SIGNAL PROCESSING ON GRAPHS: ESTIMATING THE STRUCTURE OF A GRAPH

Jonathan Mei and José M.F. Moura

Carnegie Mellon University
Department of Electrical and Computer Engineering
Pittsburgh, PA 15213

ABSTRACT

This paper presents a computationally tractable algorithm for estimating the graph structure of graph signals is presented. The algorithm is demonstrated on simulated and real network time series datasets, and the performance of the new method is compared to that of related methods for estimating graph structure. The adjacency matrices estimated using the new method are shown to be close to the true graph in the simulated data and consistent with prior physical knowledge in the real dataset.

Index Terms— Graph Signal Processing, Graph Structure, Adjacency Matrix, Network, Time Series

1. INTRODUCTION

Networked data are everywhere, from finance and social media to geology and biology. Interest in studying such networks is growing rapidly as data are generated at increasing rates. While there are many methods for processing signals that come from networks (see [1] for an overview of existing techniques), often times, the graph structure underlying such data is not known a priori and needs to be determined before further analysis is performed.

In this paper we estimate network structure for time-series data in the form of a directed, weighted adjacency matrix $\mathbf{A}$. Current work on estimating network structure largely associates graph structure with the Markovian property of the processes supported by the graph [2]. Our work instead associates the graph with causal network effects, taking inspiration from the Discrete Signal Processing on Graphs (DSP_G) framework [3, 4].

We first provide a brief overview of related prior work as well as relevant concepts and notations in section 2. Then we introduce our new network process in section 3. Next, we present algorithms to infer the network structure from data generated by such processes in section 4. Finally, we show simulation results in section 5.

This work is partially supported by AFOSR grant #FA95501210087.

2. RELATION TO PRIOR WORK

2.1. Discrete Signal Processing on Graphs

DSP_Gprovides a framework with which to analyze data with N elements for which relational information between elements is known.

2.1.1. Graph Signals

Consider a graph $G = (\mathcal{V}, \mathbf{A})$ where the vertex set $\mathcal{V} = \{v_0, \ldots, v_{N-1}\}$ and $\mathbf{A}$ is the weighted adjacency matrix of the graph. Each data element corresponds to a node v_n, and the weight $\mathbf{A}_{n,m}$ is assigned to a directed edge from v_m to v_n. A graph signal can be written as N length vectors supported on $\mathcal{V}$,

$$\mathbf{x} = \begin{pmatrix} x_0 & x_1 & \ldots & x_{N-1} \end{pmatrix}^T \in \mathbb{C}^N$$

2.1.2. Graph Filters

A graph filter is a system $\mathbf{H}(\cdot)$ that takes a graph signal $\mathbf{x}$ as input and outputs another graph signal $\widetilde{\mathbf{x}} = \mathbf{H}(\mathbf{x})$. A basic nontrivial graph filter on graph $G = (\mathcal{V}, \mathbf{A})$ called the graph shift is a local operation given by the product of the input signal with the adjacency matrix $\widetilde{\mathbf{x}} = \mathbf{A}\mathbf{x}$. Graph filters in DSP_G are matrix polynomials of the form

$$h(\mathbf{A}) = h_0\mathbf{I} + h_1\mathbf{A} + \ldots + h_L\mathbf{A}^L$$

The output of the filter is $\widetilde{\mathbf{x}} = \mathbf{H}(\mathbf{x}) = h(\mathbf{A})\mathbf{x}$. Note that graph filters are linear shift-invariant filters. For a linear combination of inputs they output the same linear combination of outputs, and consecutive application of multiple graph filters does not depend on the order of application (i.e., graph filters commute). Graph filters also have at most $L \leq N_\mathbf{A}$ taps h_ℓ, where $N_\mathbf{A} = \deg m_\mathbf{A}(z)$ is the degree of the minimal polynomial $m_\mathbf{A}(z)$ of $\mathbf{A}$.

