TIME SERIES ESTIMATION IN A SPIKED SIGNAL REGIME

BY

BENJAMIN C. ENG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Assistant Professor Lara S. Waldrop

ABSTRACT

Working with missing or incomplete data is a universal problem in all sciences. In meteorology, temperature data streams can contain missing values due to sensor malfunctions. In geophysical remote sensing, missing data can may be attributed to irregular global sampling by an orbiting spacecraft. In a collaborative filtering application, like the Netflix Challenge, data is incomplete since it is not possible for all users to provide a recommendation on all items. Though we do not have access to complete data, it is still quite possible to forecast weather, and to recommend good movies on Netflix. The development of estimation algorithms that properly handle missing data make data imputation and forecasting possible.

The design of any estimation algorithm depends on the assumptions one can make on a given set of data. This thesis addresses the problem of estimating a noisy, incomplete time series of a dynamical system with unknown state evolution. The technique presented is TSCC (Transformed Spiked Covariance Completion), a matrix completion algorithm for signal estimation that leverages the spiked signal model, an assumption that holds true for many high-dimensional datasets. The TSCC technique exploits this assumption to develop an estimator that is resilient to noise and accurately fills in missing data.

This thesis first addresses the specific estimation problem and the signal model that it follows. It then presents a survey of both standard and the state-of-the-art techniques in addition to an anlaysis of TSCC. These methods are used to solve the problem of estimating the state of dynamical system, with partial, noisy observations. Standard textbook techniques are not reliable in state estimation due to their inability to handle missing data and to generalize dynamical models. TSCC is an algorithm which addresses this estimation problem and accounts for the deficiencies. Concluding this thesis, several numerical experiments on both synthetic and real data demonstrate

that TSCC outperforms these other techniques by forming a time-lagged embedding and estimating the dynamical modes of the system.

TSCC has an advantage over other techniques as it does not require knowledge of the state dynamics and that it leverages the asymptotic behavior of noisy, low-rank matrices to perform imputation and denoising. The TSCC technique assumes that a system can be represented by several dynamical modes which is analgous to a matrix having a low rank. Overall, TSCC is a state estimation algorithm that performs estimation on noisy and incomplete data without prior model assumptions. Numerical experiments show that TSCC is an enhancement of the current, accepted techniques which address the same estimation problem.

To Mom, Dad, and Brendan
For Everything

ACKNOWLEDGMENTS

There were many individuals who contributed to the development of this Master's thesis. First and foremost, I would like to thank my advisor, Professor Lara Waldrop, for her technical guidance (and patience) throughout the duration of my graduate studies. It was an honor to work for such a passionate scientist the last two years. In addition, I would also like to thank my advisors across the quad in CSL, Professors Farzad Kamalabadi and Zhizhen Zhao. I thank them for inciteful suggestions and useful discussions regarding the signal processing aspects of this project. Without the help and encouragement of these three individuals, this work would not have been possible.

There were many individuals throughout my time here at Illinois that I would also like to acknowledge. I'd like to thank the ECE Buds: Shruti Vaidya, Umberto Ravaioli, Daisy Fong, Kin Man Lee, Vivian Hou, and Ben Kuo (Status) for those afternoons spent working in the atrium and those late nights building robots in ESPL during undergrad studies. I would also like to thank the members of the RSSS team: Jianqi Qin, Gonzalo Cucho, Yamuna Phal, Pratik Joshi, Bill Wang, Paul Lee, Shiyi Yang, Matt Grawe, Anil Agarwal, and Tony Caton for lively discussions on ionospheric physics, optics, and other tea-time related topics. In addition, I'd also like to thank the many roommates of 803, Daniel Tu, Felipe Fregoso, Abhi Deshpande, Mitchell Quigley, Top, and Kevin Dvorak for being my family away from family.

Finally, this all would not be possible without the support of my family. I thank my parents for not giving up on me and encouraging me to pursue engineering. I thank my younger brother Brendan for his encouragement, enthusiasm, and support.

The work presented in this thesis was funded in part by NASA grant NNX16AF77G.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

EBLP	Empirical Best Linear Predictor
EnKF	Ensemble Kalman Filter
KF	Kalman Filter
PCA	Principal Component Analysis
SSA	Singular Spectrum Analysis
TRMF	Temporal Regularized Matrix Factorization
TSCC	Transformed Spiked Covariance Completion

CHAPTER 1

MISSING DATA IN DYNAMICAL SYSTEMS

1.1 Missing Data: A Universal Problem

In all fields of the natural sciences, we rely upon data, whether acquired in-situ or remotely, to draw conclusions about the world. For example, a typical goal of meteorology is to forecast the temperature at a given location several days in advance from an ensemble of in-situ temperature measurements. Other examples include radar remote sensing, which typically involves tracking a moving target based on the backscattered radiation field collected by an antenna, or space-based optical remote sensing, which is often used to yield models of the geophysical environment from spectroscopic measurements of its photochemical emissions.

A common challenging task in forecasting, tracking, or modeling is the need to work with missing or incomplete data. For temperature forecasting, it is impossible to measure temperature everywhere all the time, and sensor malfunction can introduce data gaps that augment measurements associated with distributed sensor placement. In radar applications, radio frequency interference or ground clutter can introduce intermittent contamination of the backscattered signal, while the often limited viewing geometry from satellites precludes comprehensive spatial sampling of the radiation field. Despite the common occurrence of sensor outages, data contamination, or incomplete sampling in these and other data analysis applications, it is still possible to forecast the weather, track a moving object, and model the geophysical environment. These tasks are enabled by algorithms that accurately estimate the true state from partial observations of the system.

1.2 Observing Dynamical Systems

Dynamic state estimation problems arise in many domains. In dynamic imaging applications such as biomedical or solar tomography [1], the objective is to image an object in motion by measuring the time-dependent projections of that object on a sensor. Within the interval of measurement acquisition, whether an individual image or the ensemble used for the analysis, the dynamics of the system must be properly taken into account. Climatological data analysis of time-dependent geophysical signals, usually incompletely sampled, likewise requires developing a reliable approach for dynamic signal recovery in the presence of missing data.

1.3 Overview of Thesis

This thesis addresses the problem of estimating the state of a dynamical system from noisy, incomplete observations. Though many state estimation algorithms that already address this problem exist, many have shortcomings in computational capability, data scaling, improper treatmeant of partial data, and strong assumptions on parametric state evolution models. To address these issues, this thesis will discuss the Transformed Spike Covariance Completion (TSCC) algorithm, a new method that leverages the spiked signal model to develop an estimator that produces accurate imputation and denoising with limited assumptions on state evolution models. Chapter 2 first discusses some mathemtical prerequistites that are fundamental to the understanding of these estimation algorithms. Chapter 3 then discusses the signal model used in the problem in addition to the current techniques that address them. The deficiences of these current techniques are addressed by this thesis's main contribution, TSCC in Chapter 4. Chatper 5 conlcludes this thesiswith several numerical experiments that show the performance of TSCC over other standard and state-of-the-art algorithms on both synthetic and real data.

CHAPTER 2

SVD, MATRIX NORMS, AND THE LOW-RANK APPROXIMATION

This chapter will serve as a primer for some concepts in linear algebra that will be used throughout this thesis. These matrix methods are widely used across all engineering fields as they are useful in linear inverse problems, data reduction, and reduced order modeling to name a few specific application domains. The same methods can be applied to time series analysis problems.

2.1 Singular Value Decomposition (SVD)

All matrices M have a Singular Value Decomposition (SVD). The SVD breaks down a matrix M into its leftmost singular vectors U, rightmost singular vectors V, and a diagonal matrix containing the singular values Σ. The applications of the SVD will be clearly described in the sections describing SSA, EBLP, PCA, and TSCC.

$$M = U\Sigma V^T \tag{2.1}$$

More explicitly if M is an $m \times n$ marix this is

$$M = \begin{bmatrix} u_1 & u_2 & u_3 & \dots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \sigma_m \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} \tag{2.2}$$

U is a matrix $\in \mathbb{R}^{m \times m}$, whose columns contain singular vectors u_i. V is a matrix $\in \mathbb{R}^{n \times n}$, whose columns contain singular vectors v_i. Σ is a diagonal matrix $\in \mathbb{R}^{m \times n}$ which has diagonal values σ_i containing the singular values of M. In addition, most numerical packages that perform SVD have the values of σ_i arranged in decreasing order (e.g. $\sigma_1 > \sigma_2 > \sigma_3 ...$). Other special

properties, with regard to these matrices are that U and V are orthogonal, meaning that UU^T and V^TV yield the identity I.

From college algebra, it is important to note the relationship between the SVD and spectral decomposition of positive semidefinite matrices. Suppose M has decompsoiton $U\Sigma V^T$.

$$\begin{aligned} MM^T &= (U\Sigma V^T)(U\Sigma V^T)^T \\ &= (U\Sigma V^T V\Sigma^T U^T) \\ &= U\Sigma\Sigma^T U^T \\ &= U\Lambda U^T \end{aligned} \tag{2.3}$$

The diagonal of Σ is just the square root of the eigenvalues of MM^T. In addition, U contains the eigenvectors of MM^T. By a similar exercise, V contains the eigenvectors of M^TM. Given these properties, one can define several matrix norms in the next section. The previous results are common in most linear algebra textbooks.

2.2 Matrix Norms

For many applications in image and signal processing, it is useful to quantitatively know the similarity between matrices and vectors. Norms are a useful way to determine this similarity. Matrix norms are functions on matrices that assign a positive length to the given matrix much like how vector norms assign length. Many matrix norms exist and can be written but in the context of matrix completion and time series analysis; however, this thesis only discusses the Frobenius norm and the Schatten norm.

The Frobenius norm for a matrix M is written as

$$\|M\|_F = \sqrt{\sum_{i\in I}\sum_{j\in J}|m_{i,j}|^2} = \sqrt{trace(M^TM)} \tag{2.4}$$

Simply stated, the Frobenius norm is the square root of the sum of the square of the entries of the elements $m_{i,j}$ in matrix M over all rows and columns. This calculation is also equivalent to the square root of the trace of M^TM. The trace is simply the sum of the diagonal elements of a matrix.

The Frobenius norm is analogous to the assignment of a Euclidean length on vectors.

Another commonly used norm in matrix completion algorithms is the Schatten norm. This is defined as

$$\|M\|_{*p} = (\sum_{i=1}^{\min m,n} \sigma_i^p(M))^{\frac{1}{p}} \tag{2.5}$$

Here σ_i is a function that returns the i^{th} largest singular value of matrix M. The operator essentially sums the p^{th} power of each singular value. In the case where p = 1, the Schatten norm reduces to the sum of the singular values in M, namely

$$\|M\|_{*} = (\sum_{i=1}^{\min m,n} \sigma_i) \tag{2.6}$$

For the p = 1 condition, the Schatten norm is sometimes then referred to as the nuclear, trace, or Fan norm.

2.3 Low-Rank Models

One concept that is addressed in section 3.3 is that multivariate time series can be rewritten as a sum of lower-rank matrices. It is common to approximate a given matrix with a simpler, low-rank version. Many of these approximations come from the assumption that the class of data that one is working with is low-rank. In dynamical systems theory, a common assumption is that a system can be represented by a finite number of modes. Several methods can be used to find this low-rank approximation, namely by some minimization of norm error with regularization. In these cases, they are the Frobenius norm and the nuclear norm with Tikhonov regularization. This section will briefly describe these approximation methods.

The problem in low-rank approximation is the following. Given a matrix M, one seeks to find an approximate matrix $\hat{M}$, that has low rank (a small number of indepenedent columns in $\hat{M}$). This problem can be formulated as the minimization of the Frobenius norm error between M and $\hat{M}$ under the constraint that the rank of $\hat{M}$ is less than some desired rank r, an integer.

The problem can then be written as the following optimization equation

involving a Frobenius norm error and a rank constraint

$$\begin{aligned} \underset{\hat{M}}{\text{minimize}} \quad & \left\| M - \hat{M} \right\|_F^2 \\ \text{subject to} \quad & rank(\hat{M}) < r \end{aligned} \tag{2.7}$$

Typically, solving such optimization problem is difficult as the minimization of the Frobenius norm error is a non-convex problem. However, this rank minization for this problem can be solved using a sum or rank-1 matrices. Suppose a matrix M_1 has rank R_1 and suppose $R_1 > R_2$. Then M_1 can be written as

$$M_1 = U\Sigma V^T = \sum_{k=1}^{R_1} \sigma_k u_k v_k^T \tag{2.8}$$

The rank R_2 approximation of the matrix is simply calculated as the sum of the first R_2 components.

$$\hat{M}_1 = \hat{U}\hat{\Sigma}\hat{V}^T = \sum_{k=1}^{R_2} \sigma_k u_k v_k^T \tag{2.9}$$

Here $\hat{U}$ would be the first R_2 columns U, $\hat{\Sigma}$ contains the first R_2 columns of Σ and $\hat{V}$ contains the first R_2 rows V.

