GENERALIZED LAPLACIAN PRECISION MATRIX ESTIMATION FOR GRAPH SIGNAL PROCESSING

Eduardo Pavez and Antonio Ortega

Department of Electrical Engineering, University of Southern California, Los Angeles, USA

ABSTRACT

Graph signal processing models high dimensional data as functions on the vertices of a graph. This theory is constructed upon the interpretation of the eigenvectors of the Laplacian matrix as the Fourier transform for graph signals. We formulate the graph learning problem as a precision matrix estimation with generalized Laplacian constraints, and we propose a new optimization algorithm. Our formulation takes a covariance matrix as input and at each iteration updates one row/column of the precision matrix by solving a non-negative quadratic program. Experiments using synthetic data with generalized Laplacian precision matrix show that our method detects the nonzero entries and it estimates its values more precisely than the graphical Lasso. For texture images we obtain graphs whose edges follow the orientation. We show our graphs are more sparse than the ones obtained using other graph learning methods.

Index Terms— graph learning, precision matrix estimation, graph signal processing, generalized laplacian

1. INTRODUCTION

Graph signal processing (GSP) is a novel framework for analyzing high dimensional data. It models signals as functions on the vertices of a weighted graph, and extends classic signal processing techniques by interpreting the eigenvalues of the graph Laplacian as graph frequencies and the eigenvectors as a Graph Fourier Transform (GFT). Graph structures arise naturally in several domains such as sensor networks [1], brain networks [2], image de-noising [3], and image and video coding [4, 5, 6]. A major challenge in this new field is that of learning the graph structure from data. The learned graph must have a meaningful interpretation and be useful for analysis. Also, the learning algorithm must be efficient and scale nicely as dimensions increase.

In this work, we formulate graph learning as a matrix optimization problem and focus on an efficient algorithmic solution. Learning a graph from data can be posed as follows: given a set of nodes V and a set of corresponding signals on those nodes, choose an edge set E and edge weights, or equivalently, estimate a matrix \mathbf{Q} whose nonzero patterns define the edge connectivity of the graph. We propose a method that takes any symmetric matrix with positive diagonal entries, for example an empirical covariance matrix \mathbf{K} , and outputs a generalized Laplacian (GL) matrix \mathbf{Q} [7], a symmetric positive definite matrix with non positive off diagonal values.

Our solution **Q** satisfies the following properties: i) it has a probabilistic interpretation, ii) it allows for a GFT, iii) it provides compact data representation, and iv) it comes with an efficient estimation algorithm.

Using GLs instead of a combinatorial or normalized Laplacian has several advantages for this problem. First, by allowing self

loops and not restricting the first eigenpair, the optimization has fewer constraints, and still includes traditional Laplacian matrices as special cases. Second, the algorithmic solution is simpler and allows us to use a block coordinate descent algorithm that updates one row/column at each iteration. Finally, the eigenvectors of the GL also satisfy a nodal domain theorem [7, 8], which characterizes their oscilatory behavior allowing us to define a GL-based GFT.

To find an optimal GL, we solve a Gaussian maximum likelihood problem where \mathbf{Q} corresponds to an inverse covariance (precision) matrix with GL constraints. When \mathbf{K} is an empirical covariance matrix, optimizing the ML functional can be intepreted as promoting average smoothness of the data in the GL-GFT. The entries of \mathbf{Q} can be interpreted as partial correlations.

Recently, graph signals have been analyzed as random vectors with a Gaussian Markov Random Field (GMRF) distribution, whose Precision matrix is a Laplacian [4, 9, 10]. These papers assume the Laplacian matrix is known before-hand and do not propose a estimation algorithm. Our proposed Graph learning framework finds the optimal GL under the GMRF model.

Constructing a graph in which the data is smooth has been the main idea behind some recent work in graph learning for GSP [11, 12]. In [12] a smoothness functional of the combinatorial Laplacian is optimized to estimate a brain connectivity graph, while in [11] that idea is extended for noisy data and a more general statistical model with a PCA interpretation. In [13], a Gaussian ML with Laplacian constraints and ℓ_1 regularization is solved to find a sparse graph. These methods are difficult to apply because they require some parameter adjusting to balance sparsity, smoothness and data consistency, which have to be optimized using an exhaustive grid search. Furthermore, they do not come with efficient algorithms and they have to be implemented using general purpose solvers. Our proposed graph learning algorithm is a type of Coordinate Descent method, which have become very popular for their efficiency in solving high dimensional optimization problems [14].

