LAPLACIAN MATRIX LEARNING FOR SMOOTH GRAPH SIGNAL REPRESENTATION

Xiaowen Dong †*, Dorina Thanou* ‡*, Pascal Frossard* ‡ *and Pierre Vandergheynst* ‡

† Media Lab, MIT, USA
xdong@mit.edu
‡ Signal Processing Laboratories, EPFL, Switzerland
{dorina.thanou, pascal.frossard, pierre.vandergheynst}@epfl.ch

ABSTRACT

The construction of a meaningful graph plays a crucial role in the emerging field of signal processing on graphs. In this paper, we address the problem of learning graph Laplacians, which is similar to learning graph topologies, such that the input data form graph signals with smooth variations on the resulting topology. We adopt a factor analysis model for the graph signals and impose a Gaussian probabilistic prior on the latent variables that control these graph signals. We show that the Gaussian prior leads to an efficient representation that favours the smoothness property of the graph signals, and propose an algorithm for learning graphs that enforce such property. Experiments demonstrate that the proposed framework can efficiently infer meaningful graph topologies from only the signal observations.

Index Terms— Graph learning, graph signal processing, representation theory, factor analysis, Gaussian prior.

1. INTRODUCTION

Modern data processing tasks often manipulate structured data, where signal values are defined on the vertex set V of a weighted and undirected graph G. We refer to such data as graph signals. Due to the irregular structure of the graph domain, processing these signals is a challenging task that combines tools form algebraic and spectral graph theory with computational harmonic analysis [1, 2]. Currently, most of the research effort in the emerging field of signal processing on graphs has been devoted to the analysis and processing of the graph signals in both the vertex and the spectral domain of the graph. The graph however, which is crucial for the successful processing of these signals, is considered to be known a priori or naturally chosen from the application domain. However, there are cases where a good graph is not readily available. It is therefore desirable in these situations to learn the graph topology from the observed data such that it captures the intrinsic relationships between the entities. This is exactly the motivation and objective of this paper.

The key challenge in the problem of graph learning is to choose some meaningful criteria to evaluate the relationships between the signals and the graph topology. In this paper, we are interested in a family of signals that are smooth on a graph. Given a set of signals $X = \{x_i\}_{i=1}^p$, $x_i \in \mathbb{R}^n$, defined on a weighted and undirected graph G of n vertices, we would like to infer an optimal topology of G, namely, its edges and the associated weights, which results in the smoothness of these signals on that graph. More precisely, we want to find an optimal Laplacian matrix for the graph G from the signal observations.

We define the relationship between signals and graphs by revisiting the representation learning theory [3]. Specifically, we consider a factor analysis model for the graph signals, and impose a Gaussian prior on the latent variables that control the observed signals. The transformation from the latent variables to the observed signals involves information about the topology of the graph. As a result, we can define joint properties between the signals and the graph, such that the signal representation is consistent with the Gaussian prior. We then propose an algorithm for graph learning that favours signal representations which are smooth and consistent with the statistical prior defined on the data. Specifically, given the input signal observations, our algorithm iterates between the updates of the graph Laplacian and the signal estimates whose variations on the learned graph are minimized upon convergence.

We test our graph learning algorithm on synthetic data, where we show that it efficiently infers the topology of the groundtruth graphs, by recovering the correct edge positions. We further demonstrate the meaningfulness of the proposed framework on some meteorological signals, where we exploit the spectral properties of the learned graph for clustering its nodes through spectral clustering [4]. The proposed framework is one of the first rigorous frameworks to solve the challenging problem of graph learning in graph signal processing. It provides new insights into the understanding of the interactions between signals and graphs, which could be beneficial in many real world applications, such as the analysis of transportation, biomedical, and social networks.

Finally, it is important to notice that the objective of our graph learning problem is to infer a graph Laplacian operator that can be used for analysing or processing graph signals of the same class as the training signals. This is clearly different from the objective of frameworks for learning Gaussian graphical models [5, 6, 7] proposed in machine learning, where the estimated inverse covariance matrix only represents the conditional dependence structure between the random variables, and cannot be used directly for forming graph signals of given properties[1].