The approximation in equation 2.9 can be used to solve the optimization problem in equation 2.7 without more sophisticated non-convex programs. The proof of this solution to the low-rank approximation problem is shown in the well-known Ekhart-Young theorem. Approximation in this case is simply just determing the first R_2 left and right singular vectors, multipliying the left singular vector with the transpose of the right singular vectors and scaling by the associated singular value to form a rank 1 matrix. These rank 1 matrices are summed over the top R_2 singular values.

The problem in equation 2.7 can be varied by penalizing the rank through some regularization instead of setting a hard number for the rank. This is formulated in the following optimization problem:

$$\underset{\hat{M}}{\text{minimize}} \quad \left\| M - \hat{M} \right\|_F^2 + \lambda \times rank(M) \tag{2.10}$$

Here the $rank()$ function computes the rank of some input matrix M and λ is a penalty factor on the rank. Unlike equation 2.9 where one effectively sets the singular values σ_i, for $i > R_2$, to zero, the approach in equation 2.10 can be viewed as thresholding the singular values at some cutoff values. This is known as a hard thresholding.

$$\hat{\sigma}_k = \begin{cases} \sigma_k, & \text{if } \sigma_k \geq \beta \\ 0, & \sigma_k < \beta \end{cases} \tag{2.11}$$

Here β is some set threshold. The approximation of M thus is

$$\hat{M} = U\hat{\Sigma}V^T \tag{2.12}$$

where the diagonal matrix $\hat{\Sigma}$ has diagonal values $\hat{\sigma}_k$ as show in equation 2.11. The last low-rank approximation technique that this thesis will address is a soft thresholding algorithm. This is also known as singular value shrinkage. Here, instead of setting a limit to the singular values to where they suddenly become zero, this algorithm will gradually phase out the singular values. This becomes

$$\hat{\sigma}_s = \begin{cases} \sigma_k - \beta, & \text{if } \sigma_k \geq \beta \\ 0, & \sigma_k < \beta \end{cases} \tag{2.13}$$

Here, the methods subtracts or shrinks out some strength from the largest singular values; this also has the effect of setting the weaker singular values to zero. The approximation then becomes the same as equation 3.26 with the exception that $\hat{\Sigma}$ has diagonal entries of $\hat{\sigma}_s$ instead of $\hat{\sigma}_k$. This solution presented in equation 2.13 solves the following optimization problem

$$\underset{M}{\text{minimize}} \quad \left\| M - \hat{M} \right\|_F^2 + \lambda \times \left\| \hat{M} \right\|_* \tag{2.14}$$

Here $\left\| \hat{M} \right\|_*$ is the nuclear norm of $\hat{M}$ which is the sum of the singular values in $\hat{M}$. The process of determing a solution to equation 2.14 can be done through a linear program as the cost function in equation 2.14 is convex. This soft thresholding algorithm is detailed in [2].

The tools introduced in this section will be utilized in all the techniques de-

scribed in Chapter 3 for time series estimation. Essentially, one can reform a time series estimation problem as a low-rank approximation problem. These problems, as evident in this section, rely heavily on characterizing the error through a careful choice of matrix norm. The means to find these approximate matrix solutions rely ultimately on proper treatment on the shrinkage or the thresholding of the singular values.

CHAPTER 3

TIME SERIES ESTIMATION TECHNIQUES

This chapter will lay out the standard and state-of-the-art techniques used for time series estimation of dynamical systems. Given a series of measurements or observations of a system, these technqiues estimate the system's true state. Section 3.1 will first state mathematically the signal model and the problem that this research will address. Then, section 3.2 discusses the standard methods used to solve this problem, namely the Kalman filter and singular spectrum analysis. Section 3.3 presents more modern methods like the total regularized matrix factorization (TRMF) algorithm and other matrix completion algorithms. This chapter ends with a presentation of the empirical best linear predictor, an estimation technique that leverages the spiked signal model, a common characteristic in today's large datasets.

3.1 Signal Model

Estimation of the state of a dynamical system usually begins by some processing of measurements of that system from some sensor. These measurements are some linear transformation of the true state. In a discrete setting, these measurements are put into a time series indexed by some time sample index set i. In this study, dynamics will be considered to be linear. This model is summarized by the following:

$$X_{i+1} = F_i X_i + w_i \tag{3.1}$$

$$Y_i = A_i X_i + \epsilon_i \tag{3.2}$$

The dynamics of this system is described by equations 3.2 and 3.1. Here X_i is a state vector in $\in \mathbb{R}^N$ at time instance i. F_i is the state transition operator that models the next realization of the state X_{i+1}. w_i is a process noise that

accounts for errors in the state transition model. Y_i is the measurement vector collected by a sensor that also resides in $\mathbb{R}^N$. A_i is an observation matrix that maps the true state X_i to the measurement Y_i. A_i is a diagonal matrix in $\mathbb{R}^{N\times N}$ with entries either 0 or 1, accounting for whether or not a vectoral component of the system was observed. Here ϵ_i is a measurment noise. Both ϵ_i and w_i are both additive white Guassian noises.

In this estimation problem, there is no prior knowledge on the state transition matrix F_i. This problem assumes complete knowledge of the observation matrix A_i, which maps the true state to the measurement. The goal of this problem is to estimate X_i with this partial and missing data observation Y_i. In a sense, this problem, like most signal processing problems, is a denoising and imputation problem. As this thesis is interested in estimation of high-dimensional systems, the methods presented assume that the observations follow a spike signal model that will be clearly defined in section 3.6.

By inspection of the above signal model, one can see that this is the state space of linear dynamical system. Traditionally,the estimation problem this thesis addresses has solutions with the classical Kalman filter discussed in the next section.

3.2 Kalman Filter

The Kalman filter is an estimation algorithm which addresses the state space model in equation 3.1. The KF estimates the true state of a system by considering a series of noisy measurements over time and predicting the probability distribution of the state iteratively. The KF is commonly used in the controls community in technologies like navigation, robotics, and econometrics.

The textbook state-space model usually is written as

$$X_{i+1} = F_i X_i + B_i U_i + w_i \tag{3.3}$$

$$Y_i = A_i X_i + \epsilon_i \tag{3.4}$$

The only difference between equations 3.3 and 3.1 is that the state space for the KF has the term $B_i U_i$ which represents a control input U_i. In this case, U_i is 0. In this sense, equation 3.3 becomes equation 3.1. The other difference between the state spaces is that F_i is not usually known. The

traditional KF assumes known state transitions; however, extensions of the KF like the switching KF can account for these defficiencies by learning the state transitions.

Using the KF or one of its many variants, one can forumulate an estimation algorithm as the following. Suppose Q_i and R_i are the covariance matrices of the process and observation noise. In addition $\hat{X}_{i|i}$ denotes the a posteriori state estimate at time i with all observations before and at i taken into account. In addition, $P_{i|i}$ is the error covariance which is calculated to be the $Cov(X_i - \hat{X}_{i|i})$. From these matrices, estimation can be split into two parts, predicition and update. In prediction, the state $\hat{X}_{i|i-1}$ and error covariance $P_{i|i-1}$ is computed. In the update, the state estimate and the error covariance are updated based on the Kalman gain and the residual error between measurement Y_i and the predicted measurement $\hat{Y}_i$.

Prediction contains the following computations:

$$\hat{X}_{i|i-1} = F_i \hat{X}_{i-1|i-1} \tag{3.5}$$

$$P_{i|i-1} = F_i P_{i-1|i-1} F_i^T + Q_i \tag{3.6}$$

Equations 3.5 and 3.6 represent predictions of the state and error covariance with measurements up to time $i-1$.

Update contains the following computations:

$$e_i = Y_i - A_i \hat{X}_{i|i-1} \tag{3.7}$$

$$S_i = R_i + A_i P_{i|i-1} A_i^T \tag{3.8}$$

$$K_i = P_{i|i-1} A_i S_i^{-1} \tag{3.9}$$

$$\hat{X}_{i|i} = \hat{X}_{i|i-1} + K_i e_i \tag{3.10}$$

$$P_{i|i} = (I - K_i A_i) P_{i|i} (I - K_i A_i)^T + K_i R_i K_i^T \tag{3.11}$$

$$e_{i|i} = Y_i - A_i \hat{X}_{i|i} \tag{3.12}$$

Equation 3.7 represents the residual error between the measurement and the predicted measurment with observations up to $i-1$. Equation 3.8 represents the covariance of the error residual. Equation 3.9 represents the Kalman gain. Equation 3.10 represents the state estimate with observations up to and including time i. Equation 3.11 represents the error covariance with obser-

vations up to and including time i. Lastly, Equation 3.12 is a calculation of the error between the measurement and the predicted measurement with observations up to and including time i. Equations 3.5 to 3.12 represent the totality of the prediction and update process for state estimation with the KF. The derivation of these parameters are commonly found in most control theory textbooks.

The KF as modeled in equations 3.3 to 3.12, proves robust in many applications where state transition is known. In estimation problems where F_i is not known or understood well, improper treatment of the state transition can propagate error in the estimation as the steps require F_i to be known accurately to perform updates to the Kalman gain. In addition, in applications where the dimension of the state is moderate, computational constraints are not a concern. When performing imaging applications where the state could be an image of size 512×512, the dimension becomes 2^{18}. As a result, the inversion of a $2^{18} \times 2^{18}$ matrix must be considered as in equation 3.9 in the update process.

Considering computational constraints and the large uncertainty with the linear dynamics, the KF may not be a suitable choice in designing a robust algorithm for this state estimation problem. Section 3.3 discusses Singular Spectrum Analysis (SSA), a non-parametric estimation technique that does not require knowledge of these state dynamics in performing estimation.

3.3 Singular Spectrum Analysis

Singular Spectrum Analysis (SSA), is a non-parametric time series estimation technique. Unlike the Kalman filter, which is an iterative technique that requires many parameters like covariance matrices, SSA performs estimation by forming a time-lagged embedding. These embeddings are inspired by the embedding work done in dynamical systems theory as shown in [3] and [4]. The motivating theory behind SSA is that by forming a time-lagged embedding of measurements, one can decompose the time series into several dynamical modes. This is under the assumption that a system can be accurately characterized by a few modes, decoupling modes that can represent noise.

The algorithm for SSA begins by the following process. This subsection de-

scribes a univariate time series, which can be easily extended to multivariate time series. Here, the technique begins with a time series of measurements $Y_i = [Y_0, Y_1, ...Y_{T-1}]$ where T represents the total number of measurements one has access to. Each instance in Y_i is a single scalar value. This technique forms a time-lagged embedding of length L, $\hat{Y}$. This is done by forming a matrix that has lagged instances of the signal measurements. This matrix is shown below:

$$\hat{Y} = \begin{bmatrix} Y_L & Y_{L+1} & ... & Y_{L+T-1} \\ Y_{L-1} & Y_L & ... & Y_{L+T-2} \\ ... & ... & ... & ... \\ Y_1 & Y_2 & ... & Y_T \end{bmatrix} \tag{3.13}$$

Here $\hat{Y}$ is an $L \times K$ matrix where K is an $L \times T - L + 1$ matrix. One can clearly see that this forms a Toeplitz structure. For the multivariate case, this would be a block Toeplitz structure. Given this embedding, one can decompose the structure to the principal modes via an SVD. Suppose now, $\hat{Y} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$. This method can then form several rank-1 matrices

$$D_i = \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^T \tag{3.14}$$

Here i indexes the number of non-zero singular values in $\hat{\Sigma}$. A larger singular value $\tilde{\sigma}_i$ indicates that the associated eigenvector represents a dominant dynamical mode. Thus one can now approximate $\bar{Y}$ with $\tilde{D}$ to create a new trajectory matrix that is more represenative of the principal modes. $\tilde{D}$ is calculated as the following:

$$\tilde{D} = D_1 + D_2 + D_3 + \dots D_R \tag{3.15}$$

This method then sums the first R principal modes. This process is known in the literature as eigen-triple adding. To determine the final estimate of the clean signal X_i , this method then performs diagonal averaging on $\tilde{D}$. Diagonal averaging is simply summing the values on the diagonals $\tilde{D}$ and dividing by the number of occurences of the the lagged realizations $D_{i,j}$. This process is detailed in [5]

SSA is a common technique that is used in many time series applications like climatology and geophysics. This method amounts to doing a truncated

SVD on a time-lagged embedding. As a result, SSA is faster than KF as it is not iterative. Though perhaps faster than iterative techniques, SSA may not perform robustly in the face of missing data. If measurements are missing, in which case the values imputed to the trajectory matrix would be 0, the estimate of the singular vector and values may be inaccurate in the face of missing values. In this sense, the characterization of the principal eigenmodes may be skewed. Recent methods presented in [5] discuss how SSA can be updated to better account for these missing values using a dictionary-based matrix completion algorithm. Section 3.4 discusses how time series estimation can be reformed into matrix completion problems.