In the statistics community the precision matrix estimation literature is extensive. The natural candidate for a precision matrix estimator in the inverse of the empirical covariance but it may not be sparse. Also, if the number of samples is less than the dimension, the empirical covariance is singular and inversion is not possible. For high dimensional data it is common that even if the number of realizations is large, it is still not enough to obtain a good estimator and regularization is required. One of the most popular algorithms is the graphical Lasso [15, 16] because of its simple implementation and speed. It solves a gaussian ML objective with an ℓ_1 regularization term. Other state of the art algorithms include [17, 18, 19]. These matrices are only constrained to be positive semidefinite, and since spectral graph properties and nodal domain theorems are derived from the theory of non negative matrices, these arbitrary precision matrices do not allow for a Fourier like interpretation using their eigensystem. For that reason, in this paper we consider GL matrices.

This work was supported in part by NSF under grant CCF-1410009.

For a given covariance matrix \mathbf{K} we solve

$$\min_{\mathbf{Q} \succeq 0, q_{ij} \le 0, i \ne j} -\log \det(\mathbf{Q}) + \operatorname{tr}(\mathbf{KQ}).$$
(1)

Our block coordinate descent algorithm to optimize (1) is similar to the one proposed by Slawski and Hein in [20]. At each iteration they solve a non negative quadratic program (NNQP), while our method also solves a NNQP but on the dual variable which simplifies the equation updates and works only with sparse matrices. They analyze (1) from a statistical estimation point of view and do not make the connection with GL matrices or GSP applications.

This paper is organized as follows. In Section 2 we introduce notation on graphs, GSP and GL matrices. In Section 3 we analyze the solution of (1) from a GSP point of view and derive our algorithm. In Section 4 we show experimental results with synthetic data and texture images. We end with conclusions and future work in Section 5.

2. PRELIMINARIES

We denote matrices and vectors by bold letters. For a matrix $\mathbf{X} = (x_{ij})$ its entries are denoted by x_{ij} and for a vector \mathbf{x} its *i*-th entry by x_i or x[i]. If \mathbf{A} is positive semidefinite we denote it by $\mathbf{A} \succeq 0$.

An undirected graph $G = (V, E, \mathbf{Q})$ is a triplet consisting of a collection of nodes $V = \{1, 2, ..., n\}$ connected by a set of edges E and a symmetric matrix representation \mathbf{Q} , where for $i \neq j$ the link $(i, j) \notin E$ if and only if the weight $\mathbf{Q}_{ij} = 0$. Note that our graph definition is different from the ones used in recent graph signal processing literature, where \mathbf{Q} is usually restricted to be a graph Laplacian [21] or an adjacency matrix [22]. An adjacency matrix is a non negative matrix \mathbf{W} , the degree of vertex i is defined as $d_i = \sum_{j=1}^n w_{ij}$ and the corresponding degree matrix as $\mathbf{D} = \text{diag}(d_1, \cdots, d_n)$. The combinatorial Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and the normalized Laplacian $\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$. For more details of these matrices and their graphical properties see [23]. All these matrices are special cases of the Generalized Laplacian (GL) matrix,

Definition 1 ([7]). A square matrix $\mathbf{Q} = (q_{ij})$ is a GL if it is symmetric, and $\mathbf{Q} = \alpha \mathbf{I} - \mathbf{N}$ with \mathbf{N} a non-negative matrix and $\alpha \in \mathbb{R}$. If \mathbf{Q} is also positive semidefinite we call it a Generalized Laplacian Precision (GLP) matrix.

Any GL can be written as $\mathbf{Q} = \mathbf{P} + \mathbf{L}$ where $\mathbf{P} = \text{diag}(\mathbf{p})$ is a diagonal matrix and $\mathbf{L1} = \mathbf{0}$. The matrix \mathbf{L} has the same off diagonal entries as \mathbf{Q} and $\mathbf{L}_{ii} = \sum_{j \neq i} q_{ij}$, thus GLs can be interpreted as combinatorial Laplacian with self loops. GLPs are also called symmetric M-matrices, and satisfy the following:

Proposition 1. [24]. Consider a non singular GLP matrix \mathbf{Q} with orthonormal eigendecomposition $\mathbf{Q} = \mathbf{U}\mathbf{\Omega}\mathbf{U}^T$, $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$ and $\mathbf{\Omega} = \text{diag}(\omega_1, \omega_2, \cdots, \omega_n)$ where the ω_i are sorted in increasing order of magnitude. The following properties hold.

