PCA USING GRAPH TOTAL VARIATION

Nauman Shahid, Nathanael Perraudin, Vassilis Kalofolias, Benjamin Ricaud, Pierre Vandergheynst

Ecole Polytechnique Federale de Lausanne

ABSTRACT

Mining useful clusters from high dimensional data has received significant attention of the signal processing and machine learning community in the recent years. Linear and non-linear dimensionality reduction has played an important role to overcome the curse of dimensionality. However, often such methods are accompanied with problems such as high computational complexity (usually associated with the nuclear norm minimization), non-convexity (for matrix factorization methods) or susceptibility to gross corruptions in the data. In this paper we propose a convex, robust, scalable and efficient Principal Component Analysis (PCA) based method to approximate the low-rank representation of high dimensional datasets via a two-way graph regularization scheme. Compared to the exact recovery methods, our method is approximate, in that it enforces a piecewise constant assumption on the samples using a graph total variation and a piecewise smoothness assumption on the features using a graph Tikhonov regularization. Futhermore, it retrieves the low-rank representation in a time that is linear in the number of data samples. Clustering experiments on 3 benchmark datasets with different types of corruptions show that our proposed model outperforms 7 state-of-the-art dimensionality reduction models.

Index Terms— PCA, graph total variation, low-rank feature extraction, clustering

1. INTRODUCTION

In modern signal processing and machine learning applications one often needs to manipulate data for which the hidden structure can be synthesized in the form of a graph. In some cases, this type of structure reveals itself naturally for real life networks, such as the citations (of scientific articles) or social interactions. In addition, graphs are able to model a manifold embedded into a high dimensional space leading to many application for the data mining community [1]. Based on spectral graph theory and computational harmonic analysis [2], graph techniques are nowadays a traditional tool to extract low dimensional features from the data [3].

Our work is focused on PCA which is the most widely used feature extraction tool for linear dimensionality reduction and clustering. Given a data matrix $X \in \mathbb{R}^{p\times n}$ with n p-dimensional data vectors, PCA can be formulated as learning the projection $Q \in \mathbb{R}^{n\times d}$ (principal components) of X in a d-dimensional linear space characterized by an orthonormal basis $V \in \mathbb{R}^{p\times d}$ (principal directions). The product $VQ^\top$ is known as the *low-rank approximation* $U \in \mathbb{R}^{p\times n}$ of X. The dimensionality reduction and clustering properties of PCA have significantly benefited from the graph structured data representation in the past decade. Several PCA based models which benefit from the graph smoothness assumption of the principal components Q have been proposed recently [4], [5], [6]. These methods, however, are non-convex and susceptible to gross corruptions.

The techniques presented in the above mentioned works extract features based on the assumption that the data is low rank and evolves smoothly over the graph of $n \times n$ samples. However, we could also assume that the representation is *piecewise constant over the graph*. Imagine for instance a dataset made from noisy faces of 5 different people. Ideally, the low rank representations of the same person's faces should not just be close to each other (following a traditional smoothness assumption) but actually the same. This leads to the new assumption we adopt, namely that the low rank representation is piecewise constant on the underlying graph.

We propose a convex, fast and efficient clustering algorithm for corrupted low-rank signals. Our model is inspired by the two-way graph regularization scheme introduced in the context of matrix completion by Kalofolias et.al. [7]. In contrast to [4], [5], [6] and [7] which use Tikhonov graph regularization $\mathrm{tr}(x^T Lx) = \|\nabla_{\mathcal{G}} x\|_2^2)$ (G-TIK) w.r.t. the data samples, we propose to use the *graph total variation (G-TV) regularization* ($\|\nabla_{\mathcal{G}} x\|_1$) w.r.t the data samples which enforces our piecewise constant assumption. Additionally and unlike [4], [5], [6], we also propose to use a G-TIK w.r.t. the features of the data matrix. This two-way graph regularization scheme helps in 1) the extraction of a fast low-rank approximation U of the data matrix X without using the expensive nuclear norm operator (as in [8]) and 2) better clustering in the low-dimensional space due to piecewise constant assumption enforced by the G-TV. Furthermore, in contrast to the L_2 data fidelity term of [4], [5], [6], we propose to use the L_1 fidelity term which makes our model robust to outliers or gross corruptions in the dataset. Finally, our model can be solved efficiently, in time linear in the number of data samples, with a forward-backward based primal dual algorithm. We point out here that the G-TV framework has been used in previous works, such as [9], [10]. However, to the best of our knowledge it has never been used in the context of PCA.