2. FACTOR ANALYSIS FRAMEWORK

We consider the factor analysis [8, 9] model as our signal model, which is a generic linear statistical model that tries to explain observations of a given dimension with a potentially smaller number of unobserved latent variables. Such latent variables usually obey

This work was done while the first author was at EPFL. It was partially supported by the LOGAN project funded by Hasler Foundation, Switzerland.

[1] Although the work in [7] does learn a valid graph topology, their method is essentially similar to the classical approach for sparse inverse covariance estimation, but with a regularized Laplacian matrix.

given probabilistic priors and lead to effective signal representations in the graph signal processing setting, as we show next.

We start with the definition of the Laplacian matrix of a graph G. The unnormalized (or combinatorial) graph Laplacian matrix L is defined as $L = D - W$, where D is the degree matrix that contains the degrees of the vertices along the diagonal, and W is the adjacency matrix of G. Since L is a real and symmetric matrix, it can be decomposed as $L = \chi \Lambda \chi^T$, where χ is the complete set of orthonormal eigenvectors and Λ is the diagonal eigenvalue matrix where the eigenvalues are sorted in increasing order. The smallest eigenvalue is 0 with a multiplicity equal to the number of connected components of the graph [10].

We consider the following model:

$$x = \chi h + u_x + \epsilon, \tag{1}$$

where $x \in \mathbb{R}^n$ represents the observed graph signal, $h \in \mathbb{R}^n$ represents the latent variable that controls the graph signals x, χ is the representation matrix that linearly relates the two random variables, $u_x \in \mathbb{R}^n$ is the mean of x, and ϵ is a multivariate Gaussian noise with mean zero and covariance $\sigma_\epsilon^2 I_n$. The probability density function of ϵ is given by:

$$p(\epsilon) \sim \mathcal{N}(0, \sigma_\epsilon^2 I_n). \tag{2}$$

Moreover, we impose a Gaussian prior on the latent variable h. Specifically, we assume that the latent variable h follows a degenerate zero-mean multivariate Gaussian distribution with precision matrix defined as the eigenvalue matrix Λ of the graph Laplacian L:

$$p(h) \sim \mathcal{N}(0, \Lambda^\dagger). \tag{3}$$

where $\Lambda^\dagger$ is the Moore-Penrose pseudoinverse of Λ. The conditional probability of x given h, and the probability of x, are respectively given as:

$$p(x|h) \sim \mathcal{N}(\chi h + u_x, \sigma_\epsilon^2 I_n), \tag{4}$$

$$p(x) \sim \mathcal{N}(u_x, L^\dagger + \sigma_\epsilon^2 I_n), \tag{5}$$

where we have used in Eq. (5) the fact that the pseudoinverse of L, $L^\dagger$, admits the eigendecomposition $L^\dagger = \chi \Lambda^\dagger \chi^T$.

The representation in Eq. (1) leads to smoothness properties for the signal on the graph. To see this, recall that the latent variables h explain the graph signal x through the representation matrix χ, namely, the eigenvector matrix of the graph Laplacian. Given the observation x and the multivariate Gaussian prior distribution of h in Eq. (3), we are thus interested in a maximum a posteriori (MAP) estimate of h. Specifically, by applying Bayes' rule and assuming without loss of generality that $u_x = 0$, the MAP estimate of the latent variable h can be written as follows [11]:

$$\begin{aligned} h_{\text{MAP}}(x) &\coloneqq \arg\max_h p(h|x) = \arg\max_h p(x|h)p(h) \\ &= \arg\min_h \left(-\log p_E(x - \chi h) - \log p_H(h)\right). \end{aligned} \tag{6}$$

From the probability distributions shown in Eq. (2) and Eq. (3), the above MAP estimate of Eq. (6) can be expressed as:

$$h_{\text{MAP}}(x) = \arg\min_h ||x - \chi h||_2^2 + \alpha\, h^T \Lambda h, \tag{7}$$

where α is some constant parameter. In a noise-free scenario where $x = \chi h$, Eq. (7) corresponds to minimizing the following quantity:

$$h^T \Lambda h = (\chi^T x)^T \Lambda \chi^T x = x^T \chi \Lambda \chi^T x = x^T L x. \tag{8}$$

The Laplacian quadratic term in Eq. (8) is usually considered as a measure of smoothness of the signal x on G [12]. Therefore, we see that in a factor analysis model in Eq. (1), a Gaussian prior in Eq. (3) imposed on the latent variable h leads to smoothness properties for the graph signal. Similar observations can be made in a noisy scenario, where the main component of the signal x, namely, χh, is smooth on the graph. We are going to make use of the above observations in our graph learning algorithm in the following section.

