Low-Rank Tensor Constrained Multiview Subspace Clustering

Changqing Zhang[1] Huazhu Fu[3] Si Liu[2] Guangcan Liu[4] Xiaochun Cao[1,2] *
[1]School of Computer Science and Technology, Tianjin University
[2]State Key Laboratory of Information Security, IIE, Chinese Academy of Sciences
[3]Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore
[4]School of Information and Control, Nanjing University of Information Science and Technology
caoxiaochun@iie.ac.cn; zhangchangqing@tju.edu.cn

Abstract

In this paper, we explore the problem of multiview subspace clustering. We introduce a low-rank tensor constraint to explore the complementary information from multiple views and, accordingly, establish a novel method called Low-rank Tensor constrained Multiview Subspace Clustering (LT-MSC). Our method regards the subspace representation matrices of different views as a tensor, which captures dexterously the high order correlations underlying multiview data. Then the tensor is equipped with a low-rank constraint, which models elegantly the cross information among different views, reduces effectually the redundancy of the learned subspace representations, and improves the accuracy of clustering as well. The inference process of the affinity matrix for clustering is formulated as a tensor nuclear norm minimization problem, constrained with an additional $\ell_{2,1}$-norm regularizer and some linear equalities. The minimization problem is convex and thus can be solved efficiently by an Augmented Lagrangian Alternating Direction Minimization (AL-ADM) method. Extensive experimental results on four benchmark image datasets show the effectiveness of the proposed LT-MSC method.

1. Introduction

Many problems in machine learning and computer vision involve multiview data, in which each data point is represented by different information from multiple sources of features. For example, in computer vision problems, images and videos are often described by different kinds of features, such as color, texture and edge. Web pages are also able to be represented in a multiview fashion, based on the text, hyperlinks and possibly existing visual information. In general, the multiview representation can capture seamlessly the rich information from multiple data cues as well as the complementary information among different cues, and thus be beneficial to various tasks, e.g., clustering, classification, de-noising. In this paper, we focus on advancing clustering by making use of the multiview representation.

To integrate different features into a unified framework, most existing multiview clustering methods employ graph based models. Early methods focus on the setting of "two-view" [14, 1, 6]: The most recent method in [14] constructs a bipartite graph to connect two types of features, and uses a standard spectral clustering algorithm to obtain the final clustering result. The method in [1] extends k-means to handle the data of two conditionally independent views. These early methods depend on the assumption that there are only two views, and thus it is hard to extend them to the cases of three or more views. The approach in [35] fuses the information from multiple graphs with Linked Matrix Factorization (LMF).To exploit complementary information, some methods employ co-regularization and co-training strategies, e.g., [4], [5], [23], [37] and [22]. The methods [2, 7] based on dimension reduction typically use Canonical Correlation Analysis (CCA) to project high dimensional multiview data onto a low-dimensional subspace. The method [39] recovers a shared low-rank transition probability matrix as a crucial input to the standard Markov chain method for clustering. The work in [10] learns a common representation under the spectral clustering framework by combining Laplacians of different views.

While effectual, existing methods may not fully explore the advantages of multiview representation. In fact, most previous methods capture only the pairwise correlations between different views, but essentially ignore the high order correlations underlying the multiple views. Moreover, the inference procedure of existing methods often leads to non-convex optimization problems, which in general cannot guarantee to produce globally optimal solutions. The method in [38] reaches a convex formulation, but, again, is limited to the case of two-view. The proposed method, as will be shown later, formulates the inference procedure as

*Corresponding Author

a convex optimization problem and seamlessly captures the high order cross information among multiple views.

We propose a novel multiview clustering method, termed Low-rank Tensor constrained Multiview Subspace Clustering (LT-MSC). Unlike most existing methods, which generally ignore the high order information of the multiview representation, our LT-MSC considers all the subspace representations of the individual views as a high order structure, i.e., a tensor, as shown in Fig. 1. Then the tensor is equipped with a low-rank constraint, which models elegantly the high order cross information among different views, reduces effectually the redundancy of the learned subspace representations, and improves the accuracy of clustering as well. Similar to the prevalent subspace clustering methods (e.g., [15, 28, 21]), LT-MSC also adopts a two-stage framework for clustering: It firstly learns an affinity matrix from the given data points, and then uses spectral clustering to produce the final clustering results. We formulate the inference process of the affinity matrix as a tensor nuclear norm minimization problem constrained with an additional $\ell_{2,1}$-norm regularizer. The minimization problem is convex and hence can be solved efficiently by an Augmented Lagrangian Alternating Direction Minimization (AL-ADM) [26] method.

The contributions of this work are summarized as follows: 1) We propose a novel method, termed LT-MSC, for clustering the data of multiview representation. By using a low-rank tensor to integrate together all the subspace representations of the individual views, LT-MSC well captures the high order information so as to perform distinctly better than previous methods in our extensive experiments. 2) The tensor in our method is utilized to capture the global structure of all the views and explore the correlations within and across multiple views, rather than keep the spatial information of images (e.g., [16, 20, 3]). 3) We provide an effectual extension for the recently established Low-Rank Representation (LRR) [28], which is a subspace clustering method for single view data.

2. Related Work

In recent years, a large number of multiview learning approaches have been proposed. Most of existing methods focus on supervised learning by means of labeled data. For the clustering task, the exisiting approaches could be roughly categorized into three lines. The first line exploits the multiview features with graph-based models. For example, the work in [14] constructs a bipartite graph to connect the two-view features, and uses a standard spectral clustering to obtain the 2-view clustering result. The approach in [35] fuses the information from multiple graphs with Linked Matrix Factorization, where each graph is approximated by matrix factorization with a graph-specific factor and a factor common to all graphs. The second category of methods often firstly learn a common space before

