

A Connectedness Constraint for Learning Sparse Graphs

Martin Sundin, Arun Venkitaraman, Magnus Jansson, Saikat Chatterjee
 ACCESS Linnaeus Center, KTH Royal Institute of Technology, Sweden
 Email: masundi@kth.se, arunv@kth.se, janssonm@kth.se, sach@kth.se

Abstract—Graphs are naturally sparse objects that are used to study many problems involving networks, for example, distributed learning and graph signal processing. In some cases, the graph is not given, but must be learned from the problem and available data. Often it is desirable to learn sparse graphs. However, making a graph highly sparse can split the graph into several disconnected components, leading to several separate networks. The main difficulty is that connectedness is often treated as a combinatorial property, making it hard to enforce in e.g. convex optimization problems. In this article, we show how connectedness of undirected graphs can be formulated as an analytical property and can be enforced as a convex constraint. We especially show how the constraint relates to the distributed consensus problem and graph Laplacian learning. Using simulated and real data, we perform experiments to learn sparse and connected graphs from data.

I. INTRODUCTION

Graphs are naturally sparse objects which describe relations between different data sources. Graphs are classically used in network problems, such as distributed estimation [1]–[4]. Other examples are webpage ranking [5], [6] and relations in social networks [7], [8]. More recently, graphs have been used to generalize signal processing concepts, such as transforms, to signals defined on graphs [7]–[12]. Often, the graph in question is sparse. For example, in social networks, there are many users and each user is only connected to few other users, resulting in a sparse graph. In many applications, the graph is not given a-priori but needs to be learned from data [13]–[18]. When learning sparse graphs, one challenge is that promoting sparsity may make the graph disconnected. Connectedness is hard to incorporate as it is often treated as a combinatorial property [18]. In this article we show that connectedness is an analytical property that can be formulated as a convex constraint and subsequently be used efficiently in learning.

A. Preliminaries and notations

A graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ is commonly defined as a set of nodes/vertices $\mathcal{V} = \{1, 2, \dots, N\} = [N]$ and edges $\mathcal{E} \subset [N] \times [N]$ together with an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. Two nodes $k \in V$ and $l \in V$ share an edge if $(k, l) \in \mathcal{E}$. The elements of the adjacency matrix describes the *strength* of the connection between two nodes. The relation between \mathcal{E} and \mathbf{A} is: $(k, l) \in \mathcal{E} \Leftrightarrow A_{kl} \neq 0$. For undirected graphs, the adjacency matrix \mathbf{A} is symmetric. In this article we consider real symmetric adjacency matrices with non-negative components.

We denote the matrix determinant by $\det(\cdot)$, the matrix trace by $\text{tr}(\cdot)$. We use \succ (\succeq) to denote positive (semi) definiteness

of symmetric matrices and \geq to denote element-wise greater or equal. We use $\mathbf{1} = (1, 1, 1, \dots, 1)^\top \in \mathbb{R}^N$ to denote the vector consisting of only ones and \mathbf{I}_N to denote the $N \times N$ identity matrix. The k 'th largest eigenvalue of a matrix is denoted by $\lambda_k(\cdot)$ and the element-wise ℓ_1 -norm by $\|\mathbf{A}\|_1 = \sum_{k,l} |A_{kl}|$.

B. Problem statement

Many graph learning problems can be posed as finding the adjacency matrix \mathbf{A} that minimizes an objective function $g(\mathbf{A})$ under appropriate constraints. To make the adjacency matrix sparse, a sparsity promoting penalty function $s(\mathbf{A})$ is often introduced. The graph is thus obtained by solving the optimization problem:

$$\min_{\mathbf{A}} g(\mathbf{A}) + \eta s(\mathbf{A}), \text{ subject to } \textit{constraints} \quad (1)$$

where $\eta > 0$ is a regularization parameter and \mathbf{A} is symmetric. A few examples of $g(\mathbf{A})$, $s(\mathbf{A})$ and *constraints* are shown in Table I. Signal Processing on Graphs (SPG) and the graphical LASSO are data-driven problems while the distributed consensus problem only uses the topology of an underlying graph. We note that all optimization problems in Table I are convex.