2. GRAPH NOMENCLATURE

A graph is a tupple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ where $\mathcal{V}$ is a set of vertices, $\mathcal{E}$ a set of edges, and $\mathcal{W} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+$ a weight function. The vertices are indexed from $1, \ldots, |\mathcal{V}|$ and each entry of the weight matrix $W \in \mathbb{R}_+^{|\mathcal{V}|\times|\mathcal{V}|}$ contains the weight of the edge connecting the corresponding vertices: $W_{i,j} = \mathcal{W}(v_i, v_j)$. If there is no edge between two vertices, the weight is set to 0. We assume W is symmetric, non-negative and with zero diagonal. We denote by $i \leftrightarrow j$ that node v_i is connected to node v_j. For a vertex $v_i \in \mathcal{V}$, the degree $d(i)$ is defined as the sum of the weights of incident edges: $d(i) = \sum_{j\leftrightarrow i} W_{i,j}$. In this framework, a graph signal is defined as a function $s : \mathcal{V} \to \mathbb{R}$ assigning a value to each vertex. It is convenient to consider a signal

Thanks to the Swiss National Foundation (SNF) grant 200021_154350/1 for funding this work.

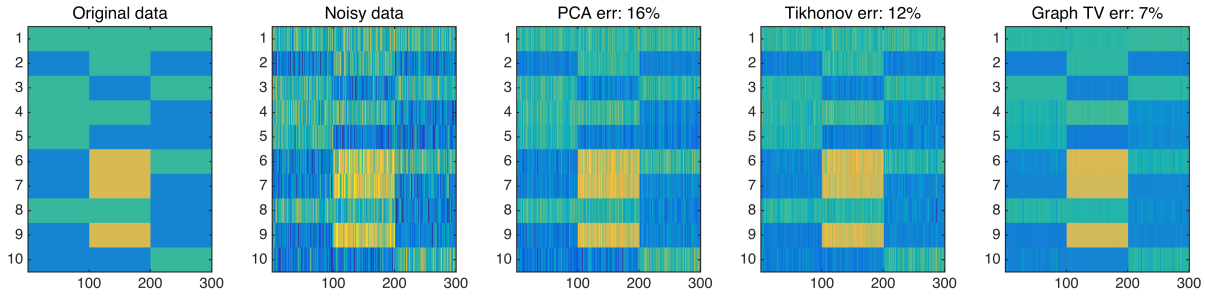

Fig. 1. Recovery of a rank 3 matrix corrupted by Gaussian noise using PCA, G-TIK and G-TV. The G-TV based model recovers the best piecewise constant low-rank representation which is closer to the original data matrix (in Frobenius norm), whereas the G-TIK model recovers a weaker low-rank representation due to the smoothness assumption.

s as a vector of size $|\mathcal{V}|$ with the i^{th} component representing the signal value at the i^{th} vertex. For a signal s living on the graph $\mathcal{G}$, the gradient $\nabla_{\mathcal{G}} : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}^{|\mathcal{E}|}$ is defined as

$$\nabla_{\mathcal{G}} s(i,j) = \sqrt{W(i,j)} \left(\frac{s(j)}{\sqrt{d(j)}} - \frac{s(i)}{\sqrt{d(i)}} \right),$$

where we consider only the pair $\{i,j\}$ when $i \leftrightarrow j$. For a signal c living on the graph edges, the adjoint of the gradient $\nabla^*_{\mathcal{G}} : \mathbb{R}^{|\mathcal{E}|} \to \mathbb{R}^{|\mathcal{V}|}$, called divergence can be written as

$$\nabla^*_{\mathcal{G}} c(i) \quad = \quad \sum_{i \leftrightarrow j} \sqrt{W(i,j)} \left(\frac{1}{\sqrt{d(i)}} c(i,i) - \frac{1}{\sqrt{d(j)}} c(i,j) \right).$$

The Laplacian corresponds to the second order derivative and its definition arises from $Ls := \nabla^*_{\mathcal{G}} \nabla_{\mathcal{G}} s$. In this work, we use the normalized graph Laplacian L defined as $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ where D is the diagonal degree matrix with diagonal entries $D_{ii} = d(i)$ and I the identity.