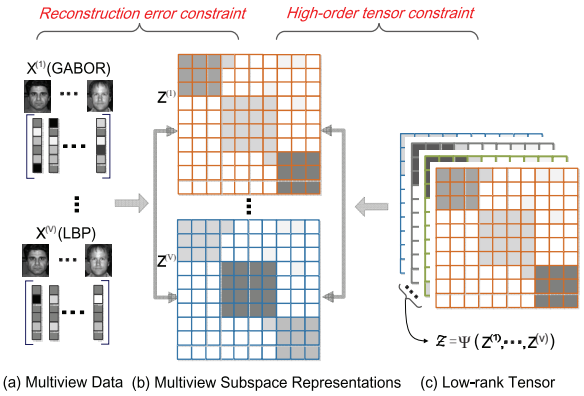


Figure 1. Overview of LT-MSC. Given a collection of data points with multiple views (a), $\mathbf{X}^{(1)} \cdots \mathbf{X}^{(V)}$, LT-MSC integrates all the subspace representations (b), $\mathbf{Z}^{(1)} \cdots \mathbf{Z}^{(V)}$, into a low-rank tensor (c), $\mathcal{Z}$, so as to involve the information of each individual views as well as the high order correlations among multiple views.

clustering. The approaches in [23], [37] co-regularize the clustering hypotheses to exploit the complementary information within the spectral clustering framework. In [22], a co-training based framework is proposed where it searches for the clusterings that agree across the different views. The third line of research is based on *Multiple Kernel Learning (MKL)*. As suggested in earlier work [11], even simply combining different kernels by adding them often leads to near optimal results as compared to more sophisticated approaches for classification. Instead of equally adding these kernels, views are expressed in terms of given kernel matrices and a weighted combination of these kernels is learned in parallel to the partitioning in [17]. Note that, the affinity matrices in these methods are independently constructed, while our LT-MSC constructs the affinity matrices jointly aiming to capture the high order correlations across different views.

Our work is closely related to the work of [15, 28, 21, 42] in that we also concentrate on subspace clustering, but differs in the data setting. In our case, the data are equipped with multiview features. The methods [15, 28, 21] have achieved the state-of-the-art performance. However, they only consider the single view feature, where the affinity matrix is constructed based on these reconstruction coefficients. The work in [18] formulates the subspace learning with multiple views as a joint optimization problem with a common subspace representation matrix and a group sparsity inducing norm. The work in [38] provides a convex reformulation of 2-view subspace learning. There are some methods based on dimensionality reduction, which usually learn a low-dimensional subspace from the multiple views and then apply any existing clustering method to get the result. The representative methods in this stream are proposed in [2, 7], which use *canonical correlation analysis* (CCA) to project the multiview high-dimensional data on-

to a low-dimensional subspace. Note that, unlike the self-representation based methods[15, 28, 21], all these multiview clustering methods pursue a low-dimensional subspace. In contrast, we extend the self-representation based subspace clustering to multiview setting by using the high order tensor constraint to explore the correlations across different views.

3. The Proposed Approach

Subspace clustering approaches cluster data into multiple subspaces. Specifically, in this paper, we consider the self-representation based subspace clustering methods, which construct affinity matrix via reconstruction coefficients. Suppose $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ is the matrix of data vectors, where each column is a D dimensional sample vector. To cluster the data into their respective subspaces, we firstly compute a subspace representation matrix $\mathbf{Z}$ following [15, 28, 21]:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} R(\mathbf{Z}) + \lambda L(\mathbf{X}, \mathbf{XZ}) \\ &\text{s.t. } \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \end{aligned} \tag{1}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N] \in \mathbb{R}^{N\times N}$ is the learned subspace representation matrix with each $\mathbf{z}_i$ being the learned subspace representation of sample $\mathbf{x}_i$, and $\mathbf{E}$ is the reconstruction error matrix. $L(\cdot,\cdot)$ denotes the loss function, $R(\cdot)$ is the regularizer and λ is the hyperparameter that controls the intensity of the loss penalty. After obtaining the self-representation matrix $\mathbf{Z}$, an affinity matrix (e.g., $(|\mathbf{Z}| + |\mathbf{Z}^T|)/2$ with $|\mathbf{Z}|$ being the matrix formed by taking the absolute values of the elements of $\mathbf{Z}$) is further obtained, and input into a spectral clustering algorithm [30] to produce the final clustering result.

In spite of the achieved promising performance, all these methods concentrate on single view data. A naive manner to extend Eq. (1) to handle multiview data is as follows:

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} \sum_{v=1}^{V} \left(R(\mathbf{Z}^{(v)}) + \lambda_v L(\mathbf{X}^{(v)}, \mathbf{X}^{(v)}\mathbf{Z}^{(v)})\right) \\ &\text{s.t. } \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)},\ v = 1, 2, ..., V, \end{aligned} \tag{2}$$

where $\mathbf{X}^{(v)}$, $\mathbf{Z}^{(v)}$ and $\mathbf{E}^{(v)}$ denote the data matrix, the subspace representation matrix and the reconstruction error matrix for the v^{th} view, respectively. λ_v is a hyperparameter which controls the intensity of loss penalty for v^{th} view. V is the number of views. However, this naive manner only considers each view independently, ignoring the correlations among different views. To address this issue, we propose to constrain these affinity matrices with tensor.

3.1. Formulation

In our efforts, we introduce a low-rank tensor constraint into the subspace clustering and propose a low-rank tensor constrained multiview subspace clustering method to learn the subspace representations of different views jointly and explore the intrinsic correlations across different views. The tensor is the generalization of the matrix concept. We give the definition of tensor nuclear norm as used in [29, 36], which generalizes the matrix (i.e., 2-mode or 2-order tensor) case (e.g., [28, 40, 41]) to higher-order tensor as

$$||\boldsymbol{\mathcal{Z}}||_* = \sum_{m=1}^{M} \xi_m ||\mathbf{Z}_{(m)}||_*, \tag{3}$$

where ξ_m's are constants satisfying $\xi_i > 0$ and $\sum_{m=1}^{M} \xi_m = 1$. $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1\times I_2\times ...\times I_M}$ is a M-order tensor, and $\mathbf{Z}_{(m)}$ is the matrix by unfolding the tensor $\boldsymbol{\mathcal{Z}}$ along the m^{th} mode defined as $\text{unfold}_m(\boldsymbol{\mathcal{Z}}) = \mathbf{Z}_{(m)} \in \mathbb{R}^{I_m\times(I_1\times ...\times I_{m-1}\times I_{m+1}...\times I_M)}$ [13, 12]. The nuclear norm $||\cdot||_*$ enforces the tensor under a low-rank constraint. In essence, the nuclear norm of a tensor is a convex combination of the nuclear norms of all matrices unfolded along each mode. With the low-rank tensor constraint, the objective function of our LT-MSC is formulated as