3.4 Matrix Completion Methods

The signal model in equation 3.1 shows that the observations Y_i are just noisy, partially measured versions of the true state X_i. In this case, the measurement operator A_i is just a diagonal matrix with either 1 or 0 on the diagonal that signifies which state feature was observed. Y_i is then multivariate time series with missing and noisy values. One can then form a matrix of measurements $Y = [Y_0, Y_1, ...Y_{T-1}]$ forming a matrix of size $N \times T - 1$. From Y, one can estimate the true time series $X = [X_0, X_1, ...X_{T-1}]$. Estimating the values in this matrix is at the core of matrix completion problems.

Matrix completion is a class of problems where measurements are put into a matrix and the missing entries of the matrix are filled in according to some desired structure of the matrix. For example, one may desire the estimated matrix to have a low-rank structure or some minimum spectral norm. Typically, these algorithms work by estimating the singular values and the left/right singular vectors of a partially observed matrix and then minimizing the Frobenius norm error between the estimated and the observed entries. In general, matrix completion techniques assume that the columns of the matrix are independently and identically distributed (iid) according to some distribution. State-of-the-art techniques such as the ones used in [6, 7, 8] impute values into the matrix with these assumptions. Other methods, e.g. [9], set the estimation of a partially observed matrix as the minimization of the nuclear norm of its estimate.

Note that these matrix completion techniques differ slightly from low-rank

approximation in the sense that error is defined over the set of observed entries. Traditionally, low-rank methods do not assume missing entries. Matrix completion methods assume low-rank structure which allow for modifications of the low-rank approximation algorithms in Chapter 2. When the low-rank assumption is properly leveraged, these techniques have high reconstruction accuracy but do not incorporate any temporal dependencies in the columns of matrices. In addition, these techniques are often employed in recommendation systems, where the measurement noise is either very low or non-existent.

For this applicaition, where one wants to estimate the state of a dynamical times series, the temporal structure of the matrix should be taken into account. For example, it is possible to reform previous algorithms like SSA into a matrix completion algorithm by modifying the treatment of missing entries. An extension of the SSA algorithm in [5] demonstrates that matrix completion can be done with SSA via dictionary learning. In this algorithm (SSA-MC), given a fully observed set of data $\bar{Y}$, a matrix, a dictionary D is learned from the normal decomposition in the traditional SSA algorithm described above. The dictionary D is taken to be the left singular vectors of the SVD of $\bar{Y}$.

Learning D and accepting new input Y, then the SSA-MC algorithm tries to best approximate and fill the entries of Y by approximating it as DL. The problem is then reformed as trying to find a matrix L that best fits the observed entries in Y. This is best interpretated in the following optimization problem:

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & rank(L) \\ \text{subject to} \quad & P_\Omega(M) = P_\Omega(DL) \end{aligned} \tag{3.16}$$

Here P_Ω is a sampling operator. This is efficiently computed through the use of Augmented Lagrangian Method (ALM) as detailed in [5]. Though this method is considered a state-of-the-art method for matrix completion in the context of time series, it has the disadvantage that a complete dataset must be used for training. This is a luxury not found in many application such as in space remote sensing, where it is nearly impossible to obtain a complete dataset. The temporal regularized matrix factorization (TRMF) is another matrix completion technique with time series but it does not require training on previous data.

3.5 Temporal Regularized Matrix Factorization

Matrix completion methods aim to impute values where data is unknown in a matrix. Usually, this is done via some rank minimization of the matrix or some nuclear norm minimization. TRMF is a matrix factorization that imputes missing values by approximating an input data matrix as the product of a feature and temporal matrix. Using the signal model described in equation 3.1, suppose that one wishes to estimate the state X_i where the measurement, Y_i is simply a noisy version of the true parameter, as in any forecasting situation. In this case the observation matrix A_i is a diagonal matrix. A zero in the j^{th} element of the diagonal indicates a missing value in the measurement. Thus one can collect the measurement y_i into a matrix Y, where the i^{th} measurement is just the column, y_i.

In this framework, the columns of Y represent a time series of measurements, which in this case are just a time series of the true signal y_i, but with noise. The TRMF algorithm assumes that the matrix Y is actually a decomposition of FM, a feature matrix $F \in \mathbb{R}^{n \times k}$ and a temporal matrix $M \in \mathbb{R}^{k \times t}$, where n is the feature dimension, t is the number of samples, and k is the latent dimension. M is commonly referred to as the latent embedding. The j^{th} measurement in Y is simply the matrix F operated on the j^{th} column of the latent embedding in M. In other words, this is $y_j = Fm_j$. The measurements of Y changes in time as M changes; future measurement y_j depends on m_j. In [10], future embeddings are modeled in an autoregressive fashion:

$$m_t = \sum_{l \in L} W^l m_{t-l} \tag{3.17}$$

Here L is a lag set and W^l is the l^{th} weight applied on the $l-lagged$ previous value of m. With the knowledge that data matrix Y has such structure, the following temporal regularization can be formulated to determine matrices F and M which constitute our original data matrix Y.

$$\underset{F,M,W}{\arg\min} \sum_{i,t \in \Omega} (Y_{it} - f_i^T m_t)^2 + \lambda_f R_f(F) + \sum_{r=1}^{k} \lambda_m T_{AR}(\bar{m}_r | L, \bar{w}_r, \eta) + \lambda_w R_w(W) \tag{3.18}$$

Examining equation 3.18, one should note that it is the sum of four terms A, B, C, D. Term A simply takes the square error between the $i^{th} feature$ at

sample t and the i^{th} row of F and the t^{th} embedding. This optimization is performed over all known values in Ω. Terms B and C are simply a Frobenius norm penalization on F and W respectively. Term C is temporal regularization which penalizes error between future embedding x_i and the sum of $W_{i-1}m_{i-1} + W_{i-2}m_{i-2} + W_{i-3}m_{i-3}....$ Minimization of this cost function is alternated between minimizing F when M, W are fixed, X when F, W are fixed, and W when F, M are fixed. The numerical methods used to solve this cost function are detailed in [10]. In summation, the imputed values are obtained when F and X are obtained by simply multiplying $FM = \hat{Y}$, where $\hat{Y}$ is an imputed approximation of X. It is important to note that weighting matrices W obtained in minimizing 3.18 can be used to predict future values of W.

Currently, in the high-dimensional time series community, TRMF serves as the state-of-the-art estimation technique due to its high prediction accuracy and its clever fitting of the autoregressive nature of the data. Though TRMF performs well on estimation in datasets that have clear periodicities that can be modeled well by an autoregressive process, this assumption can be very limiting in dynamical systems that do not follow this model. Unlike SSA, TRMF is a parametric model, meaning that there is a functional form for the linear dynamics.

Instead of fitting to a functional form, one can also consider identification of the dynamical modes embeddded within the set of observation. The following section describes a recent discovery in the geometry of high-dimensional datasets that will serve useful in developing a robust matrix completion algorithm that performs this mode identification. This thesis will then leverage this robust algorithm in the development of the main contribution of this thesis, Transformed Spiked Covariance Completion (TSCC), a non-parametric time series estimation framework.

3.6 Spiked Signal Model and PCA

In modern estimation problems, where the dimension of the data can be quite large, it is common to reduce the dimensionality of the dataset to a lower-dimensional representation. This section describes the conditions in the signal model in which this reduction can happen. This result will then

be used in developing the estimator in the following section.

Principal Component Analysis (PCA) is a common data reduction technique that is used in many facets of signal processing and machine learning. With the ever increasing demands of data processing, PCA is a robust algorithm that reduces a dataset to a possibly more meaningful, lower dimensional space. PCA works on the principle that a dataset can be projected onto a small set of eigenvectors that represent the largest variance within the data.

Suppose, there is some data matrix $X = [X_1, X_2, ...X_p]$ which lives in $\mathbb{R}^{N\times P}$. Data reduction with PCA can be done by first estimating the mean of X as

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} X_n \tag{3.19}$$

One then can estimate the covariance Σ_n to be

$$\Sigma_n = \frac{1}{N}\sum_{n=1}^{N}(X_n - \hat{\mu})(X_n - \hat{\mu})^T \tag{3.20}$$

One way to find the principal eigenvectors or the principal components, one can take an SVD of $\Sigma_n = U\Sigma U^T$ and project the first R columns of U onto X. In other words if $U = [u_1, ..., u_N]$. One can then write $U_r = [u_1, ...u_R]$. The reduced data $\hat{X} = U_r^T X$. This is one method of implmenting PCA.

PCA works in the sense that as the number of samples go to infinty, the population covariance Σ_n and the population mean $\hat{\mu}$ converge to the true covariance Σ and true mean μ. It is important to note that when examining the distribution of the eigenvalues of Σ_n, the largest eigenvalues represent the weight on the eigenvectors in the direction of greatest variance. In this case, the largest eigenvalues associated with the strongest eigenvectors do represent the true principal components. The data then lends itself as having some low-dimensional structure.

Suppose now there is the situation where n represents the dimensionality of some data and p represents the number of samples. As both $p, n \to \infty$ with a fixed ratio $p/n = \gamma \leq 1$, the strongest eigenvalue may not necessarily be associated with the true principal component. There are simply not enough samples to correctly obtain the variance in the dataset. In this case, strong

eigenvalues can be deceiving; these eigenvalues may not correspond to the true principal component.

Suppose X is an $M \times N$ iid random matrix. In random matrix theory the distribution of eigenvalues in $S_n = \frac{1}{P}XX^T$ follows the Marčenko Pastur distribution shown in [11]:

$$dF_\gamma(\lambda) = \frac{1}{2\pi}\frac{\sqrt{((\gamma_+ - \lambda)(\lambda - \gamma_-))}}{\gamma\lambda}1_{[\gamma_-,\gamma_+](\lambda)}d\lambda \tag{3.21}$$

This measure enables analysis of the distribution of eigenvalues within some dataset. In the case, where the dataset is some signal contaminated by white noise, it may not be possible to identify a principal component in the data or find the strongest eigenvalue in the dataset. In the right-hand graph in figure 3.1, in the case of very low signal to noise, it is deceiving to see that there are a few strong eigenvalues. These eigenvalues are eigenvalues associated with noise. This behavior differs from the case of high signal to noise (left-hand graph of figure 3.1). Here, it is less difficult to identify the true principal component. In the case of the high SNR regime, given a distribution of the eigenvalues, there is a popout of the largest eigenvalue from the bulk of the distribution. This popout of the eigenvalue from bulk is seen in the left-hand graph in figure 3.1.

Given a randomly generated rank-1 matrix (one principal component), with two noise levels, one can plot the distribution of the eigenvalues. It is shown in the SNR = 10 case, there is a popout of the largest eigenvalue from the distribution, meaning that a principal component can be easily identified. In the SNR = 0.01 case, one cannot see this spike effect and should be cautious in suggesting that the largest eigenvalue represents a true principal component from the original data.

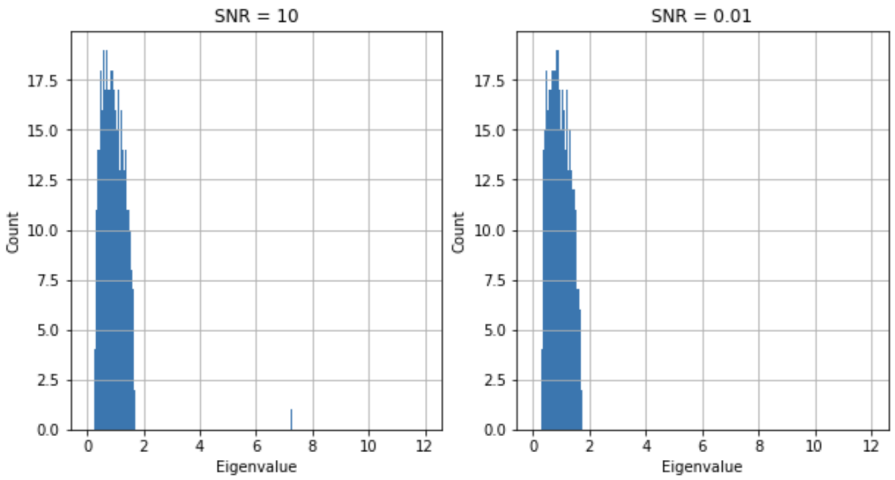

Figure 3.1: Left: In a high SNR regime, there is a popout of the largest eigenvalue from the bulk. Right: In a low SNR regime, there is no popout phenomenon and all eigenvalues are contained within the bulk.

