GRAPH LEARNING BASED ON TOTAL VARIATION MINIMIZATION

Peter Berger, Manfred Buchacher, Gabor Hannak, and Gerald Matz

Institute of Telecommunications, TU Wien (Vienna, Austria)
Email: `firstname.lastname@nt.tuwien.ac.at`

ABSTRACT

We consider the problem of learning the topology of a graph from a given set of smooth graph signals. We construct a weighted adjacency matrix that best explains the data in the sense of achieving the smallest graph total variation. For the case of noisy measurements of the graph signals we propose a scheme that simultaneously denoises the signals and learns the graph adjacency matrix. Our method allows for a direct control of the number of edges and of the weighted node degree. Numerical experiments demonstrate that our graph learning scheme is well suited for community detection.

1. INTRODUCTION

Motivation. Graph signal processing (GSP) is a highly successful new engineering paradigm for dealing with huge data sets since graphs are flexible, efficient, and scalable data models [1–3]. The main idea is that graphs and the associated signal models capture the similarity of data items in an intuitive and versatile manner. GSP has been used to deal with applications ranging from online social networks [4–6], recommender systems [7], and proteomics [8, 9] (see [1] and [3] for further application examples).

A key problem in GSP, which we tackle in this paper, is how to learn the topology of a graph from a given set of graph signals. In particular, our goal is to learn the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ of a graph $\mathcal{G} = (\{1, \ldots, N\}, \mathbf{W})$ such that a given set of M length-N graph signal vectors $\mathbf{x}_m \in \mathbb{R}^N$, $m = 1, \ldots, M$, is smooth on $\mathcal{G}$. In the noisy setting, given noisy measurements $\mathbf{y}_m = \mathbf{x}_m + \mathbf{u}_m \in \mathbb{R}^N$ with noise vectors $\mathbf{u}_m \in \mathbb{R}^N$ ($m = 1, \ldots, M$), our goal is to learn the weighted adjacency matrix $\mathbf{W}$ and simultaneously denoise the measurements to obtain estimates of the graph signals $\mathbf{x}_m$. In contrast to most existing work (see below), we quantify graph signal smoothness in terms of the total variation $\|\mathbf{x}\|_{\mathrm{TV}} = \sum_i \sum_j |x_i - x_j| \, W_{ij}$. The proposed approach complements our previous work on graph signal reconstruction from a small set of noisy signal samples based on total variation minimization [10].

Related Work. One way to approach the graph learning problem is to interpret the graph Laplacian $\mathbf{L}$ as the inverse covariance matrix of a multivariate normal distribution and to adapt sparse inverse covariance estimation for learning Laplacians [11–14]. The underlying assumption is that graph signals that are smooth in the sense that the quadratic form $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is small are more probable to be observed. However, other signal smoothness metrics on graphs such as graph total variation in general cannot be written as a quadratic form and therefore call for a different approach.

The authors of [15] learn a sparse unweighted graph with a given number of edges. In order to avoid that the learned graph is poorly connected, [16, 17] penalizes small weighted node degrees in the objective function. In [18], the connectedness constraint is directly enforced. In contrast, [19] use the Frobenius-norm of the graph Laplacian to penalize large node degrees. In [20, 21] the graph topology is estimated from diffused signals, and in [22] the graph is learned under the assumption that the signals are band-limited on the graph. The authors of [23] construct a b-matched graph where each node has the same number of neighbors. In [24] a graph with normalized node degree is obtained based on the assumption that each data point can be expressed as linear combination of its neighbors. In [25] the graph topology is learned by minimizing the Laplacian quadratic objective function with lower bounds on the node degrees and in [26] the graph on a subsampled image is constructed by connecting pixels which have small distance and similar image luminance.

