Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Graph Anomaly Detection Using Dictionary Learning

Andra Băltoiu * **Andrei Pătrașcu** ** **Paul Irofti** *,**

* *The Research Institute of the University of Bucharest (ICUB), Romania (andra.baltoiu@fmi.unibuc.ro)*
** *Department of Computer Science, University of Bucharest, Romania (andrei.patrascu@fmi.unibuc.ro, paul@irofti.net)*

Abstract:
Anomaly detection in networked signals often boils down to identifying an underlying graph structure on which the abnormal occurrence rests on. We investigate the problem of learning graph structure representations using adaptations of dictionary learning aimed at encoding connectivity patterns. In particular, we adapt dictionary learning strategies to the specificity of network topologies and propose new methods that impose Laplacian structure on the dictionaries themselves. In one adaptation we focus on classifying topologies by working directly on the graph Laplacian and cast the learning problem to accommodate its 2D structure. We tackle the same problem by learning dictionaries which consist of vectorized atomic Laplacians, and provide a block coordinate descent scheme to solve the new dictionary learning formulation. Imposing Laplacian structure on the dictionaries is also proposed in an adaptation of the Single Block Orthogonal learning method. Results on synthetic graph datasets comprising different graph topologies confirm the potential of dictionaries to directly represent graph structure information.

Keywords: anomaly detection, dictionary learning, graph Laplacian classification

1. INTRODUCTION

Ever more clever money laundering schemes are being employed and are becoming harder to track, especially considering the outnumber of anomalous behaviours or patterns by the pool of legitimate transactions in financial data. Recent years have shown great success in applying machine learning (ML) techniques when dealing with large chunks of data whose intrinsic structure is eluded. Standard ML algorithms focus on the dominant trend, but for anti-money laundering (AML) we are interested in their anomaly detection (AD) variants that have started to develop only in recent years Akoglu et al. (2015); Bolton and Hand (2002); Elliott et al. (2019). Here the focus is shifted towards the unfitted data.

Banking transactions between different entities (e.g. individuals, companies, banks, state institutions) can be represented as a directed weighted graph: the nodes are the entities connected by directed edges representing the transactions whose weight is given by various transaction and entity attributes (e.g. amount, currency, country of origin etc.). Please note that describing all transactions within a time-frame (a month, a trimester, a year) leads to a very large graph. It is common for AML techniques to look for known static patterns within the existing transactions in order to identify possible frauds. Thus the first problem that we focus on is identifying patterns, or sub-graphs, in a given graph. This task might seem daunting at first, and it is indeed NP-hard, but the use of community detection and ML techniques helps gain some traction. Community detection is used to split the graph into manageable sub-graphs and ML to perform pattern matching within these sub-graphs. Graph classification is therefore becoming a mandatory tool, both in itself and as a means to an end. A second approach is to simply perform AD without any prior knowledge such as known patterns or other inside information from the bank institutions. This is more challenging even with ML algorithms but, if successful, it has the benefit of providing new insight into present money-laundering schemes. Unlike normal ML tasks, the key insight here is to over-fit on existing data in order to strongly reject anomalous behaviours.

Numerous signal processing applications are now being recast in order to include structural information. In many cases, an underlying network exists on which the signal rests, but this topological information has been previously discarded, mostly for complexity reasons. These underlying graphs can have physical interpretations, as in the case of natural or man-made networks, or may be abstract structures, relevant to the data, as in the case of images. Either way, paying attention to the graph topology is proving fruitful in signal processing tasks Dong et al. (2019).

Dictionary learning (DL) are a class of ML methods well suited for signal processing applications such as compressed sensing, image denoising, inpainting and blind source separation. In this article we use DL to construct and identify graph structures and their connections to the signals that they produce or support.

⋆ Acknowledgements: This work was partially supported by BRD Groupe Societe Generale through Data Science Research Fellowships of 2019 and partially by a grant of Romanian Ministry of Research and Innovation CCCDI-UEFISCDI. project no. 17PCCDI/2018.

We propose two algorithmic schemes that directly work with graph structures and are therefore appropriate for applications where the anomalous topologies are available. We also deal with the more general problem of signals that lie on graphs. Thus, our third solution regards underlying hidden graphs with possibly no physical relevance, but which rather encode dependencies or correlations.

All types of approaches are suited for the task of identifying anomalous signals. Returning to our financial transactions problem, this translates in determining malicious transaction schemes relevant to money-laundering. Domain knowledge reveals several typical patterns employed in fraudulent conducts, such as clique-like or circular structures. Solutions presented in Sections 3 and 4 target precisely this case, where the known graph structure is defining for the signal's membership to the anomaly class. However, other financial crimes remain undetected and the question of exposing these schemes (and their corresponding graph structures) is an open one. Our latter proposal, presented in Section 5, is suited for this task, as it indirectly infers structural information from a signal resting on a graph.

Notations. Given matrix $\boldsymbol{A}$, we denote $\boldsymbol{A}_i$ the i^{th} (block) column of $\boldsymbol{A}$ and $\boldsymbol{A}^j$ the j^{th} (block) line of $\boldsymbol{A}$. Also, given vector x, we use notation x_ℓ or $[x]_\ell$ for ℓ^{th}(block) coordinate of x. It will clearly result from the local context and explanations if the index is either a block or not. Given the closed convex set $\mathcal{X}$ and a point z, we denote with $\pi_{\mathcal{X}}(z)$ the orthogonal projection of z onto $\mathcal{X}$. For a given function f, we denote $\nabla f(z)$ the gradient of f at z and with $\nabla_{x_i} f(z)$ the i^{th}-block of the gradient at z. For any integer n, we have $[n] = \{1, \cdots, n\}$. For any matrix $A \in \mathbb{R}^{m\times n}$, the operation $\text{vec}(A)$ returns a vector of dimension mn containing the vectorized columns of A. We denote $\|\cdot\|_0$ the l_0 pseudo-norm, which counts the number of nonzero elements in the vector or matrix. The all-ones vector is denoted by $\mathbf{1}$ and the Kronecker product by $\otimes$.

