Dual Graph Regularized Dictionary Learning

Yael Yankelevsky and Michael Elad, *Fellow, IEEE*

***Abstract*—Dictionary Learning (DL) techniques aim to find sparse signal representations that capture prominent characteristics in a given data. Such methods operate on a data matrix $Y \in \mathbb{R}^{N\times M}$, where each of its columns $y_i \in \mathbb{R}^N$ constitutes a training sample, and these columns together represent a sampling from the data manifold. For signals $y \in \mathbb{R}^N$ residing on weighted graphs, an additional challenge is incorporating the underlying geometric structure of the data domain into the learning process. In such cases, the topological graph structure may provide a crucial interpretation for the columns, while the data manifold itself may also possess a low-dimensional intrinsic structure that should be taken into account.**

In this work we propose a novel dictionary learning algorithm for graph signals that simultaneously takes into account the underlying structure in both the signal and the manifold domains. Specifically, we require that the dictionary atoms are smooth with respect to the graph topology, as encapsulated by the graph Laplacian matrix. Furthermore, we propose to learn this graph Laplacian within the dictionary learning process, adapting it to promote the desired smoothness. Utilizing the manifold structure, we propose to encourage smoothness of the sparse representations on the data manifold in a similar manner. Both these smoothness forces implicitly enhance the learned dictionary. The efficiency of the proposed approach is demonstrated on synthetic examples as well as on real data, showing that it outperforms other dictionary learning methods in typical problems such as resistance to noise and data completion.

***Index Terms*—Dictionary learning, graph Laplacian, graph signal processing, sparse approximation, manifold structure, dual graph regularization.**

I. Introduction

The era of big data introduces new challenges to classic signal processing applications. In numerous problems, the signals to be handled have an underlying complicated geometric topology, which could be represented using a graph structure. Examples of such signals can be found in applications of transportation, energy, social networks, sensor networks, and more [1].

A popular and highly effective approach taken for solving common signal processing problems such as denoising, data completion etc. is sparse representation of the signals over a trained dictionary. In this paper we shall be focusing on processing of graph-structured signals via sparse representation modeling and learned dictionaries.

Seeking a representative dictionary for graph signals, it is possible to ignore the graph structure and view the signals as vectors in $\mathbb{R}^N$, similarly to signal representation in the Euclidean domain. Then, dictionary learning approaches that adapt the dictionary to a set of signal realizations can be applied, such as the Method of Directions (MOD) [2] or K-SVD [3]. The basic dictionary learning problem is formulated as

$$\arg\min_{D,X} \|Y - DX\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq T \quad \forall i, \qquad (1)$$

where $Y \in \mathbb{R}^{N\times M}$ is the data matrix, $D \in \mathbb{R}^{N\times K}$ is an overcomplete dictionary, $X \in \mathbb{R}^{K\times M}$ is the sparse coefficients matrix, T is a sparsity threshold and x_i denotes the i-th column of the matrix X. However, these methods ignore dependencies arising from the irregular data domain, and so the learned dictionaries will neither possess an efficient structure nor explicitly incorporate the underlying topology. As some signal characteristics, such as smoothness, depend on the topology of the graph on which the signals reside, this topology should be accounted for in order to identify and exploit structure in the data. This is especially true in cases of incomplete, insufficient or corrupted data. It is therefore desired to capitalize on the prior knowledge provided by the underlying graph structure when extending dictionary learning methods to signals residing on weighted graphs.

Along this line of reasoning, analytic dictionaries for graph signals can be proposed, generalizing transform-based dictionaries from the Euclidean domain to the graph settings. These include the graph Fourier transform [4], windowed graph Fourier transform [5], diffusion wavelets [6], and spectral graph wavelets [7], among others. Such dictionaries exhibit structure derived from the graph and are less costly to apply, yet they are less adapted to the data.

To bridge the gap between analytic and dictionary learning approaches, recent work dealing with dictionary learning for graph signals imposes structure on the trained dictionary. The enforced structure is derived from the graph topology while its parameters are learned from the data. Zhang et al. [8] suggest that the dictionary should be a collection of shift-invariant filters or sub-dictionaries. Namely, each structured sub-dictionary has the form $D_s = \chi\Lambda_s\chi^T$ where χ is the eigenbasis of the graph Laplacian $\mathcal{L}$ and $\Lambda_s \succeq 0$ are some diagonal matrices. Thanou et al. [9], [10] further restrict the dictionary to a polynomial structure, $D_s = \sum_{k=0}^{K} \alpha_{s,k}\mathcal{L}^k$, with additional constraints imposed in order to control the frequency behavior of the kernels.

The graph considered thus far captures the internal structure of each signal $y \in \mathbb{R}^N$, and so describes the relation between the rows of the data matrix Y. In this context, another graph structure can be considered, describing the relations between columns of Y. To limit terminology confusion, we shall henceforth refer to this graph as the data manifold and denote its Laplacian matrix by L_c. This manifold may also possess a low-dimensional intrinsic structure that should be taken into account.

Various manifold learning methods have been proposed to explore this structure (e.g. [11], [12], [13]), their common assumption being that if two data points are close in the intrinsic data manifold, then their representations in any other domain are close as well. In recent years, the data manifold has

become prevalent in image processing for describing pairwise relationships between image pixels or patches (see e.g. [14], [15], [16], [17], [18], [19], [20]). The manifold Laplacian L_c is then used as a regularizer, promoting similar pixels to remain similar in the sparse embedded domain.

Note that smoothness of a graph signal f can be measured in terms of a quadratic form of the graph Laplacian

$$f^T L f = \frac{1}{2}\sum_{i,j} W_{ij}\left[f(i) - f(j)\right]^2 \tag{2}$$

which is merely a sum of squared differences between the signal entries, weighted by the corresponding graph weights. Using this notion, the manifold regularized sparse coding, as used for example by [21], [16], reads:

$$\min_X \|Y - DX\|_F^2 + \beta Tr(XL_cX^T) \;\text{ s.t. }\; \|x_i\|_0 \le T\ \forall i. \tag{3}$$

The added regularization limits the degree of freedom in the sparse coding task and favors solutions preserving the manifold geometry. A similar approach was taken by [22], [19] by applying the Laplacian regularization on the reconstructed data DX rather than on the sparse representation X.

Nonetheless, requiring that the obtained sparse representations X vary smoothly along the geodesics of the data manifold, Equation (3) promotes inter-signal smoothness. When the signals themselves reside on a graph or network, we propose to require intra-signal smoothness in a similar manner.

In this paper, we therefore account for the graph structure by an additional Laplacian regularization term applied to the dictionary D:

$$\begin{aligned}\min_{D,X} \|Y - DX\|_F^2 + \alpha Tr(D^T L D) + \beta Tr(XL_cX^T)\\ \text{s.t.}\quad \|x_i\|_0 \le T\ \forall i,\end{aligned} \tag{4}$$

where $L \in \mathbb{R}^{N\times N}$ is the graph Laplacian. The two regularization terms have similar forms, but serve totally different purposes. Requiring that the dictionary atoms vary smoothly along the graph geodesics implies smoothness of any signal represented over this dictionary. The additional smoothness constraint serves the purpose of reducing the degrees of freedom given to the learning algorithm, as does the explicit dictionary structure proposed by [8] and [10], yet it is simpler and less restrictive.

Furthermore, our proposed scheme suggests the additional ability of learning the graph topology, encapsulated by the matrix L, within the dictionary learning process. This is important in cases where this structure is not given, yet known to exist.

To summarize, motivated by the above discussion, in this paper we propose a dual regularized dictionary learning problem that incorporates the graph topology via a quadratic smoothness constraint imposed on the dictionary atoms, in addition to a manifold smoothness regularization applied to the sparse codes. The latter of these regularizations alters the sparse coding problem and thus calls for the development of a new pursuit technique, as described in detail in Section V. Furthermore, we propose to learn the graph Laplacian L within the dictionary learning process, adapting it to promote the desired smoothness. All these joint forces implicitly enhance the learned dictionary.

A potential application for the proposed approach is graph signal recovery from noisy or incomplete measurements. Consider for example a temperature sensor network. In this case, the rows and columns of the data matrix Y correspond to the measurements location and time, respectively. Since the temperature is expected to change gradually in both time and space, the proposed graph smoothness constraints seem very natural in both dimensions, and so incorporating the temporal and spatial structure of the data in our scheme may improve the recovery performance in cases of malfunctioning sensors. Indeed, we shall come back to this and other data sources in Section VI, demonstrating the effectiveness of the proposed learning scheme in the context of recovery from noisy and missing measurements.

The rest of the paper is organized as follows: Section II delineates the background and recalls some basic definitions on graphs. Section III presents our basic regularized dictionary learning approach, and Section IV suggests an extension that adapts the graph Laplacian along the learning process. The complete scheme that regularizes the sparse codes as well as the dictionary atoms is then described in Section V. Experimental results are presented and discussed in Section VI, covering synthetic and true-data applications. Finally, we conclude in Section VII.