Often, the learned graph becomes disconnected for large values of η . This means that the resulting graph describes two or more separate non-interacting systems. In this article, we address the issue of learning sparse connected graphs by formulating a convex constraint which preserves connectedness. Our contributions are as follows:

- 1) We analytically formulate connectedness in terms of a weighted Laplacian matrix.
- 2) For the distributed consensus problem we show that the graph splits into several components when the parameter η exceeds some value.

We illustrate the validity of the proposed constraint through numerical simulations considering both synthetic data and real world temperature data.

II. PRESERVING THE CONNECTEDNESS OF GRAPHS

The connectedness of a graph is usually described by the spectrum of the graph Laplacian matrix defined using the incidence matrix [19]. As the Laplacian is not a continuous function of the adjacency matrix \mathbf{A} , preserving connectedness in terms of the Laplacian matrix leads to combinatorial op-

Problem	$g(\mathbf{A})$	$s(\mathbf{A})$	constraints
Consensus [4], [15], [16]	$\mu(\mathbf{A}) = \max\{\lambda_2(\mathbf{A}), -\lambda_N(\mathbf{A})\}$	$\ \mathbf{A}\ _1 - \text{tr}(\mathbf{A})$	$\mathbf{A}\mathbf{1} = \mathbf{1}, \mathbf{A} \geq \mathbf{0}, A_{ij} = 0$ for $(i, j) \notin \mathcal{E}$
Graphical LASSO [13]	$\text{tr}(\hat{\mathbf{R}}\mathbf{A}) - \log \det(\mathbf{A})$	$\ \mathbf{A}\ _1$	$\mathbf{A} \succeq \mathbf{0}$
SPG [12], [17]	$\text{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top)$	$\ \mathbf{A}\ _1$	$\mathbf{L} = \text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$

TABLE I
FEW EXAMPLES OF $g(\mathbf{A})$, $s(\mathbf{A})$ AND *constraints* THAT CAN BE USED IN (1).

timization problems. For this reason we instead consider the weighted graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ with elements

$$L_{kl} = \begin{cases} \sum_{m \neq k} A_{km} & , \text{ if } k = l, \\ -A_{kl} & , \text{ if } k \neq l, \end{cases}$$

the matrix can thus be expressed as $\mathbf{L} = \text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$. We note that for $\mathbf{x} = [x_1, \dots, x_N]^\top$

$$\begin{aligned} \mathbf{x}^\top \mathbf{L} \mathbf{x} &= \sum_{k,l} A_{kl} (x_k^2 - x_k x_l) = \frac{1}{2} \sum_{k,l} A_{kl} (x_k^2 - 2x_k x_l + x_l^2) \\ &= \frac{1}{2} \sum_{k,l} A_{kl} (x_k - x_l)^2 = \sum_{k < l} A_{kl} (x_k - x_l)^2. \end{aligned}$$

The weighted Laplacian is thus positive semi-definite when $A_{ij} \geq 0$. The nullspace of the weighted Laplacian relates to the number of connected components through the following lemma.

Lemma 1 (Connected components). The number of connected components of a graph with adjacency matrix \mathbf{A} equals to the dimension of the null space of \mathbf{L} .

The proof of Lemma 1 is similar to the proof for the (unweighted) Laplacian in [19] and is therefore not repeated here. Lemma 1 gives that a graph is connected if and only if the second smallest eigenvalue of \mathbf{L} (also known as the Fiedler value) is nonzero. By noting that $\text{span}\{\mathbf{1}\} \subset \text{null}(\mathbf{L})$, we find the following proposition.