2. PRELIMINARIES ON DICTIONARY LEARNING

The sparse representation model assumes that a given signal admits a representation in which only few elements, fewer than its original dimension, are nonzero. More precisely, with the aid of an overcomplete basis called a dictionary, the signal can be written as

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X} + \boldsymbol{V}, \tag{1}$$

where $\boldsymbol{Y} \in \mathbb{R}^{m\times N}$ are the N signals, $\boldsymbol{X} \in \mathbb{R}^{n\times N}$ is representation with s nonzero elements, $\boldsymbol{D} \in \mathbb{R}^{m\times n}$ is the dictionary and $\boldsymbol{V}$ the zero mean white Gaussian noise. The columns of the dictionary are called atoms and in order to avoid the ambiguity introduced by multiplication, they are normalized.

The dictionary learning problem consists in finding both the dictionary and the representation that best characterize the signal. The objective function therefore is

$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{X}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{X}_\ell\|_0 \le s,\ 1 \le \ell \le N \\ & \|\boldsymbol{D}_j\| = 1,\ 1 \le j \le n \end{aligned} \tag{2}$$

Common practice solves the above problem in an alternate manner, that is by first fixing the dictionary and computing the representation, then fixing the representation and updating the dictionary. The former step is also known as sparse coding and is usually performed either by greedy methods that attempt to construct the sparse support Pati et al. (1993), or by relaxing the l_0 pseudo-norm in (2) into a convex formulation, such as by replacing it with l_1. As for the latter step, it is routinely solved by coordinate or gradient descent methods, of which K-SVD Aharon et al. (2006), a coordinate descent method, is a popular choice.

2.1 Classification using Dictionary Learning

The representative power of the dictionary can also be employed to classification tasks, provided some additional measures are taken, that specialize atoms to the characteristics of each class of signals. Intuitively, signals corresponding to the same class should have similar representations, that (ideally) differ significantly from those of other classes.

The formulation in (1) suggests this discriminative property does not arise naturally, however it can be pursued without modifying the objective function, by training separate (sub)dictionaries with signals from each class Skretting and Håkon Husøy (2006). Class estimation of a new signal can then be obtained by comparing its representation error with each subdictionary. The core of this scheme, Sparse Representation-based Classifier, was developed in Wright et al. (2009). Additional penalties can be included as to limit the possibility of having similar atoms in different subdictionaries or, furthermore, to explicitly impose discriminative traits on the dictionary by penalizing the representation errors on each class Mairal et al. (2008). A different approach is to indirectly force the desired properties in $\boldsymbol{D}$ by imposing that $\boldsymbol{X}$ both fulfills the signal reconstruction demands and exhibits class expressiveness. This translates in coupling representation learning with classification. Label Consistent K-SVD (LC-KSVD) Jiang et al. (2011) is such an approach, that explicitly constrains the representation to be similar for signals within the same class and that moreover imposes each dictionary atom to become specialized for one class only.

2.2 Dictionary Learning on Graphs

Most common DL adaptations that handle graph structure learning build on sparse coding applications, such as Zheng et al. (2011), that employ Laplacian learning as a smoothness-inducing factor. These methods assume that if data sits on a graph, then neighboring nodes have similar representations, a property known as local invariance. Forcing the representations to include the data manifold structure is achieved by adding a Laplacian regularization term to the sparse coding task or to the dictionary update.

Classification, too, can gain from this constraint on the representations, as intra-class similarities are also assumed to be reflected in the underlying graph structure. In Yankelevsky and Elad (2017), the constraint of LC-KSVD that ties atoms to classes is replaced with a milder regularization term that ensures smoothness over the data manifold. The suitability of dictionaries to implicitly represent structure has been tested in works such as Irofti and Stoican (2017), which train the dictionary to discriminate between several types of faults in water networks.

3. LAPLACIAN-STRUCTURED DICTIONARY LEARNING

We first consider the problem in which the classification task involves discriminating between different graph structures. Thus, in this section we design a learning algorithm based on structured dictionaries for computing efficient representations of Laplacian matrices. We assume that each matrix signal $\boldsymbol{y}^{(i)} \in \mathbb{R}^{m\times m}$ has a Laplacian structure, namely it exhibits the properties of a Laplacian matrix. Note that we have abused the notation in (1), and now m refers to each of the two signal dimensions. We intend to capture this structure using linear combinations of basic atomic Laplacians

$$\boldsymbol{y}^{(i)} \approx \boldsymbol{L}_1 x_1^{(i)} + \cdots + \boldsymbol{L}_n x_n^{(i)}$$
$$\text{where } \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{L}_j \in \mathbb{R}^{m\times m},\ \forall j \in [n].$$

More compactly, we seek the following particular representation of $\boldsymbol{y}^{(i)}$

$$\boldsymbol{y}^{(i)} \approx \boldsymbol{L}\cdot(\boldsymbol{x}^{(i)} \otimes \mathbf{1}),$$

where $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ and $\boldsymbol{L} = [\boldsymbol{L}_1 \cdots \boldsymbol{L}_n] \in \mathbb{R}^{m\times mn}$. In order to put this approximation problem in a more formal manner, we consider vectorized signals $y^{(i)} \in \mathbb{R}^{m^2}$, the atoms $\boldsymbol{D}_i = vec(\boldsymbol{L}_i)$ and derive the following constrained optimization problem with Laplacian-type constraints

$$\begin{aligned}\min_{\boldsymbol{D},\boldsymbol{X},\boldsymbol{L}}\ & \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.}\ & \boldsymbol{D}_i = \text{vec}(\boldsymbol{L}_i),\ \ 1\le i\le n,\\ & \boldsymbol{L}_i\mathbf{1}=0,\ [\boldsymbol{L}_i]_{\ell j}\le 0,\ \forall \ell\neq j,\\ & Tr(\boldsymbol{L}_i)=m.\end{aligned} \tag{3}$$

Note that the slack variables $\boldsymbol{L}$ can be easily eliminated, but we keep them for a more elegant presentation. A typical approach in the literature, for solving the problem (3), is the K-SVD family of algorithms, which proved empirically efficient when only orthogonality constraints are present; however, the extension of K-SVD scheme to the more complicated Laplacian constraints is highly nontrivial and we do not analyze this subject in our paper. Consequently, we will focus on alternating minimization and block coordinate descent schemes which are more appropriate for our current constrained setting (see Nesterov (2012); Patrascu and Necoara (2015a,b)).

