Topology Inference and Signal Representation Using Dictionary Learning

Mahmoud Ramezani-Mayiami
Department of ICT
University of Agder
Email: mahmoud.ramezani@uia.no

Karl Skretting
Dep. of Elec. Eng. and Computer Science
University of Stavanger
Email: karl.skretting@uis.no

***Abstract*—This paper presents a Joint Graph Learning and Signal Representation algorithm, called JGLSR, for simultaneous topology learning and graph signal representation via a learned over-complete dictionary. The proposed algorithm alternates between three main steps: sparse coding, dictionary learning, and graph topology inference. We introduce the ”transformed graph” which can be considered as a projected graph in the transform domain spanned by the dictionary atoms. Simulation results via synthetic and real data show that the proposed approach has a higher performance when compared to the well-known algorithms for joint undirected graph topology inference and signal representation, when there is no information about the transform domain. Five performance measures are used to compare JGLSR with two conventional algorithms and show its higher performance.**

***Index Terms*—Graph signal processing, dictionary learning, topology inference, signal recovery, multi-variate signal**

I. Introduction

Over the past couple of decades, many problems in signal processing, machine learning, and pattern recognition dealt with the high dimensional data, having signal $\mathbf{x} \in \mathbb{R}^N$. A set of such signals $\{\mathbf{x}[k]\}_{k=1}^{K}$ can be represented as an $N \times K$ data matrix $\mathbf{X}$. Matrix factorization of $\mathbf{X}$ has given significant contribution in data analysis such as data compression, dimensionality reduction, low dimensional signal representation, and information retrieval [1]–[4]. However, the above methods can not capture the possible nonlinearity or structure inherent in the data input space. In many real world applications, the data space follows a geometrical structure. By exploiting this prior information, the data representation can be more accurate. In this context, many manifold learning algorithms have been proposed, e.g. ISOMAP [5] and Locally Linear Embedding [6].

A standard way to represent pairwise connections between entities is by using affinity graphs. Signal representation over the underlying graph is well motivated by Shuman and his colleagues [7] in the seminal work of the emerging field of signal processing on graphs. In this framework, a graph signal resides on vertices of the underlying graph, where the graph captures space structure among entities. Cai et al. [8] applied an affinity graph to encode the geometrical structure of dataset and proposed a graph regularized non-negative matrix factorization algorithm. Shahid and his coworkers exploited the graph Laplacian matrix and enhanced the standard PCA to recover the low rank representation of the data matrix [9]. Zheng et. al. [10] proposed an algorithm to learn the sparse representation of the data when its local structure is taken into account.

The above mentioned approaches assumed that the underlying graph topology is known a priori. But in many applications, the graph structure is unknown and has to be estimated from the input data. In this context, some algorithms investigate the undirected network topology, e.g. [11], or directed topology inference, like [12], [13], for specific applications. Moreover, some researches investigated graph topology learning and signal representation, like [11] and [14]. In [11], the input graph signals are represented based on the classical factor analysis model and a learning algorithm, called GL-SigRep, is proposed to recover the signal and the graph topology, simultaneously. Yankelevsky and Elad [14] proposed a method, called graph Dictionary Learning (graphDL) for graph topology inference and dictionary learning. Sardellitti et al. [15] proposed a general framework for graph topology inference via the transform learning, but with the limitation of dictionary completeness and the orthonormality of atoms.

Our Contribution: We propose a new algorithm for Joint Graph Learning and Signal Representation via learning of the following items:

- The set of dictionary atoms,
- The sparse coefficients, representing the signals in the transform domain,
- The graph topology of the data space, representing the underlying geometry of the data domain.

The proposed method has no assumptions on the Gaussianity of the input data, as [11] has, or a priori knowledge about the dictionary. Moreover, JGLSR considers the importance of the atom coherence and tries to reduce the average coherence, while the graphDL uses some empirical regularization parameters to have a dictionary with reasonable coherence [14]. Also in JGLSR, the dictionary is not the eigenvector matrix of the graph Laplacian and hence the graph signals may not share a common support, like the transform learning method proposed in [15]. Totally, JGLSR is a more general approach, by including more flexible terms in the objective function.

