Graph Learning from Smooth Signal Representations

Skip Moses

5/16/2022

Introduction

Data often has an underlying structure or 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, digital filter design, parameter estimation, and optimal denoising. Historically, GSP has focused on modeling smooth signals on a graph, but the increase in availabilty of abstract data sets has led to progress in learning a valid graph given a set of signals.

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 nonnegative 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

$$x^T Lx = \sum_{(i,j)\in E} A_{i,j}(x_j - x_i)^2 \tag{1}$$

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 matirx of eigenvalues and χ is associated matrix of eigenvectors.

Note that the quadratic form above is minimized when adjacent vertices have identical signal values. This makes program 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)

$$\begin{aligned} \min_{L} \quad & \operatorname{tr}(Y^T LY) + f(L), \\ \text{s.t.} \quad & \operatorname{tr}(L) = N, \\ & L_{i,j} = L_{j,i} \le 0, \quad i \ne j, \\ & L \cdot \mathbf{1} = \mathbf{0} \end{aligned} \tag{2}$$

Graph Learning Model

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

$$\begin{aligned} \min_{L \in \mathbb{R}^{N \times N},\ Y \in \mathbb{R}^{N \times P}} & ||X - Y||_F^2 + \alpha \text{tr}(Y^T L Y) + \beta ||L||_F^2 \\ \text{s.t.} \quad & \text{tr}(L) = N, \\ & L_{i,j} = L_{j,i} \leq 0, \ \ i \neq j, \\ & L \cdot \mathbf{1} = \mathbf{0} \end{aligned} \tag{3}$$

Because the program above 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

$$\begin{aligned} \min_{L} \ \ & \alpha \text{tr}(Y^T L Y) + \beta ||L||_F^2, \\ \text{s.t.} \quad & \text{tr}(L) = N, \\ & L_{i,j} = L_{j,i} \leq 0, \ \ i \neq j, \\ & L \cdot \mathbf{1} = \mathbf{0} \end{aligned} \tag{4}$$

Next, using L from the first step we solve

$$\min_{Y} \ \ ||X - Y||_F^2 + \alpha \text{tr}(Y^T L Y) \tag{5}$$

Both steps can be cast as convex optimazation problems. Specifically, the first problem can be solved with the method of alternating direction of multipliers (ADMM). The second can be solved algebraically. 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.

Python Implementation

The program of Dong et. al. can be solved efficently using the Python package CVXPY. An implementation of the algorithm can be found here.

We can see by inspection the algorithm appears to do a decent job learning the network. Detailed results on the preformance of the algrithm can be found in Dong et al. (2016) .

Contact Info

- GitHub
- Email: skipmoses@gmail.com

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.

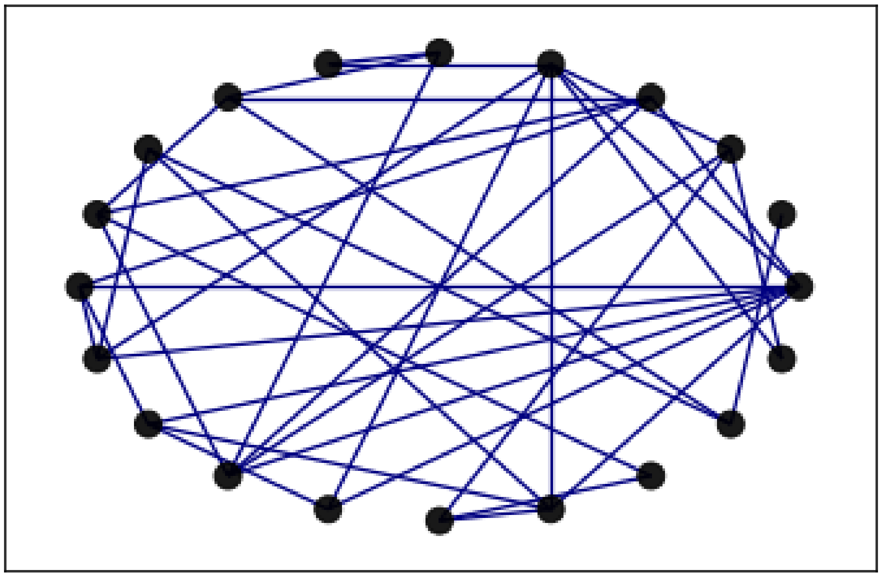

Figure 1: An Erdos-Reyni Graph on 20 nodes, with edge probability 0.2. The ground truth network used to generate 100 synthetic gaussian signals for testing.

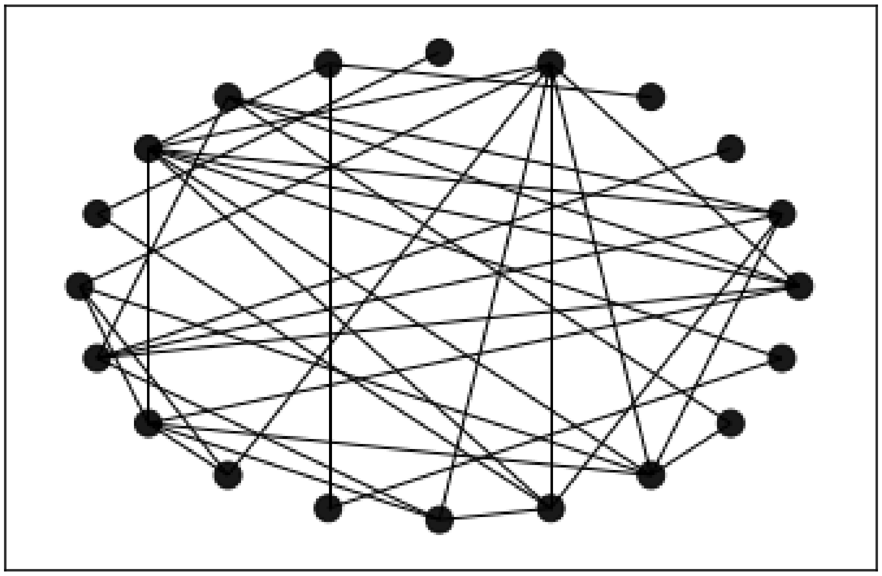

Figure 2: The estimated network. Notice the networks share many features.