- *1.* $\mathbf{Q}^{-1} \ge 0$, the inverse GLP is a non negative matrix.
- 2. The first eigenvector satisfies $\mathbf{u_1} > 0$.

The first property highlights a limitation of the GLP approach, namely, that it only makes sense if the covariance matrix is non negative, or very close to a non negative matrix. The second allows us to interpret the first eigenvector of \mathbf{Q} as the DC component of the graph spectrum. We can define the smoothness of a graph signal following what is tipically done with the combinatorial Laplacian and say that \mathbf{x} is smoother than \mathbf{y} iff $\frac{\mathbf{x}^t \mathbf{Q} \mathbf{x}}{\mathbf{x}^t \mathbf{x}} < \frac{\mathbf{y}^t \mathbf{Q} \mathbf{y}}{\mathbf{y}^t \mathbf{y}}$. The quantity $\frac{\mathbf{x}^t \mathbf{Q} \mathbf{x}}{\mathbf{x}^t \mathbf{x}}$ is minimized by the smoothest signal which is \mathbf{u}_1 , and if $\omega_i < \omega_j$

then \mathbf{u}_i is smoother than \mathbf{u}_j . The eigenvectors of a GL can be used to define the GFT of signal \mathbf{x} as $\bar{\mathbf{x}} = \mathbf{U}^t \mathbf{x}$. Another approach to visualize the GFT is by considering discrete nodal domain theorems for GL matrices, for space considerations we refer the reader to [7] for an excelent exposition on GL matrices and their properties.

The probabilistic interpretation of the GLP matrix is provided by considering a Gaussian Markov Random Field (GMRF) model for graph signals.

Definition 2. A random graph signal is a Gaussian random vector $\mathbf{x} = [x[1], x[2], \dots, x[n]]^t$ with mean $\boldsymbol{\mu}$ and GLP matrix \mathbf{Q} . And each x[i] is associated with a node of the graph $G = (V, E, \mathbf{Q})$.

The Gaussianity assumption implies that the partial correlations are given by [25]

$$\operatorname{Corr}(x[i], x[j] || x[k] : k \neq i, j) = -\frac{q_{ij}}{\sqrt{q_{ii}q_{jj}}}.$$
 (2)

Hence, the non zero pattern of \mathbf{Q} encodes the conditional independence relations within the graph signal, i.e., x[i] and x[j] are independent conditioned in all other variables iff $q_{ij} = 0$.

3. GLP MATRIX ESTIMATION

In this section we present our proposed solution to the problem of (1). We are interested in learning the graph structure of an *n* dimensional random graph signal **x** from i.i.d. realizations $\mathbf{x}_1, \dots, \mathbf{x}_N$. We assume the density of **x** is unknown and has finite second moments. For simplicity assume **x** is mean zero, and we have an estimator for its covariance $\mathbf{K} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$. We can write the right side term of (1) as

$$\operatorname{tr}(\mathbf{KQ}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x_i}^T \mathbf{Qx_i} = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} k_{ij}.$$
 (3)

By looking at the second term in (3), minimizing tr(KQ) is equivalent to promoting average smoothness. And by looking at the last equality, the minimization will enforce large off-diagonal negative values on q_{ij} when k_{ij} is large. The log det function acts as a barrier on the minimum eigenvalue of Q, thus enforcing the positive semidefinite constraint. The sign constraints can be handled using Lagrange multipliers.