II. Preliminaries

A weighted and undirected graph $\mathcal{G} = (V, E, W)$ consists of a finite set V of N vertices (or nodes), a finite set $E \subset V \times V$ of weighted edges, and a weighted adjacency matrix W. The entry W_{ij} represents the weight of the edge $(v_i, v_j) \in E$, reflecting the similarity between the nodes v_i and v_j. In general, W_{ij} is non-negative, and $W_{ij} = 0$ if v_i, v_j are not directly connected in the graph. Additionally, for undirected weighted graphs, $W_{ij} = W_{ji}$. The graph degree matrix Δ is the diagonal matrix having $\Delta_{ii} = \sum_j W_{ij}$. Δ_{ii} is the degree of the node v_i, measuring the sum of weights in the direct neighborhood of that node. The combinatorial graph Laplacian matrix L is then defined to be $L = \Delta - W$. A normalized version of the Laplacian can also be defined in the form $\mathcal{L} = \Delta^{-1/2} L \Delta^{-1/2} = I - \Delta^{-1/2} W \Delta^{-1/2}$. While we note that other normalized versions of the Laplacian are sometime used, we focus on this symmetric form for its desired properties.

Given a topological graph, we refer to graph signals as functions $f : V \to \mathbb{R}$ assigning a real value to each graph node. Any graph signal is therefore a vector in $\mathbb{R}^N$.

When the weight matrix W is not naturally defined by an application, a common construction is via a thresholded Gaussian kernel. Put explicitly,

$$W_{ij} = \begin{cases} \exp\left(\frac{-d^2(i,j)}{2\sigma^2}\right) & \text{if } d(i,j) \le \kappa \\ 0 & \text{otherwise,}\end{cases} \tag{5}$$

for some parameters σ and κ. The distance function $d(i,j)$ may represent a physical distance between nodes v_i and v_j, or the Euclidean distance between two feature vectors describing

these nodes (e.g. sensor locations). Alternatively, $d(i,j)$ may be data-dependent and measure the distance between the data signals evaluated at the nodes v_i and v_j. This is the case in the Non-Local Means (NLM) filter [23], for example. A combination of external and internal features is also possible, as suggested for the bilinear filter [24]. Besides the Gaussian kernel, another common construction method is to connect each node with its k-nearest neighbors based on either the physical or the feature space distance [1]. We further touch upon this point in Section VI.

III. Graph Regularized Dictionary Learning

We start by incorporating the internal signal structure into the training process, leading to the following graph regularized dictionary learning problem:

$$\begin{aligned} \arg\min_{D,X} \; & \|Y - DX\|_F^2 + \alpha Tr(D^T L D) \\ \text{s.t.} \quad & \|x_i\|_0 \leq T \quad \forall i, \end{aligned} \tag{6}$$

where $Y \in \mathbb{R}^{N \times M}$ is the data matrix, $D \in \mathbb{R}^{N \times K}$ is an over-complete dictionary, $X \in \mathbb{R}^{K \times M}$ is the sparse codes matrix, $L \in \mathbb{R}^{N \times N}$ is the graph Laplacian, T is a sparsity threshold and x_i denotes the i-th column of X.

The suggested smoothness regularization, based on the Laplacian Quadratic Form (LQF), is less restrictive than forcing a parametric structure on the atoms. As opposed to previously proposed Laplacian regularizations (e.g. [22]), smoothness along the graph geodesics is here imposed directly on the dictionary atoms rather than on the reconstructed signals. Clearly, smoothness of the atoms over the graph topology implies smoothness of any signal represented over the dictionary, bearing in mind that such signals are sparse combinations of these atoms. Yet a major benefit of this approach is that it significantly simplifies the learning process by relieving the additional coupling between D and X beyond their tie through the fidelity term. Moreover, an explicit constraint posed on the dictionary prevents scenarios where the sparse coefficients compensate for non-smoothness of the atoms, and therefore yields a more robust dictionary that can be better generalized for representing other sets of signals.

To solve Equation (6), we propose a dictionary learning algorithm in the spirit of K-SVD [3]. That is, the algorithm alternates between estimating the sparse coefficients X and updating the dictionary D. Since optimization over X is not impacted by the added regularization, standard sparse coding can be used. Moreover, due to the nature of the proposed regularization, each atom could still be updated independently of the rest, since $Tr(D^T L D) = \sum_{i=1}^{K} d_i^T L d_i$. Overall, by utilizing the positive semi-definite nature of the graph Laplacian, a computationally efficient learning algorithm is obtained.

Adopting the K-SVD algorithm formulation [3], Equation (6) is solved by sequential update of each atom independently. Let v_j denote the j-th column of X^T, so that v_j^T is the j-th row of X. For the j-th atom update, the error term could thus be reformulated as follows:

$$\begin{aligned} \|Y - DX\|_F^2 =& \|Y - \sum_{i \neq j} d_i v_i^T - d_j v_j^T\|_F^2 \\ =& \|E_j - d_j v_j^T\|_F^2, \end{aligned} \tag{7}$$

To preserve the representation sparsity, the update support is restricted to samples using the j-th atom by the restriction matrix P_j, that selects the subset of columns corresponding to signals using the j-th atom:

$$\|E_j P_j - d_j v_j^T P_j\|_F^2 = \|E_j^R - d_j (v_j^T)^R\|_F^2, \tag{8}$$

with $E_j^R, (v_j^T)^R$ denoting the restricted versions of E_j, v_j^T respectively. The regularized update problem for the j-th atom is hence

$$\min_{d_j, v_j^R} \|E_j^R - d_j (v_j^T)^R\|_F^2 + \alpha d_j^T L d_j, \tag{9}$$

and could be solved using a block-coordinate descent (BCD) approach, by alternating between updates of d_j and $(v_j^T)^R$ (assuming the other variables are kept fixed). Minimizing (9) leads to closed-form update rules:

$$v_j^R = (E_j^R)^T \frac{d_j}{\|d_j\|_2^2} = \frac{P_j^T E_j^T d_j}{\|d_j\|_2^2}, \tag{10}$$

$$d_j = (\|v_j^R\|_2^2 I + \alpha L)^{-1} E_j P_j v_j^R. \tag{11}$$

We observe that the graph Laplacian L is real and symmetric, hence by eigenvalue decomposition $L = Q \Lambda Q^T$,

$$(\|v_j^R\|_2^2 I + \alpha L)^{-1} = Q (\|v_j^R\|_2^2 I + \alpha \Lambda)^{-1} Q^T. \tag{12}$$

Therefore the computational complexity is limited to a single decomposition of L followed by repeating inversions of diagonal matrices. The additional cost of $\mathcal{O}(N^2)$ is negligible compared with the complexity of the pursuit which is $\mathcal{O}(N^3)$. The computational cost of our approach is therefore similar to that of K-SVD, which is anyhow bounded by the pursuit complexity. Since the decomposition of L also costs $\mathcal{O}(N^3)$ computations and is similarly required for the polynomial dictionary learning, the complexity of both algorithms is comparable.

The complete algorithm is summarized in Algorithm 1.

IV. Laplacian Learning

The graph Laplacian L has an important role in describing the structure of a graph, and its construction thus has a significant impact on the success of the dictionary learning process. Nevertheless, the choice of L was thus far rather arbitrary. Even when the choice of L is natural to the application at hand, it may not accurately reflect the true network connectivity and the intrinsic relationships between data entities. Basing the smoothness constraint on a non-representative L will evidently lead to sub-optimal performance of our proposed algorithm. Moreover, it may be the case that the underlying topology is altogether unknown.

To overcome this barrier, we propose an extended framework that adapts the arbitrarily initialized Laplacian to the data in a way that promotes atom smoothness. The suggested

Algorithm 1 Graph Regularized Dictionary Learning

Input: initial dictionary $\mathbf{D_{(0)}} \in \mathbb{R}^{N\times K}$
Iterate: for $k = 1, 2, ...$

- **Sparse Coding:** solve (e.g. using OMP)

$$\mathbf{X_{(k)}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D_{(k-1)}}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x_i}\|_0 \le T \quad \forall i$$

- **Dictionary Update:** for $j = 1, 2, ..., K$
  - Identify the samples using the atom $\mathbf{d_j}$,

$$\Omega_j = \left\{ i \mid 1 \le i \le M \,,\, \mathbf{X_{(k)}}[j,i] \neq 0 \right\}$$

  - Define the restriction operator $\mathbf{P_j}$ corresponding to Ω_j
  - Compute the residual matrix

$$\mathbf{E_j} = \mathbf{Y} - \sum_{i\neq j} \mathbf{d_i}\mathbf{v_i^T}$$

    where $\mathbf{v_i^T}$ is the i-th row of $\mathbf{X_{(k)}}$.
  - Apply alternately:
    * $\mathbf{v_j^R} = \frac{\mathbf{P_j^T E_j^T d_j}}{\|\mathbf{d_j}\|_2^2}$
    * $\mathbf{d_j} = (\|\mathbf{v_j^R}\|_2^2\mathbf{I} + \alpha\mathbf{L})^{-1}\mathbf{E_j P_j v_j^R}$

Output: $\mathbf{D_{(k)}}$

optimization of L is integrated in the dictionary learning process.