The rest of this paper is organized as follows; Section II presents a brief overview on graph signal processing. The derivation of the proposed algorithm for the simultaneous

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

graph topology inference, signal recovery, and dictionary learning is in section III. Finally, experimental results and conclusion are given in sections IV and V, respectively.

II. Graph Signal Processing

Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a simple graph (i.e. undirected, no self-loop, and no multi-edge) with the vertex set $\mathcal{V}$, including v_i's for $i = 1, 2, ..., N$ and edge set $\mathcal{E}$. The number of edges $|\mathcal{E}|$ and N are called the size and the order of the graph, respectively. The elements of the weight matrix $\mathbf{W} \in \mathbb{R}_+^{N\times N}$ are positive edge weights connecting adjacent nodes. Thus, $w_{ij} = 0$ means no connection between the node v_i and the node v_j and because of undirectionality, $w_{ij} = w_{ji} > 0$ and also ”no self loop” means $w_{ii} = 0$. The graph degree matrix $\mathbf{D}$ is a diagonal matrix defined as $\mathbf{D} = \text{diag}(\mathbf{W} \cdot \mathbf{1}_N)$, where $\mathbf{1}_N$ is the all ones vector of size $N \times 1$. The combinatorial graph Laplacian and the normalized Laplacian are defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{L}_{\text{norm}} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, respectively.

The k'th graph signal is represented by $\mathbf{f}[k]$ as follows:

$$\begin{aligned} &\mathbf{f}[k] : \mathcal{V} \to \mathbb{R}^N, v_i \mapsto f_i[k] \\ &\mathbf{f}[k] = \big(f_1[k], f_2[k], ..., f_N[k]\big)^T \in \mathbb{R}^N, \end{aligned} \tag{1}$$

where T denotes the transpose operator. In most of real applications, the signal is considered to be smooth with respect to the intrinsic structure of the graph. The ”smoothness” means that two neighbor nodes which are strongly connected to each other (large W_{ij}), have highly similar data values. The local variation for signal $\mathbf{f}[k]$ can be defined as follows

$$\mathcal{S}_l[k] := \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \|f_i[k] - f_j[k]\|_2^2 \, W_{ij}. \tag{2}$$

and then, the global variation is considered as a measure of total smoothness [7] as follows

$$\mathcal{S}_t := \sum_{k=1}^{K} \mathcal{S}_l[k] = \frac{1}{2}\sum_{k=1}^{K} \mathbf{f}[k]^T\mathbf{L}\mathbf{f}[k] = \text{Tr}\big(\mathbf{F}^T\mathbf{L}\mathbf{F}\big). \tag{3}$$

where $\mathbf{F} = \big[\mathbf{f}[1], \dots, \mathbf{f}[K]\big]$ is the $N \times K$ matrix.

III. Graph-Signal Representation

The data matrix $\mathbf{X}$ is represented as follows:

$$\mathbf{X} = \mathbf{U}\mathbf{Y} + \mathbf{E}, \tag{4}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times K}$ and $\mathbf{E}$ are the coefficient and representation error matrices, respectively and $N < M < K$ for the over-complete dictionary $\mathbf{U} \in \mathbb{R}^{N\times M}$. Our proposed method aims at solving the following minimization problem

$$\begin{aligned} \underset{\mathbf{U},\mathbf{Y},\mathbf{L}}{\text{argmin}} \;& \|\mathbf{X} - \mathbf{U}\mathbf{Y}\|_F^2 + \alpha_1\text{Tr}\big((\mathbf{U}\mathbf{Y})^T\mathbf{L}\mathbf{U}\mathbf{Y}\big) \\ & + \alpha_2\sum_{k=1}^{K}\|\mathbf{y}[k]\|_1 + \alpha_3\|\mathbf{L}\|_F^2 + \alpha_4 r(\mathbf{U}) \\ \text{s.t. } & \mathbf{L}_{ij} = \mathbf{L}_{ji}, \;\; \mathbf{L}_{ij} \le 0 \text{ if } i \ne j, \;\; \mathbf{L}\cdot\mathbf{1}_N = \mathbf{0}_N, \\ & \text{Tr}(\mathbf{L}) = c_0, \\ & \text{diag}(\mathbf{U}^T\mathbf{U}) = \mathbf{1}_M. \end{aligned} \tag{5}$$