Proposition 1 (Graph connectedness constraint). A graph with adjacency matrix $\mathbf{A} \geq \mathbf{0}$ is connected if and only if

$$\mathbf{L} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top = \text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top \succ \mathbf{0}. \quad (2)$$

Proof. Let the graph be $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$. Assume first that there exists a vector $\mathbf{x} \neq \mathbf{0}$ such that

$$\mathbf{x}^\top \left(\mathbf{L} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top \right) \mathbf{x} = \sum_{i > j} A_{ij} (x_i - x_j)^2 + \frac{1}{N} \left(\sum_{i=1}^N x_i \right)^2,$$

is zero. This means that $\sum_{i=1}^N x_i = 0$ and $x_i = x_j$ for all i and j such that $A_{ij} > 0$. Hence, $x_j = x_i$ for all nodes $j \in V$ connected to a the node $i \in V$, the vector \mathbf{x} is thus piecewise constant over the graph. However, the vector cannot be completely constant since $\sum_{i=1}^N x_i = 0$ and $\mathbf{x} \neq \mathbf{0}$ by assumption. The graph must therefore consist of at least two components.

Next, assume that the graph is disconnected. Then there exists two sets C and D such that $C \cup D = \mathcal{V}$, $C \cap D = \emptyset$ and $A_{ij} = 0$ for all $(i, j) \in C \times D$. Set the elements of

\mathbf{x} be $x_i = 1/|C|$ for $i \in C$ and $x_j = -1/|D|$ for $j \in D$ where $|C|$ and $|D|$ denotes the number of elements in C and D respectively. We find that

$$\mathbf{x}^\top (\mathbf{L} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top) \mathbf{x} = \frac{1}{N} (|C||C| - |D||D|)^2 = 0.$$

Hence $\mathbf{L} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top \not\succeq \mathbf{0}$. \square

Proposition 1 immediately implies that the solution to the graph learning problem (1) is connected when the constraint

$$\text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top \succeq \epsilon \mathbf{I}_N, \quad (3)$$

is imposed for some $\epsilon > 0$ and $A_{ij} \geq 0$.

For the graphical Lasso problem [13], the components of the adjacency matrix \mathbf{A} are not necessarily non-negative. However, the sign pattern of the solution is the same as that of $\hat{\mathbf{R}}$ in the sense that $\hat{R}_{ij} A_{ij} \geq 0$ for all (i, j) [20], [21]. One can therefore replace the adjacency matrix in (3) by an adjacency matrix $\hat{\mathbf{A}}$ with components $\hat{A}_{ij} = \text{sign}(\hat{R}_{ij}) A_{ij}$. The constraint can therefore also be used in problems with negative adjacency matrices provided that the sign pattern of \mathbf{A} is known.

We next consider the application of the constraint to signal processing on graphs and the distributed consensus problem.

III. APPLICATIONS

A. The consensus problem

In the distributed consensus problem [4], [15], [16], we iteratively compute the mean of a sequence $\{x_i(0)\}_{i=1}^N$ as

$$x_i(t+1) = \sum_j A_{ij} x_j(t); \quad t = 0, 1, 2, \dots,$$

where $A_{ij} \geq 0$ and $x_j(t)$ denotes the value at the j 'th node at the t 'th iteration. The iterations converge to the mean as $t \rightarrow \infty$ when $\mathbf{A}\mathbf{1} = \mathbf{1}$ and $\mu(\mathbf{A}) = \max\{\lambda_2(\mathbf{A}), -\lambda_N(\mathbf{A})\} < 1$ [4]. Smaller $\mu(\mathbf{A})$ leads to faster (worst case) convergence.