The extended formulation now aims at solving the following joint optimization problem:

$$\begin{aligned} \arg\min_{L,D,X} \; & \|Y - DX\|_F^2 + \alpha Tr(D^T L D) + \mu\|L\|_F^2 \\ \text{s.t.} \quad & \|x_i\|_0 \le T \quad \forall i \\ & L_{ij} = L_{ji} \le 0 \; (i \neq j) \\ & L \cdot \mathbf{1} = \mathbf{0} \\ & Tr(L) = N, \end{aligned} \tag{13}$$

where N is the number of graph nodes. We shall refer to this proposed method as **graphDL**. The first two added constraints guarantee that the resulting L is a valid Laplacian matrix, and the third is added as normalization to avoid the trivial solution. Since the trace constraint fixes the ℓ_1 norm of L, the Frobenius norm penalty is added to control the distribution of the off-diagonal entries and impact the resulting sparsity of L.

We note that Laplacian learning was also proposed in [25], [26] under a different setting of promoting smoothness of the given signals. That is, smoothness over L is imposed on the data matrix, while within the sparsity context of our approach, we employ it directly on the dictionary atoms.

While Equation (13) is non-convex with respect to (L, D, X) jointly, it is convex with respect to D and L separately, assuming the other variables are fixed. Solving by alternation, optimization over D, X (for a fixed L) reduces to Equation (6), and can be solved using the algorithm proposed in Section III. Consequently, optimization over L (assuming D and X are fixed) leads to the following problem:

$$\begin{aligned} \min_{L} \; & \alpha Tr(D^T L D) + \mu\|L\|_F^2 \\ \text{s.t.} \quad & L_{ij} = L_{ji} \le 0 \; (i \neq j) \\ & L \cdot \mathbf{1} = \mathbf{0} \\ & Tr(L) = N. \end{aligned} \tag{14}$$

By vectorizing L, Equation (14) can be cast as a quadratic optimization problem with linear constraints, which could be solved using existing convex optimization tools. As the computational complexity scales quadratically with the number of nodes N, for very large graphs an approximate solution may be sought based on splitting methods or using iterative approaches.

V. Data Manifold Regularization

Having introduced the graph regularization in Equation (6), we next construct a combined problem restricting both the rows and columns of the recovered data matrix. The network topology, representing the internal structure of the signals, is modeled by a graph Laplacian $L \in \mathbb{R}^{N\times N}$ that is applied to the dictionary D. The data manifold, representing the relations between different signals, is modeled by another Laplacian $L_c \in \mathbb{R}^{M\times M}$ that is applied to the sparse code matrix X. Enforcing a similar LQF smoothness constraint in both dimensions, the following unified problem is obtained:

$$\begin{aligned} \arg\min_{D,X} \; & \|Y - DX\|_F^2 + \alpha Tr(D^T L D) \\ & + \beta Tr(X L_c X^T) \quad \text{s.t.} \quad \|x_i\|_0 \le T \quad \forall i. \end{aligned} \tag{15}$$

To solve this problem, some modification of Algorithm 1 is required. First, the sparse coding stage will now diverge from the standard form due to the regularization applied on the sparse codes. Second, the update rule for the sparse coefficients related to the j-th atom should be altered to reflect the added restriction.

For the latter, Equation (9) now reads

$$\begin{aligned} \min_{d_j, v_j^R} \; & \|E_j^R - d_j (v_j^T)^R\|_F^2 + \alpha d_j^T L d_j \\ & + \beta (v_j^T)^R L_c^R v_j^R, \end{aligned} \tag{16}$$

where $L_c^R = P_j^T L_c P_j$ is the $M_j \times M_j$ restricted version of L_c, consisting solely of the rows and columns corresponding to the samples using the j-th atom. We emphasize that L_c^R is simply a subset selection out of the full Laplacian L_c, and does not require recomputing the weights.

Optimizing over d_j, v_j^R alternately, the modified closed-form update rule for v_j^R is

$$v_j^R = \left(\|d_j\|_2^2 I + \beta L_c^R\right)^{-1} P_j^T E_j^T d_j \tag{17}$$

while the update rule for d_j remains as given by Equation (11).

Computation-wise, we note that while different for each atom, L_c^R does not occupy the entire dimension of L_c but rather the restricted support considering only the samples using that atom. Assuming that each atom is only used by a small subset of the signals, L_c^R is of limited

dimensions, and the matrix inversion in Equation (17) can be carried out for each atom (at each iteration) in reasonable time.

Having modified the dictionary update stage, we should still tackle the graph regularized sparse coding task:

$$\begin{aligned} &\arg\min_{X} \|Y - DX\|_F^2 + \beta Tr(XL_cX^T) \\ &\text{s.t.} \quad \|x_i\|_0 \leq T \quad \forall i \end{aligned} \tag{18}$$

This problem is no longer separable, demanding joint sparse coding of the dataset signals. Previous work [16] proposed to solve Equation (18) by replacing the ℓ_0 norm with ℓ_1 and using a coordinate descent approach and subgradient methods.

We propose a different solution based on the Alternating Direction Method of Multipliers (ADMM) [27], which enables simultaneous update of all columns of X. In this approach, the non-convex sparsity constraint is separated from the rest and Equation (18) is reformulated as

$$\begin{aligned} &\arg\min_{X} \|Y - DX\|_F^2 + \beta Tr(XL_cX^T) \\ &\text{s.t.} \quad X = Z, \\ &\qquad \|z_i\|_0 \leq T \quad \forall i. \end{aligned} \tag{19}$$

The augmented Lagrangian is then given by

$$\mathcal{L}_\rho(X, Z, U) = f(X) + g(Z) + \rho\|X - Z + U\|_2^2 \tag{20}$$

where $f(X) = \|Y - DX\|_F^2 + \beta Tr(XL_cX^T)$, $g(Z) = \mathcal{I}(\|z_i\|_0 \leq T \ \forall i)$ for an indicator function $\mathcal{I}()$, and U is the scaled dual form variable.

The ADMM iterative solution consists of the following steps, with k denoting the iteration number:

$$\begin{aligned} X^{(k+1)} &= \arg\min_X \left(f(X) + \rho\|X - Z^{(k)} + U^{(k)}\|_2^2 \right) \\ Z^{(k+1)} &= \arg\min_Z \left(g(Z) + \rho\|X^{(k+1)} - Z + U^{(k)}\|_2^2 \right) \\ U^{(k+1)} &= U^{(k)} + X^{(k+1)} - Z^{(k+1)} \end{aligned} \tag{21}$$

For the sub-problem of updating X, omitting the sparsity requirement has led to a quadratic objective. By simple derivation, this problem reduces to solving the following Sylvester equation [28]:

$$(D^TD + \rho I)X + \beta XL_c = D^TY + \rho(Z - U). \tag{22}$$

It is well known (e.g. [29], [30]) that this equation has a unique solution X since the eigenvalues of $(D^TD + \rho I)$ and $(-\beta L_c)$ are distinct. A numerical solution can be efficiently obtained using the Bartels-Stewart algorithm [31], [32], based on a Schur decomposition and backward substitution. Alternatively, for large dimensions, an iterative gradient descent approach may be applied.

As for the sub-problem of updating Z, this turns out to be a shrinkage problem, requiring merely a sparse projection of $X + U$. To obtain it, hard thresholding is applied to $X + U$ such that only the T largest entries of each column are kept. We denote this projection operator by $\mathcal{P}_T$.

Upon convergence, to further improve the result while preserving the sparsity pattern, an additional least squares (LS) step is performed to update the coefficient values on the determined support. Let Ω_j denote the set of T atoms used for representing the j-th signal, $\Omega_j = \{i \mid Z_{i,j} \neq 0\}$. Minimization of $\|Y - DZ\|_F^2$ over the fixed supports Ω_j is a convex problem, leading to the final update $Z_{\Omega_j,j} = D_{\Omega_j}^\dagger y_j \quad \forall j = 1, ..., M$.

As the problem in Equation (19) is non-convex, ADMM is not guaranteed to converge, and even if it does, it need not be to a global optimum. This approach is thus relatively sensitive to the choice of parameters and initialization of X, Z, U. Nonetheless, as a heuristic, we initialize with the standard sparse coding

$$X^{(0)} = \arg\min_X \|Y - DX\|_F^2 \ \text{ s.t. } \ \|x_i\|_0 \leq T \quad \forall i, \tag{23}$$

which is empirically found to perform well.

The graph regularized sparse coding algorithm is summarized in Algorithm 2.