Contributions. We formulate the graph learning problem as a constraint quadratic program in the graph's edge weights. As opposed to most graph learning algorithms found in the literature we quantify signal smoothness in terms of total variation. All parameters in our scheme have a natural interpretation in terms of the number of neighbors and node degrees and thus we can explicitly control the number of edges of the graph, the minimum number of neighbors of each node, as well as the minimum and maximum degree of each node. In contrast, most existing approaches have several free parameters whose effect on the topology of the learned graph is only implicit, thus rendering these algorithms impractical. Our method doesn't ensure that the learned graph is connected; however, in many applications (e.g., community detection) it is not desirable to enforce connectedness. In the case of noisy data, we propose to alternate between (i) learning the graph using the current signal estimates and (ii) denoising by finding the graph signals with minimal total variation among all

Funding by WWTF Grant ICT15-119.

signals with a bounded measurement error. In our constraint formulation, the noise level can be directly incorporated into the learning problem. In contrast, an appropriate choice for the regularization parameters used, e.g., in [15, 19] seems to be difficult. We confirm the advantages of our total variation based graph learning method via numerical experiments that demonstrate its suitability for cluster/community detection.

2. NOISE-FREE GRAPH LEARNING

Problem Formulation. We consider data given in the form of a noise-free graph signal matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_M) \in \mathbb{R}^{N\times M}$. Let us define the symmetric discrepancy matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ with elements

$$D_{ij} = \sum_{m=1}^{M} |x_{m,i} - x_{m,j}|. \tag{1}$$

For an undirected graph with symmetric weighted adjacency matrix $\mathbf{W} = \mathbf{W}^T \geq \mathbf{0}$ (interpreted element-wise) the aggregate graph total variation for the data matrix $\mathbf{X}$ then equals

$$\sum_{m=1}^{M} \|\mathbf{x}_m\|_{\mathrm{TV}} = \sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij}W_{ij} = \mathrm{tr}\{\mathbf{WD}\}. \tag{2}$$

We assume $\mathrm{diag}(\mathbf{W}) = \mathbf{0}$. Minimizing $\mathrm{tr}\{\mathbf{WD}\}$ aims at enforcing smoothness of the graph signals on the graph associated to $\mathbf{W}$. In order to control the connectivity of the graph, we add the Frobenius norm of $\mathbf{W}$ as penalty term and propose to learn the unknown weight matrix $\mathbf{W}$ by solving the following convex optimization problem (with $\beta > 0$):

$$\begin{aligned} \min_{\mathbf{W}} \quad & \mathrm{tr}\{\mathbf{WD}\} + \frac{\beta}{2}\|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \mathbf{W} = \mathbf{W}^T \geq \mathbf{0}. \\ & \mathrm{diag}(\mathbf{W}) = \mathbf{0}, \\ & \|\mathbf{W}\|_1 = 2N. \end{aligned} \tag{3}$$

The last constraint is a normalization that prevents the trivial solution $\mathbf{W} = \mathbf{0}$. We note that using the Laplacian form as smoothness metric instead of total variation amounts to replacing the discrepancy matrix $\mathbf{D}$ with $\tilde{D}_{ij} = \sum_m (x_{m,i} - x_{m,j})^2$. Since $\mathbf{W}$ and $\mathbf{D}$ are symmetric with zero diagonal, the optimization problem can equivalently be formulated in terms of their upper triangular parts only, which we arrange into vectors $\mathbf{w}$ and $\mathbf{d}$ of length $K = N(N-1)/2$. We can then compactly rewrite (3) as

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^T\mathbf{d} + \frac{\beta}{2}\mathbf{w}^T\mathbf{w} \\ \text{s.t.} \quad & \mathbf{w} \geq 0, \\ & \mathbf{w}^T\mathbf{1} = N. \end{aligned} \tag{4}$$

Solution. The Lagrangian for the optimization problem (4) is given by

$$\mathcal{L}(\boldsymbol{\lambda}, \nu) = \mathbf{w}^T\mathbf{d} + \frac{\beta}{2}\mathbf{w}^T\mathbf{w} - \mathbf{w}^T\boldsymbol{\lambda} + \nu(N - \mathbf{w}^T\mathbf{1})$$

leading to the gradient conditions

$$\nabla_{w_k}\mathcal{L} = d_k + \beta w_k - \lambda_k - \nu = 0, \quad k = 1, \dots, K.$$