These plots of the eigenvalues allow one to capture the principal components of the data. This phenomenon is described in [11]. As this problem works with a transformation of a high-dimensional signal with noise, one can use knowledge of the noise model and the Marčenko Pastur distribution to aid in developing a signal estimation algorithm.

3.7 Empirical Best Linear Predictor

This thesis now discusses a solution to the problem $Y_i = A_i X_i + \epsilon_i$. The work in Emperical Best Linear Prediction (EBLP) provides a framework for creating an estimator that is resilient to noise, handles missing data, and is computationally efficient. EBLP is an estimator based on the Best Linear Unbiased Predictor (BLUP). In signal processing, BLUP is know as the Wiener filter for denoising.

EBLP is an asymptotic estimator of BLUP. This technique uses the spiked signal model. In notation within the statistics community, suppose the time series has X_i as a $p-dimensional$ vector and that there are n measurements. EBLP is BLUP in the limit that both $n, p \to \infty$ and that the ratio $\frac{p}{n} \to \gamma$. This is in the same spirit of the spike model in section 3.6. In addition, this framework treats X_i as random vectors lying in a low-dimensional space, namely:

$$X_i = \sum_{k=1}^{r} l_k^{\frac{1}{2}} z_{ik} u_k \tag{3.22}$$

Here u_k is a unit vector where l_k is a scale and z_{ik} is a mean zero random variable.

3.7.1 From BLUP to EBLP

The difference between BLUP and EBLP is clarified in this section. Recall that BLUP is a parameter estimation technique which comes from the setting of a random effects model, where the samples are drawn from populations with differing variances. It is shown in the literature that the best estimator of X_i from the measurement Y_i is given as $\hat{X}_i^{BLP}$. This is the best estimate in the mean squared error sense (i. e. $\mathbf{E}\left\|\hat{X}_i^{BLP} - X_i\right\|^2$)

$$\hat{X}_i^{BLP} = \Sigma_x A_i^T (A_i \Sigma_x A_i^T + \Sigma_\epsilon)^{-1} Y_i \tag{3.23}$$

This technique defines Σ_X to be the covariance matrix of X_i and Σ_ϵ to be the covariance matrix of the noise ϵ_i. One can write Σ_X as

$$\Sigma_X = \sum_{k=1}^{r} l_k u_k u_k^T \tag{3.24}$$

Both parameters Σ_ϵ and Σ_X must be estimated. Since this thesis is interested in the high-dimensional case of this predictor, it is difficult to exactly estimate the true population covariance of Σ_X. In addition, this estimator requires the use of a matrix inverse, which can be slow computationally in high-dimensional signals. In the limit that the number of samples tend to infinity, the estimator asymptotically converges to a new predictor that does not involve a matrix inverse. This convergence is expressed as:

$$\hat{X}_i^0 = \sum_{k=1}^{r} \eta_k^0 \langle A_i^T Y_i, \mu_k \rangle \mu_k \tag{3.25}$$

In writing this estimator, the left singular vectors are unknown and must be estimated from the population covariance which in this case is

$$\Sigma_p = \sum_{k=1}^{n} (A_i^T Y_i)(A_i^T Y_i)^T \times \frac{1}{n} \tag{3.26}$$

This process is also equivalent to taking the left singular vectors from the SVD of the matrix $[A_1^T Y_1, ...A_N^T Y_N]$. One can now write the estimator in a similar fashion to equation 3.25.

$$\hat{X}_i^\eta = \sum_{k=1}^{r} \eta_k \langle A_i^T Y_i, \hat{\mu}_k \rangle \hat{\mu}_k \tag{3.27}$$

One should note that now η_k is different from η_k^0 due to the angle between the correlation of the true singular vector with the population singular vector.

Suppose one can work with the backprojected data $B_i = A_i^T Y_i$. The estimator in equation 3.27 becomes

$$\hat{X}_i^\eta = \sum_{k=1}^{r} \eta_k \langle B_i, \hat{\mu}_k \rangle \hat{\mu}_k \tag{3.28}$$

This methods writes the data matrix estimate to be $\hat{X}^\eta = [\hat{X}_1^\eta ... \hat{X}_n^\eta]$. This estimate can also be expressed as $\hat{X}^\eta = \sum_{k=1}^{r} \eta_k \hat{\mu}_k \hat{\mu}_k^T B$. From this result, B and X^η have the same singular vectors. One can then argue that $\hat{X}_i^\eta$ is a random variable with A_i and ϵ_i. Thus, the error in predicting $\hat{X}_i^\eta$ is the same as the error in estimating the matrix $\hat{X}^\eta$. Thus, one can conclude that estimating X_i is equivalent to a matrix completion or a low-rank approximation on B, where approximation can be accomplished by singular value shrinkage.

3.7.2 EBLP Theoretical Development

This section will provide the derivation of the EBLP algorithm and then discuss the step-by-step implentation of EBLP. First, this method shall define the backprojected sample $\tilde{B}_i = M^{-1} A_i^T Y_i$ and the diagonal normalization operator $M = \mathbf{E} A_i^T A_i$. For this derivation, this method considers A_i as a random variable which samples the true state X_i. In addition, it is convenient to define $A_i A_i^T$ as equal to $M + E_i$ where E_i is a mean zero diagonal matrix. It can be show that the operator norm of E_i tends to zero in the high-dimensional limit. Beginning again with the signal model:

$$Y_i = A_i X_i + \epsilon_i \tag{3.29}$$

One can then backproject the data and use $A_i A_i^T = M + E_i$

$$\begin{aligned} B_i &= A_i^T Y_i = A_i^T A_i X_i + A_i^T \epsilon_i \\ B_i &= (M + E_i) X_i + A_i^T \epsilon_i \\ B_i &= M X_i + E_i X_i + A_i^T \epsilon_i \end{aligned} \tag{3.30}$$

In the high-dimensional limit $E_i X_i \to 0$:

$$B_i \sim M X_i + A_i^T \epsilon_i \tag{3.31}$$

Finally, the distribution of $\tilde{B}_i$ is determined to be

$$\tilde{B}_i = M^{-1} A_i^T Y_i + M^{-1} A_i^T \epsilon_i \tag{3.32}$$

The observation of the back rojected data is just the true signal with some linear transformation of the noise. Based on this result, the procedure determine X_i is a singular value shrinkage scheme that would denoise the backprojected to perform the estimation. The step-by-step scheme is decribed in the following section.

3.7.3 EBLP Algorithm

The EBLP algorithm shown in [12] can be performed using the folllowing steps:

1. As input, this algorithm has the observations Y_i, which are just linearly transformed versions of the true state with additive noise. In addition, there is an input which is the estimated rank of the data matrix r and the measurement matrices A_i.

2. Form a backprojected data matrix B from the observations. This can be written as $[A_1^T Y_1 ... A_n^T Y_n]^T$. In additon calculate the diagonal normalization matrix $\hat{M} = n^{-\frac{1}{2}} \sum_{i=1}^{n} A_i^T A_i$. Following this step, normalize B by $\tilde{B} = B M^{-1}$.

3. From step(2), one uses the SVD to calculate singular values σ_k and the top r singular vectors $\hat{u}_k$ and $\hat{v}_k$ of the matrix $\tilde{B}$.

4. This algorithm then computes the estimated true data matrix $\hat{X} = [\hat{X}_1 ... \hat{X}_n]$. This is equal to $\sum_{k=1}^{r} \hat{\lambda}_k \hat{u}_k \hat{v}_k$ where $\hat{\lambda}_k$ is calculated by $\hat{l}_k^{\frac{1}{2}} \hat{c}_k \hat{\tilde{c}}_k$. This is done by the following:

$$\begin{aligned} \hat{l}_k &= \frac{1}{\hat{D}(\sigma_k)^2} \\ \hat{c}_k^2 &= \frac{\hat{m}(\sigma_k^2)}{\hat{D}'(\sigma_k^2)\hat{l}_k} \\ \hat{\tilde{c}}_k^2 &= \frac{\hat{\tilde{m}}(\sigma_k^2)}{\hat{D}'(\sigma_k^2)\hat{l}_k} \end{aligned} \tag{3.33}$$

Here $\hat{D}$ and $\hat{D}'$ are the D transforms and $\hat{m}$ and $\hat{\tilde{m}}$ are the Stieltjes-transform-like functionals defined in [12]. Given an approximate noise level and rank estimate, these transforms allow for estimation of optimal singular values that will phase out principal components that are associated with noise.

As this algorithm generates an estimator in a high-dimensional space, this thesis willl leverage the result and the algorithm in constructing the transformed spike covariance completion algorithm.

3.8 Overview

This section explores several options that solve the signal model above. Traditional methods like KF and SSA can perform robust estimation; however, they may prove ineffective in the face of high-dimensional estimation, model uncertainty, and partial data. This section then looks at other algorithms that involve the notion of matrix completion like TRMF and EBLP to perform denoising and imputation. Chapter 4 discusses TSCC, an algorithm that is motivated by some of the facets of each of the previous methods explained.

CHAPTER 4

TRANSFORMED SPIKED COVARIANCE COMPLETION

This chapter describes a new algorithm for high-dimensional signal estimation. This algorithm is non-parametric in the sense that it does not assume any functional form for the state transition of the time series. The signal model used is that of the state space model shown in equation 3.1. In addition, the signal also follows a spiked covariance model like that in the EBLP.

4.1 Algorithm Overview

Given Y_i, a series of observations, which are a noisy, transfomed version of the true state X_i, the goal is to find an estimate $\hat{X}_i$ from Y_i. This is framed as a matrix completion problem.

4.1.1 Trajectory Matrix Formation

Each measurement Y_i is concatenated vertically to form a time-lag embedding of length L to form a trajectory matrix Z. Effectively, each measurement Y_i is concatenated with the previous $L-1$ measurements and stacked into the columns of Z:

$$Z = \begin{bmatrix} Y_L & Y_{L+1} & ... & Y_J \\ Y_{L-1} & Y_L & ... & Y_{J-1} \\ ... & ... & ... & ... \\ Y_1 & Y_2 & ... & ... \end{bmatrix} \tag{4.1}$$

The trajectory matrix Z follows a block Toeplitz structure where the measurement vector Y_i is repeated along the diagonal. Z is an $NL \times (T-L+1)$ matrix where N is the dimension of the measurement, L is the number of lagged versions of Y_i, and T denotes the number of measurements. Now, one can define Q to be a matrix, with columns Q_i, to be the true trajectory

matrix of X. The formation of trajectory matrices here are one of many examples of time-lagged embeddings found in dyamical systems theory. By creating a structure that represents lagged versions of itself, the embedding can capture the various modes of the system being observed as shown in Taken's embedding theorem [3].

If in the case where the Y_i is a full observation, that is A_i, the sampling operator, is an identity matrix, then one can just use the traditional SSA technique to obtain a low rank approximation of Z with some shrinkage on the singular values. This can be written as

$$\hat{Q} = \sum_{k=1}^{r} \eta(\sigma_k) u_k v_k^\top \tag{4.2}$$

Here η denotes a shrinking operation on the singular values. One should note that $\hat{Q}$ follows a block Toeplitz structure. In order to get a final estimate on X, this algorithm performs diagonal averaging on the diagonals of the matrix.

$$\hat{X}_i = \frac{1}{L} \sum_{j,k} \hat{Q}_{j,k} \quad \text{where} \quad k - j = i \tag{4.3}$$

Here, we take the sum over the j^{th} lagged vector at time k for $0 \leq j \leq L-1$ and $0 \leq k \leq T - L$.

Thus far, this describes a multivariate SSA method for estimation. In the signal model of interest in this thesis, A_i, the sampling operator, may not be an identity. In this technique, A_i is taken as a diagonal matrix of ones or zeros for whether or not the i^{th} feature of the state was observed. If one uses the above technique, SSA will not provide a good estimate of X due to the missing entries. The following section describes the statistcal aspect to TSCC that will handle the missing entries.