2.2. Sparse Vector Autoregressive Estimation

For time series data, sparse vector autoregressive (SVAR) estimation [5, 6, 7] recovers matrix coefficients for multivariate

processes. This problem assumes given data matrix of the form

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}[0] & \mathbf{x}[1] & \dots & \mathbf{x}[K-1] \end{pmatrix} \in \mathbb{R}^{N \times K}, \quad (1)$$

which is generated by the dynamic equation with sparse evolution matrices $\mathbf{A}^{(i)}$,

$$\mathbf{x}[k] = \sum_{i=1}^{M} \mathbf{A}^{(i)} \mathbf{x}[k-i] + \mathbf{w}[k]$$

where $\mathbf{w}[i]$ is a random noise process that is generated independently from $\mathbf{w}[j]$ for all $i \neq j$, and $\mathbf{A}^{(i)}$ all have the same sparse structure. That is, there is some sparse structure matrix $\mathbf{A}' \in \{0,1\}^{N \times N}$ such that $\mathbf{A}'_{ij} = 0 \Rightarrow \mathbf{A}^{(k)}_{ij} = 0$ for all k. Then SVAR solves the optimization,

$$\{\widehat{\mathbf{A}}^{(i)}\} = \underset{\{\mathbf{A}^{(i)}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=M}^{K-1} \left\| \mathbf{x}[k] - \sum_{i=1}^{M} \mathbf{A}^{(i)} \mathbf{x}[k-i] \right\|_2^2 + \lambda \sum_{i,j} \|\mathbf{A}_{ij}\|_2 \quad (2)$$

where $\mathbf{A}_{ij} = \begin{pmatrix} A^{(1)}_{ij} & \dots & A^{(M)}_{ij} \end{pmatrix}^T$ and the term $\|\mathbf{A}_{ij}\|_2$ promotes sparsity of the (i,j)-th entry of each $\mathbf{A}^{(i)}$ matrix simultaneously. This optimization can be solved using Group Lasso [8].

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 $\mathbf{A}'$. SVAR estimation methods estimate multiple weighted graphs $\mathbf{A}^{(i)}$ with the same sparsity structure as $\mathbf{A}'$ that can be more challenging to interpret and analyze than a single weighted graph. Also, it is not clear that the Markov property holds between nodes for various applications that can be described by graph processes. Our model produces a single weighted graph and allows estimation of non-Markov processes.

3. CAUSAL GRAPH PROCESSES

Here we present our new model for graph processes. Consider $x_n[k]$, a discrete time series on node v_n in graph $G = (\mathcal{V}, \mathbf{A})$, 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) to be a discrete time series $\mathbf{x}[k]$ on a graph $G = (\mathcal{V}, \mathbf{A})$ of the following form:

$$\begin{aligned} \mathbf{x}[k] =& \mathbf{w}[k] + \sum_{i=1}^{M} P_i(\mathbf{A}) \mathbf{x}[k-i] \\ =& \mathbf{w}[k] + \sum_{i=1}^{M} \left(\sum_{j=0}^{i} c_{ij} \mathbf{A}^j \right) \mathbf{x}[k-i] \\ =& \mathbf{w}[k] + (c_{10}\mathbf{I} + c_{11}\mathbf{A})\mathbf{x}[k-1] \\ &+ \left(c_{20}\mathbf{I} + c_{21}\mathbf{A} + c_{22}\mathbf{A}^2\right)\mathbf{x}[k-2] + \dots \\ &+ \left(c_{M0}\mathbf{I} + \dots + c_{MM}\mathbf{A}^M\right)\mathbf{x}[k-M] \end{aligned} \quad (3)$$

where $P_i(\mathbf{A})$ is a matrix polynomial in $\mathbf{A}$, c_{ij} are scalar polynomial coefficients, and $\mathbf{w}[k]$ is statistical noise. The $P_i(\mathbf{A})$ can be seen in the DSP_G framework as causal graph filters.

Note that this model allows a signal on a node at the current time to be affected through network effects by signals on other nodes at past times. This can model delays across the network, which is important for representing dynamics in many real world processes. Matrix polynomial $P_i(\mathbf{A})$ 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 by $N_\mathbf{A}$, the degree of the minimum polynomial of $\mathbf{A}$. Typically, we take the model order $M < N_\mathbf{A}$.