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} ||\boldsymbol{\mathcal{Z}}||_* + \lambda||\mathbf{E}||_{2,1}, \\ &\text{s.t. } \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)},\ v = 1, 2, ..., V, \\ &\boldsymbol{\mathcal{Z}} = \Psi(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, ..., \mathbf{Z}^{(V)}), \\ &\mathbf{E} = [\mathbf{E}^{(1)}; \mathbf{E}^{(2)}; ...; \mathbf{E}^{(V)}], \end{aligned} \tag{4}$$

where $\Psi(\cdot)$ constructs the tensor $\boldsymbol{\mathcal{Z}}$ by merging the different representations $\mathbf{Z}^{(v)}$ to a 3-order tensor with the dimensionality of $N \times N \times V$, as shown in Fig. 1(c). $\mathbf{E} = [\mathbf{E}^{(1)}; \mathbf{E}^{(2)}; ...; \mathbf{E}^{(V)}]$ is formed by vertically concatenating together along the column of errors corresponding to each view. $||.||_{2,1}$ is the $\ell_{2,1}$-norm, which encourages the columns of $\mathbf{E}$ to be zero. The underlying assumption here is that corruptions are sample-specific, i.e., some data points are corrupted and some are clean. This manner of integration will enforce the columns of $\mathbf{E}^{(1)}, \mathbf{E}^{(2)}, ..., \mathbf{E}^{(V)}$ to have jointly consistent magnitude values, and the effectiveness of which has been widely proved [8]. Note that, we perform normalization on the data matrices of different views to force the error of different views to be the same scale, which could reduce the variation in the magnitude of the error across the different views. Intuitively, the objective function in Eq. (4) seeks the lowest rank of the self-representation via collaborating multiple views jointly.

For our objective function Eq. (4), we substitute $||\boldsymbol{\mathcal{Z}}||_*$ by the definition in Eq. (3), accordingly, the optimization problem of Eq. (4) is transformed as:

$$\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} ||\mathbf{E}||_{2,1} + \sum_{m=1}^{M} \gamma_m ||\mathbf{Z}_{(m)}||_*, \tag{5}$$

where $\gamma_m = \frac{\xi_m}{\lambda} > 0$ encodes the strength of the low-rank tensor constraint, and the constraints are the same as in (4). Note that, $\mathbf{Z}^{(v)}$ is the subspace representation matrix corresponding to the v^{th} view, while $\mathbf{Z}_{(m)}$ is the m^{th} mode unfolding matrix of $\mathcal{Z}$.

3.2. Optimization Procedure

To the best of our knowledge, ALM is the fastest algorithm for solving the problem. To adopt alternating direction minimizing strategy to our problem (5), we need to make our objective function separable. Thus, we follow [36] to introduce M auxiliary variables $\mathbf{G}_m$'s, and convert it to a convex optimization problem as

$$\begin{aligned}
&\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)},\mathbf{G}_m} ||\mathbf{E}||_{2,1} + \sum_{m=1}^{M} \gamma_m ||\mathbf{G}_m||_*, \\
&\text{s.t. } \mathbf{P}_m \mathbf{z} = \mathbf{g}_m,\ m = 1, 2, ..., M, \\
&\mathcal{Z} = \Psi(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, ..., \mathbf{Z}^{(V)}), \\
&\mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)},\ v = 1, 2, ..., V, \\
&\mathbf{E} = [\mathbf{E}^{(1)}; \mathbf{E}^{(2)}; ...; \mathbf{E}^{(V)}],
\end{aligned} \tag{6}$$

where $\mathbf{z}$ is the vectorization of the tensor $\mathcal{Z}$, and $\mathbf{g}_m$ is the vectorization of the matrix $\mathbf{G}_m$. $\mathbf{P}_m$ is the alignment matrix corresponding to the mode-k unfolding which is a permutation matrix used to align the corresponding elements between $\mathbf{Z}_{(m)}$ and $\mathbf{G}_m$. The first constraint (the first two equalities) ensures the solution $\mathcal{Z}$ to be low-rank as $\mathbf{G}_m$s are enforced to be low-rank. The second constraint (the third equality) relates the data points of the same cluster, i.e., the same linear subspace, while the last constraint (the last equality) with $\ell_{2,1}$-norm gives the underlying assumption of error, i.e., sample-specific error. The optimization problem of Eq. (6) can be solved by the AL-ADM method [26], which minimizes the following augmented Lagrangian function:

$$\begin{aligned}
&\mathcal{L}_{\mu>0}(\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(V)}; \mathbf{E}^{(1)}, ..., \mathbf{E}^{(V)}; \mathbf{G}_1, ..., \mathbf{G}_M) = \\
&||\mathbf{E}||_{2,1} + \sum_{m=1}^{M} \big(\gamma_m ||\mathbf{G}_m||_* + \mu\Phi(\boldsymbol{\alpha}_m, \mathbf{P}_m\mathbf{z} - \mathbf{g}_m)\big) \\
&+ \sum_{v=1}^{V} \mu\Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}).
\end{aligned} \tag{7}$$

For simplicity, we give the definition $\Phi(\mathbf{Y}, \mathbf{C}) = \frac{1}{2}||\mathbf{C}||_F^2 + \langle \mathbf{Y}, \mathbf{C} \rangle$, where $\langle \cdot, \cdot \rangle$ represents matrix inner product and μ is a positive penalty scalar. The above problem is unconstrained. Therefore, it can be minimized with respect to the variables $\mathbf{E}^{(v)}$, $\mathbf{Z}^{(v)}$ and $\mathbf{G}_m$ by alternating minimization method, and then updating the Lagrange multipliers $\mathbf{Y}_v$ and $\boldsymbol{\alpha}_m$ accordingly. Fortunately, each subproblem has a simple closed-form solution, and hence can be computed efficiently. In this paper, the inexact ALM algorithm [26] with the alternating direction strategy is employed and outlined in Algorithm 1. The convergence properties of the algorithm could be proved similarly as those in [26]. For each iteration, we update each variable as follows:

1. $\mathbf{Z}^{(v)}$-subproblem: For updating the subspace representation $\mathbf{Z}^{(v)}$, we solve the following subproblem:

$$\begin{aligned}
\mathbf{Z}^{(v)*} &= \operatorname*{argmin}_{\mathbf{Z}^{(v)}} \sum_{m=1}^{M} \mu\Phi(\boldsymbol{\alpha}_m, \mathbf{P}_m\mathbf{z} - \mathbf{g}_m) \\
&\quad + \mu\Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}) \\
&= \operatorname*{argmin}_{\mathbf{Z}^{(v)}} \sum_{m=1}^{M} \mu\Phi(\Omega^v(\boldsymbol{\alpha}_m^T), \Omega^k(\mathbf{P}_m\mathbf{z} - \mathbf{g}_m)) \\
&\quad + \mu\Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}),
\end{aligned} \tag{8}$$

where $\Omega^v(\cdot)$ denotes the operation which selects the elements and reshapes them into a matrix corresponding to the v^{th} view. And the closed-form solution of $\mathbf{Z}^{(v)}$ is obtained by:

$$\begin{aligned}
\mathbf{Z}^{(v)*} &= (\sum_{m=1}^{M} \mathbf{B}_m^{(v)} - \sum_{m=1}^{M} \mathbf{A}_m^{(v)} + \mathbf{X}^{(v)^T}\mathbf{X}^{(v)} - \mathbf{X}^{(v)^T}\mathbf{E}^{(v)} \\
&\quad + \mathbf{X}^{(v)^T}\mathbf{Y}_v)(M\mathbf{I} + \mathbf{X}^{(v)^T}\mathbf{X}^{(v)})^{-1}
\end{aligned}$$

with

$$\mathbf{A}_m^{(v)} = \Omega^v(\boldsymbol{\alpha}_m) \text{ and } \mathbf{B}_m^{(v)} = \Omega^v(\mathbf{g}_m). \tag{9}$$

Specifically, there are M unfolding ways for a M-way tensor. For all the three unfolding modes in our model, the operator $\Omega^v(\cdot)$ only selects $N \times N$ elements corresponding to the v^{th} views and reshapes it to the $N \times N$ dimensional matrices $\mathbf{A}_m^{(v)}$ and $\mathbf{B}_m^{(v)}$ corresponding to $\mathbf{Z}^{(v)}$.

2. z-subproblem: With the updated $\mathbf{Z}^{(v)}$, we update $\mathbf{z}$ by directly replacing the corresponding elements as:

$$\mathbf{z}^* \leftarrow \mathbf{Z}^{(v)}. \tag{10}$$

Since $\mathbf{Y}_v$ and $\mathbf{E}^{(v)}$ are independent of other views in current iteration, i.e., the inner loop of Algorithm 1, updating $\mathbf{z}$ is also independent among multiple views. We can update $\mathbf{z}$ once when all $\mathbf{Z}^{(v)}$s of multiple views are obtained.

3. E-subproblem: The reconstruction error matrix $\mathbf{E}$ is optimized by:

$$\begin{aligned}
\mathbf{E}^* &= \operatorname*{argmin}_{\mathbf{E}} ||\mathbf{E}||_{2,1} \\
&\quad + \sum_{k=1}^{V} \mu\Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}) \\
&= \operatorname*{argmin}_{\mathbf{E}} \frac{1}{\mu}||\mathbf{E}||_{2,1} + \frac{1}{2}||\mathbf{E} - \mathbf{F}||_F^2,
\end{aligned} \tag{11}$$

where $\mathbf{F}$ is formed by vertically concatenating the matrices $\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{Y}^{(v)}$ together along column. This subproblem can be efficiently solved by Lemma 3.2 in [28].

4. $\mathbf{Y}_v$-subproblem: The multiplier $\mathbf{Y}_v$ is updated by:

$$\mathbf{Y}_v^* = \mathbf{Y}_v + (\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}). \quad (12)$$

Intuitively, the multiplier is updated proportionally to the violation of the equality constraint.

5. $\mathbf{G}_m$-subproblem: $\mathbf{G}_m$ is updated by:

$$\begin{aligned}\mathbf{G}_m^* &= \underset{\mathbf{G}_m}{\operatorname{argmin}} \gamma_m ||\mathbf{G}_m||_* + \mu\Phi(\boldsymbol{\alpha}_m, \mathbf{P}_m\mathbf{z} - \mathbf{g}_m) \\ &= \operatorname{prox}_{\beta_m}^{tr}(\Omega_m(\mathbf{P}_m\mathbf{z} + \boldsymbol{\alpha}_m)),\end{aligned} \quad (13)$$

where $\Omega_m(\mathbf{P}_m\mathbf{z} + \boldsymbol{\alpha}_m)$ reshapes the vector $\mathbf{P}_m\mathbf{z} + \boldsymbol{\alpha}_m$ to a matrix corresponding to the m^{th} mode unfolding. $\beta_m = \gamma_m/\mu$ denotes the thresholds of the spectral soft-threshold operation $\operatorname{prox}_{\beta_m}^{tr}(\mathbf{L}) = \mathbf{U}\max(\mathbf{S} - \beta_m, 0)\mathbf{V}^T$ with $\mathbf{L} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ being the Singular Value Decomposition (SVD) of the matrix $\mathbf{L}$, and the max operation being taken element-wise. Intuitively, the solution is truncated according to the subspace representation tensor $\boldsymbol{\mathcal{Z}}$.

6. $\mathbf{g}_m$-subproblem: Similarly to updating $\mathbf{z}$, we update $\mathbf{g}_m$ by directly replacing the corresponding elements:

$$\mathbf{g}_m^* \leftarrow \mathbf{G}_m. \quad (14)$$

7. $\boldsymbol{\alpha}_m$-subproblem: Similarly to updating $\mathbf{Y}_v$, the variable $\boldsymbol{\alpha}_m$ is updated by:

$$\boldsymbol{\alpha}_m^* = \boldsymbol{\alpha}_m + (\mathbf{P}_m\mathbf{z} - \mathbf{g}_m). \quad (15)$$

The accuracy of this estimate improves at every step. The major advantage of the ALM method is that, unlike the penalty method it is not necessary to take $\mu \to \infty$ in order to solve the original constrained problem. Instead, because of the presence of the Lagrange multiplier term, our algorithm converges fast since μ can stay much smaller. We only provide a general optimization solution for our problem in Eq. (6). In fact, any optimization algorithm can be employed to further improve our optimization scheme. Specifically, LADMPSAP [27] is a more efficient substitution for AL-ADM, which is more suitable for large scale problem. Furthermore, there are some fast methods [33, 34] for approximating matrix inversion computation and some methods [31, 19] for Singular Value Thresholding (SVT) to update $\mathbf{G}_m$, which can also be adopted for our problem.