Enforcing the complementary slackness conditions [27] $\lambda_k w_k = 0$ implies

$$\begin{aligned} &\lambda_k = 0 \quad \rightarrow \quad w_k = \frac{1}{\beta}(\nu - d_k), \\ &w_k = 0 \quad \rightarrow \quad \lambda_k = d_k - \nu \geq 0 \quad \rightarrow \quad d_k \geq \nu. \end{aligned}$$

We therefore obtain the closed-form solution

$$w_k = \frac{1}{\beta}(\nu - d_k)_+, \tag{5}$$

which amounts to a thresholding of the discrepancies d_k to obtain the graph's edge weights w_k ($z_+ = \max\{0, z\}$ denotes the positive part). This solution is similar to water-filling in communications [28, Section 9.4]. The constraint $\sum_{k=1}^{K} w_k = N$ is enforced by choosing the "water level" ν such that

$$\omega(\nu) = \sum_{k=1}^{K}(\nu - d_k)_+ = \beta N. \tag{6}$$

Since $\omega(\nu)$ is a monotonically increasing piecewise linear function with breakpoints at $\nu = d_k$, larger β results in a higher water level and thus in a graph with more edges. The edges will be placed according to increasing d_k. Let $\tilde{d}_k$ denote the discrepancies sorted according to increasing magnitude. A phase transition that amounts to an additional edge will occur whenever $\nu = \tilde{d}_l$. The associated regularization parameter equals $\beta_l = \omega(\tilde{d}_l)/N$ with

$$\omega(\tilde{d}_l) = \sum_{k=1}^{l}(\tilde{d}_l - \tilde{d}_k) = l\tilde{d}_l - \sum_{k=1}^{l}\tilde{d}_k.$$

By choosing $\beta \in (\beta_l, \beta_{l+1}]$, we can thus directly control the number of edges in the graph to be l.

Weight and Degree Constraints. We next impose additional constraints to take into account prior information regarding the topology of the graph. To enforce minimum and maximum weighted node degrees $\mathbf{W1}$ we propose to augment (3) with the constraints $\mathbf{a} \leq \mathbf{W1} \leq \mathbf{b}$ with $\mathbf{a}, \mathbf{b} \geq 0$. An additional upper bound c on the edge weights allows us to enforce a minimum number of neighbors per node. We thus arrive at the augmented quadratic program

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^T\mathbf{d} + \frac{\beta}{2}\mathbf{w}^T\mathbf{w} \\ \text{s.t.} \quad & \mathbf{0} \leq \mathbf{w} \leq c\mathbf{1}, \\ & \mathbf{w}^T\mathbf{1} = N, \\ & \mathbf{a} \leq \mathbf{Sw} \leq \mathbf{b}. \end{aligned} \tag{7}$$

Algorithm 1 TV-based graph learning and denoising

```
input: $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$, $\beta$, $\mathbf{a}$, $\mathbf{b}$, $c$, $\varepsilon$
initialize: $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}$, $n = 0$
repeat
    compute $\mathbf{d}(\hat{\mathbf{X}}^{(n)})$ according to (1)
    determine $\hat{\mathbf{w}}^{(n)}$ by solving (7)
    for $m = 1, \dots, M$ do
        compute $\hat{\mathbf{x}}_m^{(n+1)}$ via (9) with weights $\hat{\mathbf{w}}^{(n)}$
    end for
    $n = n + 1$
until stopping criterion is satisfied
output: $\widehat{\mathbf{W}} = \widehat{\mathbf{W}}^{(n-1)}$, $\hat{\mathbf{X}} = \hat{\mathbf{X}}^{(n)}$
```

Here, $\mathbf{S} \in \{0,1\}^{N \times K}$ is constructed such that $\mathbf{S}\mathbf{w} = \mathbf{W}\mathbf{1}$. As an example, consider $\mathbf{a} = \mathbf{b} = \mathbf{2}$ and $c = 2/L$; here, the solution of (7) yields a graph where all nodes have weighted node degree 2 and at least L neighbors. We note that (7) has no closed form solution but can be solved efficiently via interior point methods [27] or via ADMM [29, 30].