The Lagrangian is $-\log \det(\mathbf{Q}) + \operatorname{tr}(\mathbf{KQ}) + \operatorname{tr}(\mathbf{\Lambda Q})$, with $\mathbf{\Lambda} = \mathbf{\Lambda}^T = (\lambda_{ij}) \geq 0$ for $i \neq j$ and zero diagonal elements. Slater's condition is easy to check, thus strong duality holds, and since the problem is convex, the solution is unique [26]. Furthermore, the *Karush-Kuhn-Tucker* (KKT) conditions are necessary and sufficient for optimality and we have

$$-\mathbf{Q}^{-1} + \mathbf{K} + \mathbf{\Lambda} = 0 \qquad (4)$$

$$\mathbf{\Lambda}^{T} = \mathbf{\Lambda} \ge 0, q_{ij} = q_{ji} \le 0, \lambda_{ij} q_{ij} = 0, \forall i \neq j.$$
(5)

If **K** is not an inverse M-matrix, then from (4) we deduce the matrix of Lagrange multipliers Λ acts as a perturbation on **K** such that $\mathbf{Q} = (\mathbf{K} + \Lambda)^{-1}$ is a GLP matrix.

3.1. A dual block coordinate descent algorithm

In this section we derive a block coordinate descent algorithm for for solving (1). We analyze the KKT conditions to solve a dual of the algorithm presented in [20]. Consider a partition of \mathbf{K} , $\boldsymbol{\Lambda}$ and \mathbf{Q}^{-1} as shown below for \mathbf{Q}



Fig. 1: Texture graphs using our GLP estimation and the LsigRep method from [11]. We consider *wood* textures, both original and with 60 degree rotation. We normalize each graph by its largest entry and show only off-diagonal entries.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q_{11}} & \mathbf{q_{12}} \\ \mathbf{q_{12}}^T & q_{22} \end{pmatrix}, \tag{6}$$

where $\mathbf{Q_{11}}$ is a $(n-1) \times (n-1)$ sub-matrix, $\mathbf{q_{12}}$ is a column vector of size n-1, and q_{22} is a scalar. The inverse $\mathbf{Q^{-1}}$ has a closed form expression in terms of its block components

$$\mathbf{Q}^{-1} = \begin{pmatrix} (\mathbf{Q}_{11} - \frac{\mathbf{q}_{12}\mathbf{q}_{12}^{T}}{\frac{q_{22}}{12}})^{-1} & -\frac{\mathbf{Q}_{11}^{-1}\mathbf{q}_{12}}{c} \\ -\frac{\mathbf{q}_{12}^{T}\mathbf{Q}_{11}^{-1}}{c} & \frac{1}{c} \end{pmatrix}, \quad (7)$$

with $c = q_{22} - \mathbf{q_{12}^T} \mathbf{Q_{11}^{-1}} \mathbf{q_{12}}$. Suppose $\mathbf{Q}, \mathbf{\Lambda}$ satisfy the KKT conditions, then we can write them for their last row/column and get

$$\frac{\mathbf{Q_{11}}^{-1}\mathbf{q_{12}}}{q_{22} - \mathbf{q_{12}}^{t}\mathbf{Q_{11}}^{-1}\mathbf{q_{12}}} + \mathbf{k_{12}} + \mathbf{\lambda_{12}} = 0$$
(8)

$$q_{22} - \mathbf{q_{12}}^T \mathbf{Q_{11}}^{-1} \mathbf{q_{12}} = \frac{1}{k_{22}}.$$
 (9)

We can combine both equations with the KKT conditions for λ_{12} and q_{12}

$$\mathbf{Q_{11}}^{-1}\mathbf{q_{12}}k_{22} + \mathbf{k_{12}} + \boldsymbol{\lambda_{12}} = 0$$
(10)

$$\lambda_{12} \ge 0 \tag{11}$$

$$\mathbf{q_{12}} \le 0 \tag{12}$$

$$\boldsymbol{\lambda_{12}} \odot \mathbf{q_{12}} = 0, \tag{13}$$

where \odot denotes the Hadamard product (entrywise product). Assuming **Q**₁₁ is fixed, the set of equations (10)-(13) are the KKT conditions for the optimization problem,

$$\min_{\mathbf{q}_{12} \le 0} \frac{1}{2} \mathbf{q}_{12}{}^{T} \mathbf{Q}_{11}{}^{-1} \mathbf{q}_{12} + \frac{1}{k_{22}} \mathbf{q}_{12}{}^{T} \mathbf{k}_{12}.$$
(14)