Algorithm 1: Algorithm of LT-MSC

Input: Multiple types of feature matrices: $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, ..., $\mathbf{X}^{(V)}$, parameters γ_m's and the cluster number K

Initialize: $\mathbf{Z}^{(1)} ... = \mathbf{Z}^{(V)} = 0$;
$\mathbf{E}^{(1)} = 0, ..., \mathbf{E}^{(V)} = 0$; $\mathbf{Y}_1 = ... = \mathbf{Y}_V = 0$;
$\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2 = ... = \boldsymbol{\alpha}_M = 0$; $\mathbf{G}_1 = 0, ..., \mathbf{G}_M = 0$;
$\mu = 10^{-6}$; $\rho = 1.1$; $\varepsilon = 10^{-8}$; $\max_\mu = 10^{10}$
while *not converged* **do**
  for *each of V views* **do**
    Update $\mathbf{Z}^{(v)}$, $\mathbf{E}^{(v)}$ and $\mathbf{Y}_v$ according to Eq. (9), (11) and (12), respectively;
  end
  Update $\mathbf{z}$ according to Eq. (10);
  for *each of M modes* **do**
    Update $\mathbf{G}_m$, $\mathbf{g}_m$ and $\boldsymbol{\alpha}_m$ according to Eq. (13), (14) and (15), respectively;
  end
  Update the parameter μ by $\mu = \min(\rho\mu, \max_\mu)$;
  check the convergence conditions:
  $||\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}||_\infty < \epsilon$ and
  $||\mathbf{P}_m\mathbf{z} - \mathbf{g}_m||_\infty < \epsilon$;
end
Combine all subspace representations of each view by $\mathbf{S} = \frac{1}{V}\sum_{v=1}^{V} |\mathbf{Z}^{(v)}| + |\mathbf{Z}^{(v)^T}|$;
Apply the spectral clustering algorithm with the affinity matrix $\mathbf{S}$;
Output: Clustering result $\mathcal{C}$.

4. Experiments

The datasets used in our experiments are widely used in recent works [15, 28, 21] for face and image clustering. Fig. 2 are example images of these datasets. Specifically, we conduct our experiments using four benchmark datasets:

•**Yale** [1]**.** The Yale face dataset contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration.

•**Extended YaleB** [2]**.** The Extended YaleB dataset consists of 38 individuals and around 64 near frontal images under different illuminations for each individual. Similarly to the other work [28], we use the images for the first 10 classes, including 640 frontal face images .

•**ORL** [3]**.** There are 10 different images of each of 40 distinct subjects in the ORL face dataset. They took the images at different times, changing the lighting, facial expressions and facial details for some subjects.

•**COIL-20** [4]**.** The Columbia Object Image Library (COIL-20) dataset contains 1440 images of 20 object categories. Each category contains 72 images. All the images are normalized to 32×32 pixel arrays with 256 gray levels per pixel.

For all the datasets, we extract three types of features: intensity, LBP [32] and Gabor [24]. The standard LBP features are extracted with the the sampling density size of 8 and the blocking number of 7×8. Gabor wavelets are extracted with one scale $\lambda = 4$ at four orientations

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[2] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

[3] http://www.uk.research.att.com/facedatabase.html

[4] http://www.cs.columbia.edu/CAVE/software/softlib/

Table 1. Results (mean ± standard deviation) on *Yale*. We set $\gamma = 0.2$ in LT-MSC.

Method	*NMI*	*ACC*	*AR*	*F-score*	*Precision*	*Recall*
SPC_{best}	0.654±0.009	0.616±0.030	0.440±0.011	0.475±0.011	0.457±0.011	0.495±0.010
LRR_{best}	0.709±0.011	0.697±0.000	0.515±0.004	0.547±0.007	0.529±0.003	0.567±0.004
RTC	0.607±0.013	0.594±0.016	0.371±0.005	0.412±0.012	0.384±0.005	0.443±0.025
$FeatConcate_{PCA}$	0.665±0.037	0.578±0.038	0.396±0.011	0.434±0.011	0.419±0.012	0.450 ±0.009
PCA+LRR	0.632±0.006	0.582±0.038	0.353±0.009	0.396±0.008	0.360±0.007	0.441 ±0.008
co-Reg SPC	0.648±0.002	0.564±0.000	0.436±0.002	0.466±0.000	0.455±0.004	0.491±0.003
co-Train SPC	0.672±0.006	0.630±0.011	0.452±0.010	0.487±0.009	0.470±0.010	0.505±0.007
Min-Disagreement	0.645±0.005	0.615±0.043	0.433±0.006	0.470±0.006	0.446±0.005	0.496±0.006
ConvexReg SPC	0.673±0.023	0.611±0.035	0.466±0.032	0.501±0.030	0.476±0.032	0.532±0.029
RMSC	0.684±0.033	0.642±0.036	0.485±0.046	0.517±0.043	0.500±0.043	0.535±0.044
LT-MSC	**0.765±0.008**	**0.741±0.002**	**0.570±0.004**	**0.598±0.006**	**0.569±0.004**	**0.629±0.005**

Table 2. Results (mean ± standard deviation) on *Extended YaleB*. We set $\gamma = 100$ in LT-MSC.