3. LEARNING GRAPH LAPLACIAN UNDER SIGNAL SMOOTHNESS PRIOR

As shown above, given a Gaussian prior in the factor analysis model of the graph signals, the MAP estimate of h in Eq. (7) implies that the signal observations form smooth graph signals. Specifically, notice in Eq. (7) that both the representation matrix χ and the precision matrix Λ of the Gaussian prior distribution imposed on h come from the graph Laplacian L. They respectively represent the eigenvector and eigenvalue matrices of L. When the graph is unknown, we can therefore have the following joint optimization problem of χ, Λ and h in order to infer the graph topology:

$$\min_{\chi, \Lambda, h} ||x - \chi h||_2^2 + \alpha\, h^T \Lambda h. \tag{9}$$

Eq. (9) can be simplified with the change of variable $y = \chi h$ to:

$$\min_{L, y} ||x - y||_2^2 + \alpha\, y^T L y. \tag{10}$$

According to the factor analysis model in Eq. (1), y can be considered as a "noiseless" version of the zero-mean observation x. Due to the properties of the graph Laplacian L, the quadratic form $y^T L y$ in Eq. (10) is usually considered as a measure of smoothness of the signal y on G. Solving the problem of Eq. (10) is thus equivalent to finding jointly the Laplacian L (which is equivalent to the topology of the graph) and the signal y that is close to the observation x and at the same time smooth on the learned graph G. As a result, it enforces the smoothness property of the observed signals on the learned graph.

We propose to solve the optimization problem of Eq. (10) with the following objective function given in a matrix form:

$$\begin{aligned} &\min_{L \in \mathbb{R}^{n \times n}, Y \in \mathbb{R}^{n \times p}} ||X - Y||_F^2 + \alpha\, tr(Y^T L Y) + \beta ||L||_F^2, \\ &\text{s.t.} \quad tr(L) = n, \quad L_{ij} = L_{ji} \leq 0,\ i \neq j, \quad L \cdot \mathbf{1} = \mathbf{0}, \end{aligned} \tag{11}$$

where $X \in \mathbb{R}^{n \times p}$ contains the p input data samples $\{x_i\}_{i=1}^p$ as columns, α and β are two positive regularization parameters, and $\mathbf{1}$ and $\mathbf{0}$ denote the constant one and zero vectors. The first constraint (the trace constraint) in Eq. (11) permits to avoid trivial solutions, and the second and third constraints guarantee that the learned L is a valid Laplacian matrix. The latter is particularly important for two reasons: (i) only a valid Laplacian matrix can lead to the interpretation of the input data as smooth graph signals; (ii) a valid Laplacian allows us to define notions of frequencies in the irregular graph domain, and use successfully already existing signal processing tools on graphs [1]. Furthermore, under the latter constraints, the trace constraint essentially fixes the L^1-norm of L, while the Frobenius norm is added as a penalty term in the objective function to control the distribution of the off-diagonal entries in L, namely, the edge weights of the learned graph.

The optimization problem of Eq. (11) is not jointly convex in L and Y. Therefore, we adopt an alternating optimization scheme

where, at each step, we fix one variable and solve for the other variable. Specifically, at the first step, for a given Y (which at the first iteration is initialized as the input X), we solve the following optimization problem with respect to L:

$$\begin{aligned} &\min_{L} \ \alpha \, tr(Y^T L Y) + \beta ||L||_F^2, \\ &\text{s.t.} \quad tr(L) = n, \quad L_{ij} = L_{ji} \leq 0, \ i \neq j, \quad L \cdot \mathbf{1} = \mathbf{0}. \end{aligned} \tag{12}$$

At the second step, L is fixed and we solve the following optimization problem with respect to Y:

$$\min_{Y} \ ||X - Y||_F^2 + \alpha \, tr(Y^T L Y). \tag{13}$$

Both Eq. (12) and Eq. (13) can be casted as convex optimization problems. The first one is a quadratic program that can be solved efficiently with state-of-the-art convex optimization packages, while the second one has a closed form solution. A detailed description about solving these two problems is presented in [13]. We then alternate between these two steps to get the final solution to the problem of Eq. (11), and we generally observe convergence to a local minimum within a few iterations.