where $\alpha_i \in \mathbb{R}^+$, $i = 1, ..., 4$ are regularization parameters, and $\mathbf{0}_N$ is an all zero vector of size $N \times 1$. The operator diag$(\cdot)$ constructs a vector of the diagonal elements of its input matrix and $y[k]$ is the kth column of $\mathbf{Y}$. The first term of the objective function minimizes the Frobenius norm of the error and the second term promotes smoothness of the represented signals in the dictionary domain, i.e. $\mathbf{U}\mathbf{Y}$, over the underlying graph. The third term encourages coefficient sparsity and the fourth term controls the off diagonal entries of the Laplacian matrix while the constraint Tr$(\mathbf{L}) = c_0$ controls the diagonal elements and avoids the trivial solution, for some constant c_0. The first three constraints guarantee that the estimated $\mathbf{L}$ is a valid Laplacian. These constraints also prevent any identifiability issue for estimated dictionary and Laplacian matrices. The last constraint forces dictionary atoms to be normalized and the last term in the objective function is to encourage low coherence among atoms, defined in [16] as follows

$$r(\mathbf{U}) := -\sum_{1\le m<m'\le M} \log(1 - (\mathbf{u}[m]^T\mathbf{u}[m'])^2). \tag{6}$$

Considering the importance of dictionary coherence and represented signal smoothness (instead of observed signal smoothness) are two main ideas in the proposed minimization problem, leading to a better performance. The objective function of (5) is convex for each variable separately. In another word, if any two of $\mathbf{U}$, $\mathbf{L}$, and $\mathbf{Y}$ are kept fixed, the objective function in (5) is convex with respect to the third one. Thus, a local minimum can be achieved via an alternating method as follows; In section III-A, we keep $\mathbf{L}$ and $\mathbf{U}$ fixed and solve the sparse coding problem. Then, $\mathbf{L}$ and $\mathbf{Y}$ are kept fixed and the dictionary is estimated in section III-B. Finally, $\mathbf{U}$ and $\mathbf{Y}$ are kept fixed and (5) is solved with respect to $\mathbf{L}$ in section III-C to infer the graph topology.

A. Finding the Sparse Coefficients

The aim is to solve the following minimization problem

$$\underset{\mathbf{Y}}{\text{argmin}}\, \|\mathbf{X} - \mathbf{U}\mathbf{Y}\|_F^2 + \alpha_1\text{Tr}(\mathbf{Y}^T\tilde{\mathbf{L}}\mathbf{Y}) + \alpha_2\sum_{k=1}^{K}\|\mathbf{y}[k]\|_1 \,. \tag{7}$$

where $\tilde{\mathbf{L}} = \mathbf{U}^T\mathbf{L}\mathbf{U}$ is the graph Laplacian matrix in the dictionary domain. In other words, signal smoothness in observations domain can be considered as coefficients smoothness in the dictionary domain. Since $\mathbf{L}$ is symmetric and positive semi-definite, its eigendecomposition can be represented as $\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ where $\mathbf{Q}$ and $\mathbf{\Lambda}$ are the eigenvector and eigenvalue matrices, respectively. Thus,

$$\tilde{\mathbf{L}} = \mathbf{U}^T\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{U} = \widehat{\mathbf{U}}\mathbf{\Lambda}\widehat{\mathbf{U}}^T, \tag{8}$$

where $\widehat{\mathbf{U}} = \mathbf{U}^T\mathbf{Q}$ is the graph Fourier transform (GFT) of dictionary atoms[1].

[1] For an orthonormal complete dictionary $\mathbf{U} \in \mathbb{R}^{N\times N}$, the eigenvalues of the Laplacian and the transformed one are the same and the eigenvectors of the transformed graph Laplacian are the GFTs of the dictionary atoms. However, here, we can not use this simplification, since the dictionary is assumed over-complete, unlike what discussed in [15]

The objective function in (7) is convex but non-differentiable due to the ℓ_1-regularization term. Thus, the standard unconstrained minimization approaches can not be used to find a closed form solution. Here, a solution based on Alternating Direction Method of Multipliers (ADMM) [17] is proposed. The problem (7) can be rewritten as follows,

$$\begin{aligned}\underset{\mathbf{y}[1],\ldots,\mathbf{y}[K]}{\operatorname{argmin}} \; & \sum_{k=1}^{K} \|\mathbf{x}[k]-\mathbf{U}\mathbf{y}[k]\|_2^2 + \alpha_1 \sum_{k=1}^{K} \mathbf{y}^T[k]\tilde{L}\mathbf{y}[k] \\ & + \alpha_2 \sum_{k=1}^{K} \|\mathbf{y}[k]\|_1 .\end{aligned} \tag{9}$$

