Draft: Learning Laplacian from Binomial Signals

Skip Moses

3/12/2022

Contents

1 Introduction

Data often has an underlying structure or geometry that can be modeled as a signal on the vertices of a weighted, undirected graph. Traditionally, the emphasis was on using a underlying graph, or network, to understand the properties of signals over the vertices. Recently, there has been a surge in converse problem; of learning a graph structure from a set of signals satisfying some constraints Xia et al. (2021) Stankovic et al. (2019) Ortega et al. (2018) . A subset of graph learning that focuses on graph signal processing based methods deals with sampling, graph recovery and learning topological structure Xia et al. (2021).

The methods for learning topological structure fall into the categories of regression like models Dong et al. (2016), Tugnait (2021), Pu, Cao, et al. (2021), and Saboksayr and Mateos (2021), and machine learning techniques Pu, Chau, et al. (2021), Kalofolias (2016), and Venkitaraman, Chatterjee, and Händel (2019) . In all of these models, signals are assumed to follow multivariate Gaussian distributions, but there has been little exploration in learning a network from binomial signals.

1.1 Related Work

Dong et al. propose a modified Factor Analysis model in Dong et al. (2016) for learning a valid graph laplacian. The model proposed is given by

$$x = \chi h + \mu_x + \epsilon$$

where $h \in \mathbb{R}^N$ is the latent variable, $\mu_x \in \mathbb{R}^N$ mean of x. The noise term ϵ is assumed to follow a multivariate Gaussian distribution with mean zero and covariance $\sigma_\epsilon^2 I_N$. The key difference from traditional factor analysis is the choice of χ as a valid eigenvector matrix for a graph laplacian. Finding a maximum apriori estimate of h reduces to solving the optimization problem

$$\begin{aligned} \min_{L \in \mathbb{R}^{N\times N},\ Y \in \mathbb{R}^{N\times P}} & ||X-Y||_F^2 + \alpha \mathrm{tr}(Y^T L Y) + \beta ||L||_F^2 \\ \text{s.t.} \quad & \mathrm{tr}(L) = N, \\ & L_{i,j} = L_{j,i} \leq 0, \ \ i \neq j, \\ & L \cdot \mathbf{1} = \mathbf{0} \end{aligned} \tag{1}$$

Because (1) is not jointly convex, Dong et al. employ an alternating minimization scheme to solve the problem. Initially, set $Y = X$ and we find a suitable L by solving

$$\begin{aligned} \min_{L} \ & \alpha \mathrm{tr}(Y^T L Y) + \beta ||L||_F^2, \\ \text{s.t.} \quad & \mathrm{tr}(L) = N, \\ & L_{i,j} = L_{j,i} \leq 0, \ \ i \neq j, \\ & L \cdot \mathbf{1} = \mathbf{0} \end{aligned} \tag{2}$$

Next, using L from (2), we solve

$$\min_{Y} \ \ ||X-Y||_F^2 + \alpha \mathrm{tr}(Y^T L Y) \tag{3}$$

Both (2) and (3) can be cast as convex optimization problems. Specifically, 2 can be solved with the method of alternating direction of multipliers (ADMM). The model is applied to both synthetic and real world data, and compared to a technique used in machine learning that is similar to sparse inverse covariance estimation of Gaussian Markov Random Field models. Four evaluation criteria were used to evaluate the performance of the framework: *Precision, Recall, F-measure,* and *Normalized Mutual Information (NMI).* Kalofolias cites two weaknesses to this model: The use of the Frobenius norm is not amenable to interpretation, and it requires four constraints on the solution space Kalofolias (2016).