Algorithm 2 Graph Regularized Sparse Coding

Initialize:

$$\mathbf{X^{(0)}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 \ \text{ s.t. } \ \|\mathbf{x_i}\|_0 \leq T \quad \forall i$$

$$\mathbf{Z^{(0)}} = \mathbf{X^{(0)}}$$

$$\mathbf{U^{(0)}} = \mathbf{0}$$

Iterate: for $k = 1, 2, ...$

- Update $\mathbf{X^{(k)}}$ as the solution of

$$(\mathbf{D^TD} + \rho\mathbf{I})\mathbf{X} + \beta\mathbf{XL_c} = \mathbf{D^TY} + \rho\left(\mathbf{Z^{(k-1)}} - \mathbf{U^{(k-1)}}\right)$$

- Update $\mathbf{Z^{(k)}} = \mathcal{P}_T\left(\mathbf{X^{(k)}} + \mathbf{U^{(k-1)}}\right)$
- Update $\mathbf{U^{(k)}} = \mathbf{U^{(k-1)}} + \mathbf{X^{(k)}} - \mathbf{Z^{(k)}}$

LS update: for j=1 to M

- $\Omega_j = \left\{i \mid \mathbf{Z^{(k)}}[i, j] \neq 0\right\}$
- $\mathbf{Z^{(k)}}[\Omega_j, j] = \mathbf{D_{\Omega_j}^\dagger y_j}$
  where $\mathbf{D_{\Omega_j}}$ is the restriction of $\mathbf{D}$ to the subset Ω_j.

Output: The desired result is $\mathbf{Z^{(k)}}$.

To show the advantage of the proposed formulation, we performed simulations on a synthetic example and compared both regularized sparse coding methods (our ADMM based pursuit and the graph regularized sparse coding proposed in [16]) in representing noisy graph signals over a known dictionary. The signals were generated by combining 4 atoms of the dictionary and adding Gaussian noise with Signal to Noise Ratio (SNR) of 10. These signals were then coded over the known dictionary for different levels of sparsity using both pursuit methods, and the representation error was evaluated in terms of Root Mean Squared Error (RMSE) divided by the noise power. Since graphSC [16] uses an ℓ_1 sparsity measure, its regularization coefficient was chosen such that both methods yield the same sparsity level in terms of ℓ_0.

The results presented in Figure 1 clearly demonstrate that the ADMM approach yields lower representation errors for all the evaluated sparsity levels.

The dual graph regularized dictionary learning algorithm for solving Equation (15) is assembled by replacing the sparse

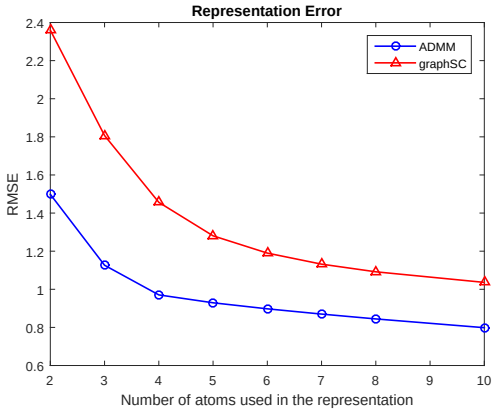


Fig. 1: Evaluation results for two graph regularized pursuit methods: the proposed ADMM solution and the graph regularized sparse coding (graphSC) of [16].

coding stage in Algorithm 1 with the procedure of Algorithm 2, and replacing the update rule from Equation (10) in the dictionary update stage with Equation (17). The resulting algorithm is described in Algorithm 3.

Algorithm 3 Dual Graph Regularized Dictionary Learning

Input: initial dictionary $\mathbf{D_{(0)}} \in \mathbb{R}^{N\times K}$
Iterate: for $k = 1, 2, ...$

- **Sparse Coding:** run Algorithm 2 to solve

$$\mathbf{X_{(k)}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D_{(k-1)}X}\|_F^2 + \beta Tr\left(\mathbf{XL_cX^T}\right)$$
$$\text{s.t.} \quad \|\mathbf{x_i}\|_0 \le T \quad \forall i$$

- **Dictionary Update:** for $j = 1, 2, ..., K$
  - Identify the samples using the atom $\mathbf{d_j}$,
$$\Omega_j = \left\{i \mid 1 \le i \le M\,,\, \mathbf{X_{(k)}}[j,i] \neq 0\right\}$$
  - Define the restriction operator $\mathbf{P_j}$ corresponding to Ω_j
  - Compute the residual matrix
$$\mathbf{E_j} = \mathbf{Y} - \sum_{i\neq j} \mathbf{d_i v_i^T}$$
where $\mathbf{v_i^T}$ is the i-th row of $\mathbf{X_{(k)}}$.
  - Apply alternately:
    * $\mathbf{v_j^R} = (\|\mathbf{d_j}\|_2^2\mathbf{I} + \beta\mathbf{L_c^R})^{-1}\mathbf{P_j^T E_j^T d_j}$
    * $\mathbf{d_j} = (\|\mathbf{v_j^R}\|_2^2\mathbf{I} + \alpha\mathbf{L})^{-1}\mathbf{E_j P_j v_j^R}$

Output: $\mathbf{D_{(k)}}, \mathbf{X_{(k)}}$

Finally, the proposed extensions may be merged together by adding the Laplacian learning, which results in the following optimization problem:

$$\begin{aligned}\arg\min_{L,D,X} \; & \|Y - DX\|_F^2 + \alpha Tr(D^T L D) \\ & + \beta Tr(X L_c X^T) + \mu\|L\|_F^2 \\ \text{s.t.} \quad & \|x_i\|_0 \le T \quad \forall i \\ & L_{ij} = L_{ji} \le 0 \; (i \neq j) \\ & L \cdot \mathbf{1} = \mathbf{0} \\ & Tr(L) = N,\end{aligned} \tag{24}$$

which we refer to as **graph^{2}DL**. This problem could then be solved using a fused procedure, alternating between optimization over D, X using Algorithm 3, and optimization over L by solving Equation (14).

As stated in the introduction, this combined learning framework offers a symmetric two-dimensional analysis of the data while jointly optimizing the graph Laplacian and the representation dictionary.

Finally, we note that we could theoretically learn L_c similarly to learning L, which would result in a fully symmetric problem formulation. This was not attempted in the scope of this work mainly for focusing on the new proposed regularization that uses L, and due to the larger typical dimensions of L_c.

VI. Experiments and Applications

In this section, we demonstrate the effectiveness of our method on synthetic examples and on real network data and show its potential use in data analysis applications.

The main problem we discuss is dealing with faulty sensors, producing missing or corrupted measurements. Specifically, we evaluate the ability of the proposed approach to recover the true underlying signals from noisy or incomplete samples. This application is demonstrated on two sensor networks: traffic loads and temperatures. Consequently, we revisit the problem of image denoising and demonstrate the capability of our method to improve denoising performance while successfully inferring the underlying patch structure.

Throughout this section, the parameters used for the various compared algorithms are chosen empirically, by exhaustive search over different sets of values.

A. A Synthetic Experiment

We first carry out experiments on a synthetic setup, where the generating dictionary and underlying graph are known, such that their recovery by our algorithm can be quantitatively assessed. Note that due to the complexity of the model and the inherent coupling between D and L, these matrices can not be drawn independently and rather require a more careful construction.

We generated a random graph consisting of $N = 100$ nodes that are randomly distributed in the square $[0, \sqrt{5}] \times [0, \sqrt{5}]$. The edge weights between each pair of nodes were determined based on the Euclidean distances between them and using the Gaussian Radial Basis Function (RBF) $W_{ij} = \exp\left(\frac{-d^2(i,j)}{2\sigma^2}\right)$ with $\sigma = 0.5$. Consequently, edges with weights smaller than

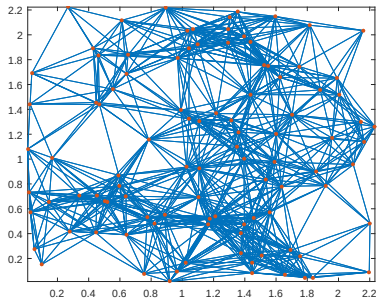


Fig. 2: Synthetic experiment ground-truth graph

0.5 were removed, keeping about 17% of the overall edges. An illustration of the resulting graph is provided in Figure 2.

The graph Laplacian $L = \Delta - W$ was then computed (for the diagonal degree matrix Δ), and multiplied by a constant normalization factor such that for the resulting L, $Tr(L) = N$. This form of normalization is only needed to enable fair comparison with the graph learned in our method, which is restricted to have a predefined trace.

To construct the dictionary $D \in \mathbb{R}^{N\times K}(K = 2N)$ with smooth atoms, an initial random dictionary D_0 was drawn and then D was obtained as the solution of

$$\arg\min_{D} \|D - D_0\|_F^2 + \lambda Tr(D^T L D), \tag{25}$$

or put explicitly, $D = (I + \lambda L)^{-1} D_0$.

We note that the choice of λ is important. The larger it is, the smoother the generated atoms, yet a strong enforcement of structure dramatically increases the mutual coherence of D, which may result in convergence problems for the Orthogonal Matching Pursuit (OMP) incorporated in our method. In our experiment, we used $\lambda = 5$, which empirically resulted in a reasonable coherence.

The data matrix $Y \in \mathbb{R}^{N\times 40N}$ was generated by drawing a random sparse coefficient matrix X with a predefined sparsity of $T = 4$ atoms per signal, and setting $\hat{Y} = DX$. Each signal was then normalized to have unit norm, and contaminated by an additive Gaussian noise with Signal to Noise Ratio (SNR) of 10.