4.1.2 Linear Estimation with Spiked Covariance Model

This technique utilizes a signal model in which the observation Y_i is a linear transformation of the true state $A_i X_i$ with additive white Gaussian noise ϵ_i. Given this model, one can write an estimator like the EBLP in [12] to denoise and fill in the entries of trajectory matrix Z.

Effectively, this method would perform imputation on the missing entries of Z. First, for notation, $\tilde{A}_i$ is the truncation matrix for each column of Z. $\tilde{A}_i$ is of dimension $NL \times NL$ with diagonal values of either one or zero. The additive white Gaussian noise for each column Z_i is denoted by $\tilde{\epsilon}_i$. In the same spirit as the proof in section 3.7, this algorithm defines the diagonal normalization matrix as

$$M = \frac{1}{T-L+1} \sum_{i=1}^{T-L+1} \tilde{A}_i \tilde{A}_i^\top \tag{4.4}$$

Essentially, the signal model can be rewritten as:

$$Z_i = \tilde{A}_i Q_i + \tilde{\epsilon}_i \tag{4.5}$$

Similar to the derivation of the EBLP, one can work with the backprojected data $\tilde{A}_i Z_i$.

$$B_i = \tilde{A}_i^\top Z_i = \tilde{A}_i^T \tilde{A}_i Q_i + \tilde{A}_i^T \tilde{\epsilon}_i \tag{4.6}$$

This method writes $\tilde{A}_i \tilde{A}_i^T = M + E_i$. Here E_i is the deviation of $\tilde{A}_i \tilde{A}_i^T$ from the ensemble mean from M. In the high-dimensional limit, i.e., $NL \to \infty$, $T - L + 1 \to \infty$, and $\frac{NL}{T-L+1} \to \gamma$, the operator norm of the matrix with rows $\frac{E_i Q_i}{\sqrt{T-L+1}}$ vanishes. The backprojected data B_i becomes

$$B_i = \tilde{A}_i^\top Z_i = MQ_i + E_i Q_i + \tilde{A}_i^\top \tilde{\epsilon}_i \sim MQ_i + \tilde{A}_i^\top \tilde{\epsilon}_i \tag{4.7}$$

Because M is full rank with high probability, M has inverse M^{-1}. It naturally follows that

$$\tilde{B}_i = M^{-1} B_i \sim Q_i + M^{-1} \tilde{A}_i^\top \tilde{\epsilon}_i \tag{4.8}$$

It is now clear that the backprojected data is just the true signal contaminated by colored noise $M^{-1} \tilde{A}_i^T \tilde{\epsilon}_i$. One uses the empircal best linear estimatior in [12] to estimate Q_i from $\tilde{B}_i$. This linear prediction is known in signal processing as the Wiener filter. This algorithm then estimates the singular vectors u_k and v_k and singular values σ_k by taking an SVD of $\tilde{B}$. Following SVD, σ_k is shrunk using the plug-in equations 3.33 to find the optimal singular values $\hat{\sigma}_k$ from singular value shrinkage using the Marčenko Pastur distribution and random matrix theory.

Finally, this algorithm truncates and shrinks the singular values of $\tilde{B}$ using

random matrix theory and generalized Marčenko Pastur distribution [11]. The singular values after shrinkage are denoted by λ_k and the estimated Q is

$$\hat{Q} = \sum_{k=1}^{r} \lambda_k u_k v_k^\top \tag{4.9}$$

This algorithm obtains an estimate of X_i by diagonal averaging $\hat{Q}$. This process is summarized in the following pseudocode.

Result: $\hat{X}$
Input : $Y \in \mathbb{R}^{N \times T}$, Measurements;
Input : Lag parameter L;
Input: Rank Estimate R;
Input: Noise Level ϵ;
Input: Measurment Indicator $\bar{I} \in \mathbb{R}^{N \times T}$
Allocate $Z \in \mathbb{R}^{LN \times T+L-1}$;
Allocate $\hat{I} \in \mathbb{R}^{LN \times T+L-1}$;
for *t in T+L-1* **do**
    LaggedMeasurement = $vec(Y_{t:t+L-1})$;
    Z_t = LaggedMeasurement;
    LaggedIndicator = $vec(\bar{I}_{t:t+L-1})$;
    $\hat{I}_t$ = LaggedIndicator
end
Allocate $\hat{M}_i \in \mathbb{R}^{LN \times LN}$ **for** *t in T+L-1* **do**
    LaggedI = $vec(\bar{I}_{t:t+L-1})$;
    $\hat{M}_i = \hat{M}_i + diag(LaggedI)$;
end
$\hat{M}_i = \hat{M}_i/(T + L - 1)$;
$\bar{Z} = M_i^{-1} Z$ //Back project data;
Estimate Q from $\bar{Z}$ using EBLP ;
Estimate $\hat{X}$ by Diagonally Averaging Q;

Algorithm 1: TSCC

4.2 Computational Complexity

In addition to higher accuracy, the TSCC technique can be a faster algorithm when compared to TRMF. The complexity for TSCC is $O(\min(LN, T-L+1)^2 \times \max(LN, T-L+1))$. Like in the signal model, N is the dimension of the state vector, T is the number of measurements, and L is lag parameter for the trajectory matrix. This complexity comes mainly from the fact that EBLP performs an SVD with other $O(NT)$ calculations in generating the shrinkage coefficients.

TRMF is an iterative algorithm whose single update complexity for each iteration is $O(NTk^2 + L(T-L+1)k^2 + (L^3 + TL^2))$. Here the variables k and L respectively represent the latent dimension and the lag parameter (of the autoregressive model). Note that the complexity is the sum of three terms. TRMF decomposes a data matrix to two matrices that are constrained by an optimization problem that minimizes the Frobenius norm error between the observations and the approximation and the model error [10]. It is important to note that as the dimensions of the matrix increases, the number of iterations increases.

Since TRMF assumes a model for the dynamical system, to properly find its factorization, one would need to search through various model parameters, namely k and L to best fit the observations. TSCC, a non-iterative algorithm, can potentially have a better complexity compared to TRMF. The TSCC complexity is dominated by a SVD operation; new algorithms like the one presented in [13] show that a randomized SVD for the low-rank approximation is linear with $O(lMN)$ where l is slightly larger than rank r of the matrix. Potentially, the incorporation of this algorithm into the current TSCC framework can lead to a complexity that is faster than TRMF.

4.3 Advantages

The design of this algorithm was based on several constraints involving signal dimensionality, robustness to high noise levels, and ability to handle missing data. Traditonally the Kalman filter or ensemble Kalman filter would have been the algorithm of choice; however, this would have required an accurate knowledge of the state transition, a luxury unknown in many

applications. SSA is another alternative traditional technique that is non-parametric, meaning there is no model for the state dynamics. SSA simply embeds time-lagged samples and performs a low-rank approximation to estimate the signal. This technique can be expensive in high dimensions as performing an SVD is an $O(\min(M, N))^2$ operation. This algorithm may do a poor job in gap filling if the singular values and vectors are inaccurately estimated. TRMF is another method that is considered the state-of-the-art for the high-dimensional time series estimation problem; however, TRMF may not be suitable in applications where the dynamics can be captured well by an AR model.

TSCC is an algorithm that fixes many of the shortcomings in the algorithms presented in Chapter 3. TSCC does not rely on any specific parametric model. TSCC does not require any priors on the state dynamics. The algorithm essentially captures the dynamics in an embedding, such that the resulting representation still has the qualities of the spiked covariance model, allowing the use of the EBLP to estimate the signal. TSCC differs from direct usage of EBLP, as that method does not assume any prior structure on the estimated matrix nor does it assume that the signal is a time series. As shown in the section 4.2, the TSCC algorithm can be made more computationally efficient with methods like randomized SVD.

CHAPTER 5

NUMERICAL EXPERIMENTS

5.1 Experimental Overview

This chapter presents three experiments that demonstrate the performance of TSCC over the other algorithms described previously, namely the standard SSA, TRMF, and EBLP. Each of the described algorithms produces an output, the estimated state of the system, given noisy, partially observed data. The input in each experiment is a dataset that is noisy and has missing values. The resulting output of each algorithm is evaluated on either the accuracy of the estimate or by examining the physicality of the result.

Experiment I uses synthetic data generated by using an autoregrressive model with a control of the noise level and the number of missing values. Experiment II uses traffic sensor data collected in San Francisco. In this case, there is no control on the noise but a control on the number of missing values. Both experiments I and II have a ground truth in which the results of each estimation method can be compared against numerically. Accuracy is evaluated by using the normalized Frobenius norm error or the Root Mean Square Error (RMSE) between the estimate and the ground truth given by

$$Err = \frac{||X - \hat{X}||_F}{||X||_F} \tag{5.1}$$

Experiment III uses data from NASA's Sounding of the Atmosphere using Boradband Emission Radiometery (SABER). The quantity that this experiment examines is the derived hydrogen density derived from measurements taken from the Thermosphere Ionosphere Mesophere Energetics and Dynamics (TIMED) spacecraft. This experiment will evaluate the physicality of the data imputation results across the algorithms described.

5.2 Experiment I: Synthetic Data

Experiment I compares four methods. The first method is TRMF as seen in [10]. TRMF performs time series estimation via matrix completion under the autoregressive assumption. The second method is direct estimation of matrix M with EBLP only. The third technique is the TSCC technique detailed in Chapter 4. The fourth technique is the standard SSA technique [5].

The input to each technique is a noisy data matrix Y generated from the autoregression in equation 3.17. For TRMF, the number of iterations is set to 10, the point where the algorithm converges. The simulated data has 50 state features and 250 time samples for N and T respectively. The input data is created by first generating 100 clean data matrices (no noise). To test various levels of noise, Gaussian noise vectors with mean 0 and standard deviations of 0.1, 0.3, 0.5, 0.8 and 1.0 are added to the clean matrix. In total, with 100 unique clean matrices with six noise levels (noise deviations), there are a total of 600 input matrices tagged by a unique level of noise and a unique clean matrix. To simulate missing data, 20% of the entries of each column of Y are randomly set to zero. Each of the mentioned algorithms evaluates all 600 matrices.

Figure 5.1 displays the algorithm output with the unique clean matrix 0 at the six levels of noise deviation. Each column represents (excluding the Clean Data column) the output of each algorithm at a given noise level. In each panel, the output contains two dimensions, a horizontal dimension representing time samples and a vertical dimension representing features.

The error is calculated for each method by averaging the RMSE error of each of the mentioned methods at each given noise deviation over each unique clean data matrix. Figure 5.2 displays the average error as a function of the input noise level. As expected, error increases with increasing noise. The TSCC method error curve is below each of the error curves for all noise levels.

