm," in *Proc. of Conf. on Artificial Intelligence and Statistics*, 2009, pp. 456–463. 3.3.2

[60] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999. 3.3.2

[61] Y. Lucet, "Faster than the fast Legendre transform, the linear-time Legendre transform," *Numerical Algorithms*, vol. 16, no. 2, pp. 171–185, Mar. 1997. [Online]. Available: https://link.springer.com/article/10.1023/A:1019191114493 3.3.2, 3.3.2

[62] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," *Statistical Science*, vol. 27, no. 4, pp. 538–557, Nov. 2012. [Online]. Available: http://projecteuclid.org/euclid.ss/1356098555 1

[63] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, Apr. 2011. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/090761793 3

[64] E. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010. 3

[65] C. Hedge, *Summary of February 1-3, 2011 Central and Eastern U.S. Winter Storm*. Weather Prediction Center, National Oceanic and Atmospheric Administration, Feb. 2011. [Online]. Available: http://www.wpc.ncep.noaa.gov/winter_storm_summaries/event_reviews/2011/Feb1-3_Central_Eastern_Winterstorm.pdf 3.5.2

[66] D. Hamrick, *Mid-Atlantic and Northeast U.S. Winter Storm January 26-27, 2011*. Weather Prediction Center, National Oceanic and Atmospheric Administration, Jan. 2011. [Online]. Available: http://www.wpc.ncep.noaa.gov/winter_storm_summaries/event_reviews/2011/Mid-Atlantic_Northeast_WinterStorm_Jan2011.pdf 3.5.2

[67] "Healthy Ride Pittsburgh," https://healthyridepgh.com/data/, Oct. 2016. [Online]. Available: https://healthyridepgh.com/data/ 3.5.3

[68] A. S. Willsky, E. B. Sudderth, M. I. Jordan, and E. B. Fox, "Nonparametric

Bayesian learning of switching linear dynamical systems," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 457–464. [Online]. Available: http://papers.nips.cc/paper/3546-nonparametric-bayesian-learning-of-switching-linear-dynamical-systems.pdf 4.1.1

[69] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An Introduction to MCMC for Machine Learning," *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, Jan. 2003. [Online]. Available: https://link.springer.com/article/10.1023/A:1020281327116 4.1.1

[70] R. Dahlhaus, "Locally stationary processes," *Handbook of Statistics*, vol. 30, pp. 351–412, 2012. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=9wXOytMWHQoC&oi=fnd&pg=PA351&dq=locally+stationary+processes+dahlhaus&ots=5iSnYa6gZM&sig=c2dndyjRfnKwKMq-dy7bv1WD-Yw 4.1.1

[71] J. Fan and I. Gijbels, "Local linear smoothers in regression function estimation," North Carolina State University, Technical Report 2055, 1991. 4.1.2

[72] P. Qiu, "A jump-preserving curve fitting procedure based on local piecewise-linear kernel estimation," *Journal of Nonparametric Statistics*, vol. 15, pp. 437–453, 2003. 4.1.2, 4.2.1

[73] I. Gijbels, A. Lambert, and P. Qiu, "Jump-preserving regression and smoothing using local linear fitting: a compromise," *Annals of the Institute of Statistical Mathematics*, vol. 59, pp. 235–272, 2007. 4.1.2, 4.2.1

[74] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*. SIAM, Jan. 1997, google-

Books-ID: JaPtxOytY7kC. 4.1.3

[75] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *In ICML 08: Proceedings of the 25th International Conference on Machine Learning*, 2008. 4.1.3

[76] D. Dueck and B. Frey, "Probabilistic sparse matrix factorization," *Univ. Toronto Tech. Rep. PSI-2004*, 2004. [Online]. Available: http://www.psi.toronto.edu/psi/pubs/2004/PSI-TR-2004-23.pdf 4.1.3

[77] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016 4.1.3, 4.3

[78] D. Hajinezhad, T. H. Chang, X. Wang, Q. Shi, and M. Hong, "Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 4742–4746. 4.1.3, 4.3