The last linear equality constraint of (3) couples the lines of the Laplacian matrices and it is imposed with the purpose of avoiding the trivial solution (see Yankelevsky and Elad (2017)). Since the problem scales with Laplacian dimension m^2 and data dimension N, simple algorithms with computationally cheap iteration are ideal in our high dimensional setting. Although the block coordinate descent (BCD) methods are one of the most appropriate schemes at hand (Nesterov (2012); Patrascu and Necoara (2015a,b)), they require constraints separability which do not hold in our case due to the last equality. To remedy this issue, we transfer the last equality constraint in the objective through quadratic penalization, which will further avoid the trivial solution, and subsequently apply BCD inner schemes. Also, since we aim to obtain parsimonious representations of the vectorized Laplacians, we will additionally impose ℓ_0 sparsity constraints on the representations matrix X, obtaining the following problem

Algorithm 1: Alternating Minimization (AM)

Data: signals $\boldsymbol{Y} \in \mathbb{R}^{m^2\times N}$,
initial dictionary $\boldsymbol{D}^{[0]} \in \mathbb{R}^{m^2\times n}$

Initialize $k = 0$
while *stopping criterion do not hold* **do**
  Compute $\boldsymbol{X}^{[k]}$ by solving (4), with $\boldsymbol{D} = \boldsymbol{D}^{[k]}$ fixed
  Compute $\boldsymbol{D}^{[k+1]}$ by solving (4), with $\boldsymbol{X} = \boldsymbol{X}^{[k]}$ fixed
  $k = k + 1$

$$\begin{aligned}\min_{\boldsymbol{D},\boldsymbol{X},\boldsymbol{L}}\ & \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2 + \frac{\rho}{2}\sum_{i=1}^{n}(Tr(\boldsymbol{L}_i)-m)^2 \\ \text{s.t.}\ & \boldsymbol{D}_i = \text{vec}(\boldsymbol{L}_i),\ \ 1\le i\le n,\\ & \boldsymbol{L}_i\mathbf{1}=0,\ [\boldsymbol{L}_i]_{\ell j}\le 0,\ \forall \ell\neq j,\\ & \|\boldsymbol{X}_i\|_0 \le s,\end{aligned} \tag{4}$$

where s is the prefixed number of nonzeros in each representation. Notice that, for high enough values of the penalty parameter $\rho > 0$, the trace of each atomic Laplacian $\boldsymbol{L}_i$ approaches m. Since the resulted problem has nonconvex objective function and also nonconvex constraints, one of the most simple and efficient outer strategies in this case is the alternating minimization (AM) scheme, which, at each iteration, alternates the minimization over $\boldsymbol{D}$ and $\boldsymbol{X}$. Since none of the steps of AM are computable in closed-form, we will consider AM (presented in Algorithm 1) as an outer optimization scheme and provide further appropriate algorithms to solve the local problems from each step.

Computing the dictionary. The optimization problem in $\boldsymbol{D}$ is convex, with a linear least squares type objective and linear constraints, which can be approached in principle with standard QP algorithms. Given the representation matrix from previous iteration $\boldsymbol{X}^{[k]}$, the new dictionary is the solution of following problem

$$\begin{aligned}(\boldsymbol{D}^{[k+1]},\boldsymbol{L}^{[k+1]}) = \arg\min_{\boldsymbol{D},\boldsymbol{L}}\ & f_\rho(\boldsymbol{D},\boldsymbol{L}) := \left\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}^{[k]}\right\|_F^2 + \\ & + \frac{\rho}{2}\sum_{i=1}^{n}(Tr(\boldsymbol{L}_i)-m)^2 \\ \text{s.t.}\ & \boldsymbol{D}_i = \text{vec}(\boldsymbol{L}_i),\ \ 1\le i\le n,\\ & \boldsymbol{L}_i\mathbf{1}=0,\ [\boldsymbol{L}_i]_{\ell j}\le 0,\ \forall \ell\neq j.\end{aligned} \tag{5}$$

Since this particular problem scales with m^2 and N, most of general QP iterative algorithms (e.g. gradient algorithms Nesterov (1983), interior point methods) require at each iteration at least $\mathcal{O}(m^2N)$ operations, which even for medium dimensions (m, N) might be prohibitively high. Therefore, we further provide a BCD scheme with $\mathcal{O}(mn + m\log(m))$ complexity per iteration, which efficiently computes the dictionary even in high dimensions. The BCD algorithm uses the constraints separability of (5)(see Nesterov (2012)): each line of the Laplacian $\boldsymbol{L}_i$ is subjected to a simplex-type set (one linear equality and positivity constraints). Since the projection of simplex-type sets can be efficiently computed in $\mathcal{O}(m\log m)$, then a projected coordinate gradient step can also have a fast iteration. Therefore, our scheme performs the following:

(*i*) Chooses randomly at each iteration an atom $\boldsymbol{D}_i$ and a $m-$size block $[\boldsymbol{D}_i]_l$ from the atom. This randomly

chosen block represents a line from the vectorized Laplacian $\boldsymbol{L}_i$.

(*ii*) Once the randomized selections are made, a projected coordinate gradient descent step is performed on the objective function.