3. PROPOSED MODEL (PCA-GTV)

We associate two different graphs to our data $X \in \mathbb{R}^{p \times n}$. The columns of X live on the first graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1, \mathcal{W}_1)$ that connects different samples of X. The rows of X live on a second graph $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2, \mathcal{W}_2)$ that connects the different features. Let $L_1 \in \mathbb{R}^{n \times n}$, $L_2 \in \mathbb{R}^{p \times p}$ be the graph Laplacians and $\nabla_{\mathcal{G}_1}$, $\nabla_{\mathcal{G}_2}$ be the gradients of the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. The construction of these two graphs is described in Section 4. Let $U \in \mathbb{R}^{p \times n}$ be the low-rank noiseless matrix that needs to be recovered from the data X, then our proposed model can be written as:

$$\min_U \|X - U\|_1 + \gamma_1 \|\nabla_{\mathcal{G}_1} U^T\|_1 + \gamma_2 \|\nabla_{\mathcal{G}_2} U\|_F^2. \qquad (1)$$

where γ_1 and γ_2 control the trade-off between the two regularization terms, $\|\cdot\|_1$ and $\|\cdot\|_F$ denote the matrix L_1 and frobenius norms respectively. The term $\|X - U\|_1$ helps in retrieving a low-rank representation U that is robust to outliers or sparse gross errors in the data set. We call model (1) *graph total variation PCA* (PCA-GTV). This model has important implications in recovering robust approximate low-rank representation of the data.

The first graph regularization term $\|\nabla_{\mathcal{G}_1} U^\top\|_1$, also known as graph total variation (G-TV), enforces sparse gradients w.r.t $\mathcal{G}_1$, where the non-zeros tend to occur when there is a transition from one cluster to another. In other words, it uses the underlying structure between the samples encoded in $\mathcal{G}_1$ and then enforces a piecewise constant assumption along the columns of U. This automatically enforces the matrix U to acquire a low-rank structure. This is in contrast to the graph Tikhonov (G-TIK) regularization term $\|\nabla_{\mathcal{G}_1} U^\top\|_F^2 = \text{tr}(U L_1 U^\top)$ used in [4], [5], [6] which allows the gradients to vary smoothly w.r.t the graph $\mathcal{G}_1$.

We motivate the use of G-TV over G-TIK with a small experiment demonstrated in Fig. 1 where we show the recovery of a rank 3 matrix corrupted with Gaussian noise. We use the standard PCA, G-TV with L_1 data fidelity term ($\|X - U\|_1$) and G-TIK with L_1 data fidelity term to recover the low-rank structure. Clearly, the G-TV based model recovers a piecewise constant low-rank matrix which has lower error as compared to other models.

The role of the second graph can be defined in a similar manner in that it uses the underlying structure between the features encoded in L_2. The prior $\|\nabla_{\mathcal{G}_2} U\|_F^2 = \text{tr}(U^T L_2 U)$, (G-TIK) on the features then enforces smoothness of U in the Laplacian basis of L_2. Used together, these two priors push towards a matrix U that is close to low-rank in both the columns and the rows.

4. GRAPHS CONSTRUCTION

The graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are built using a two-step standard K-nearest neighbor strategy. In the first step the search for the closest neighbours for all the samples is performed using the Euclidean distance metric. Each x_i is connected to its K nearest neighbors x_j, resulting in $|\mathcal{E}|$ number of connections. In the second step the graph weight matrix W is computed as

$$W_{i,j} = \begin{cases} \exp\left(- \frac{\|B_{i,j} \circ (x_i - x_j)\|_2^2}{\|B_{i,j}\|_1 \sigma^2} \right) & \text{if } x_j \text{ is connected to } x_i \\ 0 & \text{otherwise.} \end{cases}$$

where $B_{i,j} \in \{0,1\}^p$ is the vector mask corresponding to the intersection of uncorrupted values in x_i and x_j and $\circ$ denotes the Hadamard product. The use of mask B (if available) makes the graph robust to gross corruptions in the dataset. The parameter σ can be set experimentally as the average distance of the connected samples. Finally, the third step consists of constructing the normalized graph Laplacian $L = I - D^{-1/2} W D^{-1/2}$, where D is the degree matrix. This procedure has a complexity of $\mathcal{O}(ne)$ and each $W_{i,j}$ can be computed in parallel.