This objective function is separable over $\mathbf{y}[k]$'s. Thus, each $\mathbf{y}[k]$ is updated individually while keeping all other ones fixed. To solve (9) for the k'th variable, we have the following vector optimization problem,

$$\underset{\mathbf{y}[k]}{\operatorname{argmin}} \|\mathbf{x}[k]-\mathbf{U}\mathbf{y}[k]\|_2^2 + \alpha_1 \mathbf{y}^T[k]\tilde{L}\mathbf{y}[k] + \alpha_2 \|\mathbf{y}[k]\|_1 . \tag{10}$$

In ADMM approach, the non-differentiable sparsity term is separated from the rest and (10) is rewritten as

$$\begin{aligned}&\underset{\mathbf{y}[k],\mathbf{z}[k]}{\operatorname{argmin}} \; l(\mathbf{y}[k]) + g(\mathbf{z}[k]) \\ &\text{s.t. } \mathbf{y}[k]-\mathbf{z}[k] = \mathbf{0}_M,\end{aligned} \tag{11}$$

where $l(\mathbf{y}[k]) = \|\mathbf{x}[k]-\mathbf{U}\mathbf{y}[k]\|_2^2 + \alpha_1\mathbf{y}^T[k]\tilde{L}\mathbf{y}[k]$ and $g(\mathbf{z}[k]) = \alpha_2\|\mathbf{z}[k]\|_1$. The scaled form of ADMM algorithm consists of the following steps:

$$\begin{aligned}\mathbf{y}^{\tau+1}[k] &:= \underset{\mathbf{y}[k]}{\operatorname{argmin}}\Big(l(\mathbf{y}[k]) + \frac{\rho}{2}\|\mathbf{y}[k]-\mathbf{z}^{\tau}[k]+\mathbf{v}^{\tau}[k]\|_2^2\Big) \\ \mathbf{z}^{\tau+1}[k] &:= \underset{\mathbf{z}[k]}{\operatorname{argmin}}\Big(g(\mathbf{z}[k]) + \frac{\rho}{2}\left\|\mathbf{z}[k]-\mathbf{y}^{\tau+1}[k]-\mathbf{v}^{\tau}[k]\right\|_2^2\Big) \\ \mathbf{v}^{\tau+1}[k] &:= \mathbf{y}^{\tau+1}[k]+\mathbf{v}^{\tau}[k]-\mathbf{z}^{\tau+1}[k],\end{aligned} \tag{12}$$

where $\rho > 0$, τ and $\mathbf{v}[k]$ are called the penalty parameter, the ADMM iteration number, and the scaled dual variable, respectively [17]. To solve the sub-problem of updating $\mathbf{y}[k]$, by simple derivation, the following system of equation is obtained

$$\Big(\mathbf{U}^T\mathbf{U} + \frac{\rho}{2}\mathbf{I}_M + \alpha_1\tilde{\mathbf{L}}\Big)\mathbf{y}[k] = \mathbf{U}^T\mathbf{x}[k] + \frac{\rho}{2}(\mathbf{z}^{\tau}[k]-\mathbf{v}^{\tau}[k]). \tag{13}$$

where due to the positive definiteness of the matrix on the lefthand side, a Cholesky decomposition can solve efficiently.

For the sub-problem of updating $\mathbf{z}[k]$, a closed form solution is given by using subdifferential calculus [18] as follows

$$\mathbf{z}^{\tau+1}[k] = \mathcal{S}_{\frac{\alpha_2}{\rho}}(\mathbf{y}^{\tau+1}[k]+\mathbf{v}^{\tau}[k]), \tag{14}$$

where the element-wise soft thresholding operator $\mathcal{S}_\kappa(a)$ is the proximity operator of the ℓ_1-norm and given as below

$$\mathcal{S}_\kappa(a) = \begin{cases} a-\kappa, & a > \kappa \\ 0, & |a| \leq \kappa \\ a+\kappa & a < -\kappa. \end{cases} \tag{15}$$