The block coordinate descent algorithm proposed in [20] iterates over all row/columns of \mathbf{Q} and solves (14). Direct inversion of $\mathbf{Q_{11}}$

at each iteration is avoided by defining $\Sigma = \mathbf{Q}^{-1}$ and updating both of them at each iteration using Schur complements and the block partitioned matrix inverse formula. To avoid updating a possibly dense Σ at each iteration, we instead solve the dual of (14) by rewriting (10)-(13). By multiplying both sides of (10) by \mathbf{Q}_{11} we get

$$\mathbf{q_{12}} + \frac{1}{k_{22}}\mathbf{Q_{11}}(\mathbf{k_{12}} + \boldsymbol{\lambda_{12}}) = 0.$$
 (15)

Considering λ_{12} as the optimization variable, the KKT conditions (11)-(13) and (15) also characterize the solution of

$$\min_{\boldsymbol{\lambda_{12}} \ge 0} (\mathbf{k_{12}} + \boldsymbol{\lambda_{12}})^T \mathbf{Q_{11}} (\mathbf{k_{12}} + \boldsymbol{\lambda_{12}}).$$
(16)

Once λ_{12} is found, $\mathbf{q_{12}}$ can be updated using (15). The diagonal element q_{22} can be updated combining (9) and (15) to get

$$q_{22} = \frac{1}{k_{22}} (1 - \mathbf{q_{12}}^T (\mathbf{k_{12}} + \boldsymbol{\lambda_{12}})) = \frac{1}{k_{22}} (1 - \mathbf{q_{12}}^T \mathbf{k_{12}}).$$
(17)

This procedure can be repeated by iterating over all rows/columns until convergence as shown in Algorithm 1. Notice that if we start with a sparse \mathbf{Q} , and at each iteration set to zero the entries of $\mathbf{q_{12}}$ that have a positive Lagrange multiplier, there will be lower computation and memory use as we are working with sparse matrices and vectors. This is in contrast with the algorithm from [20], in which the dense Σ is updated at each iteration. The methodology to analyze the graphical Lasso algorithm and to derive the primal graphical Lasso (P-GLasso) and dual primal graphical Lasso (DP-GLasso) algorithms from [16] is very similar to what we propose here. Furthermore, the algorithm proposed in [20] has a counterpart in the P-GLasso, while ours corresponds to the DP-GLasso.

3.2. Incorporating additional information

In some cases there is additional information we would like to incorporate. For example in sensor networks, each node represents a Algorithm 1 GLP estimation

Require: $\mathbf{Q}^{(0)} = (\operatorname{diag}(\mathbf{K}))^{-1}$ 1: while not converged do 2: Choose $i_t \in \{1, \dots, n\}$ 3: $\mathbf{Q}_{i_t} \leftarrow \operatorname{remove} i_t$ -th row/column of $\mathbf{Q}^{(t-1)}$ 4: $\mathbf{k}_{i_t} \leftarrow i_t$ -th column from \mathbf{K} and remove its i_t -th entry 5: $k_{i_t} \leftarrow (i_t, i_t)$ entry of \mathbf{K} 6: $\lambda_{i_t} \leftarrow \arg\min_{\beta \ge 0} (\mathbf{k}_{i_t} + \beta)^T \mathbf{Q}_{i_t} (\mathbf{k}_{i_t} + \beta)$ 7: $\mathbf{q}_{i_t} \leftarrow -\frac{1}{k_{i_t}} \mathbf{Q}_{i_t} (\lambda_{i_t} + \mathbf{k}_{i_t})$ (15) 8: $q_{i_t} \leftarrow \frac{1}{k_{i_t}} (1 - \mathbf{q}_{i_t}^T \mathbf{k}_{i_t})$

- 9: $\mathbf{Q}^{(t)} \leftarrow \text{update with } \mathbf{Q}_{i_t}, \mathbf{q}_{i_t} \text{ and } q_{i_t}$
- 10: $t \leftarrow t+1$



Fig. 2: Estimation error of precision matrices using GLP estimation and DP-GLasso from i.i.d. realizations of a Gaussian process with GLP matrix.

sensor, and the graph signal could be a measurement, e.g. temperature. One would expect that sensors that are close to each other should be more strongly connected in the graph. Or in image processing, the distances between pixels should influence the topology of the graph. To add such information, we re-weight the trace term

$$\operatorname{tr}(\mathbf{Q}(\mathbf{K} \odot \mathbf{Z})) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} k_{ij} z_{ij}.$$

A reasonable choice is the kernel $z_{ij} = \exp(-||\mathbf{y}_i - \mathbf{y}_i||^2/\sigma^2)$ where \mathbf{y}_i is the coordinate vector of the *i*-th sensor or pixel. This kernel penalize correlations between sensors that are far apart and leads to a larger prior when two sensors are nearby. Algorithm 1 does not need to be modified, since the only thing that changes is its input. This re-weighting method is similar the ones used in adaptive image filters [27].