Depending on the values of n and p the above computation can be done in two different ways.

Small n, p: In this case the above strategy can be used directly for both $\mathcal{G}_1$ and $\mathcal{G}_2$. This is true for the case even when the dataset is corrupted and we know the mask for corruptions. It is worth mentioning here that the computation of W is $\mathcal{O}(n^2)$ but with sufficiently small n and p, the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be easily computed in the order of a few seconds.

Large n, p: For this case a similar strategy can be used but the computations can be made efficient using the FLANN library (Fast Library for Approximate Nearest Neighbors searches in high dimensional spaces) [11]. However, the FLANN library does not support the mask operator so the corruptions are included in the graphs construction. Therefore, the quality of the graphs constructed using this strategy is slightly lower as compared to strategy 1 due to 1) the approximate nearest neighbor search method and 2) corruptions (if any) even if the mask information is known.

Image vs. non-image data: For the non-image datasets, the graph $\mathcal{G}_2$ is simply constructed between the features of X. However, for the case of images it is more reasonable to enforce smoothness on the patch level rather than on the pixel level. As a first step, the patches that correspond to the same position for all the images in the dataset are vectorized. Let l^2 be the size of each square patch which is centered at the pixel under consideration, then we form p data samples each of size nl^2 . These transformed data samples are then fed into the graph construction algorithm described earlier.

5. OPTIMIZATION SOLUTION

5.1. Forward-backward based primal dual

We use proximal splitting methods in order to solve problem (1). The particularity of these methods is that they cut a complex problem into smaller and easier subproblems which are solved using proximal operators. The proximal operator of a function λh is defined as

$$\text{prox}_{\lambda h}(y) = \underset{x}{\text{argmin}} \frac{1}{2}\|x-y\|_2^2 + \lambda h(x).$$

General information about these methods can be found in [12, 13]. Let us cast our problem in the general form

$$\underset{x}{\text{argmin}}\, f(x) + g(Ax) + h(x). \tag{2}$$

The first term of (2), $f : \mathbb{R}^{np} \to \mathbb{R}$ is a convex function defined as $f(U) = \|X - U\|_1$. The proximal operator of the function f is the ℓ_1 soft-thresholding given by the elementwise operations.

$$\text{prox}_{\lambda f}(U) = X + \text{sgn}(U-X) \circ \max(|U-X| - \lambda, 0). \tag{3}$$

The second term of (2), $g : \mathbb{R}^{|\mathcal{E}_1|p} \to \mathbb{R}$, where $|\mathcal{E}_1|$ denotes the cardinality of $\mathcal{E}_1$, is a convex function defined as $g(D) = \gamma_1\|D\|_1$. The proximal operator of function g is

$$\text{prox}_{\lambda g}(D) = \text{sgn}(D) \circ \max(|D| - \lambda\gamma_1, 0). \tag{4}$$

The third term of (2), $h : \mathbb{R}^{np} \to \mathbb{R}$ is a convex, differentiable function defined as $h(U) = \gamma_2\|\nabla_{G_2} U\|_F^2 = \gamma_2 \,\text{tr}(U^\top L_2 U)$. This function has a β-Lipschitz continuous gradient

$$\nabla_h(U) = 2\gamma_2 U^\top L_2 U.$$

Note that $\beta = 2\gamma_2\|L_2\|_2$ where $\|L\|_2$ is the spectral norm (or maximum eigenvalue) of L. Finally, in (2), A is a linear operator, in our case ∇_{G_1}.

Using these tools, we can use the forward backward based primal dual approach presented in [12] to define Algorithm 1 where τ_1, τ_2, τ_3 are convergence parameters (we use $\tau_1 = \frac{1}{\beta} = \frac{1}{\gamma_2\|L_2\|_2}$, $\tau_2 = \frac{\beta}{2\|\nabla_{\mathcal{G}_1}\|_2^2} = \frac{\gamma_2\|L_2\|_2}{\|L_1\|_2}$ and $\tau_3 = 0.99$), ϵ the stopping tolerance and J the maximum number of iterations. δ is a very small number to avoid a possible division by 0.