In Kalofolias (2016), Kalofolias provides complementary ideas to improve on the work of Dong et al. In particular, Kalofolias considers the more natural problem of leaning the weighted adjacency matrix A, and show an extra smoothness condition leads to the learned graph being sparse. Define, Z to be the matrix of pairwise distances of the signal, namely, $Z_{i,j} = ||x_i - x_j||^2$. Under this definition we can recast our smoothness term as

$$\mathrm{tr}(X^T L X) = \frac{1}{2}\mathrm{tr}(AZ) = \frac{1}{2}||A \circ Z||_{1,1}; \tag{4}$$

where $\circ$ is the hadamard product and $||A||_{1,1}$ is the element wise 1-norm. We can now recast 5 as

$$\begin{aligned} \min_{A} \ & ||A \circ Z||_{1,1} + f(A), \\ \text{s.t.} \quad & ||A||_{1,1} = N, \\ & A_{i,j} = A_{j,i} \geq 0, \ \ i \neq j, \\ & \mathrm{diag}(A) = \mathbf{0} \end{aligned} \tag{5}$$

The model proposed by Kalofolias is given by specifying $f(A) = -\alpha \mathbf{1}^T \log(W\mathbf{1}) + \beta ||W||_F^2$. An appealing feature of this model is the fact that for any $\lambda > 0$ and solution $F(Z, \alpha, \beta)$ the following equalities hold

$$F(Z, \alpha, \beta) = \lambda F(Z, \frac{\alpha}{\lambda}, \beta\lambda) = \alpha F(Z, 1, \alpha\beta). \tag{6}$$

This means we can fix a scale $||A|| = s$ for an arbitrary norm and solve for β in 4 with $\alpha = 1$, and then re-normalize our solution. Furthermore, the log barrier term only acts on the degrees. This provides an

advantage over the model 1, because it promotes connectivity in the learned graph. Kalofolias goes onto show 4 can be solved efficiently for many choices of $f(A)$ via primal dual algorithms.

In Venkitaraman, Chatterjee, and Händel (2019) machine learning concepts are employed to learn a graph from training data. The goal is given some data x_n to model a target t_n by the linear basis model for regression

$$y_n = \mathbf{W}^T \phi(x_n), \tag{7}$$

where $\phi(x_n)$ is some known function and $\mathbf{W}$ denotes the regression coefficient matrix. Venkitarman et al. assume only the target y_n is smooth over the graph. The input x_n does not have to be a signal over the graph and can even be agnostic to a graph. The cost function for the model when the underlying graph is known is given by

$$C(\mathbf{W}) = \sum_{n=1}^{N} \left(||t_n - y_n||_2^2 + \beta y_n^T L y_n \right) + \alpha \text{tr}(\mathbf{W}^T \mathbf{W}). \tag{8}$$

Next, it is shown KRG induces a smoothing effect on the target by a shrinkage in the direction of eigenvectors of L that have eigenvalues less than α. In order to learn an unknown graph, a term is added to the cost function $C(\mathbf{W}, L) = C(\mathbf{W}) + \nu \text{tr}(L^T L)$. Venkitarman et al. use the model to predict temperature using air-pressure observations collected in Sweden, next day temperature from current day in Sweden, and fMRI voxel intensities of the cerebellum region. The latter two experiments are compared the method of kernel-ridge regression (KRR) by using a normalized-mean-square-error to measure the performance of the prediction. It is concluded that the method is well suited for training with noise data sets that may be small.

1.2 Preliminaries

A weighted, undirected graph is a triple $G = (V, E, \omega)$ of two sets $V = \{1, \ldots, |V| = N\}$ and $E \subset V \times V$ and a weighting function $\omega(i, j) = \omega_{i,j}$ that assigns a non-negative real number to each edge. We can represent a graph by its adjacency matrix A where $A_{i,j} = \omega_{i,j}$ if $(i, j) \in E$ and 0 otherwise. A signal on a graph G is a function $f : V \to \mathbb{R}$ that can be represented as vector $x \in \mathbb{R}^N$. The Laplacian of a graph is the matrix $L = D - A$ where D is the degree matrix. The Laplacian acts as a difference operator on signals via it's quadratic form

$$x^T L x = \sum_{(i,j) \in E} A_{i,j} (x_j - x_i)^2. \tag{9}$$

The Laplacian is positive semi definite, so it has a complete set of orthornormal eigenvectors, and real non negative eigenvalues. Thus, we can diagonalize $L = \chi^T \Lambda \chi$ where Λ is the diagonal matrix of eigenvalues and χ is associated matrix of eigenvectors.