Formally, by eliminating the slack variables $\boldsymbol{L}$, recall that we obtain a new form of penalized objective function

$$f_\rho(\boldsymbol{D}) := \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \\ + \frac{\rho}{2}\sum_{i=1}^{n}\left(\sum_{j=0}^{m-1}[\boldsymbol{D}_i]_{j(m+1)+1} - m\right)^2.$$

By further denoting $\boldsymbol{e_I} = vec(\boldsymbol{I}_m)$ and $\boldsymbol{R}^{(i)} = \sum_{j\neq i}\boldsymbol{D}_j\boldsymbol{X}^j$, we then have the gradient on atom $\boldsymbol{D}_i$

$$\begin{aligned}\nabla_{\boldsymbol{D}_i} f_\rho(\boldsymbol{D}) &= (\boldsymbol{D}_i\boldsymbol{X}^i - \boldsymbol{R}^{(i)})(\boldsymbol{X}^i)^T + \\ &+ \rho\boldsymbol{e_I}\left(\sum_{j=0}^{m-1}[\boldsymbol{D}_i]_{j(m+1)+1} - m\right) \\ &= \boldsymbol{D}_i\left\|\boldsymbol{X}^i\right\|^2 - \boldsymbol{R}^{(i)}(\boldsymbol{X}^i)^T + \\ &+ \rho\boldsymbol{e_I}\left(\sum_{j=0}^{m-1}[\boldsymbol{D}_i]_{j(m+1)+1} - m\right).\end{aligned}$$

The gradient $\nabla_{\boldsymbol{D}_i} f_\rho(\boldsymbol{D})$ is Lipschitz continuous in $\boldsymbol{D}$ and in order to determine the Lipschitz constants we make the following observations. Given two matrices $(\boldsymbol{D}, \tilde{\boldsymbol{D}})$ which differ only on the $\ell-$th block of the i-th atom with difference vector $\boldsymbol{h} \in \mathbb{R}^m$, then

$$[\tilde{\boldsymbol{D}}_j]_t = \begin{cases}[\boldsymbol{D}_j]_t + \boldsymbol{h} & \text{if } j = i, t = \ell \\ [\boldsymbol{D}_j]_t & \text{otherwise}\end{cases}$$

But observing that

$$\left\|[\nabla_{\boldsymbol{D}_i} f(\boldsymbol{D})]_\ell - [\nabla_{\boldsymbol{D}_i} f(\tilde{\boldsymbol{D}})]_\ell\right\| \leq (\left\|\boldsymbol{X}^i\right\|^2 + \rho)\|\boldsymbol{h}\|, \\ \forall \boldsymbol{D} \in \mathbb{R}^{m^2\times n}, \boldsymbol{h} \in \mathbb{R}^m,$$

then it is straightforward to estimate the Lipschitz constants as $\mathcal{L}_i = \left\|\boldsymbol{X}^i\right\|^2 + \rho$.

The traditional constant stepsize used for most of the first-order algorithms, including BCD schemes, is inverse proportional with the Lipschitz constant (see Patrascu and Necoara (2015a)). Therefore, we will further use the derived Lipschitz constant estimate in the stepsize of $\ell-$th block-atom update in block coordinate gradient descent (BCGD) scheme as follows

$$\left[\boldsymbol{D}_i^{[t+1]}\right]_\ell = \left[\boldsymbol{D}_i^{[t]}\right]_\ell - \frac{1}{\mathcal{L}_i}\left[\nabla_{\boldsymbol{D}_i} f(\boldsymbol{D}^{[t]}, \boldsymbol{X}^{[k]})\right]_\ell.$$

Denoting $\mathcal{X}_\ell = \{d \in \mathbb{R}^m : \mathbf{1}^T d = 0, d_\ell \geq 0, d_j \leq 0, \forall j \neq \ell\}$, we provide the complete BCGD iteration in Algorithm 2.

The main computational effort is comprised in the gradient and projection steps. The first one can be straightforwardly estimated to take $\mathcal{O}(mn)$. However, if we consider that the matrix $X^{[k]}(X^{[k]})^T$ is available, then the complexity can be further reduced. For the projection step we use Kiwiel's algorithms (Kiwiel (2007)), which requires $\mathcal{O}(m\log(m))$ operations. Notice that, the stopping criterion is usually based on the diminishing gradients norm of the objective function f. Regarding the total computational complexity, sublinear and dimension-dependent convergence rates for multiple BCGD schemes have been provided in Nesterov (2012); Patrascu and Necoara (2015a,b) for convex and non-convex (sparse) optimization problems.

Algorithm 2: Block Coordinate Gradient Descent

Data: signals $\boldsymbol{Y} \in \mathbb{R}^{m^2\times N}$,
initial dictionary $\boldsymbol{D}^{[0]} \in \mathbb{R}^{m^2\times n}$,
representations $X^{[k]} \in \mathbb{R}^{n\times N}$

Initialize $t = 0$
while *stopping criterion do not hold* **do**
  Choose randomly $1 \leq i_t \leq n, 1 \leq \ell_t \leq m$
  Compute gradient step:
  $\boldsymbol{\mathcal{D}} = \left[\boldsymbol{D}_{i_t}^{[t]}\right]_{\ell_t} - \frac{1}{\mathcal{L}_{i_t}}\left[\nabla_{\boldsymbol{D}_{i_t}} f(\boldsymbol{D}^{[t]}, \boldsymbol{X}^{[k]})\right]_{\ell_t}$
  Compute projection step:
  $\left[\boldsymbol{D}_j^{[t+1]}\right]_\ell = \begin{cases}\pi_{\mathcal{X}_\ell}(\boldsymbol{\mathcal{D}}) & \text{if } j = i_t, \ell = \ell_t \\ \left[\boldsymbol{D}_j^{[t]}\right]_\ell & \text{otherwise}\end{cases}$
  t = t + 1

Computing the representation. Secondly, the representation problem in $\boldsymbol{X}$ is a nonconvex ℓ_0-constrained quadratic sparse representation problem which can be suboptimally solved with standard schemes such as OMP.

$$\boldsymbol{X}^{[k]} = \arg\min_X \left\|\boldsymbol{Y} - \boldsymbol{D}^{[k]}\boldsymbol{X}\right\|_F^2 \\ \text{s.t.} \quad \|\boldsymbol{X}_i\|_0 \leq s. \tag{6}$$

Remark 1. Proper theoretical analysis of the overall computational complexity of the outer AM scheme, using the convergence rates of the inner BCGD algorithm, will be provided in a future work.