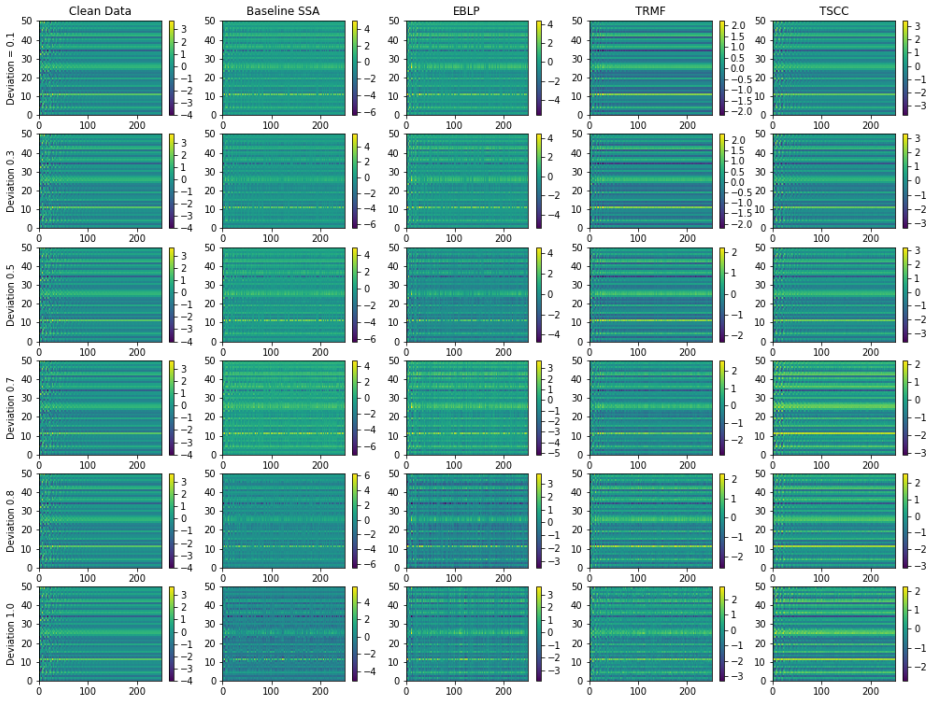

Figure 5.1: Experiment I - The output of each algorithm for the synthetic data at each level of noise deviation. Each column represents (excluding the Clean Data column) the output of each algorithm at a given noise level. For all cases, TSCC accurately reconstructs the original dataset in comparision to SSA, EBLP and TRMF as they all contain vertical striations

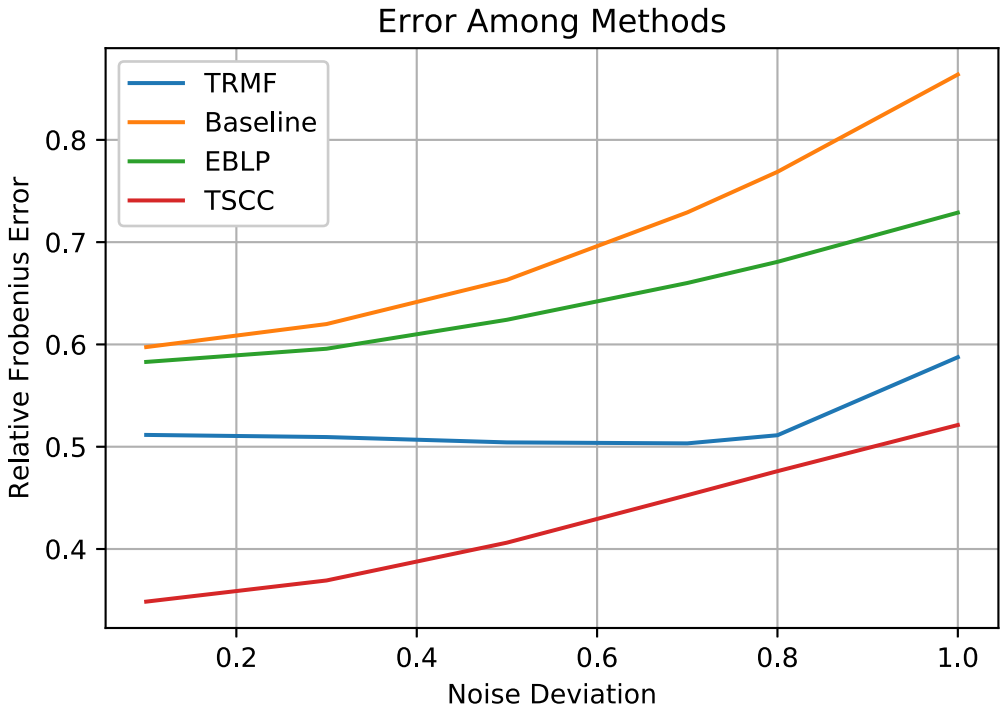

Figure 5.2: average error over all noise levels - The RMSE between estimate and ground truth are averaged over all 100 unique clean data matrices at a given noise level.

The TRMF and the TSCC methods capture the decaying oscillatory behavior of the clean data. The EBLP method by itself is not robust to missing data for this example as shown with the striation through feature dimenions of the matrix. These striations are also seen in the vertical direction of the TRMF subplots in figure 5.1. TSCC performs better than EBLP by reforming Y as a trajectory matrix. TSCC then has increased estimation accuracy due the mode capture capability of the embedding.

In these experiments, the average compute time was calculated by dividing the time taken to estimate the matrices over the number of data matrices. The average compute time was 294 ms for TRMF, 56 ms for EBLP, and 64 ms for TSCC on a 3.3 GHz Intel i7 processor. Experimentally, TSCC performs faster than the state-of-the-art TRMF.

5.3 Experiment II: San Francisco Traffic Data

The San Francisco Traffic Data set taken from the experiments in [10] consists of the averaged hourly wait time of a car idling at stoplights at 50 different (dispersed) sensors in the greater San Francisco area. A partition of the original dataset is seen in figure 5.3. This shows a matrix with the vertical and horizontal axes representing the spatial position of sensors and time sample index respectively. Each pixel in the matrix is a normalized wait time; brighter colors represent a longer wait time and darker colors represent short wait times. The periodic nature of weekday traffic is captured in hourly time samples 125 to 250 where there is an alternation between bright and dark pixels indicating rush hour and nighttime traffic. Hourly time samples 250 to 300 show relatively lighter pixel intensities indicating the lighter traffic during the weekend.

Experiment II tests the imputation capability of each algorithm. Data is removed by simulating a sensor blackout where 20% of the data in each column is set to 0 at a single hourly time sample index. For simplicity of data removal, there is also an additional constraint that the removal of data must be continguous (i.e. sensors 1, 2, 3, 4, 5 are removed from a total of 50 sensors). The removal of data is seen in figure 5.4. The inputs into each algorithm are the artificially incomplete dataset and masking matrix, a matrix indicating where and when a sensor blackout (data removal) took place. The accuracy output of each estimation algorithm is evaluated by the Root Mean Square Error (RMSE) defined in equation 5.1.

TSCC has the highest reconstruction among all four algorithms. The increase in accuracy between TSCC and EBLP is due to EBLP not accounting for spatiotemporal correlations in the data. The baseline SSA output in figure 5.5 does inpute data at a high fidelity due to improper estimation of singular vectors as discussed in section 3.8. TRMF recovers the general temporal structure of the traffic pattern as shown in figure 5.6. There is a significant mismatch in relative intensity values at time samples 0 through 30 due to poor initial condition estimation. TSCC outperforms TRMF due to this initialization failure in addition to the poor model assumption as TRMF assumes the dynamics are autoregressive. TSCC also outperforms EBLP as shown in figure 5.7 as TSCC incoprorates a time-lagged embedding. Figure 5.8 shows that TSCC has the smallest reconstruction error, performing

slightly better than EBLP.

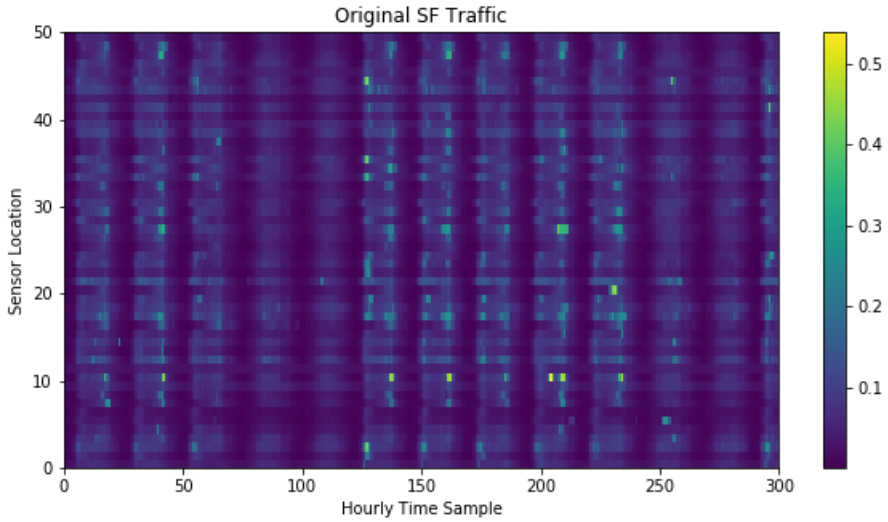

Figure 5.3: Original traffic data [10] - vertical axis shows spatial position of sensors. The horizontal axis represents hourly time sample indices. The pixel intensities indicate the level of traffic intensity, brighter color showing greater traffic and darker colors meaning lighter traffic

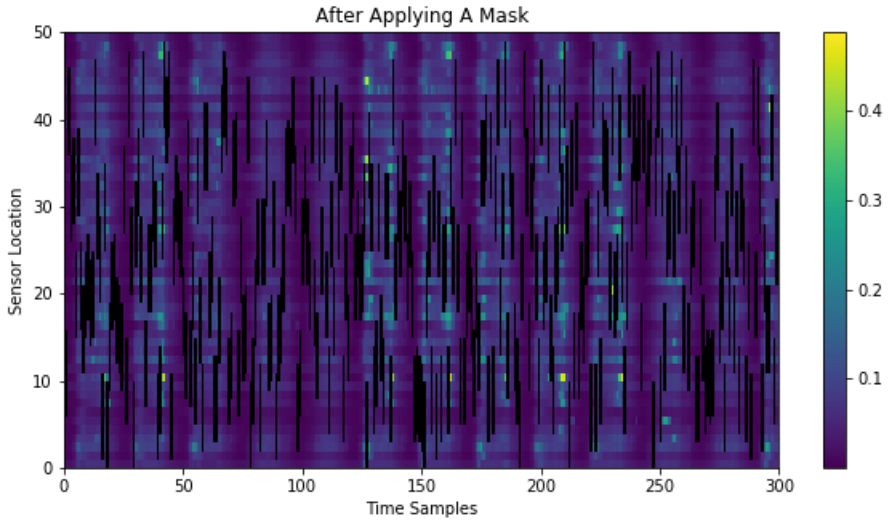

Figure 5.4: 20% of all traffic data is randomly removed at each hourly time sample by setting continguous sensor locations to 0. This is seen in the dark vertical stripes.

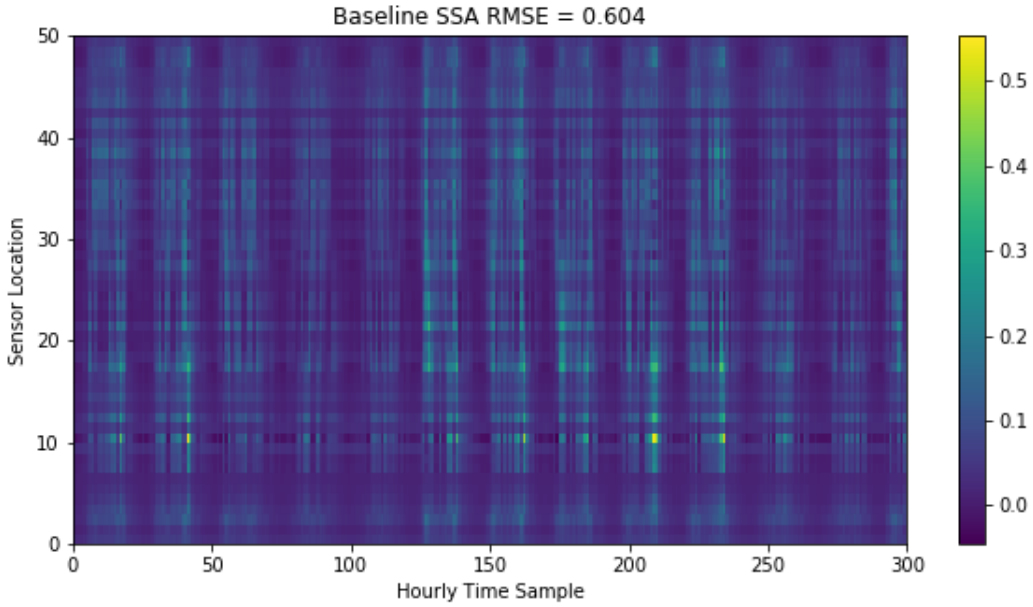

Figure 5.5: The output of the baseline SSA method has an RMSE of 0.604. The vertical striations are artifacts of the missing data.