[79] T. Pock and S. Sabach, "Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems," *SIAM Journal on Imaging Sciences*, vol. 9, no. 4, pp. 1756–1787, Jan. 2016. [Online]. Available: https://epubs.siam.org/doi/10.1137/16M1064064 4.1.3, 4.3

[80] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*,

vol. 146, no. 1-2, pp. 459–494, Aug. 2014. [Online]. Available: https://link.springer.com/article/10.1007/s10107-013-0701-9 4.3

[81] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, Jan. 2010. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/090769521 4.3

[82] "U.S. Senate: Roll call votes 112th Congress-2nd session (2010)," https://www.senate.gov/legislative/LIS/roll_call_lists/vote_menu_111_2.htm, Dec. 2010. [Online]. Available: https://www.senate.gov/legislative/LIS/roll_call_lists/vote_menu_111_2.htm 4.4.2

[83] "U.S. Senate: Roll call votes 112th Congress-1st session (2011)," https://www.senate.gov/legislative/LIS/roll_call_lists/vote_menu_112_1.htm, Dec. 2011. [Online]. Available: https://www.senate.gov/legislative/LIS/roll_call_lists/vote_menu_112_1.htm 4.4.2, 4.4.2

[84] C. F. Karpowitz, J. Q. Monson, K. D. Patterson, and J. C. Pope, "Tea Time in America? The impact of the Tea Party Movement on the 2010 midterm elections," *PS: Political Science & Politics*, vol. 44, no. 2, pp. 303–309, Apr. 2011. [Online]. Available: https://www.cambridge.org/core/journals/ps-political-science-and-politics/article/tea-time-in-america-the-impact-of-the-tea-party-movement-on-the-2010-midterm-elections/9AB13FC062670E3635A656F5BCB1617D 4.4.2

[85] "Arkansas Presidential Election Voting History," https://www.270towin.com/states/Arkansas, Apr. 2018. [Online]. Available: https://www.270towin.com/states/Arkansas 4.4.2

[86] "Indiana Presidential Election Voting History," https://www.270towin.com/states/Indiana, Apr. 2018. [Online]. Available: https://www.270towin.com/states/Indiana 4.4.2

[87] V. Williamson, T. Skocpol, and J. Coggin, "The Tea Party and the remaking of Republican conservatism," *Perspectives on Politics*, vol. 9, no. 1, pp. 25–43, Mar. 2011. [Online]. Available: https://www.cambridge.org/core/journals/perspectives-on-politics/article/tea-party-and-the-remaking-of-republican-conservatism/BDF68005B52758A48F7EC07086C3788C 4.4.2, 4.4.2

[88] B. T. Gervais and I. L. Morris, "Reading the Tea leaves: Understanding Tea Party caucus membership in the US House of Representatives," *PS: Political Science & Politics*, vol. 45, no. 2, pp. 245–250, Apr. 2012. [Online]. Available: https://www.cambridge.org/core/journals/ps-political-science-and-politics/article/reading-the-tea-leaves-understanding-tea-party-caucus-membership-in-the-us-house-of-representatives/63C5B673B0C3956749830ED519A82360 4.4.2

[89] P.-L. Loh and M. J. Wainwright, "High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity," *The Annals of Statistics*, vol. 40, no. 3, pp. 1637–1664, Jun. 2012. [Online]. Available: http://projecteuclid.org/euclid.aos/1346850068 A.2.1, A.2.1, A.2.3

[90] D. G. Feingold and R. Varga, "Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem," *Pacific Journal of Mathematics*, vol. 12, no. 4, pp. 1241–1250, Dec. 1962. [Online]. Available: http://msp.org/pjm/1962/12-4/p08.xhtml

A.2.3

[91] M. Rudelson and R. Vershynin, "Hanson-Wright inequality and sub-Gaussian concentration," *Electronic Communications in Probability*, vol. 18, no. 82, pp. 1–9, Oct. 2013. [Online]. Available: http://ecp.ejpecp.org/article/view/2865 A.2.3, A.2.5