4. SEPARABLE LAPLACIAN CLASSIFICATION

Our second contribution is an adaptation of the Separable Dictionary Learning problem that integrates topological information and is aimed at identifying anomalous graph structures.

The DL framework can also handle multidimensional signals, however the standard formulation requires the signals to be vectorized. This operation breaks the correlations between successive columns of the data matrix, be they spatial, as in the case of images, or temporal, as in multiple measurements applications. The alternative, developed in Hawe et al. (2013), is to train one dictionary for each dimension of the signal, such that each captures the patterns occurring along it. Consider the case of 2D signals, $\boldsymbol{Y} \in \mathbb{R}^{m_1\times m_2\times N}$. The DL problem can be expressed in terms of two dictionaries, $\boldsymbol{D}_1 \in \mathbb{R}^{m_1\times n_1}$ and $\boldsymbol{D}_2 \in \mathbb{R}^{m_2\times n_2}$, and the representation $\boldsymbol{X} \in \mathbb{R}^{n_1\times n_2\times N}$

$$\boldsymbol{Y} = \boldsymbol{D}_1\boldsymbol{X}\boldsymbol{D}_2^T + \boldsymbol{V} \tag{7}$$

The model is equivalent to (1) considering $\boldsymbol{D} = \boldsymbol{D}_2 \otimes \boldsymbol{D}_1$. As such, (7) can be plugged in the objective function (2) to obtain the straightforward 2D adaptation. Note that the normalizing constraint on the dictionary atoms now applies to both $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$.

Both computing the representation and updating the dictionary can benefit from working with a structured, smaller problem. Several extensions of classical methods

Algorithm 3: Graph Laplacian Classification using Separable Dictionary Learning

Data: train signals $\boldsymbol{Y}_{train} \in \mathbb{R}^{m_1 \times m_2 \times N_{train}}$,
test signals $\boldsymbol{Y}_{test} \in \mathbb{R}^{m_1 \times m_2 \times N_{test}}$,
train labels, test labels, sparsity level s, classes C,
initial dictionaries $\boldsymbol{D}_{\boldsymbol{1}}^{(c)} \in \mathbb{R}^{m_1 \times n_1}$, $\boldsymbol{D}_{\boldsymbol{2}}^{(c)} \in \mathbb{R}^{m_2 \times n_2}$

Result: estimated test labels, $\hat{\boldsymbol{l}}$

for $c = 1 : C$ **do**

Train class dictionaries: $\boldsymbol{D}_{\boldsymbol{1}}^{(c)}, \boldsymbol{D}_{\boldsymbol{2}}^{(c)}$ using $\boldsymbol{Y}_{train}^{(c)}$ and s

Compute representation $\boldsymbol{X}^{(c)}$ of all $\boldsymbol{Y}_{test}$ using $\boldsymbol{D}_{\boldsymbol{1}}^{(c)}, \boldsymbol{D}_{\boldsymbol{2}}^{(c)}$

Compute representation errors: $\boldsymbol{\epsilon}_c = \left\| \boldsymbol{Y}_{test} - \boldsymbol{D}_{\boldsymbol{1}}^{(c)} \boldsymbol{X}^{(c)} \boldsymbol{D}_{\boldsymbol{2}}^{(c)T} \right\|_F^2$

Classify test signals: $\hat{\boldsymbol{l}}_{test} = \arg\min(\boldsymbol{\epsilon})$

have been proposed for the separable case, with evident advantages in what complexity is concerned and similar, if not equal, performance compared to the standard algorithms. We briefly describe the ones used later in our simulations. The sparse coding step can be efficiently computed via an adaptation of the OMP algorithm Fang et al. (2012) that considers pairs of atoms from $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ when updating the support, resulting in a significant complexity reduction of a factor of m compared to applying standard OMP on the vectorized formulation. As for the dictionary learning step, Pairwise Approximate K-SVD Irofti and Dumitrescu (2019) performs an alternate update of the two dictionaries. Keeping one dictionary fixed when renewing the atoms of the other ensures that competing pairs that share one atom do not hinder convergence. The update itself is in the spirit of the original AK-SVD.

Our next strategy is to exploit the two dimensional structure of a graph Laplacian in order to learn connectivity patterns that are specific to each class of graphs. The adaptation of the classification method previously presented to the separable structure case proceeds as follows.

The training signals from each class are used separately to train one pair of dictionaries. Classifying a new, test signal, will account to evaluating which pair of dictionaries is better at representing the signal. This ability is assessed by computing the root mean square error, defined as $\text{RMSE} = \frac{1}{\sqrt{m_1 m_2 N}} \left\| \boldsymbol{Y} - \boldsymbol{D_1} \boldsymbol{X} \boldsymbol{D_2}^T \right\|_F$. The smallest of the c errors associated to each signal marks the estimated class. Algorithm 3 resumes the steps above.

We use the superscript (c) notation to indicate variables relating to class c. Dictionaries $\boldsymbol{D}_{\boldsymbol{1}}^{(c)}$ and $\boldsymbol{D}_{\boldsymbol{2}}^{(c)}$ are trained on signals that belong to c, $\boldsymbol{Y}^{(c)}$.

5. GRAPH ORTHONORMAL BLOCKS CLASSIFICATION

We now turn to the problem of classifying signals that have an underlying network structure. Topology is no longer explicitly given, however we wish to take advantage of the existing structural clues. For the task, we adapt a DL scheme such that the dictionary atoms incorporate network information.