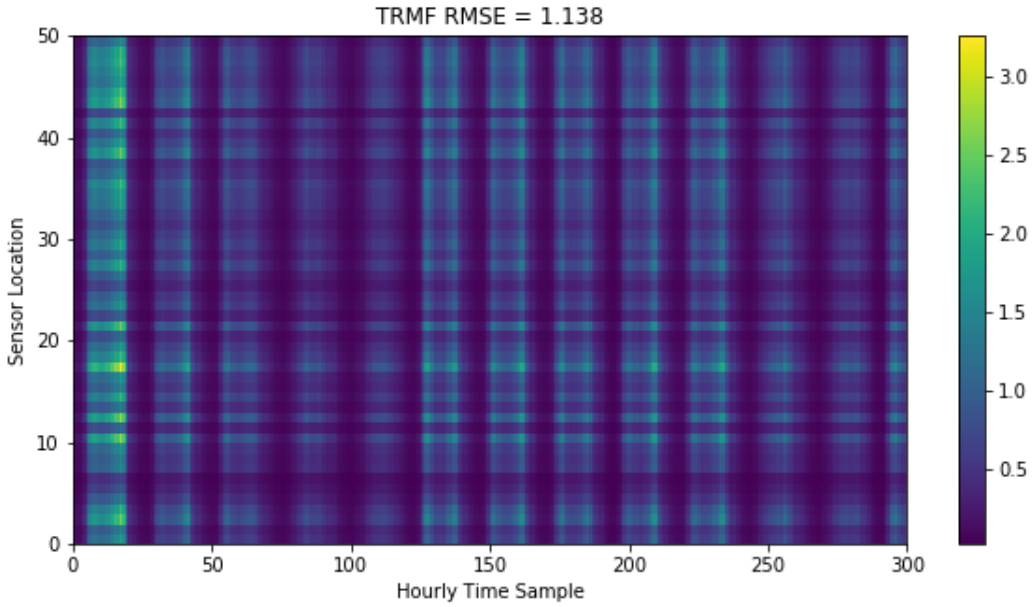

Figure 5.6: The output of TRMF has an RMSE of 1.138. The result captures the general periodicity of the traffic. The pixel intensities are much brighter at samples 0 through 30 when compared to the true data.

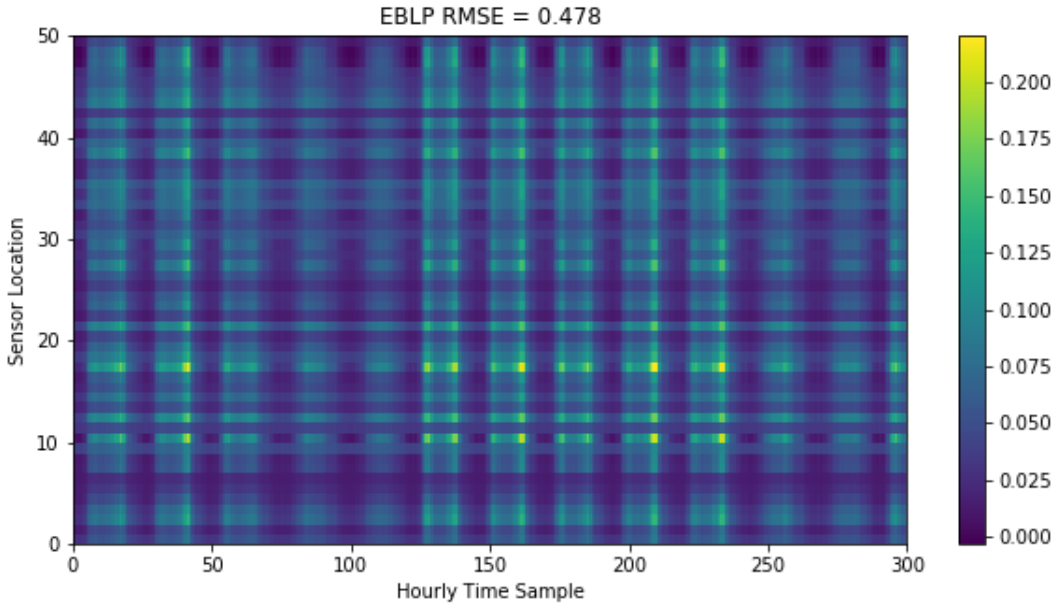

Figure 5.7: The output of EBLP has an RMSE of 0.478. The periodicity of the traffic pattern and relative intensity visually matches the original traffic data.

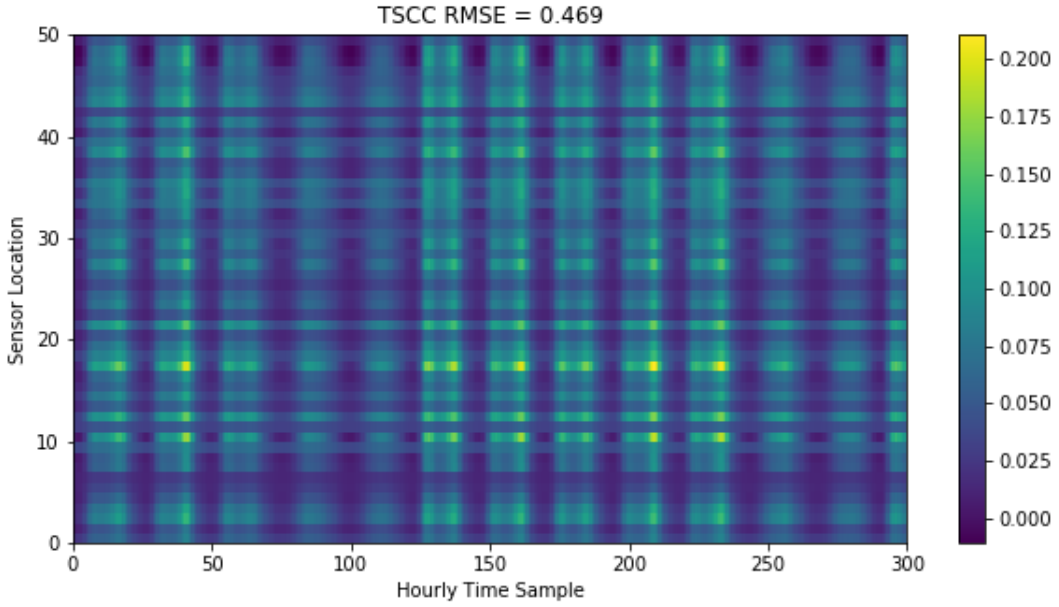

Figure 5.8: The output of TSCC has an RMSE of 0.469. The performance of TSCC is slightly better than EBLP most likely due to the incorporation of a time-lagged embedding.

5.4 Experiment III: SABER H-Density

The Sounding of Atmsophere using Broadband Emission Radiometry (SABER) is an imaging system onboard NASA's Thermosphere Ionosphere Mesosphere Energetic and Dynamics (TIMED) spacecraft. This instrument collects the infrared emission from Earth's troposphere and stratosphere. The infrared emissions are used to derive atomic hydrogen density in Earth's mesosphere near the 85 Km altitude. This dataset examines hydrogen densities taken over 60 latitudinal bins. This data is taken over 20 days or 300 orbits, where each orbit is approximately 90 minutes long. Similar to the traffic data in experiment II, the vertical dimension of this SABER dataset as shown in figure 5.9 encodes spatial information while the horizontal dimension represents time.

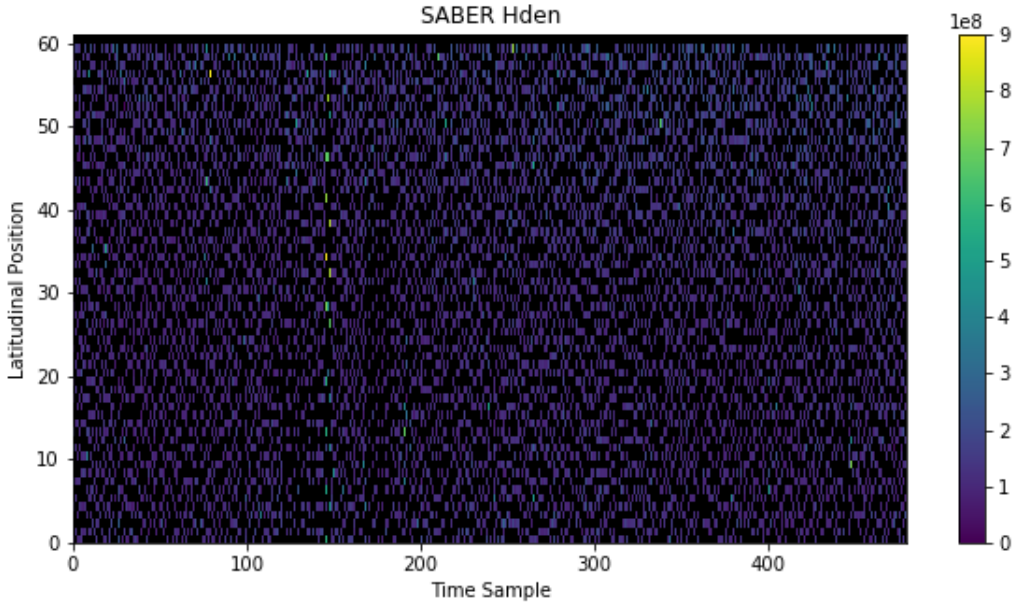

Figure 5.9: SABER hydrogen density is depicted as a matrix which has vertical dimensions representing latitudinal position and the horizontal dimension showing time sample indices. The intensity of each pixel represents hydrogen density in atoms per cubic centimeter. Black pixels represent time and latitudinal positions where there is missing data

In this case, time represent hourly samples the hydrogen density at a 85 Km latitude. The missing values in this dataset represents instances in time where no data is available due to the sampling periodicity of the spacecraft.

The other missing values are due to sensor non-idealities such as having a star in the line of sight. Unlike the traffic data, this atmospheric density dataset does not have a ground truth so evaluation of accuracy cannot be evaluated as shown in the RMSE equation 5.1. This experiment evaluates the fidelity of each algorithm's estimate by the physicality of the output (non-negative values, no growing oscillatory behavior, etc.)

The algorithms that are applied to this dataset are interpolation, autoregressive model (AR), autoregressive moving average (ARMA) model, and TSCC. In this experiment, the interpolation, AR, and ARMA methods are used instead SSA, EBLP, and TRMF as the former methods are generally accepted techniques used in data imputation in the remote sensing community. The interpolation, AR, and ARMA methods are all applied on the rows in matrix shown in figure 5.9; their implementations are a standard commonly found in statistical signal processing textbooks. The input into each of these algorithms is the dataset shown in figure 5.9.

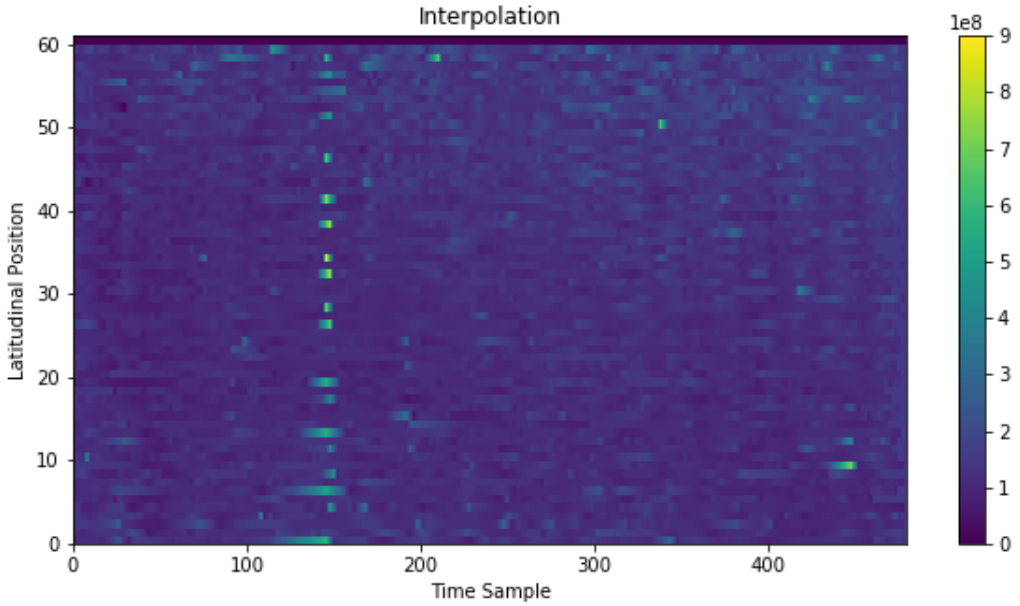

Figure 5.10: Application of the interpolation method fills in the data gaps but induces blurring within the image. This blurring may suggest a time localized disturbance that may not actually be occur.