Method	*NMI*	*ACC*	*AR*	*F-score*	*Precision*	*Recall*
SPC_{best}	0.360±0.016	0.366±0.059	0.225±0.018	0.303±0.011	0.296±0.010	0.310±0.012
LRR_{best}	0.625±0.004	0.615±0.013	0.451±0.002	0.508±0.004	0.481±0.002	**0.539**±0.001
RTC	0.373±0.001	0.360±0.000	0.215±0.005	0.291±0.003	0.287±0.005	0.294±0.002
$FeatConcate_{PCA}$	0.152±0.003	0.232±0.005	0.069±0.002	0.161±0.002	0.158±0.001	0.164±0.002
PCA+LRR	0.568±0.005	0.569±0.012	0.400±0.003	0.463±0.002	0.433±0.002	0.498±0.002
Co-Reg SPC	0.151±0.001	0.224±0.000	0.066±0.001	0.160±0.000	0.157±0.001	0.162±0.000
Co-Train SPC	0.302±0.007	0.186±0.001	0.043±0.001	0.140±0.001	0.137±0.001	0.143±0.002
Min-Disagreement	0.186±0.003	0.242±0.018	0.088±0.001	0.181±0.001	0.174±0.001	0.189±0.002
ConvexReg SPC	0.163±0.022	0.216±0.019	0.072±0.012	0.164±0.010	0.163±0.010	0.165±0.011
RMSC	0.157±0.019	0.210±0.013	0.060±0.014	0.155±0.012	0.151±0.012	0.159±0.013
LT-MSC	**0.637±0.003**	**0.626±0.010**	**0.459±0.03**	**0.521±0.006**	**0.485±0.001**	**0.539**±0.002

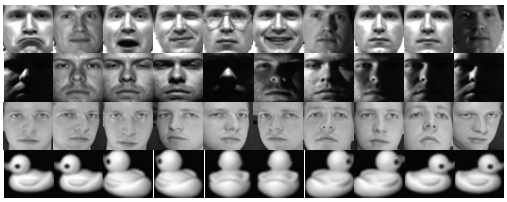


Figure 2. Example images of the four datasets used in this paper (the rows from top to bottom correspond to Yale, Extended YaleB, ORL and COIL-20, respectively).

$\theta = \{0^o, 45^o, 90^o, 135^o\}$. Accordingly, the dimensionalities of LBP and Gabor are 3304 and 6750, respectively.

Most existing clustering approaches perform standard spectral clustering algorithm after obtaining the affinity matrix, which is equipped with k-means. Therefore, we compare our approach with the 10 methods by running these methods 30 times and reporting the average performance and standard derivation. Specifically, the compared methods include 3 single view and 7 multiview ones:

• **SPC_{best}**. The method employs the most informative view with the standard spectral clustering algorithm [30].

•**LRR_{best}** [28]. Low-rank constraint and the best performed single view feature are used in the method LRR.

•**RTC** [3]. The method utilizes tensor to represent images and it is robust to the outliers.

•**$FeatConcate_{PCA}$**. The method concatenates all views and employ PCA to reduce the feature dimension to 300.

•**PCA+LRR**. The method concatenates all views and employ PCA to reduce the feature dimension to 300, on which LRR is applied.

•**Co-Reg SPC** [23]. The method co-regularizes the clustering hypotheses to enforce corresponding data points to have the same cluster membership.

•**Co-Training SPC** [22]. The method uses the co-training manner within the spectral clustering framework.

•**Min-Disagreement** [14]. The idea of "minimizing-disagreement" is realized based on a bipartite graph.

•**RMSC** [39]. The method recovers a shared low-rank transition probability matrix for clustering.

•**ConvexReg SPC**[10]. The method learns a common representation for all views.

Six evaluation metrics are used to evaluate the performances: Normalized Mutual Information (NMI), Accuracy (ACC), Adjusted Rand index (AR), F-score, Precision and Recall, which are widely used in clustering evaluation [9, 25]. For all these metrics, a higher value indicates better clustering quality. Each metric penalizes or favors different properties in the clustering, and hence we report results on these diverse measures to perform a comprehensive evaluation. These metrics have been widely used for evaluating clustering performance. For example, the compared methods, Co-Train SPC [22] and LRR [28], also use the same metrics for evaluating. Specifically, Co-Train SPC uses F-score, Precision, Recall, NMI, AR and LRR uses accuracy(ACC) for evaluating clustering task.

The inner product kernel is used for computing the graph

Table 3. Results (mean ± standard deviation) on *ORL*. We set $\gamma = 0.1$ in LT-MSC.

Method	*NMI*	*ACC*	*AR*	*F-score*	*Precision*	*Recall*
SPC_{best}	0.884±0.002	0.726±0.025	0.655±0.005	0.664±0.005	0.610±0.006	0.728±0.005
LRR_{best}	0.895±0.006	0.773±0.003	0.724±0.020	0.731±0.004	0.701±0.001	0.754±0.002
RTC	0.792±0.001	0.601±0.000	0.450±0.002	0.465±0.002	0.388±0.003	0.581±0.001
$FeatConcate_{PCA}$	0.835±0.004	0.675±0.028	0.564±0.010	0.574±0.010	0.532±0.011	0.624±0.008
PCA+LRR	0.867±0.003	0.750±0.033	0.650±0.007	0.658±0.007	0.624±0.007	0.696±0.008
Co-Reg SPC	0.853±0.003	0.715±0.000	0.602±0.004	0.615±0.000	0.567±0.004	0.666±0.004
Co-Train SPC	0.901±0.003	0.730±0.005	0.656±0.007	0.665±0.007	0.612±0.008	0.727±0.006
Min-Disagreement	0.876±0.002	0.748±0.051	0.654±0.004	0.663±0.004	0.615±0.004	0.718±0.003
ConvexReg SPC	0.883±0.013	0.734±0.031	0.668±0.032	0.676±0.035	0.628±0.041	0.731±0.030
RMSC	0.872±0.012	0.723±0.025	0.645±0.029	0.654±0.028	0.607±0.033	0.709±0.027
LT-MSC	**0.930±0.002**	**0.795±0.007**	**0.750 ±0.003**	**0.768±0.007**	**0.766±0.009**	**0.837±0.004**

Table 4. Results (mean ± standard deviation) on *COIL-20*. We set $\gamma = 1$ in LT-MSC.