Algorithm 1 Forward-backward primal dual for PCA-GTV

INPUT: $U_0 = X, V_0 = \nabla_{\mathcal{G}_1} X, \epsilon > 0$
for $j = 0, \dots J-1$ **do**
  $P_j = \text{prox}_{\tau_1 f}\left(U_j - \tau_1\left(\nabla_h(U_j) + \nabla^*_{\mathcal{G}_1} V_j\right)\right)$
  $T_j = V_j + \tau_2 \nabla_{\mathcal{G}_1}(2P_j - U_j)$
  $Q_j = T_j - \tau_2 \,\text{prox}_{\frac{1}{\tau_2} g}\left(\frac{1}{\tau_2} T_j\right)$
  $(U_{j+1}, V_{j+1}) = (U_j, V_j) + \tau_3\left((P_j, Q_j) - (U_j, V_j)\right)$
  if $\frac{\|U_{j+1}-U_j\|_F^2}{\|U_j\|_F^2+\delta} < \epsilon$ and $\frac{\|V_{j+1}-V_j\|_F^2}{\|V_j\|_F^2+\delta} < \epsilon$ **then**
    BREAK
  end if
end for

6. COMPUTATIONAL COMPLEXITY

Complexity of Graph Construction: The computational complexity of the FLANN algorithm for n p-dimensional vectors and fixed K and l^2 is $\mathcal{O}(pn\log(n))$ for the graph $\mathcal{G}_1$ between the samples and $\mathcal{O}(pn\log(p))$ for the graph $\mathcal{G}_2$ between the features [11].

Algorithm Complexity: Let J denote the number of iterations for the algorithm to converge (usually independent from the number of samples), p the data dimension, n the number of samples and d the rank of the low-dimensional space. For K-nearest neighbours graphs, the computational cost of our algorithm is linear in the number of data samples n, i.e. $\mathcal{O}(JKnp)$.

Final SVD: In order to preserve convexity, our model finds an approximately low-rank solution U without factorizing it. However for clustering applications we might need to provide explicitly the low dimensional representation in a factorized form. This can be done by computing an "economic" SVD of U for $p \ll n$. Let $U = V\Sigma Q^\top$ the SVD of U. The orthonormal basis V can be computed by the eigenvalue decomposition of the small $p \times p$ matrix $UU^\top = VEV^\top$ that also reveals the singular values $\Sigma = \sqrt{E}$ since $UU^\top$ is s.p.s.d. and therefore E is non-negative diagonal. Given V and Σ the sample projections are computed as $Q = \Sigma^{-1}V^\top U$. The complexity of this SVD is $\mathcal{O}(np^2)$.

Overall Complexity: The complexity of our algorithm is $\mathcal{O}(J\max\{pn, |\mathcal{E}_1|n\})$, the graph G_1 is $\mathcal{O}(pn\log(n))$, G_2 is $\mathcal{O}(pn\log(p))$ and the final SVD step is $\mathcal{O}(np^2)$. Given that $p \ll n$, the overall complexity of our algorithm is $\mathcal{O}(pn(\log(n)+p+JK))$.

7. RESULTS

We use clustering experiments to validate the quality of the low-rank features extracted using our model. In fact such types of experiments have become a standard practice in the PCA community [5], [14], [4], [6], [15]. We perform our clustering experiments on 3 benchmark databases: CMU PIE, ORL and COIL20 using two open-source toolboxes: the UNLocBoX [16] for the optimization part and the GSPBox [17] for the graph creation.

In order to evaluate the robustness of our model to gross corruptions we corrupt the datasets with two different types of errors 1) block occlusions and 2) random missing pixels. Block occlusions of three different sizes, i.e, 15%, 25% and 40% of the total size of the image are placed uniformly randomly in all the images of the datasets. Similarly, all the images of the datasets are also corrupted by removing 15%, 25%, and 35% pixels uniformly randomly. Separate clustering experiments are performed for each of the different types of corruptions.

We compare the clustering performance of our model with 7 other models: 1) k-means on original data 2) Laplacian Eigenmaps (LE) [3] 3) Standard PCA 4) Graph Laplacian PCA (GLPCA) [5] 5) Manifold Regularized Matrix Factorization (MMF) [4] 6) Non-negative Matrix Factorization (NMF) [18] and 7) Graph Regularized Non-negative Matrix Factorization (GNMF) [19].