Note that 9 is minimized when adjacent vertices have identical signal values. This makes 9 well suited for measuring the smoothness of a signal on a graph. We can cast the problem of learning a graph by the optimization problem found in Kalofolias (2016)

$$\begin{aligned} \min_{L} \quad & \text{tr}(Y^T L Y) + f(L), \\ \text{s.t.} \quad & \text{tr}(L) = N, \\ & L_{i,j} = L_{j,i} \leq 0, \quad i \neq j, \\ & L \cdot \mathbf{1} = \mathbf{0} \end{aligned} \tag{10}$$

2 Methods

2.1 Signals on the graph and model specification

We consider a weighted undirected graph $G = (V, E)$, with the vertices set $V = 1, 2, \ldots, N$, and edge set E. Let $\mathbf{A}$ denote the weighted adjacency matrix of G. In the case of weighted undirected graph, $\mathbf{A}$ is a square and symmetric matrix.

Let $\mathbf{Y}_{i,j}$ denote the signal on the node i of graph G at round j, where $j = 1, \ldots, M$, and $i = 1, \ldots, N$. We assume that $\mathbf{Y}_{i,j}$ is a binary signal that can be 1, or 0. Suppose the signals at stratum j denoted by $\mathbf{Y}[,j]$ for all N nodes are independent of the signals at stratum k denoted by $\mathbf{Y}[,k]$, for $j \neq k$. Let $p_{i,j}$ denote the probability of $\mathbf{Y}_{i,j} = 1$. Our model assumes

$$\text{logit}(p_{i,j}) = (\mathbf{A}h)_i, \tag{11}$$

where $\mathbf{A}$ is the adjacency matrix from the graph G, h is a vector of latent factors that governs $p_{i,j}$ through $\mathbf{A}$ and assumed to be a standard normal random vector.

2.2 Maximum likelihood for one stratum

Consider the probability mass function for a given $\mathbf{A}$ and signal $y = \mathbf{Y}[,k]$

$$\begin{aligned} P_{\mathbf{A}h}(y_i) &= p^{y_i}(1-p)^{1-y_i} && (12)\\ &= \left(\frac{e^{\mathbf{A}[i,]h}}{1+e^{\mathbf{A}[i,]h}}\right)^{y_i} \left(1 - \frac{e^{\mathbf{A}[i,]h}}{1+e^{\mathbf{A}[i,]h}}\right)^{1-y_i} && (13)\\ &= \left(\frac{e^{\mathbf{A}[i,]h}}{1+e^{\mathbf{A}[i,]h}}\right)^{y_i} \left(\frac{1}{1+e^{\mathbf{A}[i,]h}}\right)^{1-y_i} && (14)\\ &= \frac{e^{y_i\mathbf{A}[i,]h}}{1+e^{\mathbf{A}[i,]h}} && (15) \end{aligned}$$

Therefore, our Likelihood function will be given by

$$\mathcal{L}(h) = \prod_{i=1}^{N} \left(\frac{e^{y_i\mathbf{A}[i,]h}}{1+e^{\mathbf{A}[i,]h}}\right) \tag{16}$$

In order to maximize we consider the natural logarithm of our likelihood

$$\log(\mathcal{L}(h)) = \sum_{i=1}^{N} \Big(y_i(\mathbf{A}[i,]h) - \log(1+e^{\mathbf{A}[i,]h})\Big) \tag{17}$$