4. ESTIMATING ADJACENCY MATRICES

Given a time series $\mathbf{x}(t)$ on graph $G = (\mathcal{V}, \mathbf{A})$ with *unknown* $\mathbf{A}$, we wish to estimate the graph structure given by $\mathbf{A}$. A first approach to estimate $\mathbf{A}$ can be formulated as the following optimization problem:

$$(\mathbf{A}, \mathbf{c}) = \underset{\mathbf{A},\mathbf{c}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=M}^{K-1} \left\| \mathbf{x}[k] - \sum_{i=1}^{M} P_i(\mathbf{A}) \mathbf{x}[k-i] \right\|_2^2 + \lambda_1 \|\text{vec}(\mathbf{A})\|_1 + \lambda_2 \|\mathbf{c}\|_1 \quad (4)$$

where $\mathbf{c} = \begin{pmatrix} c_{10} & c_{11} & \dots & c_{ij} & \dots & c_{MM} \end{pmatrix}^T$ is a vector collecting all the c_{ij}'s. Assuming that $c_{11} \neq 0$, then without further loss of generality we can fix $c_{10} = 0$ and $c_{11} = 1$ to ensure that $\mathbf{A}$ is uniquely recoverable.

In equation (4), the regularizing term $\|\text{vec}(\mathbf{A})\|_1$ promotes sparsity of the estimated adjacency matrix, and the term $\|\mathbf{c}\|_1$ also promotes sparsity in the matrix polynomial coefficients. Unfortunately, the matrix polynomial in the first term makes this problem nonconvex and can be potentially numerically unstable.

Instead, this estimation can be broken down into three separate, more tractable steps:

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

4.1. Solving for $P_i(\mathbf{A})$

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

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

This leads to the block-coordinate optimization problem,

$$\begin{aligned}\widehat{\mathbf{R}}_i = \underset{\mathbf{R}_i}{\operatorname{argmin}} \; \frac{1}{2} \sum_{k=M}^{K-1} \left\| \mathbf{x}[k] - \sum_{i=1}^{M} \mathbf{R}_i \mathbf{x}[k-i] \right\|_2^2 \\ + \lambda_1 \|\mathrm{vec}(\mathbf{R}_1)\|_1 + \lambda_2 \sum_{j \neq i} \|[\mathbf{R}_i, \mathbf{R}_j]\|_F^2\end{aligned} \tag{5}$$

where $\widehat{\mathbf{R}}_i$ is an approximation for $P_i(\mathbf{A})$. While this is still overall 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 can be solved using standard methods for ℓ_1 regularized least squares.

4.2. 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, the information has already been incorporated during the iterations when solving for $\widehat{\mathbf{R}}_i$. A second approach is also possible, explicitly using all the $\widehat{\mathbf{R}}_i$ together to find $\mathbf{A}$,

$$\begin{aligned}\widehat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \; \left\| \widehat{\mathbf{R}}_1 - \mathbf{A} \right\|_2^2 + \lambda_1 \|\mathrm{vec}(\mathbf{A})\|_1 \\ + \lambda_2 \sum_{i=2}^{M} \left\| \left[\mathbf{A}, \widehat{\mathbf{R}}_i \right] \right\|_F^2\end{aligned} \tag{6}$$

This can be seen as running one additional step further in the block coordinate descent to find $\widehat{\mathbf{R}}_i$.

4.3. Estimating c_{ij}

With $\widehat{\mathbf{A}}$ and the data $\mathbf{X}$, we can now estimate c_{ij} ,

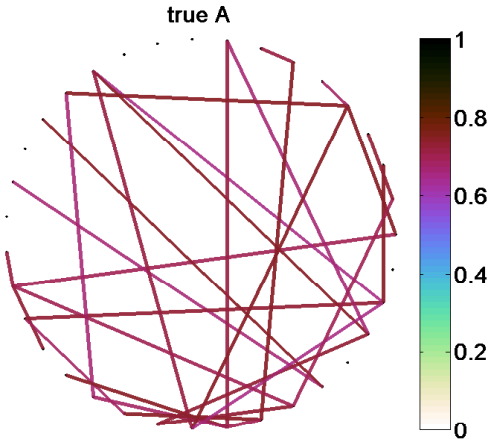
$$\widehat{\mathbf{c}} = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{Y} - \mathbf{B}\mathbf{c}\|_F^2 + \lambda_3 \|\mathbf{c}\|_1$$