In some scenarios, for example when the number of communication links is limited, it is desirable to obtain a sparse graph [15], [16]. This can be done by setting [16]

$$s(\mathbf{A}) = \sum_{i \neq j} A_{ij} = N - \text{tr}(\mathbf{A}) = \text{tr}(\mathbf{I}_N - \mathbf{A}) = \text{tr}(\mathbf{L}).$$

The optimization problem (1) then becomes [16]:

$$\begin{aligned} &\min \mu(\mathbf{A}) + \eta \text{tr}(\mathbf{I}_N - \mathbf{A}), \\ &\text{subject to } \mathbf{A}\mathbf{1} = \mathbf{1}, \mathbf{A} \geq \mathbf{0}, \mathbf{A}^\top = \mathbf{A} \\ &A_{kl} = 0, \text{ for all } (k, l) \notin \mathcal{E}. \end{aligned} \quad (4)$$

The optimization problem (4) does not ensure that the graph is connected. In fact, we now prove that the solution to (4) is guaranteed to be disconnected for certain values of η .

Proposition 2 (Consensus graph splitting with sparsity penalty). In the consensus problem (4), the graph with N nodes and adjacency matrix \mathbf{A} splits up into at least k connected components when

$$\eta > 1/(N - k + 1), \text{ for } k = 1, 2, \dots, N.$$

Proof. We prove the proposition by induction over k . The graph consists of at least one connected component, so the proposition holds for $k = 1$. Assume that the proposition holds for $k - 1$. Since $\mathbf{L} \succeq \mathbf{0}$, the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N = 0$ of \mathbf{L} are non-negative. The induction hypothesis gives that $\lambda_{N-k+2} = \lambda_{N-k+3} = \dots = \lambda_N = 0$. We find that

$$\begin{aligned} 1 &= \mu(\mathbf{I}_N) \geq \mu(\mathbf{A}) + \eta \text{tr}(\mathbf{I}_N - \mathbf{A}) = \mu(\mathbf{I}_N - \mathbf{L}) + \eta \text{tr}(\mathbf{L}) \\ &\geq 1 - \lambda_{N-k+1} + \eta(\lambda_{N-k+1} + \lambda_{N-k+2} + \dots + \lambda_1) \\ &\geq 1 + \lambda_{N-k+1} (\eta(N - k + 1) - 1). \end{aligned}$$

So $\lambda_{N-k+1} (\eta(N - k + 1) - 1) \leq 0$. Since $\lambda_{N-k+1} \geq 0$, we find that $\lambda_{N-k+1} = 0$ if $\eta > \frac{1}{N-k+1}$. Hence, for $\eta > \frac{1}{N-k+1}$, $\dim(\text{null}(\mathbf{L})) \geq k$ and the graph consists of at least k connected components. \square

Proposition 2 generalizes the result from [16] which states that the graph becomes completely disconnected (consists of N independent components) when $\eta > 1$.

Next we examine the effect of imposing the connectedness constraint (3) on the consensus problem. The constraint can be rewritten as

$$\mathbf{A} - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \preceq (1 - \epsilon) \mathbf{I}_N.$$

This gives us that

$$\lambda_2(\mathbf{A}) = \lambda_1 \left(\mathbf{A} - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \right) \leq 1 - \epsilon.$$

This shows that the graph is connected when $\lambda_2(\mathbf{A}) \leq \mu(\mathbf{A}) < 1$. Convergence of the consensus rule ($\mu(\mathbf{A}) < 1$) thus implies connectedness, whereas the reverse is not true. For example, when $\lambda_N(\mathbf{A}) = -1$ and $\lambda_2(\mathbf{A}) < 1$, the graph is connected but the consensus rule is not guaranteed to converge.

B. Signal processing on graphs

Signal processing on graphs (SPG) [7], [8], [11], [12] is an emerging field which deals with processing and analysis of data defined over graphs. Concepts such as sampling, filters and transforms have been generalized to graph signals [9]–[12], [22], [23]. In many problems the graph needs to be learned from data [14], [17], [24]. The aim is then to find a sparse adjacency matrix that describes similarities in a given data set. For undirected graphs, a commonly used measure of graph signal smoothness is the Laplacian quadratic form