3. NOISY CASE

When the observed data $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$ are noisy measurements of unknown actual graph signals $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$, we propose to simultaneously learn the graph and denoise the data by modifying (7) as follows:

$$
\begin{aligned}
\min_{\mathbf{w}, \mathbf{X}} \quad & \mathbf{w}^T \mathbf{d}(\mathbf{X}) + \frac{\beta}{2} \mathbf{w}^T \mathbf{w} \\
\text{s.t.} \quad & \mathbf{0} \le \mathbf{w} \le c\mathbf{1}, \\
& \mathbf{w}^T \mathbf{1} = N, \\
& \mathbf{a} \le \mathbf{S}\mathbf{w} \le \mathbf{b}, \\
& \|\mathbf{x}_m - \mathbf{y}_m\|_2 \le \varepsilon,
\end{aligned}
\tag{8}
$$

where the discrepancies $\mathbf{d}(\mathbf{X})$ depend on the signals $\mathbf{x}_m$ and ε controls the amount of measurement noise. The critical term in (8) is the total variation

$$
\mathbf{w}^T \mathbf{d}(\mathbf{X}) = \frac{1}{2} \sum_{m=1}^{M} \|\mathbf{x}_m\|_{\mathrm{TV}} = \sum_{i<j} \sum_{m=1}^{M} W_{ij} \, |x_{m,i} - x_{m,j}|,
$$

which is linear in the weights $\mathbf{w}$ and convex in the graph signals $\mathbf{X}$ but not jointly convex in $\mathbf{W}$ and $\mathbf{X}$. Following [19], we search for a local minimum of (8) by alternately performing the (individually convex) minimizations with respect to $\mathbf{w}$ and $\mathbf{X}$, which amounts to the following iterations (summarized in Algorithm 1, n is the iteration index):

1. learn graph weights $\hat{\mathbf{w}}^{(n)}$ via (7) based on the current graph signal estimates $\hat{\mathbf{X}}^{(n)}$;
2. obtain new graph signal estimates $\hat{\mathbf{X}}^{(n+1)}$ by denoising the data $\mathbf{Y}$ using the the current graph weights $\hat{\mathbf{w}}^{(n)}$, i.e., for $m = 1, \dots, M$, solve the convex problem

$$
\begin{aligned}
\min_{\mathbf{x}_m} \quad & \|\mathbf{x}_m\|_{\mathrm{TV}} = \sum_{i=1}^{N} \sum_{j=1}^{N} \widehat{W}_{ij}^{(n)} |x_{m,i} - x_{m,j}| \\
\text{s.t.} \quad & \|\mathbf{x}_m - \mathbf{y}_m\|_2 \le \varepsilon,
\end{aligned}
\tag{9}
$$