Given the noisy data Y, the data manifold graph L_c was constructed using an RBF kernel for the Euclidean distance function $d(i, j) = \|y_i - y_j\|_2$, where y_i is the i-th column of Y.

An initial graph Laplacian L_i was constructed using the same approach as building L_c, this time based on the Euclidean distances between rows of Y and with $\sigma = 10$. The same thresholding and normalization were applied as in the ground truth graph.

To evaluate the influence of the individual components, we provide results for three versions of our algorithm: graphDL which relies on the initial Laplacian L_i and does not update it, graphDL which learns L as well, and graph^{2}DL, which also exploits the relation between the example signals via L_c. The dictionaries learned by our algorithm (for empirically chosen parameters $\alpha = 0.1, \beta = 0.6, \mu = 0.08, T = 4$) were compared against the K-SVD [3], the manifold regularized dictionary (graphSC) [16] and the polynomial dictionary [10]. Each of the dictionaries was evaluated with respect to the ground truth generating dictionary by measuring the amount of recovered atoms. For that matter, an atom d_i of the ground truth dictionary is considered to be recovered if for any atom $\hat{d}_j$ in the learned dictionary, $|\langle \hat{d}_j, d_i \rangle| > 0.99$.

Additionally, we assess the ability of the dictionaries to sparsely represent a set of test signals with a known sparsity of $T = 4$ atoms. Two normalized measures of quality are presented: (i) The representation error, i.e. the residual energy for representing the test set using 4 atoms, divided by the additive noise power; and (ii) The denoising factor, which shows the relative noise remaining in the test signals after denoising (a value below 1 implies effective denoising).

The dictionary comparison results are presented in Table I, indicating that graph^{2}DL best recovers the generating dictionary and also yields the lowest representation error while achieving noise reduction by a factor of 1.6. Furthermore, a gradual improvement is demonstrated between the different versions of our algorithm, indicating that each component has a significant contribution to the overall outcome.

Next we compare the graph Laplacian learned by our graph^{2}DL algorithm (denoted L_D), with the one learned directly from the data signals [26] (denoted L_Y) and with the initial graph L_i.

For a quantitative evaluation, we compare the sparsity of the learned graphs and assess the recovery of the edges positions using the F-measure achieved by each algorithm with respect to the ground truth graph. We use the relation:

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{26}$$

where

$$\text{Precision} = \frac{TP}{TP + FP} \quad ; \quad \text{Recall} = \frac{TP}{TP + FN} \tag{27}$$

such that TP, FP, FN represent the true-positive, false-positive and false-negative percentages.

To further evaluate the estimated edge weights, we compare the Frobenius norm $\|L - L_{GT}\|_F$ for each learned L with respect to the ground truth Laplacian L_{GT}. The results are summarized in Table II, indicating that our algorithm successfully recovers 86% of the edges and is thus comparable to and even slightly outperforms the other methods in terms of the evaluation criteria.

TABLE II: Graph learning performance. The number of edges in the learned graphs should be compared with the ground truth graph having 855 (17.27%) edges.

	L_D	L_Y	L_i
No. edges (%)	870 (17.58)	886 (17.90)	9874 (49.87)
F-measure	0.861	0.858	0.307
$\|L - L_{GT}\|_F$	2.13	2.25	3.22

Therefore, we conclude that our joint learning approach is able to capture the underlying structure of the data in terms of both the generating dictionary and the graph Laplacian.

B. Traffic Network Data

The proposed approach was further evaluated on the Caltrans Performance Measurement System (PeMS) database that

TABLE I: Dictionary comparison in terms of atom recovery percentage, representation error (with respect to the given noisy signals) and denoising error (with respect to the ground truth clean signals). Errors are presented in units of $\frac{RMSE}{\sigma_n}$.

	Pol	graphSC	KSVD	graphDL	graphDL (learned L)	graph^{2}DL
atom recovery %	0	84	82.5	83.5	86.5	87
recovered atoms (out of 200)	0	168	165	167	173	174
representation error	2.317	1.189	1.141	1.134	1.12	1.098
denoising factor	2.178	0.769	0.706	0.691	0.664	0.625

provides traffic information throughout all major metropolitan areas of California [33]. The dataset consists of 2892 signals, representing the daily average bottlenecks measured at $N = 578$ predefined locations in Alameda County's transportation network, over the time period spanned from 2007 to 2014.

In particular, the nodes of the graph consist of detector stations where bottlenecks were identified over the period under consideration. The initial graph Laplacian L is designed by connecting stations when the distance between them is smaller than a threshold of $\theta = 0.08$, corresponding to approximately 13 kilometers. The distance is set to be the Euclidean distance of the GPS coordinates of the stations and the edge weights are set to be inversely proportional to the distance.

A bottleneck could be any location where there is a persistent drop in speed, such as merges, large on-ramps, and incidents. The signal on the graph is the average duration in minutes that a bottleneck was active for each specific day. Some exemplary signals are depicted in Figure 3.

The data manifold graph L_c is constructed using an RBF kernel (5) for the Euclidean distance function $d(i,j) = \|y_i - y_j\|_2$, where y_i, y_j are the signals measured at the i-th and j-th days respectively.

The proposed approach is compared with the parametric polynomial dictionary [10] and with the non-regularized K-SVD [3].

A random subset of 1500 signals constitutes the training set, and the rest are used for testing. The added regularization parameters in Equation (24) were empirically chosen to be $\alpha = 0.2, \beta = 1, \mu = 0.16$. For consistency with [10], the learned polynomial dictionary consists of $S = 2$ sub-dictionaries, each of which is a tenth order polynomial of the normalized graph Laplacian. For the training phase, a sparsity threshold of $T = 6$ was used across all methods, and all signals were normalized with respect to the one having the maximal energy.

We start by evaluating the fit of the learned dictionaries by sparsely representing the testing set signals over each of these dictionaries for different sparsity levels (number of used atoms). The obtained representation errors are presented in Figure 5a. It can be observed that the proposed graph^{2}DL yields lower errors compared with the other evaluated methods.

Henceforth we challenge the learned models and assume that the observed measurements Y are the outcome of some corruption of the underlying signals Z, manifested as additive noise and missing samples. Put formally,

$$y_i = M_i z_i + \eta_i \quad \forall i \tag{28}$$

where the mask matrices M_i indicate the missing samples (which may differ between signal observations) and $\eta_i \sim \mathcal{N}(0, \sigma_n^2)$ is an additive Gaussian noise.

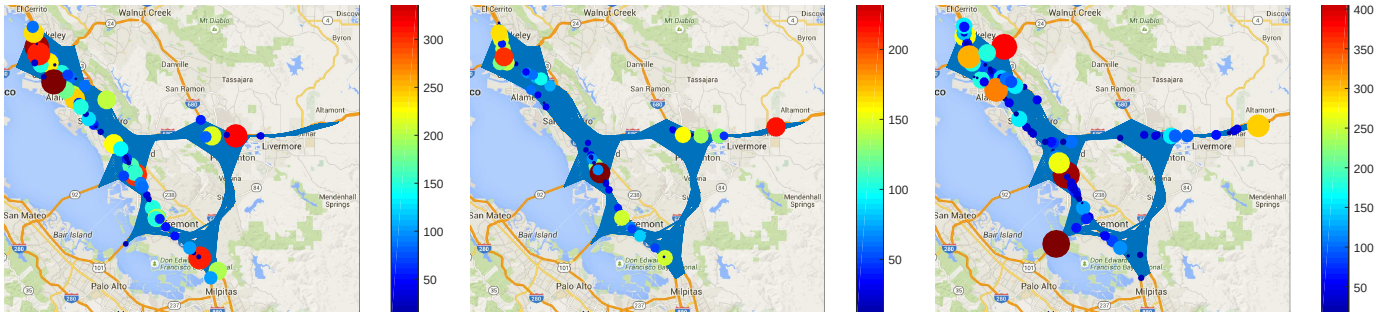
For assessing the potential benefit of the new dictionary for the common signal denoising problem, Gaussian noise of different levels σ_n was added to the test signals (assuming $M_i = I \ \forall i$) and recovery using the previously learned dictionaries was compared in terms of the Root Mean Squared Error (RMSE). Since the noise is random and does not adhere to the graph topology, the regularized dictionary is more likely to separate it from the signal. Indeed, the proposed dictionary outperforms the other methods for all the different noise levels, as illustrated in Figure 5b.

Next, we evaluate the performance for the signal inpainting (data completion) problem. In practice, missing samples may arise either from a budget restricted data acquisition, or from faulty sensors. For this scenario, we set $\sigma_n = 0$ and draw M_i to randomly subsample the test signals, preserving various predefined percentages of samples. The results presented in Figure 5c are similar to those obtained in the reconstruction and denoising experiments, and it can be observed that the regularized dictionary yields lower errors even in the extreme case where only 10% of the samples remain.