We finally remark that the proposed learning framework has some similarity with the one in [14], where the authors have proposed a similar objective as the one in Eq. (11), based on a smoothness or fitness metric of the signals on graphs. However, we rather take here a probabilistic approach that is analogous to the one in the traditional signal representation setting with the factor analysis model. This gives us an extra data fitting term $||X - Y||_F^2$ in the objective function of the optimization problem of Eq. (11). In practice, when the power of Laplacian is chosen to be 1, the problem in [14] corresponds to finding the solution to a single instance of the problem of Eq. (12) by assuming that $X = Y$.

4. EXPERIMENTS

4.1. Experimental settings

We denote the proposed algorithm as **GL-SigRep** and test its performance by comparing the graph learned from sets of synthetic or real world observations to the groundtruth graph. We provide both visual and quantitative comparisons, where we compare the existence of edges in the learned graph to the ones of the groundtruth graph. In our experiments, we solve the optimization of Eq. (12) using the convex optimization package CVX [15, 16]. The experiments are carried out on different sets of parameters, namely, for different values of α and β in Eq. (11). Finally, we prune insignificant edges that have a weight smaller than 10^{-4} in the learned graph.

We compare the proposed graph learning framework to a state-of-the-art approach for estimating a sparse inverse covariance matrix for Gaussian Markov Random Field (GMRF). Specifically, the works in [5, 6] propose to solve the following L^1-regularized log-determinant program:

$$\min_{L_{\text{pre}} \in \mathbb{R}^{n \times n}} \ tr(S L_{\text{pre}}) - \log \det(L_{\text{pre}}) + \lambda ||L_{\text{pre}}||_1, \tag{14}$$

where L_{pre} is the inverse covariance matrix (or precision matrix) to estimate, $S = XX^T$ is the sample covariance matrix, λ is a regularization parameter, $\det(\cdot)$ denotes the determinant, and $||\cdot||_1$ denotes the L^1-norm. The problem of Eq. (14) is conceptually similar to the problem of Eq. (11), in the sense that both can be interpreted as estimating the precision matrix of a multivariate Gaussian distribution. An important difference is however that the precision matrix in our framework is a valid graph Laplacian, while the one in Eq. (14) is not. Therefore, L_{pre} cannot be interpreted as a graph topology for defining graph signals; it rather only reflects the partial correlations between the random variables that control the observations. As a result, the learning of L_{pre} is not directly linked to the desired properties of the input graph signals. In our experiments, we solve the L^1-regularized log-determinant program of Eq. (14) with the ADMM [17]. We denote this algorithm as **GL-LogDet**. We test **GL-LogDet** based on different choices of the parameter λ in Eq. (14). In the evaluation, all the off-diagonal non-zero entries whose absolute values are above the threshold of 10^{-4} are considered as valid correlations. These correlations are then considered as learned "edges" and compared against the edges in the groundtruth graph for performance evaluation.

4.2. Results on synthetic data

We first carry out experiments on a synthetic graph of 20 vertices. More specifically, we generate the coordinates of the vertices uniformly at random in the unit square, and compute the edge weights between every pair of vertices using the Euclidean distances between them and a Gaussian radial basis function (RBF): $\exp\left(-d(i,j)^2/2\sigma^2\right)$, with the width parameter $\sigma = 0.5$. We remove all the edges whose weights are smaller than 0.75. We then compute the graph Laplacian L and normalize the trace according to Eq. (11). Moreover, we generate 100 signals $X = \{x_i\}_{i=1}^{100}$ that follow the distribution shown in Eq. (5) with $u_x = 0$ and $\sigma_\epsilon = 0.5$. We then apply **GL-SigRep** and **GL-LogDet** to learn the graph Laplacian or the precision matrix, respectively, given only X.