where $\mathbf{Y} = \mathrm{vec}\,(\mathbf{X}_M)$,

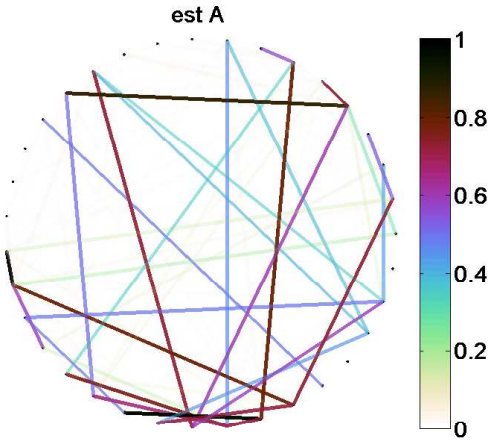
$$\mathbf{B} = \left(\mathrm{vec}\,(\mathbf{X}_{M-1}) \; \ldots \; \mathrm{vec}\left(\widehat{\mathbf{A}}^i \mathbf{X}_{M-j}\right) \; \ldots \; \mathrm{vec}\left(\widehat{\mathbf{A}}^M \mathbf{X}_0\right) \right),$$

$$\mathbf{X}_m = \begin{pmatrix} \mathbf{x}[m] & \mathbf{x}[m+1] & \ldots & \mathbf{x}[m+K-M-1] \end{pmatrix},$$

which can also be solved using standard ℓ_1-regularized least squares methods [9].



(a) True weighted $\mathbf{A}$ matrix



(b) Estimated weighted $\widehat{\mathbf{A}}$ matrix

Fig. 1. Synthetic simulated adjacency matrix

5. EXPERIMENTS

The algorithm was tested on a simulated example (varying N and K) and a temperature sensor network dataset ($N = 150$ and $K = 365$). To solve the regularized least squares iterations for estimating the CGP matrices, we used Gradient Projection for Sparse Reconstruction [9]. To estimate the MRF matrices, we implemented a proximal gradient descent algorithm. Although the estimated matrices are *directed*, undirected graphs are shown in this section for the sake of presentation.

5.1. Simulated Network

The simulated dataset was generated by first creating a random sparse $\mathbf{A}$ matrix for a graph with N nodes and $\mathbf{c}$ coefficients that corresponded to a stable system. The $\mathbf{A}$ in our sim-

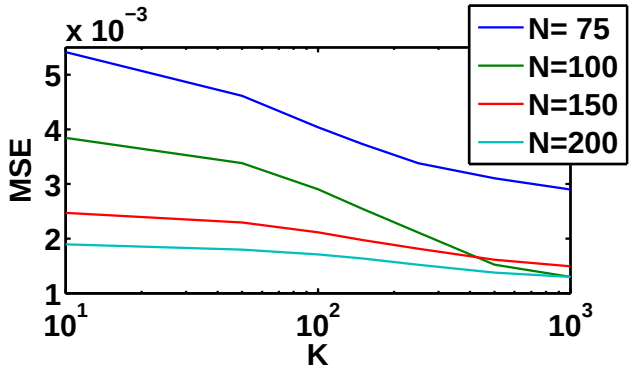


Fig. 2. MSE vs number of Samples K for simulated data

ulations had each off-diagonal element independently drawn from a unit Normal distribution and made sparse by thresholding $1.6 \leq \mathbf{A}_{ij} \leq 1.8$. The diagonal elements were generated from a uniform distribution $\mathcal{U}(0.5, 1)$. Then data matrix $\mathbf{X}$ was formed by generating random initial states and additive white Gaussian noise $\mathbf{w}[k]$, and computing K samples of $\mathbf{x}[k]$ according to (3).