$$\mathbf{x}^\top \mathbf{L} \mathbf{x} = \sum_{k < l} A_{kl} (x_k - x_l)^2,$$

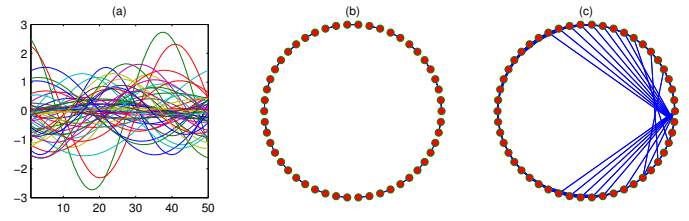


Fig. 1. Graph learning from data. (a) shows the sinusoids used to learn the graph, (b) shows reconstruction without the connectedness constraint and (c) shows reconstruction with the connectedness constraint. Edges with weights less than 0.001 have been truncated.

where $A_{kl} \geq 0$. A graph signal \mathbf{x} with a small $\mathbf{x}^\top \mathbf{L} \mathbf{x}$ varies smoothly across the edges of the graph [9], [12], [17], [25]. Given N time series in a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$, one can find graphs describing the smoothness in the data by setting

$$g(\mathbf{A}) = \frac{1}{M} \sum_{t=1}^M \sum_{i>j} A_{ij} (X_{ti} - X_{tj})^2 = \frac{1}{M} \text{tr}(\mathbf{X} \mathbf{L} \mathbf{X}^\top).$$

Self-edges can be removed by setting $A_{ii} = 0$ for all $i \in \mathcal{V}$ and the rows can be normalized by setting $\mathbf{A} \mathbf{1} = \mathbf{1}$ to prevent the trivial solution $\mathbf{A} = \mathbf{0}$, giving that $\|\mathbf{A}\|_1 = N$. The connected graph learning problem thus becomes

$$\begin{aligned} \min_{\mathbf{A}} \quad & \frac{1}{M} \text{tr}(\mathbf{X} (\mathbf{I}_N - \mathbf{A}) \mathbf{X}^\top) \\ \text{subject to} \quad & \mathbf{I}_N - \mathbf{A} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top \succeq \epsilon \mathbf{I}_N, \\ & A_{ii} = 0, A_{ij} \geq 0, \mathbf{A} \mathbf{1} = \mathbf{1}, \mathbf{A}^\top = \mathbf{A}. \end{aligned} \quad (5)$$

where we used that $\mathbf{L} = \mathbf{I}_N - \mathbf{A}$.

IV. NUMERICAL SIMULATIONS

We here show numerically that the constraint (3) ensures connectedness. We consider the problem of learning the weighted graph Laplacian from data using synthetic data and a data set of temperatures at Swedish cities [26]. In the experiments we solved the optimization problem (5) using the CVX toolbox [27] with $\epsilon = 0.01$.

A. Experiments using synthetic data

In the first experiment we synthetically generated the signals $\mathbf{X} \in \mathbb{R}^{M \times N}$ as

$$X_{kl} = d_k \sin(f_k \cdot l + \phi_k),$$

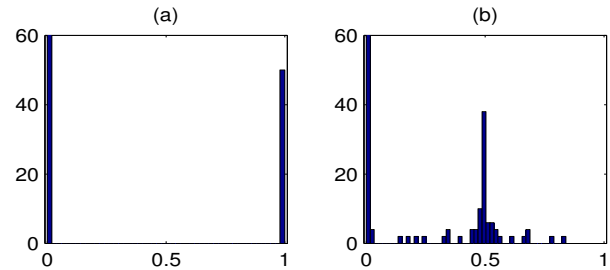


Fig. 2. Histogram of edge weights for learning graphs from data without the connectedness constraint (a) and with the constraint (b). The y -axis has been reduced to show the number of non-zero components more clearly.