The standard DL problem designs dictionary $\boldsymbol{D}$ as a set of n independent atoms, but for certain classes of signals it has been shown Irofti (2015) that imposing structure on $\boldsymbol{D}$ and its atoms improves its representation power. A common approach is to structure the dictionary as a union of orthonormal blocks Lesage et al. (2005); Rusu and Dumitrescu (2013). The DL problem becomes

$$\begin{aligned} \min_{\boldsymbol{Q}_1 \ldots, \boldsymbol{Q}_B, \boldsymbol{X}} \quad & \|\boldsymbol{Y} - [\boldsymbol{Q}_1 \ \boldsymbol{Q}_2 \ \ldots \ \boldsymbol{Q}_B] \, \boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{X}_i\|_0 \leq s, \ 1 \leq i \leq N \\ & \boldsymbol{Q}_{\boldsymbol{j}}^{\boldsymbol{T}} \boldsymbol{Q}_{\boldsymbol{j}} = \boldsymbol{I}_m, \ 1 \leq j \leq B \end{aligned} \tag{8}$$

where each block of atoms $\boldsymbol{Q}$ represents a set of atoms, forming an orthogonal basis, that are optimized together. It is common to instill these blocks with extra properties that allow for faster or better approximation algorithms. Single Block Orthogonal (SBO) algorithm [Rusu and Dumitrescu (2013)] builds the dictionary as a union of L orthogonal basis (or blocks) $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$. Given signal $\boldsymbol{y}$, the representation stage identifies the basis that represents it best and uses thresholding to impose sparsity.

Remark 2. Due to orthogonality, the hard-thresholding operation of canceling all but the s absolute largest coefficients of $\boldsymbol{x} = \boldsymbol{Q}^T \boldsymbol{y}$ is optimal.

Thus, the s-sparse representation $\boldsymbol{x}$ of a signal $\boldsymbol{y}$ using block $\boldsymbol{Q}$ can be simply implemented through the partial sorting function: $\boldsymbol{x} = \text{SELECT}(\boldsymbol{Q}^T \boldsymbol{y}, s)$.

Proposition 1. (Dumitrescu and Irofti, 2018, Chapter 7) Given a union of B orthogonal blocks, the best basis j to represent a given signal $\boldsymbol{y}$ is picked by computing the energy of the resulting representation coefficients and selecting the block where the energy is highest. More precisely, if

$$\boldsymbol{x}_i = \text{SELECT}(\boldsymbol{Q}_i^T \boldsymbol{y}, s) \tag{9}$$

is the representation using block $\boldsymbol{Q}_i$, and $E_i = \|\boldsymbol{x}_i\|^2$ is its energy, then the best orthogonal block is given by

$$j = \arg \max_{i=1:B} (E_i). \tag{10}$$

■

During the dictionary refinement stage, each basis $\boldsymbol{Q}_j$ is updated based on the N_j signals using it for representation (in the same spirit as K-SVD). The minimization of the representation error with such a dictionary is called the orthogonal Procrustes problem.

Proposition 2. (Dumitrescu and Irofti, 2018, Chapter 4) Given the matrices $\boldsymbol{Y}, \boldsymbol{X} \in \mathbb{R}^{m \times N_j}$, if $\boldsymbol{Q}_j \in \mathbb{R}^{m \times m}$ is orthogonal, then the approximation error $\|\boldsymbol{Y} - \boldsymbol{Q}_j \boldsymbol{X}\|_F$ is minimized by

$$\boldsymbol{Q}_j = \boldsymbol{V} \boldsymbol{U}^T, \tag{11}$$

where the matrices $\boldsymbol{U}$, $\boldsymbol{V}$ are obtained from the singular value decomposition

$$\boldsymbol{X} \boldsymbol{Y}^T = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T. \tag{12}$$

Here $\boldsymbol{\Sigma}$ is diagonal, $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal and all are $m \times m$. ■

A special characteristic of the SBO algorithm is that it starts with an initial set of bases, which it then expands during training. Before adding a basis to the existing set, a given percentage of the worst represented signals are collected in $\boldsymbol{W}$ and the new basis is initialized by performing a few rounds of training on $\boldsymbol{W}$.

Table 1. Circular Subgraph Anomalies. Classification Accuracy (%).

L-structured DL	Separable L-Class	DL Classification	OC-SVM
91.31	90.64	89.55	81.1

Remark 3. (Improvements via Parallelism). The special dictionary structure of SBO and its effects on the representation and training stages make it a good candidate for parallelization. Indeed, this is demonstrated by the GPU implementation from Irofti (2016) where P-SBO is proposed (P stands for parallel). P-SBO expands the set of bases by more than one at a time which, besides improving parallelism, has been shown to also reduce the representation error.

Remark 4. (Advantages over AK-SVD). Due to the simplicity of the representation stage, $O(m^2)$ for representation and $O(m)$ for thresholding, the main computation demand of SBO is the dictionary training stage, whose complexity is driven by the final number of bases in the dictionary and the number of rounds necessary to train them. On the other hand, OMP is usually the algorithm of choice when performing representation with unstructured dictionaries due to its speed and representation power. However, it is also the main computational bottleneck of the K-SVD family, having a complexity of $O(ms(Bm + s^2))$, which makes AK-SVD pale in comparison with SBO in terms of execution time for both DL and signal representation (see the execution times in Irofti (2016)).

Imposing Laplacian structure on the dictionary can also be achieved by adapting the SBO algorithm presented previously. Indeed, during training, we initialize each orthogonal basis with a fixed, orthogonalized Laplacian matrix. The SBO training rounds performed on each basis will further adapt and refine this original laplacian to better fit the signals using the current basis.

The end goal is to classify signals that are known to lie on a graph whose structure is unknown. This is achieved at the end of the learning process by employing a SRC-like scheme for handling the classification task, that can be now performed by simply using the basis allocation scheme described in Proposition 1. As such, our application performs a separate SBO instance for each class specific signal training set. The key is to initialize the bases with the orthogonalized true Laplacian of the corresponding class and then we can proceed with the standard P-SBO. The resulting union of bases of each class $Q^{(c)}$ are collected together and classification consists in applying (10) on the set of all bases such that the chosen basis indicates the signal class.

6. RESULTS

The aim of the following two synthetic experiments is to show how transferring structural information from the graphs to the dictionary atoms themselves improves the performance in classification tasks when compared to standard DL methods that are oblivious to the nature of the signals. The first considers the case where the signals directly express network topologies, the second deals with more generic signals that rest on a graph.