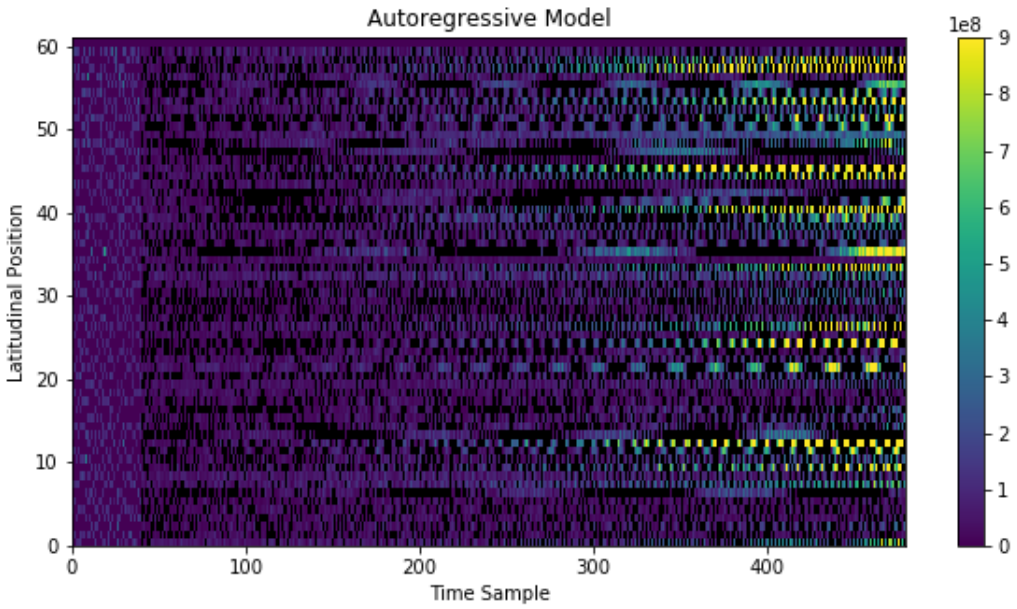

Figure 5.11: Application of an autoregressive model creates an output that has growing oscillatory behavior in the horizontal. The exponential growth of the hydrogen density through time is not physical.

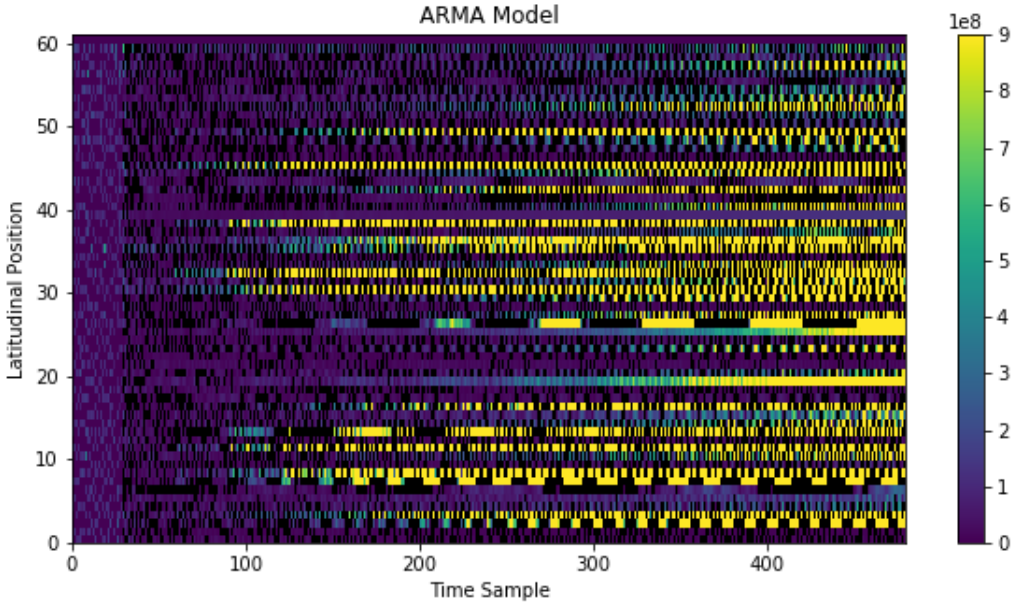

Figure 5.12: Similar to the AR model case, the ARMA output also has this growing oscillatory behavior which is not physical.

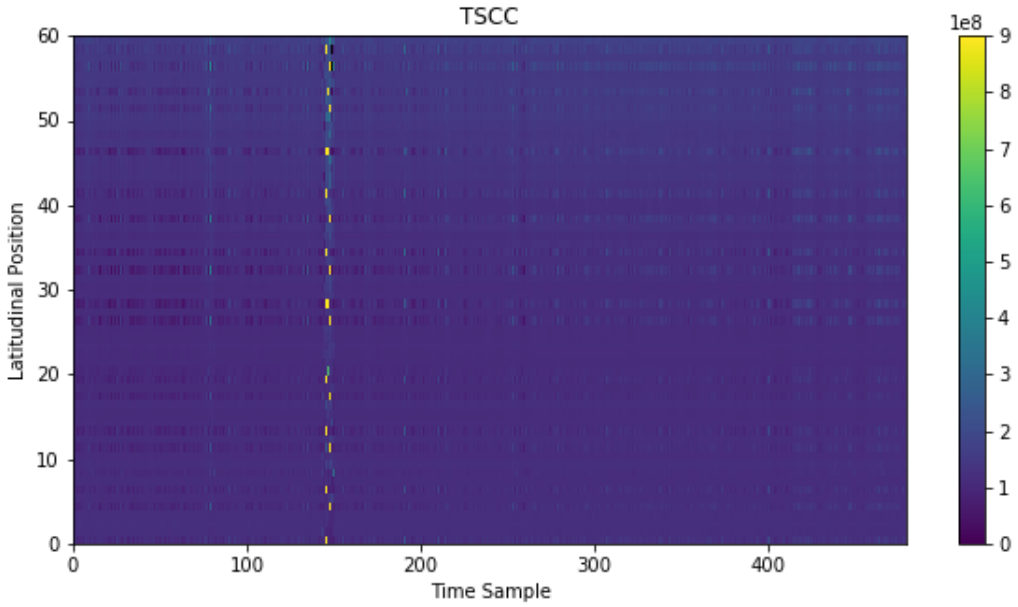

Figure 5.13: The output of TSCC fills in all missing values completely with hydrogen densities on the order of 10^9 atoms per cubic centimeters which is expected at 85 Km.

Applying interpolation on the rows of the matrix in figure 5.9 is equivalent to applying a low-pass filter on the rows in the matrix. This results in the blurring effect seen in the interpolation SABER H density results in figure 5.10. Though this result produces a physical density value, the density is smeared across time. This smearing contradicts the observed data in figure 5.9 and may suggest some localized disturbance which may not actually be occuring. In both the AR and ARMA cases, the polynomial order was set to 30 and the forecast (extrapolation) was executed from the first input 30 samples of figure 5.9. In both cases, the autoregressive coefficients learned lead to the models that produce unphysical hydrogen densities as seen in the AR and ARMA model outputs. In both figures 5.11 and 5.12 , oscillatory behavior occurs growing to density values that are not physically realizable. The unphysical reconstruction of both models stem from the learning of the AR and ARMA coefficients. The AR and ARMA models learn from sequences with large gaps of missing data which leads to learning coefficients that induce exponential growth. In the TSCC result, the inputed data values are within the same range as the values of the observed hydrogen densities in figure 5.9. Unlike interpolation, the TSCC result does not have a smearing

of the densities in time. The low-rank approximation in TSCC suggests that the data encodes certain dynamical modes. The output in figure 5.13 shows these repeated modes in the columns of the output matrix. Based on the results of each algorithm, TSCC produces an output which does not produce unphysical features or extreme behavior in the estimate. These results demonstrate that TSCC produces an output that seems natural when compared against the original input dataset.

CHAPTER 6

CONCLUSION AND FUTURE WORK

For the problem of estimating the true, underlying signal from observations of a dynamical system, this thesis presents TSCC, a method that reconstructs a dynamical time series from noisy, partial measurements. This method is evaluated against standard and state-of-the-art techniques through three numerical experiments. The results demonstrate that TSCC is more robust to noise and missing data in comparision to the accepted algorithms that address the same problem. In addition, runtime analysis shows that TSCC also has better computational tractability in comparison to the other mentioned methods. The main advantage of this technique is that multivariate time series can be represented as an embedding whose noise covariance follows a spiked covariance model, allowing usage of a low-rank linear estimator that is effective at both denoising and matrix completion.

6.1 TSCC Performance and Limitations

The design of TSCC was based on two limiting factors in the estimation problem. The first factor is the size of the state dimension, and the second factor is the lack of prior knowledge of a system's dynamical model. Because of the large size of datasets in application domains such space-based imaging, traditional methods like SSA and the Kalman filter are not computationally tractable due to the storage and inversion of large matrices. Further, when a dynamical model is not known, the traditional methods like the textbook Kalman filter and TRMF, cannot be used because the methods require strong assumptions on the dynamical model. TSCC generalizes the dynamics by converting the time series estimation problem to a matrix completion problem and leverages recent results in the spiked covariance model to perform matrix imputation and estimation. TSCC accounts for the model

uncertainities by using a time-lagged embedding to characterize the modes of the system instead of following strict parametric model assumptions such as in TRMF.

This method is evaluated on several datasets that are both synthetic and real. These results show that TSCC outperforms the accepted time series estimation frameworks in terms of reconstruction accuracy. As shown in experiment I, the generated synthetic data follows an autoregressive model. In this case, TSCC still outperforms TRMF and all other methods though these methods were designed with the assumptions that the dynamics follow an autoregressive model. This experiment demonstrates that TSCC is able to generalize linear dynamical models. In addition, for experiments II and III where real datasets representing the dynamics in vehicle traffic and exospheric hydrogen are used, TSCC shows that the values it imputes are physically reasonable and do not induce unnatural blurring or exponential oscillatory behavior in comparison to other methods.

6.2 Future Work

The TSCC algorithm can be extended in several ways to improve performance both in computation and in flexibility for a greater range of dynamical systems. Currently, in the estimation of the true trajectory matrix, empirical best linear prediction is applied to a partially observed trajectory matrix. Note that the true trajectory matrix has a Toeplitz structure. For the implementation of TSCC in this thesis, the estimated trajectory does not necessarily hold the Toeplitz structure. Regularization technqiues can be added in a future implementation of TSCC to mantain this Toeplitz structure. In addition, another possible extension of TSCC lies in the treatment of the structure of the latent embedding. Currently, the embedding is just a time-lagged embedding. It is possible to apply kernel functions on these embeddings to better account for non-linear dynamical systems. The current implmentation of TSCC is based on linear models. Future work can generalize this linear assumption.

REFERENCES

[1] M. D. Butala, R. A. Frazin, Y. Chen, and F. Kamalabadi, "Tomographic imaging of dynamic objects with the ensemble Kalman filter," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1573–1587, July 2009.

[2] J.-F. Cai, E. J. Candes, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *ArXiv e-prints*, Oct. 2008.

[3] F. Takens, "Detecting strange attractors in turbulence," *Lecture Notes in Mathematics, Berlin Springer Verlag*, vol. 898, p. 366, 1981.

[4] T. Sauer, J. Yorke, and M. Casdagli, "Embedology," *Journal of Statistical Physics*, vol. 65, no. 3, pp. 579–616, Nov 1991.

[5] G. Tsagkatakis, B. Beferull-Lozano, and P. Tsakalides, "Singular spectrum-based matrix completion for time series recovery and rrediction," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 66, May 2016. [Online]. Available: https://doi.org/10.1186/s13634-016-0360-0

[6] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix Completion from a Few Entries," *ArXiv e-prints*, Jan. 2009.

[7] N. B. Shah, S. Balakrishnan, and M. J. Wainwright, "Low Permutation-rank Matrices: Structural Properties and Noisy Completion," *ArXiv e-prints*, Aug. 2017.

[8] J. Andén, E. Katsevich, and A. Singer, "Covariance Estimation Using Conjugate Gradient for 3D Classification in Cryo-EM," *ArXiv e-prints*, Dec. 2014.

[9] E. J. Candes and Y. Plan, "Matrix Completion with Noise," *ArXiv e-prints*, Mar. 2009.

[10] H.-F. Yu, N. Rao, and I. S. Dhillon, "High-dimensional Time Series Prediction with Missing Values," *ArXiv e-prints*, Sep. 2015.

[11] F. Götze and A. Tikhomirov, "On the Rate of Convergence to the Marchenko–Pastur Distribution," *ArXiv e-prints*, Oct. 2011.

[12] E. Dobriban, W. Leeb, and A. Singer, "Optimal Prediction in the Linearly Transformed Spiked Model," *ArXiv e-prints*, Sep. 2017.

[13] N. Halko, P. Martinsson, Y. Shkolnisky, and M. Tygert, "An algorithm for the principal component analysis of large data sets," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2580–2594, 2011.