B. Dictionary Learning

To estimate $\mathbf{U}$, (5) is reduced to the following problem

$$\begin{aligned}&\underset{\mathbf{U}}{\operatorname{argmin}} \; \|\mathbf{X}-\mathbf{U}\mathbf{Y}\|_F^2 + \alpha_1\mathrm{Tr}(\mathbf{Y}^T\mathbf{U}^T\mathbf{L}\mathbf{U}\mathbf{Y}) + \alpha_4 r(\mathbf{U}), \\ &\quad \text{s.t. } \operatorname{diag}(\mathbf{U}^T\mathbf{U}) = \mathbf{1}_M.\end{aligned} \tag{16}$$

where the constraint normalizes the dictionary atoms. This minimization problem can be solved by conventional algorithms, like the L-BFGS [19] which is used here for simulations.

C. Graph Topology Inference

The minimization problem with respect to $\mathbf{L}$ is given as

$$\begin{aligned}&\underset{\mathbf{L}}{\operatorname{argmin}} \; \alpha_1\mathrm{Tr}(\mathbf{Y}^T\mathbf{U}^T\mathbf{L}\mathbf{U}\mathbf{Y}) + \alpha_3\|\mathbf{L}\|_F^2 \\ &\quad \text{s.t. } \mathbf{L}_{ij} = \mathbf{L}_{ji},\; \mathbf{L}_{ij} \leq 0 \text{ if } i \neq j,\; \mathbf{L}\cdot\mathbf{1}_N = \mathbf{0}_N, \\ &\qquad \mathrm{Tr}(\mathbf{L}) = c_0.\end{aligned} \tag{17}$$

Since the optimization variable $\mathbf{L}$ is a symmetric matrix, this problem may only be solved for the entries on and below the main diagonal. By vectorizing the lower triangular part of $\mathbf{L}$ and representing the half-vectorization and vectorization of $\mathbf{L}$ by $\operatorname{vech}(\mathbf{L}) \in \mathbb{R}^{\frac{N(N+1)}{2}}$ and $\operatorname{vec}(\mathbf{L}) \in \mathbb{R}^{N^2}$, respectively, and applying the duplication matrix $\mathcal{M}_{dup}$ [20], we have

$$\mathcal{M}_{dup}\cdot\operatorname{vech}(\mathbf{L}) = \operatorname{vec}(\mathbf{L}). \tag{18}$$

Moreover, the following identities are applicable to rewrite the problem of (17) in a vector form

$$\mathrm{Tr}(\mathbf{Y}^T\mathbf{U}^T\mathbf{L}\mathbf{U}\mathbf{Y}) = \operatorname{vec}(\mathbf{U}\mathbf{Y}\mathbf{Y}^T\mathbf{U}^T)^T\cdot\operatorname{vec}(\mathbf{L}), \tag{19}$$

$$\|\mathbf{L}\|_F^2 = \operatorname{vec}(\mathbf{L})^T\cdot\operatorname{vec}(\mathbf{L}). \tag{20}$$

By applying (18) together with (19) and (20), (17) can be rewritten in the following vector optimization problem

$$\begin{aligned}&\underset{\operatorname{vech}(\mathbf{L})}{\operatorname{argmin}} \; \alpha_1\operatorname{vec}(\mathbf{U}\mathbf{Y}\mathbf{Y}^T\mathbf{U}^T)^T\cdot\mathcal{M}_{dup}\cdot\operatorname{vech}(\mathbf{L}) \\ &\qquad + \alpha_3\operatorname{vech}(\mathbf{L})^T\cdot\mathcal{M}_{dup}^T\cdot\mathcal{M}_{dup}\cdot\operatorname{vech}(\mathbf{L}) \\ &\quad \text{s.t. } \mathbf{B}\cdot\operatorname{vech}(\mathbf{L}) \leq \mathbf{0}_{\frac{N(N-1)}{2}}, \\ &\qquad \big(\mathbf{1}_N^T\otimes\mathbf{I}_N\big)\cdot\mathcal{M}_{dup}\cdot\operatorname{vech}(\mathbf{L}) = \mathbf{0}_N \\ &\qquad \big(\operatorname{vec}(\mathbf{I}_N)\big)^T\cdot\mathcal{M}_{dup}\cdot\operatorname{vech}(\mathbf{L}) = c_0.\end{aligned} \tag{21}$$

where $\otimes$ denotes the Kronecker product and $\mathbf{B}$ is the matrix that handles the inequality constraint in (17). The minimization problem (21) is a quadratic convex problem and can be solved efficiently via several methods [17], [21]. Here, we used CVX, a package for specifying and solving convex programs [22].