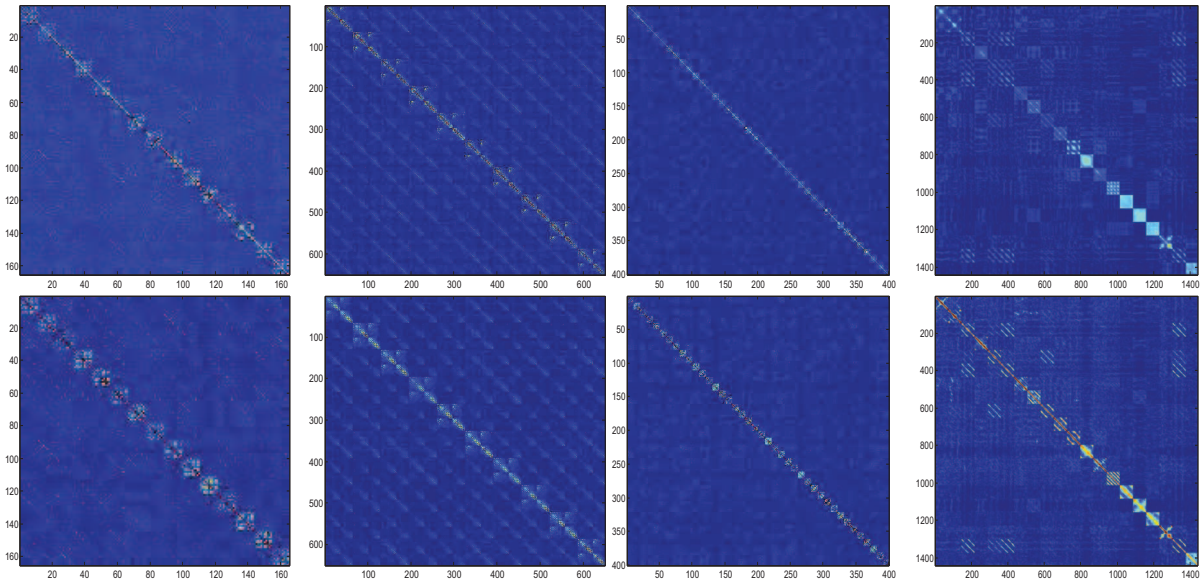
Method	*NMI*	*ACC*	*AR*	*F-score*	*Precision*	*Recall*
SPC_{best}	0.806±0.008	0.661±0.061	0.619±0.018	0.640±0.017	0.596±0.021	0.692±0.013
LRR_{best}	0.829±0.006	0.761±0.003	0.719±0.020	0.734±0.004	0.717±0.001	0.751±0.002
RTC	0.755±0.002	0.654±0.021	0.543±0.001	0.568±0.005	0.522±0.002	0.623±0.000
$FeatConcate_{PCA}$	0.810±0.005	0.701±0.044	0.635±0.010	0.654±0.009	0.614±0.013	0.702±0.008
PCA+LRR	0.832±0.004	0.770±0.031	0.718±0.007	0.732±0.011	0.725±0.004	0.739±0.011
Co-Reg SPC	0.765±0.001	0.560±0.000	0.568±0.003	0.593±0.000	0.558±0.003	0.627±0.002
Co-Train SPC	0.813±0.005	0.648±0.016	0.604±0.012	0.625±0.011	0.588±0.016	0.671±0.005
Min-Disagreement	0.789±0.002	0.661±0.052	0.597±0.005	0.619±0.005	0.579±0.007	0.666±0.003
ConvexReg SPC	0.815±0.023	0.693±0.049	0.647±0.055	0.666±0.051	0.622±0.071	0.720±0.033
RMSC	0.801±0.018	0.685±0.045	0.637±0.044	0.656±0.042	0.620±0.057	0.698±0.026
LT-MSC	**0.862±0.002**	**0.804±0.011**	**0.748±0.004**	**0.761±0.007**	**0.741±0.009**	**0.776±0.006**

similarity in all experiments if not stated otherwise. On all the four datasets, for the parameters of our method, we simply set the M parameters with equal value, i.e., $\gamma_1 = .. = \gamma_M = \gamma$, and accordingly tune the parameter γ. We run each task for 30 times and report the mean performance and standard deviation. For all the compared methods, we have tuned the parameters to the best.

We report the detailed clustering results on four benchmark datasets in Tables 1-4. On Yale, our approach excels all the baselines. The most competitive multiview clustering method, RMSC, has achieved a relatively promising results, however, given the best feature, the performance of LRR is even better. Our method further gains significant improvements around 5.6%, 3.6%, 5.5%, 5.1%, 4.0% and 6.2% over LRR in terms of NMI, ACC, AR, F-score, Precision and Recall, respectively. Besides, according to the results, directly concatenating features with PCA is not a promising manner since it does not always performs better than that of the best single view. Our method also outperforms two most recently published methods [39, 10], and the performance on ORL and COIL-20, as shown in Table 3-4, further demonstrates the effectiveness of our method.

Note that, the low performances of most comparisons are relatively low except the self-representation based methods (e.g., LRR) on Extended YaleB. The major reason is the large variation of illumination. Take the intensity feature for example, the subspace clustering methods are robust for the advantage of the self-representation, while traditional distance-based methods degrade sharply. As shown in Table 2, LRR performs best among the baselines (e.g., Min-Disagreement, Co-Reg SPC, and Co-Training SPC). However, our method further significantly outperforms *PCA+LRR* thanks to the high order low-rank tensor constraint. We also note that the improvement of LT-SMC over LRR on Extended YaleB is not such significant as on the other three datasets. The main reason is that the features LBP and Gabor are obviously worse than the best feature, i.e., intensity, which affects the performance of LT-MSC, as shown in Fig. 4. Fig. 4 shows the comparison between LRR with different single view feature and our LT-SMC with multiview features. LRR achieves promising performance given the best single view feature. However, as shown in Fig. 4, the performances of LRR with different features are of great difference on different datasets. For example, LBP performs as the best on ORL and COIL-20, while the performance degrades significantly on Extended YaleB. Hence, it is unreasonable to select the same feature for different datasets. In contrast, LT-MSC directly employs all types of features and achieves competitive results, while the performances of other multiview clustering methods degrade.

Fig. 3 compares the affinity matrices between LT-MSC and the naive manner with LRR, which constructs the affinity matrices of different views independently and then adds them. We plot these affinity matrices according to the intended clusters. The matrices corresponding to LT-MSC reveal the underlying clustering structures more clearly,



(a) Yale (b) Extended YaleB (c) ORL (d) COIL-20

Figure 3. Affinity matrices of using naive manner with LRR (top row) as in Eq. 2 and LT-SMC (bottom row).