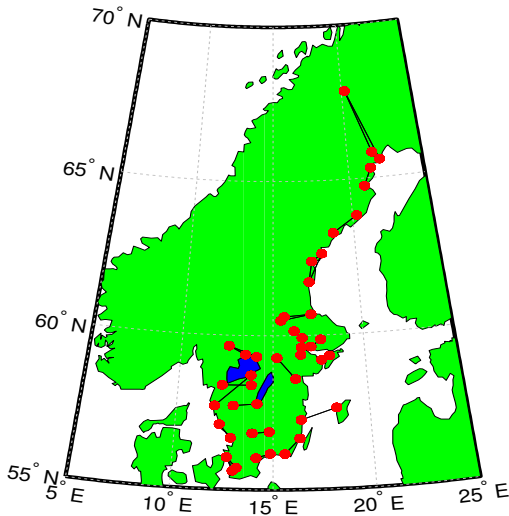


Fig. 3. Learned graph for the Swedish temperature dataset without the connectedness constraint.

where $d_k \sim \mathcal{N}(0, 1)$, $f_k \sim \mathcal{U}[\frac{2\pi}{N} - 0.05, \frac{2\pi}{N} + 0.05]$, $\phi_k \sim \mathcal{U}[-\pi, \pi]$ and $\mathcal{U}[a, b]$ denotes the uniform distribution on an interval $[a, b]$. In the experiment we set $N = M = 50$. The data is shown in Figure 1 (a). We construct a graph describing the smoothness and periodicity of the data [12], [24], [25] with and without the connectedness constraint. In Figure 1 (b) and (c) we see that without the connectedness constraint the graph shows local smoothness of the signals and is disconnected while the graph with the constraint shows local smoothness as well as some periodicity. In Figure 1, the edges with weight less than 0.001 were truncated. To examine if the graph is sensitive to the truncation we show the histogram of the adjacency matrices in Figure 2. We find that without the constraint, only a few edges have large weights while the remaining are small. With the constraint, the weights assume a wider range of values.

B. Experiments using real data

In this experiment, we use time series of daily temperature data from 45 Swedish cities from October to December of 2014 [26]. Our task is to find a graph that shows which cities have similar temperatures. Note that in the experiment, the algorithm only has access to the temperatures and not the locations of the cities. We truncated edges with weights less than 0.05. In Figure 3 we see that the graph learned without the connectedness constraint does identify some neighboring cities but is disconnected while the graph with the constraint in Figure 4 does identify neighboring cities from the temperature data. The graph also shows that cities on the same latitude have related temperatures. We show the histogram of edge weights in Figure 5. We find that the edge weights have a wider range of values when the constraint is enforced.

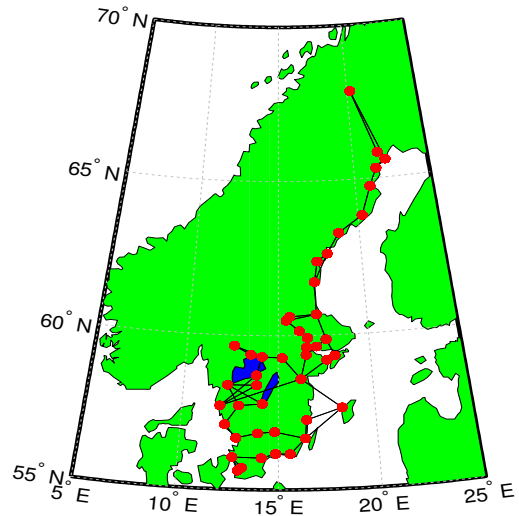


Fig. 4. Learned graph for the Swedish temperature dataset with the connectedness constraint.

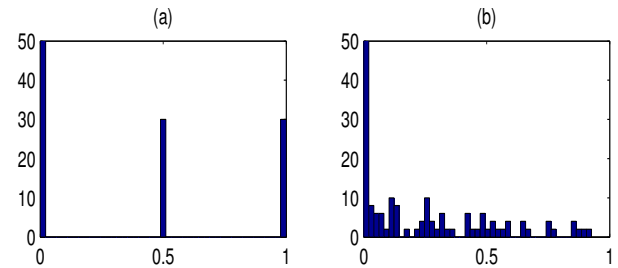


Fig. 5. Histogram of edge weights for learning graphs from temperature data without the connectedness constraint (a) and with the constraint (b). The y -axis has been reduced to show the number of non-zero components more clearly.