2.3 Optimization Program

Taking inspiration from the above derivation we will solve for estimated $\mathbf{A}$ by maximizing the following

$$\begin{aligned} &\max_{\mathbf{A},h} \sum_{j=1}^{M}\sum_{i=1}^{N} \Big(y_{i,j}(\mathbf{A}[i,]h) - \log(1+e^{\mathbf{A}[i,]h})\Big) - \alpha|L|_F \\ \text{s.t.}\quad &\mathbf{A}_{i,j} = 0 \text{ if } i = j;\ \mathbf{A}_{i,j} \geq 0 \text{ if } i \neq j \\ &\max(h) \leq a;\ \min(h) \geq b\ ;\nVdash h^T = 0 \end{aligned} \tag{18}$$

where α is a tuning parameter for controlling the sparsity and $\mathbf{A}$ and b are tuning parameters for restricting the spread of the values of h.

We presented our optimization algorithm below. In each iteration of step 5) and 6) of Algorithm 1, the optimization program gives a Disciplined Concave Program that can be solved efficiently in Python with the CVXPY library.

Algorithm 1: Binary Signal Graph Learning

1) **Input:** Input a signal $\mathbf{Y}$.
2) **Output:** Output an estimated $\mathbf{A}$.
3) **Initialization:** $h_{i,0} \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, N$
4) **for** $t = 1, \ldots, iter$:
5) **Update A given** h :
6) *Fix* h *in Optimization Program and solve* $\mathbf{A}$
7) **Update** h **given A** :
8) *Fix* $\mathbf{A}$ in Optimization Program and solve h
9) **end for**

3 Results

3.1 Synthetic Data Generation

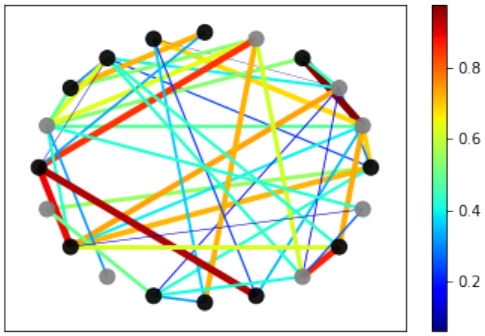

Figure 1: The ground truth graph with an example signal shown.

Let $\mathbf{A}$ be the adjacency matrix of the graph in Figure 1, and $h \in \mathbb{R}$ be such that $h_i \sim \mathcal{N}(0,1)$. Set p as a logistic function of $\mathbf{A}h$, and compare p to a vector t, where each t_i follows a random uniform distribution $\mathcal{U}(0,1)$. If $t_k < p_k$, then node k takes the signal value of 1, and 0 otherwise. We generate $M = 100$ synthetic signals, and use them to learn an adjacency matrix.

3.2 Graph Learning Results

The learned graph is found with parameter values $\alpha = .2$, $a = 1$ and $b = -1$ by implementing Algorithm 1. We can see the learned adjacency matrix shares some similar features to the ground truth (left), but the learned adjacency matrix is sparser than the ground truth (right).

4 Discussion

To our knowledge, this is the first work on learning graph topology from binary signals. We have developed a method to effectively learn the graph adjacency matrix from binary signals. This model can be easily extended for learning binomial signals in general. Incorporating parameter optimization techniques would be a natural improvement on this model. Moreover, it would be desirable to develop scalable algorithms that