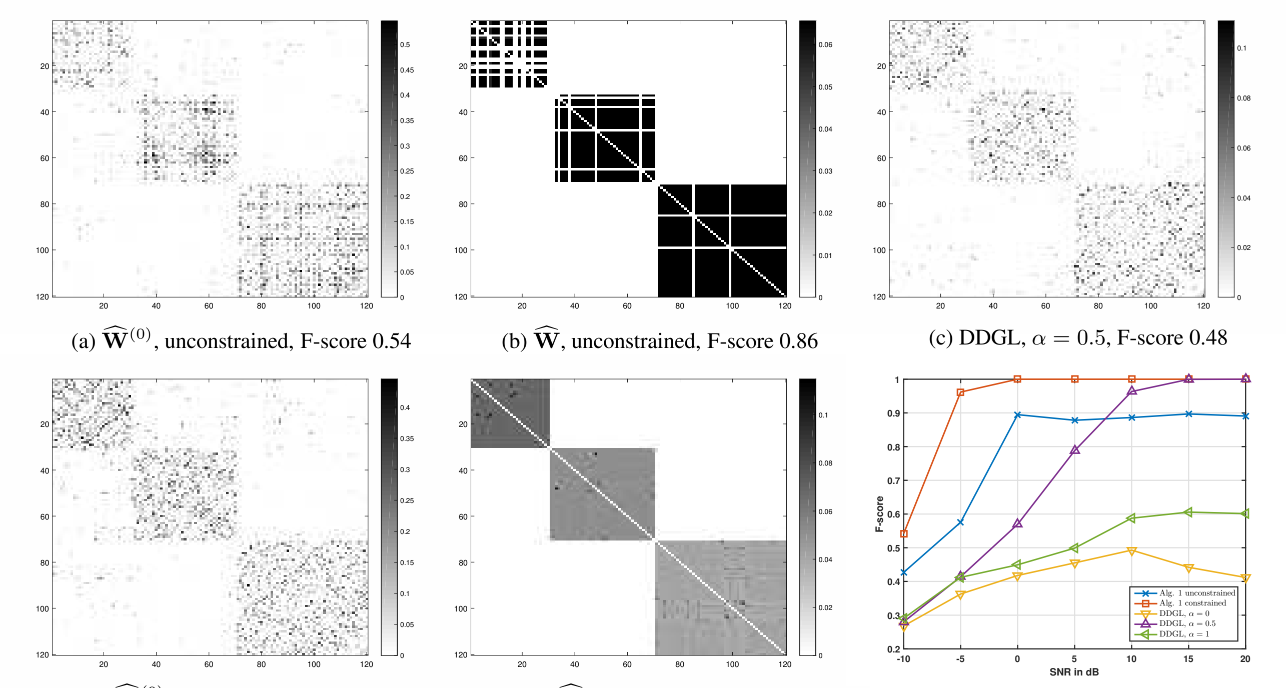
The procedure is initialized with $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}$. For $n \ge 1$ the numerical solvers used can be warm-started using the estimates $\hat{\mathbf{w}}^{(n-1)}$ and $\hat{\mathbf{X}}^{(n)}$ from the previous iteration.

The optimization problem (9) could be solved via first-order primal-dual algorithms [10]. However, due to the advantages provided by the varying penalty strategy [31, Section 2.3] and the reasonable stopping criterion, we rather advocate using the augmented ADMM method from [31] with the closed form expressions for the proximal operators elaborated in [10] and the scaling matrix [31] chosen as $s\mathbf{I}$ with $s = 4 \max_i (\sum_j (\widehat{W}_{ij}^{(n)})^2)$.

4. NUMERICAL EXPERIMENTS

We next demonstrate that our graph learning scheme is well suited for cluster/community detection. We consider a graph consisting of three clusters with 30, 40 and 50 nodes, respectively ($N = 120$, $K = 7140$). Nodes within the same cluster are all connected and there are no edges between different clusters. The noise-free graph signal values within each cluster are identical (i.e.,the graph signals are piecewise constant on the clusters). The cluster signal values are drawn from a standard Gaussian distribution. The observed data consists of $M = 50$ graph signals that are corrupted by zero-mean Gaussian noise with average power σ^2. Since the average graph signal power is one, the signal-to-noise ratio is $\mathrm{SNR} = \frac{1}{\sigma^2}$. Accordingly, the bound on the empirical errors in (9) was set to $\varepsilon = \sqrt{N}\sigma$. Unless stated otherwise, the signal-to-noise ratio was $-3\,\mathrm{dB}$ (i.e., more noise power than signal power). The reconstruction performance is assessed via the F-score [11] (i.e., the harmonic mean of edge precision and recall). The better the learned graph approximates the ground truth, the closer the F-score is to 1.

We first learn the graph without degree and weight constraints ($\mathbf{a} = \mathbf{0}$, $\mathbf{b} = \infty \cdot \mathbf{1}$ and $c = \infty$). This means that step 1 (weight learning) in our scheme uses the explicit solutions (5) for the weights and only step 2 (denoising) requires a numerical solver. Choosing the initial water level as $\nu_0 > \max_i \{\min_{j \ne i} D_{ij}(\mathbf{Y})\}$ guarantees that for each node at least one weight is nonzero and hence there won't be any isolated nodes in $\widehat{\mathbf{W}}^{(0)}$. In all experiments we set $\beta = 1.001\beta_0$ where β_0 is determined from ν_0 via (6). Since the denoising step (9) tends to decrease the elements of the discrepancy matrix $\mathbf{D}(\mathbf{X}^{(n)})$, the water level may decrease during the iterations and possibly result in isolated nodes. We stopped



