1 2 3

Graph Learning from Binomial Signals'

Skip Moses, Jing Guo

***Abstract*—Data often has an underlying structure of Geometry that can be modeled as a signal on the vertices of a weighted, undirected graph. There are several analogies between traditional signal processing and algebraic graph theory that translates many of the tools of discrete signal processing such as spectral analysis of multichannel signals, system transfer function, and digital filter design to name a few. Historically, GSP has focused on modeling smooth signals on a graph, but the increase in the availability of abstract data sets has motivated algorithms for learning a valid graph given a set of signals. In this work, we attempt to design a novel methodology for learning a valid graph topology given a set of binary signals on the graph. This is accomplished by extending the ideas of Logistic Regression and Maximum Likelihood with constraints that enforce a valid graph topology on the regression coefficient matrix.**

***Index Terms*—graph learning, graph signal processing, binary signals, logistic regression, Laplacian matrix**

I. 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 an underlying graph, or network, to understand the properties of signals over the vertices. Recently, there has been a surge in the converse problem; of learning a graph structure from a set of signals satisfying some constraints [**?**] [**?**] [**?**]. A subset of graph learning that focuses on graph signal processing-based methods deals with sampling, graph recovery, and learning topological structure [**?**].

The methods for learning topological structure fall into the categories of regression-like models [**?**], [**?**], [**?**], and [**?**], and machine learning techniques [**?**], [**?**], and [**?**]. In all of these models, signals are assumed to follow multivariate Gaussian distributions, but there has been little exploration into learning a network from binomial signals.

In this work, we suppose the odds of a node having a signal value of 1 or 0 depend on the signal value of its neighboring nodes in order to estimate the underlying graph topology. Our methodology generalizes the framework of traditional logistic regression in order to learn an adjacency matrix from a set of binary signals.

II. Related Work

Dong et al. propose a modified factor analysis model in [**?**] for learning a valid graph laplacian. The model proposed is given by

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

where $h \sim \mathcal{N}(0, \Lambda^{\dagger})$ is the latent variable,Λ is the eigenvalue matrix of L, and $\mu_x \in \mathbb{R}^N$ 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 a posterior estimate of h reduces to solving the optimization problem

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

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

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

Next, using L from (2), we solve

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

Both (2) and (3) can be cast as convex optimization problems. Specifically, 2 can be solved with the alternating direction method 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 on the Laplacian has reduced interpretability, and it requires four constraints on the solution space [**?**]. These extra constraints make computational scalability difficult.

In [**?**], Kalofolias provides complementary ideas to improve on the work of Dong et al. In particular, Kalofolias considers the more natural problem of learning 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

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

This work was supported by the Research, Scholarship, and Creative Activity (RSCA) Award from CSU, Chico.

Skip Moses is currently a graduate student in the College of Science and Mathematics, California Polytechnic University, San Luis Obispo, CA 93407 USA (e-mail: skipmoses@gmail.com).

The corresponding author Jing Guo is with the Department of Mathematics and Statistics, California State University, Chico, CA 95929 USA (e-mail: jguo2@csuchico.edu).

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

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

The model proposed by Kalofolias is given by specifying $f(A) = -\alpha \mathbf{1}^T \log(A\mathbf{1}) + \beta ||A||_F^2$. 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 [?] machine learning concepts are employed to learn a graph from training data. Let $\{x_n\}_1^N$ be a set of N input observations with each x_n paired with a target $t_n \in \mathbb{R}^M$. The authors model a set of smooth signals y_n and t_n via the following:

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

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. 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. uses the model to predict temperature using air-pressure observations collected in Sweden, next-day temperature from the current day in Sweden, and fMRI voxel intensities of the cerebellum region. The latter two experiments are compared to 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 [?].

III. Preliminaries

A weighted, undirected graph is a triple $G = (V, E, \omega)$ of two sets $V = \{1, \dots, |V| = N\}$ 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 and semi-definite, so it has a complete set of orthonormal eigenvectors and real nonnegative eigenvalues. Thus, we can diagonalize $L = \chi^T \Lambda \chi$ where Λ is the diagonal matrix of eigenvalues and χ is the 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 [?]

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

IV. Methods

A. Signals on the graph and model specification

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

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

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

where α is a tuning parameter for controlling the sparsity and a and b are tuning parameters for restricting the spread of the values of h.

3) Algorithm: We present our optimization algorithm below. In each iteration of steps 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 binary signals on a graph $\mathbf{Y}$.
2) Output: Output an estimated $\mathbf{A}$.
3) Initialization: $h_{i,0} \sim \mathcal{U}(-1,1)$ for $i = 1, \dots, N$
4) for $t = 1, \dots, iter$ :
5) Update $\mathbf{A}$ given $h$ :
6)   Fix $h$ in Optimization Program and solve $\mathbf{A}$
7) Update $h$ given $\mathbf{A}$ :
8)   Fix $\mathbf{A}$ in Optimization Program and solve h
9) end for
```

V. Results

A. Synthetic Data Generation

Let $\mathbf{A}$ be the adjacency matrix of the graph in Figure **??**fig:GTG), and $h \in \mathbb{R}$ be such that $h_i \sim \mathcal{U}(-1, 1)$. 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.

B. Graph Learning Results

The optimal α value, $\alpha = 0.2$ is found by performing a grid search and selecting the α value that maximizes the F-measure of the learned adjacency matrix with the ground truth adjacency matrix. The parameter values for $a = 1$ and $b = -1$ were selected to be the ground truth values to simplify parameter optimization. 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).

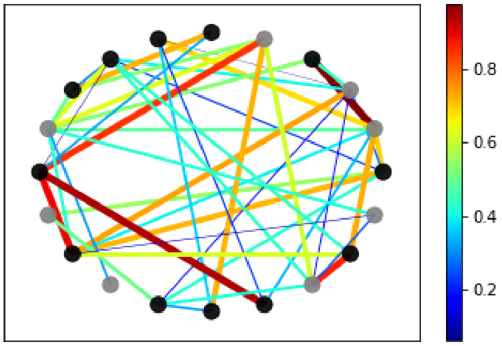
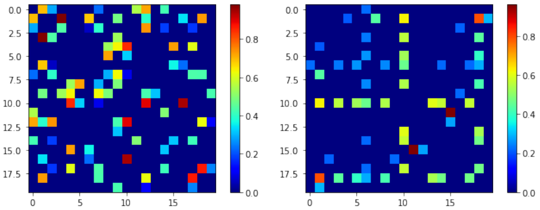


Fig. 1: Ground truth graph used to generate synthetic data.



VI. 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 in this model. Moreover, it would be desirable to develop scalable algorithms that can be used to learn a topology for a large graph from

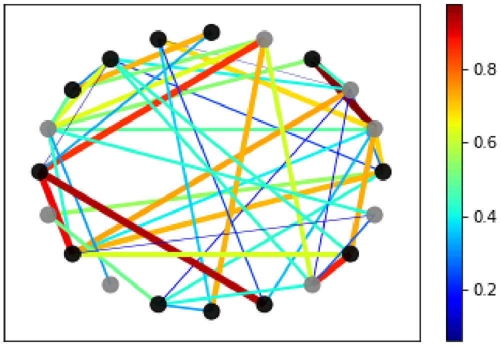


Fig. 3: Graph estimated from synthetic data.

signals. Recently, in [?] the author shows fast proximal-gradient iterations can be applied to the framework given by [?] improving the overall runtime.

Acknowledgment

This 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.