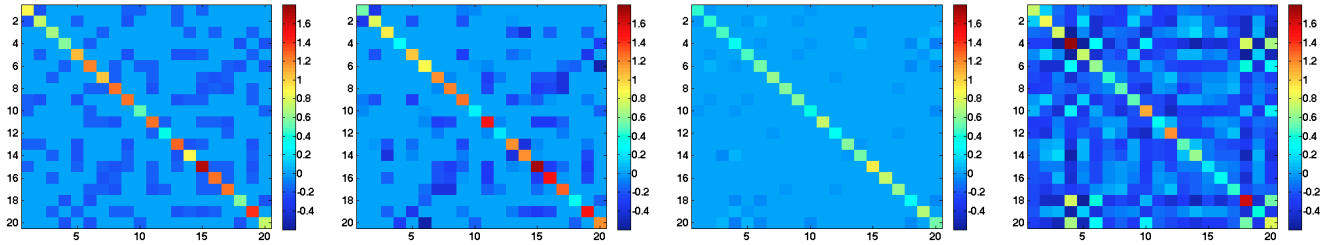
In Fig. 1, we show visually, from the left to the right columns, the Laplacian matrix of the groundtruth graph, the graph Laplacian learned by **GL-SigRep**, the precision matrix learned by **GL-LogDet**, and the sample covariance matrix $S = XX^T$, for one random instance of the Gaussian RBF graph[2]. We see clearly that the graph Laplacian matrix learned by **GL-SigRep** is visually more consistent with the groundtruth data than the precision matrix learned by **GL-LogDet** and the sample covariance matrix.

Next, we evaluate quantitatively the performance of our graph learning algorithm in recovering the positions of the edges in the groundtruth, and we compare to that obtained by **GL-LogDet**. In Table 1, we show the best *F-measure*, *Precision*, *Recall* and *Normalized Mutual Information (NMI)* [18] scores achieved by the two algorithms averaged over ten random instances of the Gaussian RBF graph with the associated signals X. Our algorithm clearly outperforms **GL-LogDet** in terms of all the evaluation criteria. Especially, **GL-SigRep** achieves an average *F-measure* score close to 0.9, which means that the learned graphs have topologies that are very similar to the groundtruth ones. Further discussions about the influence of the parameters in the algorithms, the number of training signals, and the noise level, are presented in [13].

4.3. Learning meteorological graph from temperature data

We now test the proposed graph learning framework on real world data. Specifically, we consider the average monthly temperature data collected at 89 measuring stations in Switzerland during the period between 1981 and 2010. This leads to 12 signals (i.e., one per month), each of dimension 89, which correspond to the average temperatures at each of the measuring stations. By applying the

[2]These results are obtained based on the parameters, namely, α and β in **GL-SigRep** and λ in **GL-LogDet**, that lead to a similar number of edges as the ones in the groundtruth graph. The values of the sample covariance matrix are scaled before the visualization.



(a) Gaussian RBF: Groundtruth (b) Gaussian RBF: **GL-SigRep** (c) Gaussian RBF: **GL-LogDet** (d) Gaussian RBF: Sample covariance

Fig. 1. The learned graph Laplacian or precision matrices. From the left to the right columns are the groundtruth Laplacian, the Laplacian learned by **GL-SigRep**, the precision matrix learned by **GL-LogDet**, and the sample covariance.

Table 1. Performance comparison for **GL-SigRep** and **GL-LogDet**.

Algorithm	F-measure	Precision	Recall	NMI
GL-SigRep	0.8803	0.8535	0.9108	0.5902
GL-LogDet	0.4379	0.2918	0.8851	0.0220

proposed graph learning algorithm, we would like to infer a graph where stations with similar temperature evolutions across the year are connected. In other words, we aim at learning a graph on which the observed temperature signals are smooth. In this case, the natural choice of a geographical graph based on physical distances between the stations does not seem appropriate for representing the similarity of temperature values between these stations. Indeed, we observe that the evolution of temperatures at most of the stations follows very similar trends across the year and are thus highly correlated, regardless of the geographical distances between them. On the other hand, it turns out that altitude is a more reliable source of information to determine temperature evolutions. For instance, as we observed from the data, temperatures at two stations, Jungfraujoch and Piz Corvatsch, follow similar trends that are clearly different from other stations, possibly due to their similar altitudes (both are more than 3000 metres above sea level). Therefore, the goal of our experiment is then to hopefully learn a graph that reflects the altitude relationship between the stations given the observed temperature signals.