IV. Simulations

JGLSR is tested by synthetic data, generated based on a known graph, and a real set of temperature data from the states of the USA mainland [23].

The synthetic data is constructed via a Gaussian Radial Basis Function (RBF) following the scenario explained in [14]. This class of graphs models several real applications networks. In each trial, the coordinates of $N = 25$ vertices are

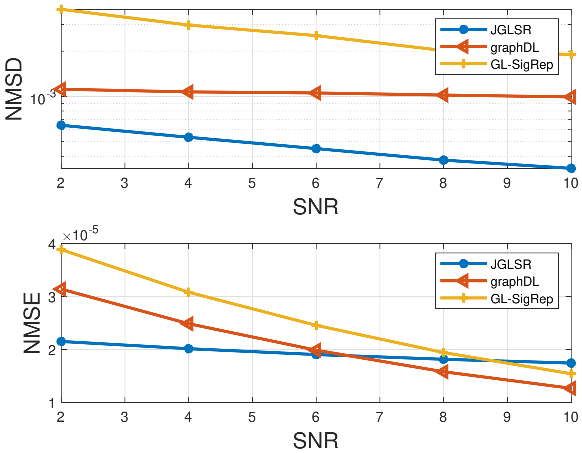

Fig. 1: The normalized mean squared deviation and error of graph topology estimation from synthetic data.

TABLE I: Averages of performance measures over different SNRs (synthetic data set).

	Recall	Precision	F-measure
JGLSR	0.90	0.87	0.89
graphDL	0.61	0.32	0.42
GL-SigRep	0.36	0.99	0.52

generated uniformly at random in the square $[0, \sqrt{5}] \times [0, \sqrt{5}]$ and the edge weights were determined with an RBF, i.e. $\exp\big(-d(i,j)^2/2\sigma^2\big)$ where $d(i,j)$ is the distance between vertices i and j. We set $\sigma = 0.5$ and remove the edges whose weights are smaller than 0.5 to keep around 17%. The graph Laplacian is computed and normalized by its trace. To construct a smooth dictionary over the graph, we generate an initial random dictionary $\mathbf{U}_0$ and then put $\mathbf{U} = \big(\mathbf{I}_N + \lambda\mathbf{L}\big)^{-1}\mathbf{U}_0$, where $\lambda = 5$ [14]. A random sparse coefficient matrix $\mathbf{Y} \in \mathbf{R}^{N \times K}$ for $K = 1000$ samples with a predefined sparsity is drawn and multiplied by the dictionary and then contaminated by an independent noise with different signal to noise ratios (SNR) to construct $\mathbf{X}$. Given $\mathbf{X}$, the underlying graph topology is estimated by JGLSR, graphDL [14] and GL-SigRep [11]. The results are compared by using the following performance measures

- Normalized Mean Squared Deviation of graph topology estimation: $\text{NMSD} = \frac{1}{N^2} \cdot \frac{\|\mathbf{L}-\hat{\mathbf{L}}\|_2^2}{\|\mathbf{L}\|_2^2}$, where $\hat{\mathbf{L}}$ denotes the estimated Laplacian matrix,
- Normalized Mean Squared Error of signal reconstruction: $\text{NMSE} = \frac{1}{N \cdot K} \cdot \frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_2^2}{\|\mathbf{X}\|_2^2}$,
- Precision: the number of truly recovered edge to the total reconstructed edges in the estimated graph,
- Recall: the number of truly recovered edge to the number of edges in the ground-truth graph,
- F-measure $= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$.

In synthetic data simulations, the regularization parameters are selected empirically, by exhaustive search over different sets of values for the various compared algorithms. We run simulations for 100 trials and averaged the results. Fig. 1 shows the higher performance of JGLSR for different SNR. Table 1 also compares three performance measures by averaging the results over different SNRs.