4. EXPERIMENTS

4.1. Synthetic Data

In this section we compare our GLP matrix estimation algorithm to learn a graph from simulated data, and compare it to the DP-GLasso algorithm. We consider a GL matrix generated using the Erdos-Renyi model with n = 100 nodes and link probability $l_p = 0.3$ to create an undirected graph. For each edge we assign a random weight uniformly in [0, 1]. We construct a combinatorial Laplacian L and create a GLP as $\mathbf{Q} = \mathbf{I} + \mathbf{L}$, then we compute $\mathbf{K} = \mathbf{Q}^{-1}$. We generate $N \in \{25, 50, 100, 200, 400, 800, 1600, 3200, 6400\}$ i.i.d. realizations of a Gaussian distribution with covariance **K**, then compute the empirical covariance matrix and input that to Algorithm 1. For each N we run the experiment 100 times to find an estimate $\bar{\mathbf{Q}}$ and compute the average between all relative errors $\|\mathbf{Q} - \bar{\mathbf{Q}}\|_F / \|\mathbf{Q}\|_F$. We plot the average relative errors of our method and the DP-GLasso with different regularization parameters in figure 2. For the case N < n, where the empirical covariance is singular, the GL constraints regularize the solution and the GLP estimator behaves like the DP-GLasso with a fixed regularization parameter. For N > n, the empirical covariance estimate is better our method estimates the precision matrix more accurately. Since in that regime, regularization is not necessary the DP-GLasso error converges to a positive value, deviating from the true precision matrix.

4.2. Graph of a texture image

We consider textures from the USC-SIPI Brodatz¹ dataset. In particular, we use two rotated versions of the wood image at 0 and 60 degree angles. We take 8×8 image blocks, vectorize them and construct a covariance matrix K and apply Algorithm 1 with input K, $K \odot W_{\mathbf{Gaussian}},$ and $K \odot W_{\mathbf{8conn}}$ where $W_{\mathbf{Gaussian}}$ is a Gaussian kernel with $\sigma = 2$ that computes distances between pixel coordinates, and W_{8conn} is a mask matrix with values 1 or 0 and $w_{ij} = 1$ iff the *j*-th and *i*-th pixels are vertical or diagonal neighbors and zero otherwise. In figure 1 we show the graphs for each image which consist of the magnitude of the off-diagonal elements of the estimated Precision matrices. In the last columns we show the method from [11]. We manually choose the regularization parameter $\beta = 6$ which roughly controls sparsity, and since we use noiseless data we set denoising parameter $\alpha = 0$. The GLP graphs allow for any type of connection hence the larger weights are well aligned with the texture orientation. By including the Gaussian and 8 connected mask, the GLP estimation algorithm is told to only look at covariances with pixels in a neighborhood, thus encouragint those solutions to connect pixels to their close neighbors. That effect can be seen clearly for the wood060 image, whose graphs lose directionality when constructed using re-weighted covariances. For the wood000 texture the use of re-weighted covariance matrices makes the resulting graphs more regular. All GLP matrices are sparse and have around 100 edges, while a fully connected graph has 2016 edges. In the last column we show the graphs constructed using the method from [11], which also follow the texture orientation precisely.

5. CONCLUSION

In this paper, we formulate the graph learning problem as a precision matrix estimation with Generalized Laplacian constraints. We proposed a novel block coordinate descent algorithm to solve the optimization and apply to precision matrix estimation and texture graph learning. We observe our approach accurately estimates the non zeros as well as the entries of the GLP matrix. For texture images, our algorithm learns a meaningful and sparse graph that follows the texture orientation. Possible extensions include developing techniques to accelerate the optimization and study convergence properties of block coordinate descent.