6.1 Anomalous Subgraph Patterns

Our first experiment is designed to test the ability of our proposed methods to distinguish between different network structures. As such, we test our Separable Laplacian Learning and the Laplacian-structured method on a synthetic graph 2-class classification dataset, constructed with the application of anomaly detection in mind. All signals are generated to have the same network topology, while on a small number of signals we implant an anomaly, namely a structurally distinct subgraph. We design two different types of anomalous subgraphs that are relevant to fraud detection.

The normal graphs are constructed using a stochastic block model with 8 modules, strong intra-module connectivity (controlled by the diagonal dominance of the probability matrix) and inter-module probability of 0.05. Each graph has $m_n = 50$ nodes. The first type of anomalies consists in Watts-Strogatz graphs of $m_a = 10$ nodes, with mean degree $k = 4$ and rewiring probability $\beta = 0.2$. The configuration leads to a network with prominent circular structure. All weight values, regardless of whether the edges are in the normal or anomalous part of the graph, are distributed normally in $[0, 100]$. We construct $N_n = 5000$ normal graphs and $N_a = 500$ anomalies and compute their Laplacian matrices. We only consider the case of undirected graphs, however the solutions can be used for directed graphs, as well.

We test the ability of two of our approaches in learning to classify these Laplacians and compare with the classic SRC-like scheme, where, as described in Section 2.1, we train different dictionaries on the signals of each class. Moreover, we compare the DL methods with One-Class Support Vector Machine (OC-SVM) (see Tian et al. (2018)), a reliable unsupervised anomaly detection method, that has been successfully employed for anti-money laundering tasks (see T. Jun (2005)). OC-SVM takes the true anomaly ratio (10%) as input parameter in order to derive the decision bound between the normal signals and the outliers.

In order to suit the classical DL and OC-SVM algorithms, we vectorize the Laplacian matrices. Vectorization is also needed in our proposed Laplacian-structured Dictionary Learning method. Our Separable Laplacian Classification adaptation, on the other hand, works directly on the matrices. The appropriate sparsity level is not given before hand and optimal values are usually determined through empirical tests. There is however the typical value of $\sqrt{m}$, which is known to yield good results in some applications. An adaptation of this popular choice to the problem of Laplacian learning asks for setting s to be roughly the square root of the number of edge connections in the graph. However, since this value differs slightly in different network realizations, for consistency reasons we set $s = 30$ in all experiments. All the algorithms were applied on 80% of the signals. The remaining 20% are used to test the classification performance.

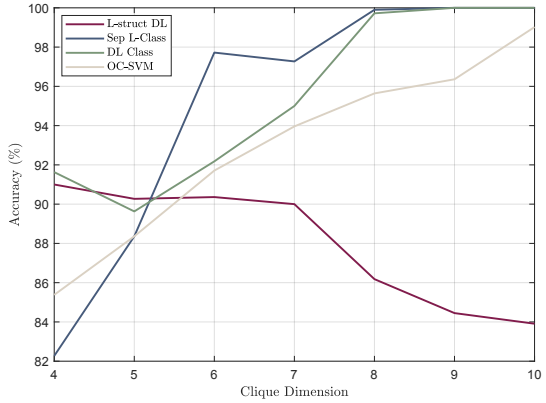

Fig. 1. Classification success on clique subgraph anomalies

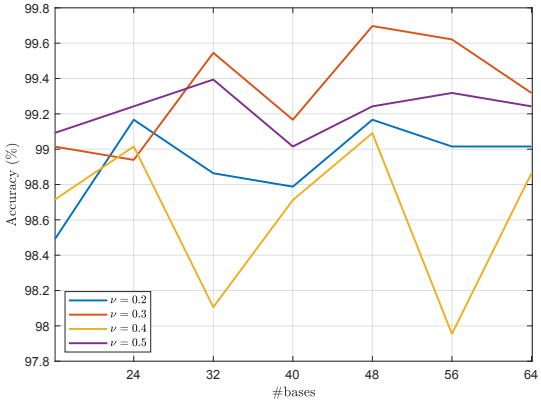

Fig. 2. Influence of SBO parameters on performance

Classification accuracy (i.e. percentage of correctly labeled graphs) results presented in Table 6 show that imposing or exploiting structure leads to increased performance. For compactness, we have use the "L-structured DL" abbreviation to denote our Laplacian-structured dictionary learning method; "Separable L-Class" to denote our Separable Laplacian Classification algorithm and refer to the standard SRC-like classification method as "DL Classification". Results show that when the DL model incorporates known information on the signals (i.e. their Laplacian structure), the solution leads to a more accurate identification of anomalous graphs, compared to the "blind" SRC-like method and OCSVM.

The same testing procedure is applied to a second type of subgraph anomalies, namely cliques. We now implant clique-structured subgraphs in the normal graphs, which we construct as before. The location of the implanted anomaly is chosen at random. We assess the performance on different sized cliques. In the case of our Separable L-Class algorithm, the smaller anomalies (of 4 and 5 nodes respectively) require an adjustment of the dictionary overcompleteness factor (which dictates the number of atoms with respect to signal size). In those cases, the best performance is obtained for $n = 6m_n$, while for the rest $n = 2m_n$ suffices. Recall that m_n represents the number of nodes in the normal graph. Figure 1 shows the classification accuracy of our two proposed algorithms, Laplacian-Structured Dictionary Learning and Separable Laplacian Classification compared to the plain Dictionary Learning Classification and One-Class SVM.

Money laundering schemes most often entail unnatural connections between the nodes of the transactions graph, such as circular or clique patterns, addressed in our synthetic experiment. The performance of the proposed methods is encouraging with respect to identifying other anomalous network structures as well.

6.2 Anomalous Signals on Graphs

In a second set of experiments we test our adaptation of SBO to the problem of classifying signals that sit on different graph topologies. It is common practice in synthetic DL experiments to generate the data as a linear combination of s atoms of a known random dictionary. In our case, we also require the signals to rest on a known graph with Laplacian $\boldsymbol{L}$. We follow the principle described in Yankelevsky and Elad (2016) that ensures a proper coupling between the generating Laplacian and dictionary and construct the signals of each class using the following scheme.