(a) $\widehat{\mathbf{W}}^{(0)}$, unconstrained, F-score 0.54

(b) $\widehat{\mathbf{W}}$, unconstrained, F-score 0.86

(c) DDGL, $\alpha = 0.5$, F-score 0.48

(d) $\widehat{\mathbf{W}}^{(0)}$, constrained, F-score 0.61

(e) $\widehat{\mathbf{W}}$, constrained, F-score 1

(f) F-score versus SNR in dB

Fig. 1: Initial and the final weight matrix estimate $\mathbf{W}_0$ and $\hat{\mathbf{W}}$, (a), (b) without constraints, (d), (e) with uniform weighted node degree constraint; (c) the DDGL weight matrix estimate; (f) F-score vs. SNR for different graph learning schemes.

Algorithm 1 as soon as the relative change between two successive weight matrix estimates was smaller than $\delta = 1\%$, i.e., $\|\widehat{\mathbf{W}}^{(n)} - \widehat{\mathbf{W}}^{(n-1)}\|_{\mathrm{F}} \leq \delta\,\|\widehat{\mathbf{W}}^{(n)}\|_{\mathrm{F}}$. In this setup, Algorithm 1 stopped after five iterations. Figures 1(a) and (b) show the learned weight matrices $\widehat{\mathbf{W}}^{(0)}$ and $\widehat{\mathbf{W}} = \widehat{\mathbf{W}}^{(4)}$, which achieve F-scores of 0.54 and 0.86, respectively. Given the large amount of noise in the graph signal measurements, this performance is surprisingly good.

We next incorporate the prior knowledge that the graph has no isolated nodes into our learning algorithm by choosing the upper and lower bound on the weighted node degrees as $\mathbf{a} = \mathbf{b} = 21$, which forces each node to have the same weighted node degree and to have at least one neighbor. The weight matrices $\widehat{\mathbf{W}}^{(0)}$ (F-score 0.61) and $\widehat{\mathbf{W}} = \widehat{\mathbf{W}}^{(4)}$ (F-score 1) obtained with Algorithm 1 are shown in Figures 1(d) and (e), respectively. Here, the additional constraints on the weighted node degrees allow all clusters to be correctly identified via the final weight matrix estimate $\hat{\mathbf{W}}$.

We finally compare the graph recovery performance of our method with the graph learning method based on diagonally dominant generalized Laplacian (DDGL) from [11]. Figure 1(c) shows the weight matrix of the graph learned by the DDGL algorithm with regularization parameter $\alpha = 0.5$, which achieves an F-score of 0.48, which is much worse than the results obtained with our method. In our final experiment, we compare the graph reconstruction quality of Algorithm 1 and DDGL (with regularization parameters $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1$) at different noise levels (corresponding to SNRs from -10 to 20 dB). All results have been averaged over 5 independent realizations of the signals and the noise. Figure 1(f) shows the F-scores achieved versus SNR. Algorithm 1 with uniform weighted node degree constraint clearly performs best and accurately recovers the clusters at SNRs as low as -5 dB. Without node degree constraints, Algorithm 1 performs a bit worse, with an F-score that saturates in the high SNR regime. The best regularization parameter for DDGL is $\alpha = 0.5$, which substantially outperforms $\alpha = 0$ and $\alpha = 1$. At very high SNR, DDGL with $\alpha = 0.5$ even outperforms the unconstrained version of our scheme. However, its performance remains worse than that of the constrained version of Algorithm 1 at all SNRs.

5. CONCLUSION

We considered graph learning based on total variation minimization with noise-free and noise-corrupted data. We provided insights regarding the effects of all free parameters and explained how they should be chosen. This renders our framework transparent and practically easy to use in real-world applications. Our numerical experiments illustrated that our learning approach is well suited for cluster identification even in high noise regimes.

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 Process. Mag.*, 30(3):83–98, May 2013.