¹http://sipi.usc.edu/database/

6. REFERENCES

- Hilmi E Egilmez and Antonio Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 1085–1089.
- [2] Ed Bullmore and Olaf Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.
- [3] Amin Kheradmand and Peyman Milanfar, "Motion deblurring with graph laplacian regularization," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 94040C–94040C.
- [4] Cha Zhang and Dinei Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," *Signal Processing Letters, IEEE*, vol. 20, no. 1, pp. 106–109, 2013.
- [5] Wei Hu, Gene Cheung, and Antonio Ortega, "Intra-prediction and generalized graph fourier transform for image coding," *Signal Processing Letters, IEEE*, vol. 22, no. 11, pp. 1913– 1917, Nov 2015.
- [6] Eduardo Pavez, Hilmi E. Egilmez, Yongzhe Wang, and Antonio Ortega, "Gtt: Graph template transforms with applications to image coding," in *Picture Coding Symposium (PCS)*, 2015, May 2015, pp. 199–203.
- [7] Türker Bıyıkoglu, Josef Leydold, and Peter F. Stadler, "Laplacian eigenvectors of graphs," *Lecture notes in mathematics*, vol. 1915, 2007.
- [8] E. Brian Davies, Graham M. L. Gladwell, Josef Leydold, and Peter F. Stadler, "Discrete nodal domain theorems," *Linear Algebra and its Applications*, vol. 336, no. 1, pp. 51–60, 2001.
- [9] Cha Zhang, Dinei Florêncio, and Philip A Chou, "Graph signal processing–a probabilistic framework," *Technical Report*.
- [10] Akshay Gadde and Antonio Ortega, "A probabilistic interpretation of sampling theory of graph signals," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, April 2015, pp. 3257–3261.
- [11] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," arXiv preprint arXiv:1406.7842, 2014.
- [12] Chenhui Hu, Lin Cheng, Jorge Sepulcre, Georges El Fakhri, Yue M. Lu, and Quanzheng Li, "A graph theoretical regression model for brain connectivity learning of alzheimer's disease," in *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*. IEEE, 2013, pp. 616–619.
- [13] Brenden Lake and Joshua Tenenbaum, "Discovering structure by learning sparse graph," in *Proceedings of the 33rd Annual Cognitive Science Conference*. Citeseer, 2010.
- [14] Stephen J Wright, "Coordinate descent algorithms," Mathematical Programming, vol. 151, no. 1, pp. 3–34, 2015.
- [15] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [16] Rahul Mazumder and Trevor Hastie, "The graphical lasso: New insights and alternatives," *Electronic journal of statistics*, vol. 6, pp. 2125, 2012.

- [17] Cho-Jui Hsieh, Mátyás A. Sustik, Inderjit S. Dhillon, Pradeep K. Ravikumar, and Russell Poldrack, "Big & quic: Sparse inverse covariance estimation for a million variables," in Advances in Neural Information Processing Systems, 2013, pp. 3165–3173.
- [18] Tony Cai, Weidong Liu, and Xi Luo, "A constrained 1 minimization approach to sparse precision matrix estimation," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 594–607, 2011.
- [19] Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury, "Covariance selection for nonchordal graphs via chordal embedding," *Optimization Methods & Software*, vol. 23, no. 4, pp. 501–520, 2008.
- [20] Martin Slawski and Matthias Hein, "Estimation of positive definite m-matrices and structure learning for attractive gaussian markov random fields," *Linear Algebra and its Applications*, vol. 473, pp. 145–179, 2015.
- [21] 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," *Signal Processing Magazine*, *IEEE*, vol. 30, no. 3, pp. 83–98, May 2013.
- [22] Aliaksei Sandryhaila and Jose M.F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 80–90, 2014.
- [23] Fan R.K. Chung, Spectral graph theory, vol. 92, American Mathematical Soc., 1997.
- [24] Ky Fan, "Topological proofs for certain theorems on matrices with non-negative elements," *Monatshefte für Mathematik*, vol. 62, no. 3, pp. 219–237, 1958.
- [25] Havard Rue and Leonhard Held, Gaussian Markov random fields: theory and applications, CRC Press, 2005.
- [26] Stephen Boyd and Lieven Vandenberghe, Convex optimization, Cambridge university press, 2004.
- [27] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 106–128, Jan 2013.