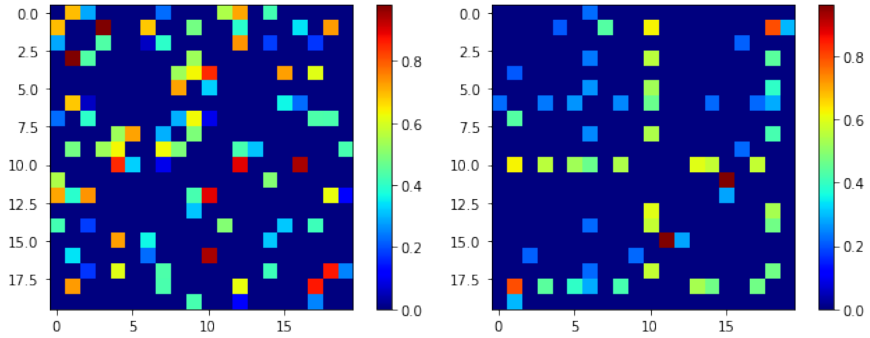

Figure 2: (Left) A heatmap of the ground truth adjacency matrix. (Right) A heatmap of the estimated adjacency matrix.

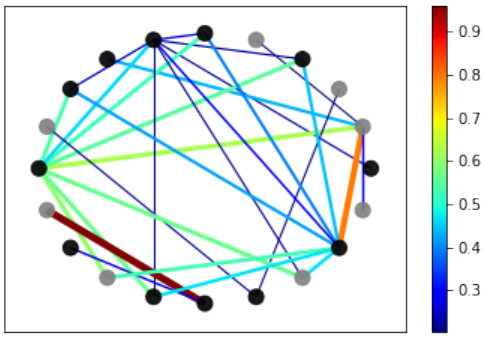

Figure 3: Circular Embedding of Estimated graph. Note the overall weight of each edge is smaller than the edge weights of the ground truth.

can be used to learn a topology for a large graph from signals. Recently, in Saboksayr and Mateos (2021) the author shows fast proximal-gradient iterations can be applied to the framework given by Kalofolias (2016) improving the over-all runtime.

5 Acknowledgement

his work is supported by Research, Scholarship, and Creative Activity (RSCA) Award 2021-2022 through Chico State Enterprises. Thank you to Dr. Robin Donatello, Ph.D., for approving this work for Data Science Capstone Project

6 References

Dong, Xiaowen, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. 2016. "Learning Laplacian Matrix in Smooth Graph Signal Representations." *IEEE Transactions on Signal Processing* 64 (23): 6160–73.

Kalofolias, Vassilis. 2016. "How to Learn a Graph from Smooth Signals." In *Artificial Intelligence and Statistics*, 920–29. PMLR.

Ortega, Antonio, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. 2018. "Graph Signal Processing: Overview, Challenges, and Applications." *Proceedings of the IEEE* 106 (5): 808–28.

Pu, Xingyue, Tianyue Cao, Xiaoyun Zhang, Xiaowen Dong, and Siheng Chen. 2021. "Learning to Learn Graph Topologies." *Advances in Neural Information Processing Systems* 34.

Pu, Xingyue, Siu Lun Chau, Xiaowen Dong, and Dino Sejdinovic. 2021. "Kernel-Based Graph Learning from Smooth Signals: A Functional Viewpoint." *IEEE Transactions on Signal and Information Processing over Networks*.

Saboksayr, Seyed Saman, and Gonzalo Mateos. 2021. "Accelerated Graph Learning from Smooth Signals." *IEEE Signal Processing Letters* 28: 2192–96.

Stankovic, Ljubisa, Danilo P Mandic, Milos Dakovic, Bruno Scalzo, Milos Brajovic, Ervin Sejdic, and Anthony G Constantinides. 2019. "Vertex-Frequency Graph Signal Processing: A Review." *arXiv Preprint arXiv:1907.03471*.

Tugnait, Jitendra K. 2021. "Sparse Graph Learning Under Laplacian-Related Constraints." *IEEE Access* 9: 151067–79.

Venkitaraman, Arun, Saikat Chatterjee, and Peter Händel. 2019. "Predicting Graph Signals Using Kernel Regression Where the Input Signal Is Agnostic to a Graph." *IEEE Transactions on Signal and Information Processing over Networks* 5 (4): 698–710.

Xia, Feng, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. 2021. "Graph Learning: A Survey." *IEEE Transactions on Artificial Intelligence* 2 (2): 109–27.