Finally, we compare the atoms of the different learned dictionaries. Figure 4 visualizes the 3 atoms in each of those dictionaries that were most commonly included by OMP in sparse decomposition of the testing signals.

It can be observed that the atoms learned by our approach are smoother over the graph compared with those learned by K-SVD [3], though not as smooth or localized as those learned for the polynomial dictionary [10].

In conclusion, our results demonstrate that the graph regularized dictionary outperforms the other dictionaries in terms of both representation error and signal recovery from noisy or missing samples. Integrating the Laplacian optimization, an additional improvement over the basic graphDL method can be observed in all simulated scenarios. Intuitively, the learned Laplacian in this example may reflect the road lengths connecting each pair of sensors rather than the plain Euclidean distances assumed in the initial graph construction, hence it better coincides with the smoothness of traffic load propagation. It is



(a) (b) (c)

Fig. 3: Characteristic graph signals demonstrating the daily traffic level (minutes of bottlenecks) across Alameda County, California, on three different days. The graph nodes are the detector stations and the connectivity is defined based on the Euclidean distance between the GPS coordinates of the stations. The size and color of each ball indicate the value of the graph signal at that node.

also evident that the dual regularized graph^{2}DL, incorporating both smoothness constraints in the learning process, further improves the performance of the proposed method and results in an overall significant enhancement compared with the two reference methods.

C. Temperature Data

We consider a dataset of daily temperature measurements collected during the years 2011 to 2013 by $N = 150$ weather stations across the mainland United States [34]. Each graph signal represents the average temperatures (in degrees Fahrenheit) measured across the sensor network on a single day. The dataset contains $M = 1096$ graph signals, constituting three full years of measurements.

We construct a graph whose nodes represent the sensors, with the edge weights set to be inversely proportional to the geographic distances between sensors. The graph is then pruned such that stations are connected when the distance between them is smaller than a threshold of $\theta = 5$, corresponding to approximately 450 kilometers. The underlying assumption is that nearby sensors will have highly correlated temperatures. The temperature graph with some typical graph signals are illustrated in Figure 6.

The manifold graph L_c is constructed in a similar manner to the procedure described for the previous dataset.

The proposed approach was again compared with K-SVD [3] for reconstruction error, noise removal and data completion applications, following the same procedure adopted in the previous subsection. Due to the previous results, the polynomial dictionary was omitted from this comparison. A random subset of 730 signals constitutes the training set, and the rest are used for testing. For the training phase, a sparsity threshold of $T = 2$ was used across all methods, and all signals were normalized with respect to the one having the maximal energy.

In accordance with the previous experiments, the results presented in Figure 7 demonstrate that the dual regularized dictionary yields lower errors for all the simulated scenarios.

D. A Glimpse at Image Processing

We conclude the section by revisiting the task of image denoising. However, the objective of this experiment is not achieving optimal denoising, but rather being able to better identify the inner structure of the data and exploit it to improve denoising performance in challenging conditions.

A 512×512 image was contaminated by random Gaussian noise with standard deviation $\sigma = 25$ and divided into overlapping 8×8 patches that constitute the columns of the data matrix. An evaluation of the proposed approach on this data shows that for a limited training set of 1000 patches, the added structure constraint slightly improves the performance of K-SVD denoising [35]. The results obtained for two different images are presented in Figure 8, demonstrating PSNR improvement of 0.15[dB]. Obviously, the more limited the training set, the more significant the improvement of graphDL over K-SVD denoising, however the overall final outcome is of lower quality.

More importantly, the learned Laplacian, having been initialized with $L = I$, captures the internal patch structure rather well. Figure 9 displays the learned graphs for both synthetic and natural images. The adjacency matrix corresponding to the learned graph Laplacian is presented in the form of a 8×8 patch, for convenience.

These results indicate that our algorithm successfully recovers a recurring pattern from its noisy observations and learns the pattern orientation instead of the local neighborhood correlations. In a natural image containing a mixture of textures, the learned graph is biased towards the included orientations. Moreover, when the image does not include a dominant texture, the learned graph structure is almost accurately the 8-nearest-neighbor relation between pixels within the patch.

We emphasize that Figure 9 does not display the learned dictionary atoms but rather the estimated underlying graph whose nodes are the 64 pixels within a patch. The dictionary itself is quite similar to the one learned by K-SVD, as demonstrated for example in Figure 10.

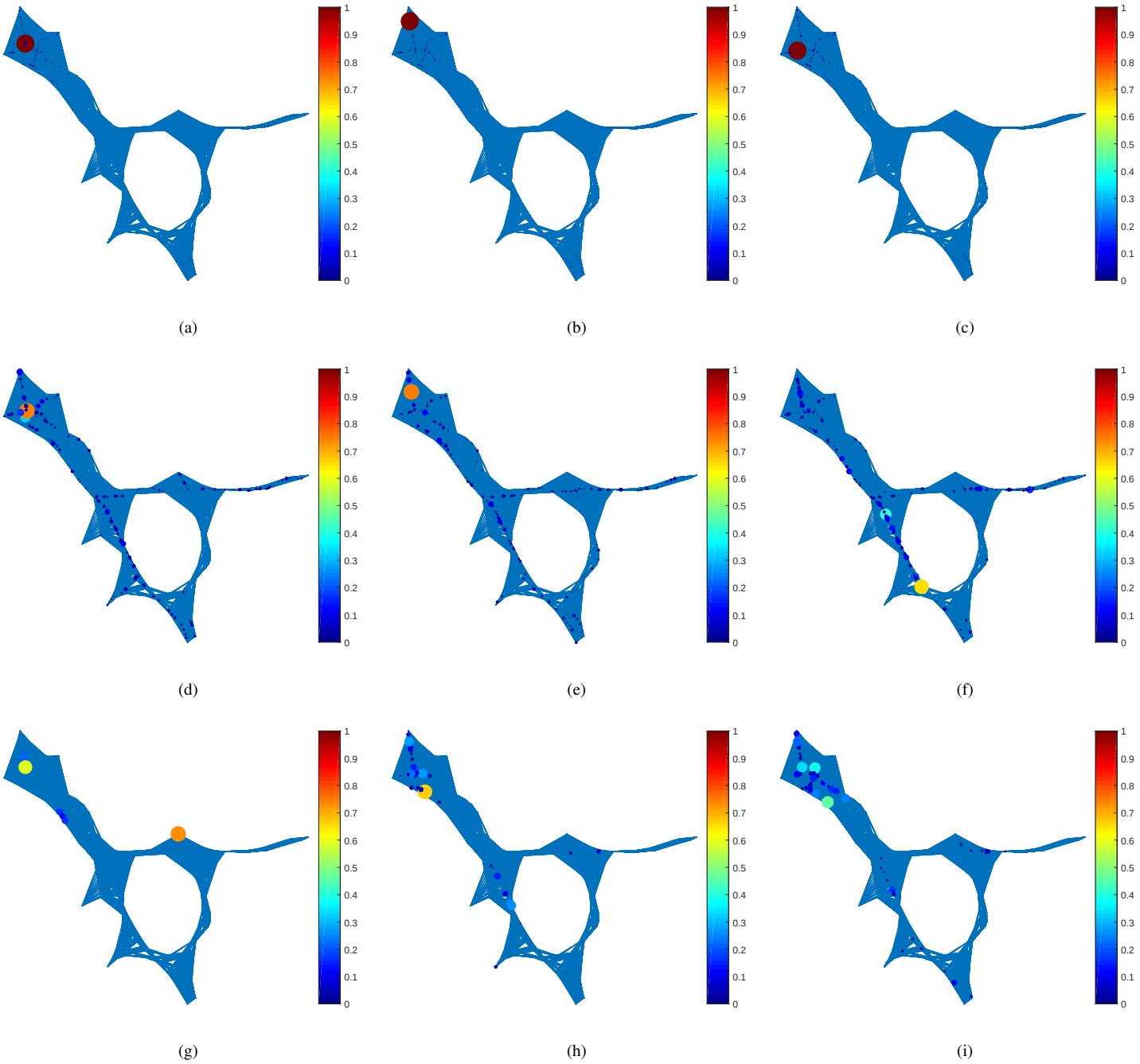
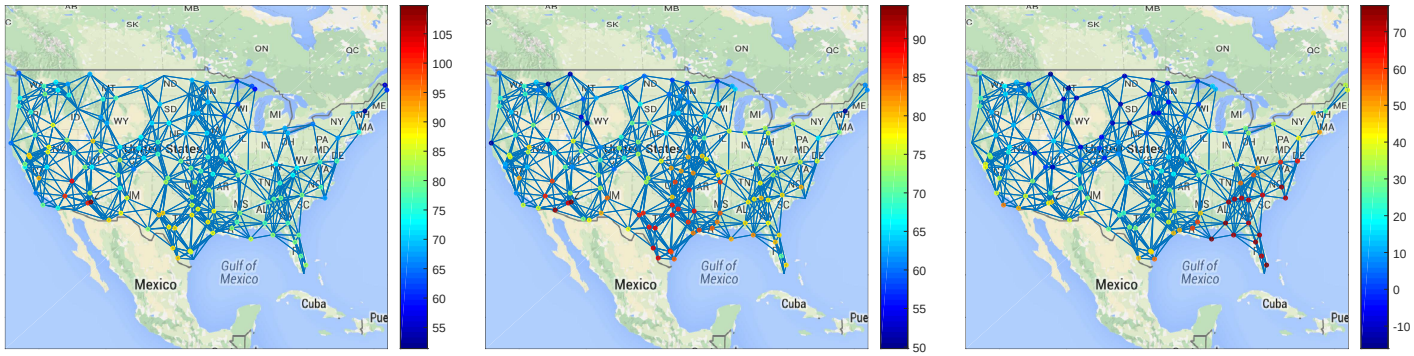


Fig. 4: Comparison of the top used atoms in each of the learned dictionaries. The first row displays atoms of the Polynomial dictionary [10], the second - atoms learned by K-SVD [3], and the third - atoms learned by graphDL. The size and color of each ball indicates the value of the atom at that graph node.