We transform all datasets to zero-mean and unit standard deviation along the features. For MMF the samples are additionally normalized to unit-norm. For NMF and GNMF only the unit-norm normalization is applied to all the samples of the dataset.

We use *clustering error* as a metric to compare clustering performance of various models. LE, PCA, GLPCA, MMF, NMF and GNMF are matrix factorization models that explicitly learn the principal components Q. The clustering error for these models is evaluated by performing k-means on the principal components. Our model on the other hand learns the low-rank matrix U. Thus, the clustering error for our model is evaluated by performing k-means on the principal components Q obtained by the economic SVD of the low-rank matrix $U = V\Sigma Q^\top$, as described in Section 6. Furthermore, our model does not recover an exact low-rank representation. It only shrinks singular values and therefore only recovers an approximate low-rank representation U. Thus, the dimension of the subspace (number of columns of Q) is decided by selecting the number of singular values such that the lowest selected singular value is 10% of the maximum singular value. Due to the non-deterministic nature of k-means, it is run 10 times and the minimum error over all runs is reported.

Each model has several parameters which have to be selected in the validation stage of the experiment. To perform a fair validation for each of the models we use a range of parameter values. For a given dataset, each of the models is run for each of the parameter tupples and the parameters corresponding to minimum clustering error are selected for testing purpose. Furthermore, PCA, GLPCA, MMF, NMF and GNMF are non-convex models so they are run 10 times for each of the parameter tupple, whereas our model is convex so it is run only once.

As mentioned in Section 4, there are two different strategies for graph construction. To perform a fair evaluation of our proposed model for different datasets we use the first strategy (small n, p) for the construction of both graphs for all the experiments reported in this work. We use normalized graph Laplacians and the following parameters for graphs $\mathcal{G}_1$ and $\mathcal{G}_2$: K-nearest neighbors = 10, $\sigma^2 = 1$ and patch size $l^2 = 25$. It is important to point out here that different types of data might call for slightly different parameters for graphs. However, for a given dataset, the use of same graph parameters (same graph quality) for all the graph regularized models ensures a fair comparison.

Table 1 presents a comparison of the clustering error of our model with various other state-of-the-art dimensionality reduction models for three datasets. The best result (lowest clustering error) for each case of the corruption is highlighted in bold. It is quite obvious that our proposed model outperforms other models in most of the scenarios. Another observation is that the clustering error of our model is quite stable as compared to other models even for the large fraction of random missing pixels. This is an interesting result because even with a higher fraction of missing pixels the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ play their respective roles and help in attaining a low clustering error. This proves that good quality graphs possess the ability to recover the data samples even in the presence of a large fraction of gross corruptions.

Table 1. A comparison of clustering error of our model with various dimensionality reduction models. The image data sets include: 1) CMU PIE 2) COIL20 and 3) ORL. The compared models are: 1) k-means 2) Laplacian Eigenmaps (LE) [3] 3) Standard Principal Component Analysis (PCA) 4) Graph Laplacian PCA (GLPCA) [5] 5) Non-negative Matrix Factorization [18] 6) Graph Regularized Non-negative Matrix Factorization (GNMF) [19] and 7) Manifold Regularized Matrix Factorization (MMF) [4]. Two types of corruptions are introduced in the data: 1) Block occlusions and 2) Random missing values. The best results are highlighted in bold.