In figure 1a, we see the structure of the $\mathbf{A}$ matrix with $N = 35$ nodes used in the simulated graph to generate $\mathbf{X}$ with $K = 100$ samples according to a CGP. In figure 1b, we wee the structure of the $\widehat{\mathbf{A}}$ matrix estimated from the data $\mathbf{X}$. We see that the estimated $\widehat{\mathbf{A}}$ matrix has almost the same support as the true $\mathbf{A}$ matrix visually. The MSE computed as

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

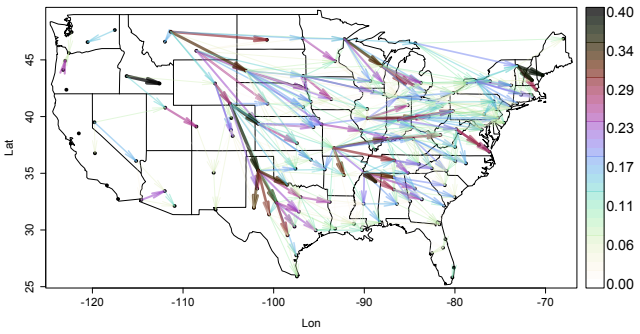
for the estimate shown in figure 1 is 0.004.

In figure 2, we see mean squared error (MSE) for $\widehat{\mathbf{A}}$ across different numbers of time samples K on the same graph, as well as across different numbers of nodes N corresponding to distinct graphs with the same level of sparsity. The MSE is averaged over 50 Monte-Carlo simulations of data matrix $\mathbf{X}$ for each problem size. As expected, the MSE for each problem size decreases as the number of samples K increases.

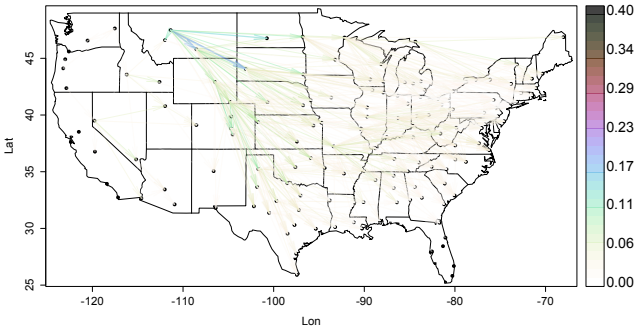
5.2. Temperature Sensor Network

The temperature dataset is a collection of daily average temperature measurements taken over 365 days in 2003 at 150 locations around the continental United States [10].

In figure 3, we compare the temperature networks estimated using CGP and MRF models. We first detrend the time signals at each sensor for seasonal effects and then recover the $\mathbf{A}$ matrix using a model order $M = 2$ and the same level of sparsity ($N_0/N^2 = 0.05$, where N_0 is the number of zero elements in $\mathbf{A}$) for both models. 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 $-110 \leq x \leq -100$.



(a) Graph Estimated using CGP



(b) Graph Estimated using MRF

Fig. 3. Estimated temperature network using order $M = 2$ model

This easy interpretation is consistent with knowledge of geographic and meteorological features. In contrast, while the MRF model also captures this overall geographic structure, the weights are much weaker at the same fixed sparsity level.

We have demonstrated that our method is able to estimate graphs from simulated data and shown its behavior when varying the dataset size. We have also empirically shown that adjacency matrices estimated from our CGP model can represent real processes better than those estimated from the MRF model.

6. CONCLUSIONS

We have presented a new method to estimate the graph structure capturing dependence in networked data and a computationally tractable algorithm for estimating such causal networks. The algorithm was demonstrated on a small simulated example and a real temperature sensor network dataset. The estimated adjacency matrix in the small example was shown to be close to the true graph, and in the real dataset consistent with prior physical knowledge.

Future work includes studying the convergence conditions and properties of the multi-convex optimization, as well as relation of estimation errors to time series length, number of nodes, and graph sparsity or structure.

7. REFERENCES

[1] D.I. Shuman, S.K. 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] Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu, "High-dimensional covariance estimation by minimizing l_1-penalized log-determinant divergence," *Electronic Journal of Statistics*, vol. 5, pp. 935–980, 2011.

[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.

[4] 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, June 2014.

[5] 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.

[6] Jitkomut Songsiri and L. Vandenberghe, "Topology selection in graphical models of autoregressive processes," *J. Mach. Learn. Res.*, vol. 11, pp. 2671–2705, 2010.

[7] A. Bolstad, B.D. Van Veen, and R. Nowak, "Causal network inference via group sparse regularization," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628–2641, June 2011.

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–41, July 2008.

[9] M.A.T. Figueiredo, R.D. Nowak, and S.J. 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.

[10] "National climactic data center," `ftp://ftp.ncdc.noaa.gov/pub/data/gsod`, 2011.