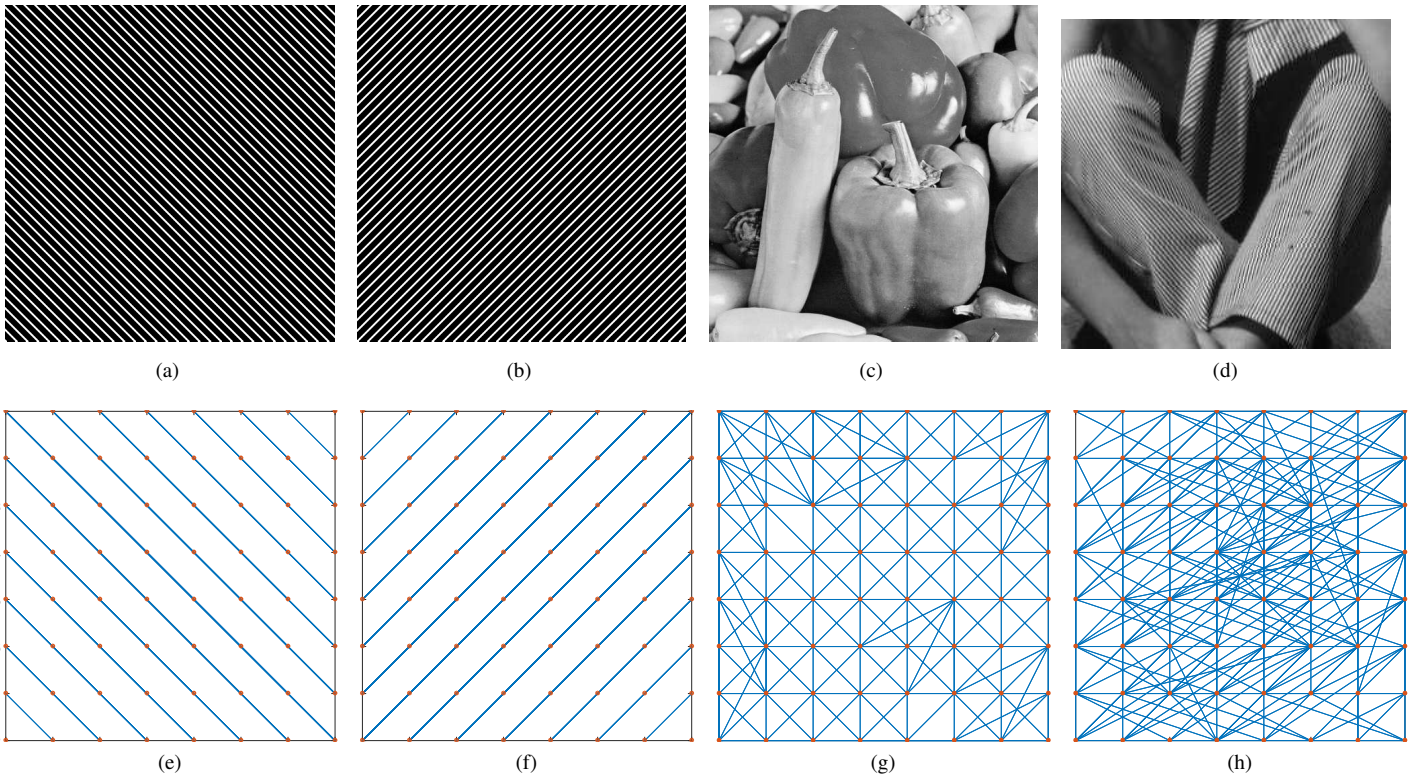
(a) (b) (c)

Fig. 6: Characteristic graph signals demonstrating the daily mean temperature (in degrees Fahrenheit) across the United States for 3 different days. The graph nodes are the detector stations and the connectivity is defined based on the Euclidean distance between the GPS coordinates of the stations. The color of each ball indicates the value of the graph signal at that node.

(a) (b) (c) (d)

(e) (f) (g) (h)

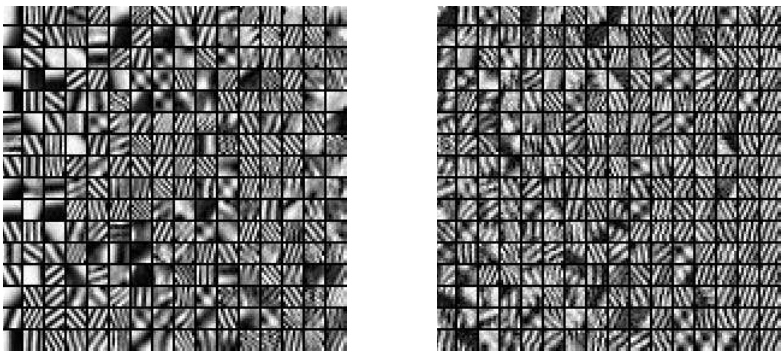
Fig. 8: Image denoising results for the images barbara and peppers: (a),(e) Original image, (b),(f) Noisy image, (c),(g) K-SVD denoising, (d),(h) graphDL denoising with optimized L.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 9: Learned graphs for different images. The top row shows the original (clean) images, and the bottom row - the corresponding patch structure graphs learned from a limited sample of noisy patches.



(a) (b)

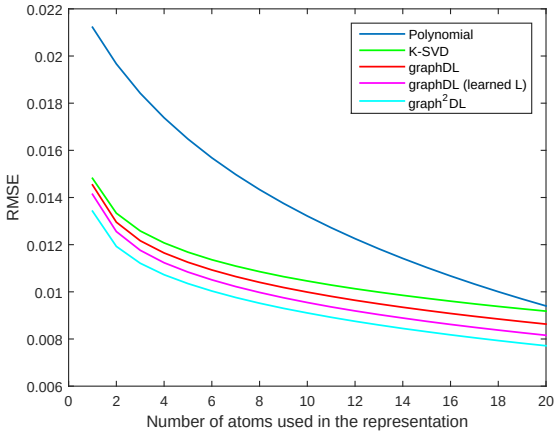
Fig. 10: Learned dictionaries for the image Barbara, using K-SVD (left) and graphDL (right).

VII. Conclusions

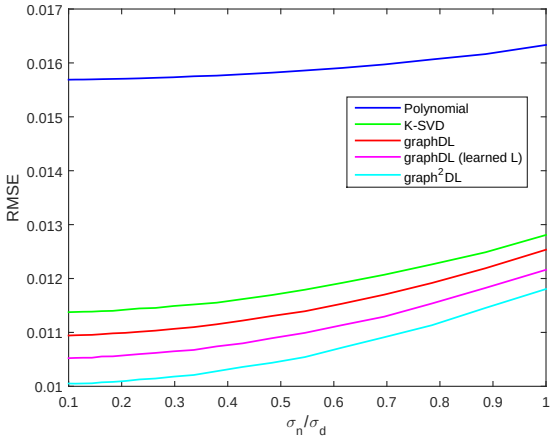
This work presented a dictionary learning algorithm for graph signals that incorporates the underlying topological prior.

The first contribution is the introduction of a Laplacian based regularization that is applied directly to the learned dictionary. This constraint, combined with the common manifold regularization that is applied to the sparse codes, leads to a symmetric problem formulation. Additional novelty therefore lies in the resulting unified framework considering the data matrix rows to be of equal significance to its columns, and treating them both in a similar manner by promoting smoothness using a Laplacian based regularization. In the network data used for our simulations, these two axes/dimensions represent the spatial and temporal domains. The dual graph regularized formulation thus captures both spatial dependence among nodes through the network topology, and the temporal evolution of the individual processes occurring at each node through the manifold structure of the training data.