We verify our results by separating these measuring stations into disjoint clusters based on the graph learned by **GL-SigRep**, such that different clusters correspond to different characteristics of the stations. In particular, since the learned graph is a valid Laplacian, we can apply the spectral clustering algorithm [4] to partition the vertex set into two disjoint clusters. The results are shown in Fig. 2, where the red and blue dots represent two different clusters of stations. As we can see, the stations in the red cluster are mainly those built on the mountains, such as those in the Jura Mountains and Alps, while the ones in the blue cluster are mainly stations in flat regions. It is especially interesting to notice that, the blue stations in the Alps region (from centre to the bottom right of the map) mainly lie in the valleys along main roads (such as those in the canton of Valais) or in the Lugano region. This shows that the obtained clusters indeed capture the altitude information of the measuring stations hence confirms the quality of the learned graph topology.

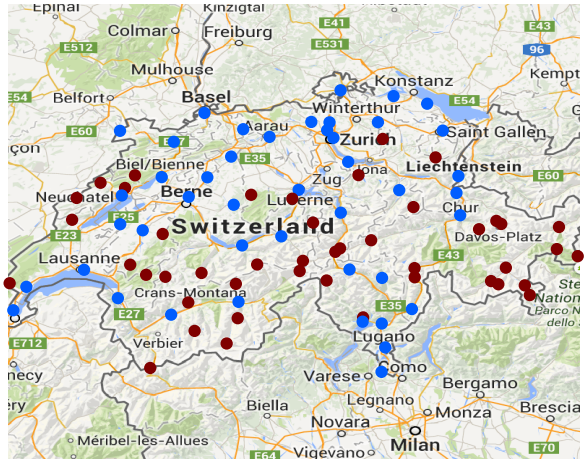


Fig. 2. Two clusters of the measuring stations obtained by applying spectral clustering to the learned graph. The red and blue clusters include stations at higher and lower altitudes, respectively.

5. CONCLUSION

We have presented a framework for learning graph topologies from the signal observations under the assumption that the resulting graph signals are smooth. The framework is based on the factor analysis model and leads to the learning of a valid graph Laplacian matrix that can be used for analysing and processing graph signals. We have demonstrated through experimental results the efficiency of our algorithm in inferring meaningful graph topologies. We believe that the proposed graph learning framework can open new perspectives in the field of signal processing on graphs and can also benefit applications where one is interested in exploiting spectral graph methods for processing data whose structure is not explicitly available.

6. 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] 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.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[4] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001, pp. 849–856.

[5] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, Jun 2008.

[6] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Jul 2008.

[7] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graph," in *Proceedings of the 33rd Annual Cognitive Science Conference*, 2010.

[8] D. J. Bartholomew, M. Knott, and I. Moustaki, "Latent variable models and factor analysis: A unified approach, 3rd Edition," *Wiley*, Jul 2011.

[9] A. Basilevsky, "Statistical factor analysis and related methods," *Wiley*, Jun 1994.

[10] F. R. K. Chung, "Spectral graph theory," *American Mathematical Society*, 1997.

[11] R. Gribonval, "Should penalized least squares regression be interpreted as maximum a posteriori estimation?," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2405–2410, May 2011.

[12] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *ICML Workshop on Statistical Relational Learning*, 2004, pp. 132–137.

[13] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Graphs from Signal Observations under Smoothness Prior," in *arXiv:1406.7842*, 2014.

[14] C. Hu, L. Cheng, J. Sepulcre, G. E. Fakhri, Y. M. Lu, and Q. Li, "A graph theoretical regression model for brain connectivity learning of Alzheimer's disease," in *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, 2013.

[15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," `http://cvxr.com/cvx`, September 2013.

[16] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited, 2008, `http://stanford.edu/~boyd/graph_dcp.html`.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[18] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Cambridge University Press*, 2008.