V. CONCLUSION

In many problems, it is useful to represent relations between data using graphs. Often, the graph is not given a-priori and needs to be learned from data. Since graphs are often preferred to be sparse, it is a challenge to preserve connectedness of the graph while simultaneously enforcing sparsity. In this paper, we showed that connectedness is an analytical property that it can be imposed on a graph as a convex constraint making it possible to guarantee connectedness when learning sparse graphs. For the consensus problem, we showed that the graph is guaranteed to be disconnected for certain values of the regularization parameter when no constraint is imposed. We illustrated the effect of the constraint when learning a graph for synthetic data and for temperature data.

ACKNOWLEDGEMENT

This work was partially supported by the Swedish Research Council under contract 2015-05484. We also want to thank the anonymous reviewers for providing useful comments.

REFERENCES

- [1] P. K. Varshney, "Distributed Bayesian detection: Parallel fusion network," in *Distributed Detection and Data Fusion*, pp. 36–118. Springer, 1997.
- [2] A. H. Sayed, "Diffusion adaptation over networks," *Academic Press Library in Signal Processing*, vol. 3, pp. 323–454, 2013.
- [3] 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.
- [4] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.
- [5] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web.," Tech. Rep., 1999.
- [6] A. N. Langville and C. D. Meyer, *Google's PageRank and beyond: The science of search engine rankings*, Princeton University Press, 2011.
- [7] 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 Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sept 2014.
- [8] A. Sandryhaila and J. M.-F. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [9] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, Sept 2015.
- [10] A. Venkitaraman, S. Chatterjee, and P. Händel, "On Hilbert transform of signals on graphs," in *2015 International Conference on Sampling Theory and Applications (SampTA)*, May 2015.
- [11] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *Signal Processing, IEEE Transactions on*, vol. 62, no. 12, pp. 3042–3054, 2014.
- [12] 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, 2013.
- [13] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical LASSO," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [14] J. Mei and J. M. F. Moura, "Signal processing on graphs: Estimating the structure of a graph," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5495–5499.
- [15] F. Lin, M. Fardad, and M. R. Jovanovic, "Identification of sparse communication graphs in consensus networks," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012, pp. 85–89.
- [16] G. Gnecco, R. Morisi, and A. Bemporad, "Sparse solutions to the average consensus problem via l_1 -norm regularization of the fastest mixing markov-chain problem," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 2228–2233.
- [17] D. Xiaowen, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning graphs from signal observations under smoothness prior," *CoRR*, vol. abs/1406.7842, 2014.
- [18] C. Hegde, P. Indyk, and L. Schmidt, "A nearly-linear time framework for graph-structured sparsity," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 928–937.
- [19] F. K. Chung, *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, vol. 92. AMS, 1997.
- [20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Applications of the LASSO and grouped LASSO to the estimation of sparse graphical models," Tech. Rep., 2010.
- [21] Somayeh Sojoudi, "Equivalence of graphical LASSO and thresholding for sparse graphs," *Journal of Machine Learning Research*, vol. 17, no. 115, pp. 1–21, 2016.
- [22] A. Jung, P. Berger, G. Hannak, and G. Matz, "Scalable graph signal recovery for big data over networks," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2016, pp. 1–6.
- [23] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.
- [24] J. Hou, L. P. Chau, Y. He, and H. Zeng, "Robust Laplacian matrix learning for smooth graph signals," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1878–1882.
- [25] S. Chen, R. Varma, A. Singh, and J. Kova, "Representations of piecewise smooth signals on graphs," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6370–6374.
- [26] Swedish Meteorological and Hydrological Institute (SMHI), "Swedish temperature data: October to December 2014," <http://opendata-catalog.smhi.se/explore/>, Accessed: 2016-08-15.
- [27] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.