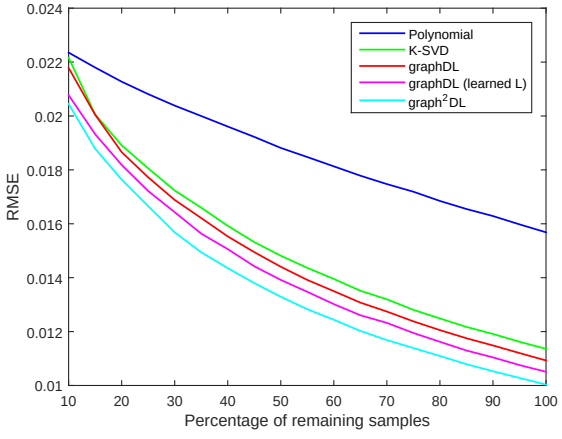
Furthermore, we proposed an extended setting in which the graph Laplacian is learned jointly with the dictionary, to overcome errors resulting from inaccurate graph construction where the underlying topology is not readily known. The



(a)

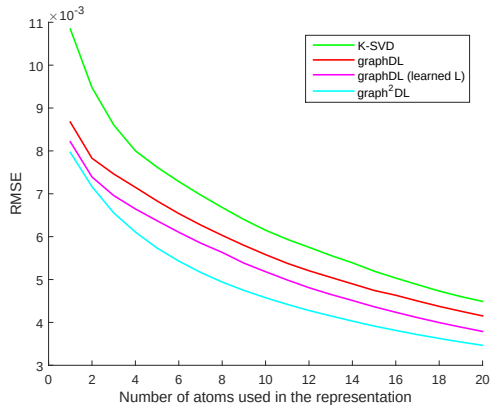


(b)

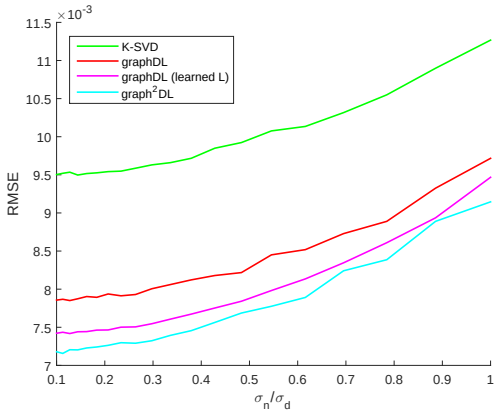


(c)

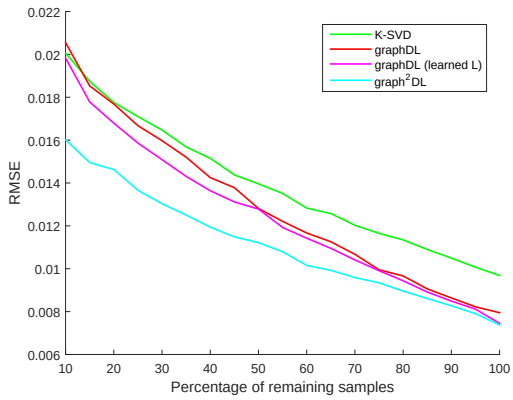
Fig. 5: Comparison of the learned dictionaries in terms of RMSE for three different applications tested on the traffic dataset: (a) representation error for different sparsity levels, (b) denoising error for different noise levels σ_n (with respect to the data STD σ_d), (c) data completion error for different percentages of remaining samples.



(a)



(b)



(c)

Fig. 7: Comparison of the learned dictionaries in terms of RMSE for three different applications tested on the temperature dataset: (a) representation error for different sparsity levels, (b) denoising error for different noise levels σ_n (with respect to the data STD σ_d), (c) data completion error for different percentages of remaining samples.

Laplacian learning problem bears similarity to other highly researched problems, such as sparse inverse covariance estimation for Gaussian graphical models and metric learning. The former may provide a probabilistic interpretation to the learned dictionary, as a Gaussian Markov Random Field (GMRF) with respect to a graph whose Laplacian is the inverse covariance matrix. We plan to further study these relations in our future work.

The effectiveness of the proposed method was demonstrated on synthetic data as well as on real network data, and compared with the parametric polynomial dictionary [10] and with K-SVD [3]. Our simulations indicate that while resulting in a relatively simple and efficient algorithm, this approach successfully infers the underlying topology, and is advantageous in the achieved representation error over a collection of graph signals, and in typical signal processing applications such as denoising and inpainting.

ACKNOWLEDGMENTS

We thank the authors of [10] for providing the code for the Polynomial dictionary learning.

The research leading to these results has received funding from the European Research Council under European Union's Seventh Framework Program, ERC Grant agreement no. 320649, and from the Israel Science Foundation (ISF) grant number 1770/14.

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.*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of Optimal Directions for Frame Design," in *ICASSP*, vol. 5, 1999, pp. 2443–2446.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[4] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs: Graph Fourier Transform," in *ICASSP*, 2013, pp. 6167–6170.

[5] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "A Windowed Graph Fourier Transform," in *IEEE Statistical Signal Processing Workshop (SSP)*, Aug. 2012, pp. 133–136.

[6] R. R. Coifman and M. Maggioni, "Diffusion Wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, 2006.

[7] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on Graphs via Spectral Graph Theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[8] X. Zhang, X. Dong, and P. Frossard, "Learning of Structured Graph Dictionaries," in *ICASSP*, 2012, pp. 3373–3376.

[9] D. Thanou, D. I. Shuman, and P. Frossard, "Parametric Dictionary Learning for Graph Signals," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, Texas, Dec. 2013.

[10] ——, "Learning Parametric Dictionaries for Signals on Graphs," *IEEE Trans. Signal Proc.*, vol. 62, no. 15, pp. 3849–3862, Aug. 2014.

[11] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *SCIENCE*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[12] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[13] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5 – 30, 2006.

[14] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal Discrete Regularization on Weighted Graphs: A Framework for Image and Manifold Processing," *IEEE Trans. Image Proc.*, vol. 17, no. 7, pp. 1047–1060, July 2008.

[15] S. Bougleux, A. Elmoataz, and M. Melkemi, "Local and Nonlocal Discrete Regularization on Weighted Graphs for Image and Mesh Processing," *International Journal of Computer Vision*, vol. 84, no. 2, pp. 220–236, 2009.

[16] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, "Graph Regularized Sparse Coding for Image Representation," *IEEE Trans. Image Proc.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

[17] P. Milanfar, "A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, Jan 2013.

[18] A. Kheradmand and P. Milanfar, "A General Framework for Regularized, Similarity-Based Image Restoration," *IEEE Trans. Image Proc.*, vol. 23, no. 12, pp. 5136–5151, Dec 2014.

[19] S. M. Haque, G. Pai, and V. M. Govindu, "Symmetric Smoothing Filters from Global Consistency Constraints," *IEEE Trans. Image Proc.*, 2014.

[20] X. Liu, D. Zhai, D. Zhao, G. Zhai, and W. Gao, "Progressive Image Denoising Through Hybrid Graph Laplacian Regularization: A Unified Framework," *IEEE Trans. Image Proc.*, vol. 23, no. 4, pp. 1491–1503, Apr. 2014.

[21] K. Ramamurthy, J. Thiagarajan, P. Sattigeri, and A. Spanias, "Learning Dictionaries with Graph Embedding Constraints," in *Signals, Systems and Computers (ASILOMAR)*, Nov. 2012, pp. 1974–1978.

[22] P. A. Forero, K. Rajawat, and G. B. Giannakis, "Prediction of Partially Observed Dynamical Processes Over Networks via Dictionary Learning," *IEEE Trans. Signal Proc.*, vol. 62, no. 13, pp. 3305–3320, July 2014.

[23] A. Buades, B. Coll, and J.-M. Morel, "A Non-Local Algorithm for Image Denoising," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR, Washington, DC, USA, 2005, pp. 60–65.

[24] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV. IEEE Computer Society, 1998, pp. 839–846.

[25] 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 *Proc. International Symposium on Biomedical Imaging (ISBI)*, San Francisco, CA, 7-11 Apr. 2013.

[26] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Laplacian Matrix Learning for Smooth Graph Signal Representation," in *ICASSP*, 2015.

[27] 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.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[28] J. Sylvester, "Sur l'equations en matrices $px = xq$," *Comptes Rendus Acad. Sci. Paris*, vol. 99, no. 2, pp. 67–71,115–116, 1884.

[29] R. Bhatia and P. Rosenthal, "How and why to solve the operator equation axxb = y," *Bull. London Math. Soc.*, vol. 29, no. 1, pp. 1–21, 1997.

[30] R. Bhatia, *Matrix Analysis*. Springer-Verlag, New York, 1997.

[31] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax + xb = c," *Comm. ACM*, vol. 15, no. 9, pp. 820–826, Sep. 1972.

[32] G. Golub, S. Nash, and C. Van Loan, "A hessenberg-schur method for the problem ax + xb= c," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, Dec 1979.

[33] T. Choe, A. Skabardonis, and P. Varaiya, "Freeway Performance Measurement System (PeMS): An Operational Analysis Tool," in *Proceedings of the 81st Transportation Research Board Annual Meeting, National Academies, Washington, D.C.*, Jan 2002.

[34] "National climatic data center," ftp://ftp.ncdc.noaa.gov/pub/data/gsod/.

[35] M. Elad and M. Aharon, "Image Denoising via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Trans. Image Proc.*, vol. 15, no. 12, pp. 3736–3745, Dec 2006.