[2] A. Sandryhaila and J. M. F. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Process. Mag.*, 31(5):80–90, Sept. 2014.

[3] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Trans. Signal Process.*, 61(7):1644–1656, Apr. 2013.

[4] S. Cui, A. Hero, Z.-Q. Luo, and J. M. F. Moura. *Big Data over Networks*. Cambridge Univ. Press, 2016.

[5] P. S. Dodds and D. J. Watts. Universal behavior in a generalized model of contagion. *Physical Review Letters*, 92(21):218701, May 2004.

[6] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, July 2012.

[7] W. Xing and A. Ghorbani. Weighted pagerank algorithm. In *Proc. Annual Conf. Commun. Networks and Services Research*, pages 305–314, Fredericton, Canada, May 2004.

[8] R. Saidi, M. Maddouri, and E.M. Nguifo. Protein sequences classification by means of feature extraction with substitution matrices. *BMC Bioinformatics*, 11(1):1–13, Apr. 2010.

[9] P. Bühlmann and S. v. d. Geer. *Statistics for High-Dimensional Data*. Springer, 2011.

[10] P. Berger, G. Hannak, and G. Matz. Graph signal recovery via primal-dual algorithms for total variation minimization. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):842–855, Sept. 2017.

[11] H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under structural and laplacian constraints. *arXiv:1611.05181*, 2016.

[12] M. Slawski and M. Hein. Estimation of positive definite M-matrices and structure learning for attractive Gaussian Markov random fields. *Linear Algebra Appl.*, 473:145–179, May 2015.

[13] E. Pavez and A. Ortega. Generalized laplacian precision matrix estimation for graph signal processing. In *Proc. IEEE ICASSP*, pages 6350–6354, Shanghai, China, Mar. 2016.

[14] Brenden M. Lake and Joshua B. Tenenbaum. Discovering structure by learning sparse graph. In *Proceedings of the 32rd Annual Cognitive Science Conference*, pages 778–783, Portland, Oregon, USA, Aug. 2010.

[15] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero. Learning sparse graphs under smoothness prior. In *icassp*, pages 6508–6512, New Orleans, USA, Mar. 2017.

[16] V. Kalofolias. How to learn a graph from smooth signals. In *Proc. Int. Conf. on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 920–929, Cadiz, Spain, May 2016.

[17] V. Kalofolias and N. Perraudin. Large scale graph learning from smooth signals. *arXiv:1710.05654*, 2017.

[18] M. Sundin, A. Venkitaraman, M. Jansson, and S. Chatterjee. A connectedness constraint for learning sparse graphs. In *25th European Signal Processing Conference, EUSIPCO*, pages 161–165, Kos island, Greece, Aug. 2017.

[19] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning Laplacian matrix in smooth graph signal representations. *IEEE Trans. Signal Process.*, 64(23):6160–6173, Dec. 2016.

[20] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro. Network topology inference from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):467–483, Sep. 2017.

[21] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat. Characterization and inference of graph diffusion processes from observations of stationary signals. *IEEE Transactions on Signal and Information Processing over Networks*, PP(99):1–1, Aug. 2017.

[22] S. Sardellitti, S. Barbarossa, and P. D. Lorenzo. Graph topology inference based on transform learning. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 356–360, Dec. 2016.

[23] T. Jebara, J. Wang, and S. Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 441–448, Montreal, Quebec, Canada, June 2009.

[24] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, Jan. 2008.

[25] S. I. Daitch, J. A. Kelner, and D. A. Spielman. Fitting a graph to vector data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 201–208, June 2009.

[26] S. Colonnese, M. Biagi, R. Cusani, and G. Scarano. Graph adjacency matrix learning for irregularly sampled markovian natural images. In *25th European Signal Processing Conference, EUSIPCO*, pages 395–399, Kos island, Greece, Aug. 2017.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2004.

[28] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.

[30] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. *IEEE Trans. Automat. Control*, 60(3):644–658, Mar. 2015.

[31] Y. Zhu. An Augmented ADMM Algorithm With Application to the Generalized Lasso Problem. *J. Comput. Graph. Statist.*, 26(1):195–204, Feb. 2017.