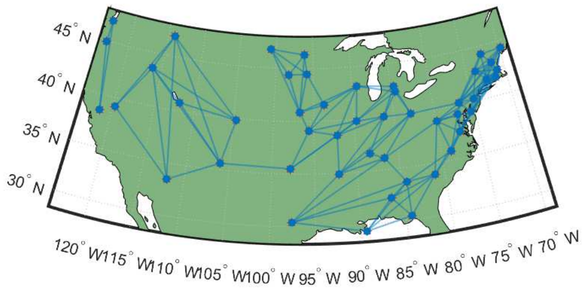

Fig. 2: The learned graph topology from real temperature data: the background map is the mainland of the USA.

For real data experiment, we store daily temperatures of $N = 48$ states for the years 2011 to 2013, i.e. $K = 1096$. In other words, graph signals are average daily temperatures measured across the states. Unlike the first simulation, here we do not access to the true underlying graph, but we consider a geographical based graph, described as follows, as the groundtruth graph. We construct a graph whose nodes represent the states, with the edge weights computed by the Gaussian RBF of each two states distance. Given the input data matrix, all three algorithms are run to infer the temperature network topology. The parameters are set as $\alpha_1 = 0.001, \alpha_2 = 0.1, \alpha_3 = 5, \alpha_4 = 100$, and $\sigma = 0.001$. The F-measure for GL-SigRep, graphDL, and JGLSR are 0.51, 0.59, and 0.64, respectively. Regarding the graph topology estimation error, NMSD for all methods is in the order of 10^{-4}. Fig. 2 shows the learned graph from this data set for the states of the USA mainland. From the learned graph, we can infer that a state weather can affect the neighboring states, which is also probable in practice. The edge concentration is mainly in the right part of the map, illustrating more dense regions and closed by cities, when compared to the west coast. Moreover, there are few edges in the middle of the figure which can be probably due to the Rocky mountains.

V. Conclusion

In this paper, an algorithm is proposed to represent a set of multi-variate signal measurements by a learned dictionary. Moreover, the recovered signals are smooth with respect to the underlying graph structure which is unknown and has to be learned. We used an alternating method to learn the dictionary, the sparse coefficients, and the graph topology, simultaneously. Besides, the transformed graph in the dictionary domain was introduced and its relation to the signal smoothness and GFT was shown. Simulation results from the synthetic data set show that the proposed method has a better performance, when compared with the conventional algorithms. Moreover, the experimental results from real data sets confirm that JGLSR is a practical algorithm to model temperature sensor network.

REFERENCES

[1] P. C. Hansen, “The truncated SVD as a method for regularization,” *BIT Numerical Mathematics*, vol. 27, no. 4, pp. 534–553, 1987.

[2] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[3] Y.-X. Wang and Y.-J. Zhang, “Nonnegative matrix factorization: A comprehensive review,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[4] W. Xu, X. Liu, and Y. Gong, “Document clustering based on non-negative matrix factorization,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, pp. 267–273.

[5] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[6] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[7] 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.

[8] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized nonnegative matrix factorization for data representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[9] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, “Fast robust PCA on graphs,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 740–756, 2016.

[10] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, “Graph regularized sparse coding for image representation,” *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1327–1336, 2011.

[11] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, Dec 2016.

[12] M. Ramezani-Mayiami and B. Beferull-Lozano, “Graph recursive least squares filter for topology inference in causal data processes,” in *Proc. IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 1–5.

[13] M. Ramezani-Mayiami, “Joint graph learning and signal recovery via Kalman filter for multivariate auto-regressive processes,” in *Proc. IEEE 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 907–911.

[14] Y. Yankelevsky and M. Elad, “Dual graph regularized dictionary learning,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 611–624, 2016.

[15] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, “Graph topology inference based on transform learning,” in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2016, pp. 356–360.

[16] S. Hawe, M. Seibert, and M. Kleinsteuber, “Separable dictionary learning,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 438–445.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “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] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.

[19] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[20] K. M. Abadir and J. R. Magnus, *Matrix algebra*. Cambridge University Press, 2005, vol. 1.

[21] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[22] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” http://cvxr.com/cvx, Mar. 2014.

[23] “National climatic data center,” ftp://ftp.ncdc.noaa.gov/pub/data/gsod/.