Data set	Model	No Corruptions	Full Corruptions					
			Occlusions (% of image size)			Missing (% of image pixels)		
			15%	25%	40%	15%	25%	35%
CMUPIE	k-means	72.1	84.3	84.4	84.1	70.7	71.6	73.2
	LE	83.7	79.3	80.3	79.3	84.5	82.9	83.4
	PCA	24.2	68.7	76.9	74.0	28.9	31.6	39.9
	GLPCA	25.5	31.7	33.2	31.8	26.7	26.0	26.4
	NMF	45.8	84.9	84.2	85.7	48.3	51.8	55.7
	GNMF	35.8	53.7	42.3	85.7	31.7	32.0	35.2
	MMF	57.6	53.7	55.2	52.4	52.0	48.7	45.8
	proposed	**22.4**	**29.3**	**30.6**	**27.3**	**22.6**	**22.3**	**25.3**
COIL20	k-means	39.2	54.8	64.9	69.8	39.4	41.4	42.3
	LE	83.7	80.7	81.8	81.5	79.2	85.3	81.5
	PCA	38.2	57.2	57.0	63.7	44.3	47.2	47.7
	GLPCA	17.2	19.2	31.9	**38.2**	20.4	20.2	21.2
	NMF	38.9	59.6	69.4	73.4	38.0	39.4	35.7
	GNMF	12.7	**13.6**	27.9	40.9	13.6	**12.8**	14.9
	MMF	25.3	32.1	34.9	**38.2**	30.2	28.8	33.0
	proposed	**12.4**	14.2	**27.1**	**38.2**	**13.2**	13.5	**14.2**
ORL	k-means	29.7	67.9	70.7	72.4	31.3	36.2	42.3
	LE	21.3	18.3	15.7	22.3	21.7	20.3	19.3
	PCA	35.3	66.3	69.0	70.7	38.7	38.3	42.0
	GLPCA	13.7	11.7	14.7	16.0	14.7	14.3	14.3
	NMF	28.7	76.0	76.0	76.3	30.7	34.3	41.3
	GNMF	24.7	25.0	28.0	30.0	20.3	23.3	21.3
	MMF	18.0	17.0	16.3	76.3	14.7	**13.7**	15.0
	proposed	**13.0**	**11.3**	**14.3**	**15.3**	**14.0**	14.0	**13.0**

8. CONCLUSION

In this paper we present a fast, efficient, scalable and convex dimensionality reduction algorithm for clustering on the low-rank signals. The proposed method benefits from the two-way graph regularization scheme along the rows and columns of the data matrix. We propose to use a total variation graph regularization along the samples and a graph Tikhonov regularization along the features of the data matrix. The underlying assumption is that the low-rank representation of the signals is piecewise constant on the graph between the samples and piecewise smooth on the graph between the features of the data matrix. The proposed algorithm has a linear complexity with respect of the number of data samples. Furthermore, it outperforms several state-of-the-art methods in clustering task.

9. REFERENCES

[1] Mikhail Belkin and Partha Niyogi, "Towards a theoretical foundation for laplacian-based manifold methods," in *Learning theory*, pp. 486–500. Springer, 2005.

[2] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, 2013.

[3] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[4] Zhenyue Zhang and Keke Zhao, "Low-rank matrix approximation with manifold regularization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 7, pp. 1717–1729, 2013.

[5] Bo Jiang, Chris Ding, and Jin Tang, "Graph-laplacian pca: Closed-form solution and robustness," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3492–3498.

[6] Taisong Jin, Jun Yu, Jane You, Kun Zeng, Cuihua Li, and Zhengtao Yu, "Low-rank matrix factorization with multiple hypergraph regularizers," *Pattern Recognition*, 2014.

[7] Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst, "Matrix completion on graphs," *arXiv preprint arXiv:1408.1717*, 2014.

[8] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[9] Xavier Bresson and Arthur D Szlam, "Total variation, cheeger cuts," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 1039–1046.

[10] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *Image Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1047–1060, 2008.

[11] Marius Muja and David Lowe, "Scalable nearest neighbour algorithms for high dimensional data," 2014.

[12] Nikos Komodakis and Jean-Christophe Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *arXiv preprint arXiv:1406.5429*, 2014.

[13] Patrick L Combettes and Jean-Christophe Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.

[14] Liang Tao, Horace HS Ip, Yinglin Wang, and Xin Shu, "Low rank approximation with sparse integration of multiple manifolds for data representation," *Applied Intelligence*, pp. 1–17, 2014.

[15] Yong Peng, Bao-Liang Lu, and Suhang Wang, "Enhanced low-rank representation via sparse manifold adaption for semi-supervised learning," *Neural Networks*, 2015.

[16] N. Perraudin, D. Shuman, G. Puy, and P. Vandergheynst, "UNLocBoX A matlab convex optimization toolbox using proximal splitting methods," *ArXiv e-prints*, Feb. 2014.

[17] Nathanaël Perraudin, Johan Paratte, David Shuman, Vassilis Kalofolias, Pierre Vandergheynst, and David K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.

[18] Daniel D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[19] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.