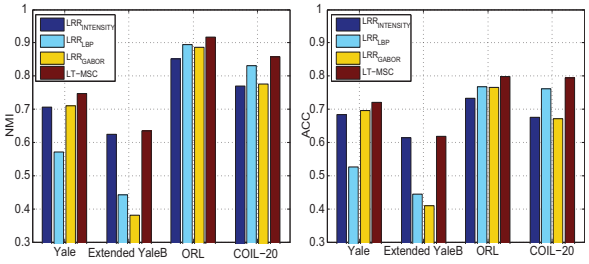


Figure 4. Comparison between LRR with the single view feature and our LT-SMC in terms of NMI and accuracy.

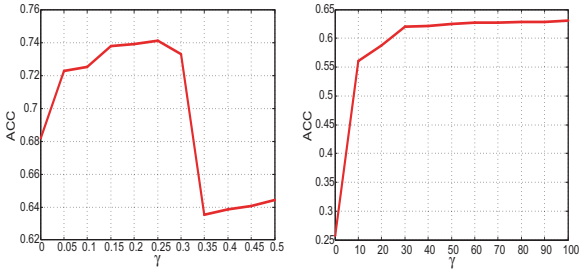


Figure 5. Parameter tuning in terms of accuracy on Yale (left) and Extended YaleB (right). Since we set $\gamma_1 = ... = \gamma_M = \gamma$ in our experiment, we only should tune the parameter γ.

which further validates the advantages of exploring the high order correlations among different views.

The parameter tuning examples are shown in Fig. 5. Note that, the value of γ is obviously larger on Extended YaleB than on Yale for promising performances. This mainly due to that the Extended YaleB dataset is more challenging for the large variation of illumination, hence a more strengthened regularization is essential to relieve the influence of illumination. For example, without regularization ($\gamma = 0$), the performance on Extended YaleB is relatively low, while it it not the case for Yale.

5. Conclusion

In this paper, we have shown how to discover the underlying structure of data via exploiting the complementary information of different views jointly. To this end, the tensor is employed to explore the high order correlations. We have formulated the problem in a unified optimization framework and proposed an efficient algorithm to find the optimal solution. The experimental results, compared to the state-of-the-art methods, have demonstrated the clear advantages of the proposed method on four benchmark datasets. Moreover, the proposed method is relatively robust for noisy views, which guarantees the high quality of clustering results. In the future, we will focus on incorporating the low-rank tensor decomposition into our method.

Acknowledgment

This work was supported by National High-tech R&D Program of China (2014BAK11B03), National Natural Science Foundation of China (No. 61422213, 61332012), National Basic Research Program of China (2013CB329305), 100 Talents Programme of The Chinese Academy of Sciences, and "Strategic Priority Research Program" of the Chinese Academy of Sciences (XDA06010701).

References

[1] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, 2004.

[2] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *CVPR*, 2008.

[3] X. Cao, X. Wei, Y. Han, Y. Yang, and D. Lin. Robust tensor clustering with non-greedy maximization. In *AAAI*, 2013.

[4] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang. Diversity-induced multi-view subspace clustering. In *CVPR*, 2015.

[5] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh. Constrained multi-view video face clustering. *TIP*, 24(11):4381 – 4393, 2015.

[6] K. Chaudhuri and S. M. Kakade. Multi-view clustering via canonical correlation analysis. In *ICML*, 2009.

[7] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, 2009.

[8] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, 2011.

[9] M. Christopher D., P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[10] M. D. Collins, J. Liu, J. Xu, L. Mukherjee, and V. Singh. Spectral clustering with a convex regularizer on millions of images. In *ECCV*, 2014.

[11] C. Cortes, M. Mohri, and A. Rostamizadeh. Learning non-linear combination of kernels. In *NIPS*, 2009.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r_1, r_2,..., r_n) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[14] V. R. de Sa. Spectral clustering with two views. In *ICML*, 2005.

[15] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI*, 35(11):2765–2781, 2013.

[16] Y. Fu, J. Gao, D. Tien, and Z. Lin. Tensor LRR based subspace clustering. In *IJCNN*, 2014.

[17] T. Grigorios and L. Aristidis. Kernel-based weighted multi-view clustering. In *ICDM*, 2012.

[18] Y. Guo. Convex subspace representation learning from multi-view data. In *AAAI*, 2013.

[19] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 52(2):217–288, 2011.

[20] X. He, D. Cai, H. Liu, and J. Han. Image clustering with tensor representation. In *ACMMM*, 2005.

[21] H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, 2014.

[22] A. Kumar and H. Daumé III. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.

[23] A. Kumar, P. Rai, and H. Daumé III. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.

[24] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Comput*, 42(3):300–311, 1993.

[25] H. Lawrence and A. Phipps. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[26] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1010.0789*, 2010.

[27] Z. Lin, R. Liu, and H. Li. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning*, 99(2):287–325, 2015.

[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *TPAMI*, 35(1):171–184, 2013.

[29] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *TPAMI*, 35(1):208–220, 2013.

[30] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[31] T.-H. Oh, Y. Matsushita, Y.-W. Tai, and I. S. Kweon. Fast randomized singular value thresholding for nuclear norm minimization. In *CVPR*, 2015.

[32] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[33] E. Quintana. A note on parallel matrix inversion. *SIAM Journal on Scientific Computing*, 22(5):1762–1771, 2001.

[34] F. Soleymani. A fast convergent iterative solver for approximate inverse of matrices. *Numerical Linear Algebra with Applications*, 21(3):439–452, 2013.

[35] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *ICDM*, 2009.

[36] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

[37] H. Wang, C. Weng, and J. Yuan. Multi-feature spectral clustering with minimax optimization. In *CVPR*, 2014.

[38] M. White, X. Zhang, D. Schuurmans, and Y.-l. Yu. Convex multi-view subspace learning. In *NIPS*, 2012.

[39] R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2014.

[40] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja. Low-rank sparse coding for image classification. In *ICCV*, 2013.

[41] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision*, 111(2):171–190, 2014.

[42] C. Zhou, C. Zhang, H. Fu, R. Wang, and X. Cao. Multi-cue augmented face clustering. In *ACM MM*, 2015.