Starting with a random initial $\boldsymbol{D}^{[0]}$ (i.e. having random vectors as atoms), we obtain our true dictionary as $\boldsymbol{D} = (\lambda \boldsymbol{I} + \boldsymbol{L})^{-1}\boldsymbol{D}^{[0]}$, where $\boldsymbol{L}$ is the graph Laplacian of the corresponding class. We use the same normal and circular topologies as in the previous experiment. The parameter λ controls the smoothness of the dictionary atoms. While atom smoothness is not crucial to our application, the choice of the parameter will be reflected in how well the signals adhere to the underlying graph structure. As per the above mentioned reference, we set $\lambda = 5$. Following the construction of $\boldsymbol{D}$, we generate the signals using $s = 4$ random atoms and add Gaussian noise of level $SNR = 20$. We set $N_1 = 6000$ for the normal class and $N_2 = 600$ for the anomalies. As previously, 80% of the signals are used for training, while the remaining for testing. To initialize the union of orthogonal blocks, we orthogonalize the true generating Laplacians. These initial 2 blocks are subsequently refined using $N/8$ signals of the corresponding class. Figure 2 shows the influence that the number of blocks and fraction ν of badly represented signals used for basis construction have on performance. We perform 7 rounds of base refinement and learn 6 bases in parallel (see Remark 3).

Best results, **99.70**% classification accuracy, are obtained when working with 48 bases for each class and retraining on 30% of the signals. Applying the SRC-like classification on the dataset yields **99.77**%. The small difference is however compensated by the definite complexity advantage of SBO over SRC Irofti (2016), as detailed in Remark 4.

7. CONCLUSIONS

We proposed three dictionary learning methods for exploiting the structural information of graphs in order to improve the network classification task. When working directly with the structure of the graphs, our method of imposing Laplacian structure on the dictionary atoms has yielded better results compared to the standard dictionary classification algorithm and to OCSVM. Our adaptation of the separable dictionary learning problem, which takes into account vicinity patterns in 2D data also constitutes a better alternative to the classical solution, which is obliv-

ious to the underlying structure, as well as to OCSVM. As for the more general problem of signals that lie on graphs, our adaptation of a block orthogonal algorithm, that imposes a Laplacian-like structure on the dictionary has yielded similar performance compared to the classic dictionary classification method, however with known computational advantages.

REFERENCES

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11), 4311–4322.

Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3), 626–688.

Bolton, R.J. and Hand, D.J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235–255.

Dong, X., Thanou, D., Rabbat, M., and Frossard, P. (2019). Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3), 44–63.

Dumitrescu, B. and Irofti, P. (2018). *Dictionary Learning Algorithms and Applications*. Springer.

Elliott, A., Cucuringu, M., Luaces, M.M., Reidy, P., and Reinert, G. (2019). Anomaly detection in networks with application to financial transaction networks. *arXiv preprint arXiv:1901.00402*.

Fang, Y., Wu, J., and Huang, B. (2012). 2D sparse signal recovery via 2D orthogonal matching pursuit. *Science China Information Sciences*, 55(4), 889–897.

Hawe, S., Seibert, M., and Kleinsteuber, M. (2013). Separable dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 438–445.

Irofti, P. (2015). Sparse denoising with learned composite structured dictionaries. In *19th International Conference on System Theory, Control and Computing*, 331–336.

Irofti, P. (2016). Efficient parallel implementation for single block orthogonal dictionary learning. *Journal of Control Engineering and Applied Informatics*, 18(3), 101–108.

Irofti, P. and Dumitrescu, B. (2019). Pairwise approximate k-svd. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 3677–3681.

Irofti, P. and Stoican, F. (2017). Dictionary learning strategies for sensor placement and leakage isolation in water networks. In *The 20th World Congress of the International Federation of Automatic Control*, 1589–1594.

Jiang, Z., Lin, Z., and Davis, L. (2011). Learning A Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1697–1704.

Kiwiel, K.C. (2007). On linear-time algorithms for the continuous quadratic knapsack problem. *Journal of Optimization Theory and Applications*, 134, 549–554.

Lesage, S., Gribonval, R., Bimbot, F., and Benaroya, L. (2005). Learning unions of orthonormal bases with thresholded singular value decomposition. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, 293–296 Vol. 5.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2008). Discriminative learned dictionaries for local image analysis. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady Akademii Nauk*, 269, 543–547.

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22, 341–362.

Pati, Y., Rezaiifar, R., and Krishnaprasad, P. (1993). Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *27th Asilomar Conference on Signals, Systems and Computers*, volume 1, 40–44.

Patrascu, A. and Necoara, I. (2015a). Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization*, 61(1), 19–46.

Patrascu, A. and Necoara, I. (2015b). Random coordinate descent methods for ℓ_0 regularized convex optimization. *IEEE Transactions on Automatic Control*, 60(7), 1811–1824.

Rusu, C. and Dumitrescu, B. (2013). Block orthonormal overcomplete dictionary learning. In *21st European Signal Processing Conference (EUSIPCO 2013)*, 1–5.

Skretting, K. and Håkon Husøy, J. (2006). Texture classification using sparse frame-based representations. *EURASIP Journal on Advances in Signal Processing*, 2006.

T. Jun, Y.J. (2005). Developing an intelligent data discriminating system of anti-money laundering based on svm. In *ICMLC*, volume 6, XIV, 284.

Tian, Y., Mirzabagheri, M., Bamakan, S.M.H., and H. Wang, Q.Q. (2018). Ramp loss one-class support vector machine; a robust and effective approach to anomaly detection problems. *Neurocomputing*, 310, 223–235.

Wright, J., Yang, A., Ganesh, A., Sastry, S., and Yu, L. (2009). Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31, 210–227.

Yankelevsky, Y. and Elad, M. (2016). Dual graph regularized dictionary learning. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4), 611–624.

Yankelevsky, Y. and Elad, M. (2017). Structure-aware classification using supervised dictionary learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4421–4425.

Zheng, M., Bu, J., Chen, C., Wang, C., Zhang, L., Qiu, G., and Cai, D. (2011). Graph regularized sparse coding for image representation. *IEEE Transactions on Image Processing*, 20(5), 1327–1336.