2014 IEEE International Conference on Data Mining

Low-Rank Common Subspace for Multi-view Learning

Zhengming Ding[†], and Yun Fu[†‡]
[†]Department of Electrical and Computer Engineering, Northeastern University, Boston, USA
[‡]College of Computer and Information Science, Northeastern University, Boston, USA
allanding@ece.neu.edu, yunfu@ece.neu.edu

***Abstract*—Multi-view data is very popular in real-world applications, as different view-points and various types of sensors help to better represent data when fused across views or modalities. Samples from different views of the same class are less similar than those with the same view but different class. We consider a more general case that prior view information of testing data is inaccessible in multi-view learning. Traditional multi-view learning algorithms were designed to obtain multiple view-specific linear projections and would fail without this prior information available. That was because they assumed the probe and gallery views were known in advance, so the correct view-specific projections were to be applied in order to better learn low-dimensional features. To address this, we propose a Low-Rank Common Subspace (LRCS) for multi-view data analysis, which seeks a common low-rank linear projection to mitigate the semantic gap among different views. The low-rank common projection is able to capture compatible intrinsic information across different views and also well-align the within-class samples from different views. Furthermore, with a low-rank constraint on the view-specific projected data and that transformed by the common subspace, the within-class samples from multiple views would concentrate together. Different from the traditional supervised multi-view algorithms, our LRCS works in a weakly supervised way, where only the view information gets observed. Such a common projection can make our model more flexible when dealing with the problem of lacking prior view information of testing data. Two scenarios of experiments, robust subspace learning and transfer learning, are conducted to evaluate our algorithm. Experimental results on several multi-view datasets reveal that our proposed method outperforms state-of-the-art, even when compared with some supervised learning methods.**

***Keywords*-Multi-view; low-rank; common subspace;**

I. Introduction

A great deal of attention, as of recent, has been focused on multi-view learning [1], [2], [3], as multi-view data is of abundance in reality. When data is observed from different viewpoints, or captured by different sensors, the data from the same class could become multiple distinct samples, even heterogeneous. Let's take face recognition for example. A face image could have been taken from various viewpoints, resulting in multi-pose face images [3]. A face can be captured by different sensors, or even hand-sketched by artists. All of these variations lead to a heterogeneous set of face images [4]. Multi-view data brings up a challenging classification problem. Data from different views of the same class lie in quite different spaces, therefore cannot directly be compared. In this paper, we mainly consider multi-pose and multi-modality problems in multi-view learning. So all the views here represent images, either different poses or modalities.

Most of the previous multi-view learning work [1], [2], [3] set out to find a common space by learning multiple view-specific projections. The most typical approach is Canonical Correlation Analysis (CCA) [1], which learned two projections, one for each view, to respectively couple samples into a common space. Both projections were obtained by maximizing the cross correlation between two views. Multi-view CCA [5] extends this to multiple view cases. Following this, Multi-view Discriminant Analysis (MvDA) [2] had been proposed to seek a discriminant common space by jointly learning multiple view-specific linear projections for robust object recognition from multiple views, while introducing the Fisher criteria in parallel. Those works mainly solved the multi-view problem by using one view as the probe and another view as the gallery. That is, the view information of probe and gallery is known in advance so the specific projection would be applied to the correct view, which leaves a lot to ask for prior from the real-world applications. However, we do not have access to the view information of testing data ahead of time in most situations, because the evaluation data is only available at running time. For example, a probe face image could have come at any unknown viewpoint. Therefore, traditional multi-view algorithms would fail in such case. Besides, recent research [2], [3] also introduce the label information to better couple the multiple views of the same class.

When addressing new multi-view dataset without any prior information, e.g. label and view information, we need to borrow a well-learned source multi-view dataset. Transfer learning [6] shows promise in handling such problems with a good performance [7], [8], [9], [10], [11], [12], [13]. One solution to transfer learning is to discover a better feature representation for the data in one or two domains that mitigates the different marginal or conditional distribution between either domain. When solving multi-view learning problems, with no prior view information and limited labeled data, we find an auxiliary, well-established multi-view dataset to help the recognition task in the target dataset by transferring the well-learned multi-view knowledge.

Recently, low-rank representation (LRR) [14], [15] was introduced to transfer learning [16], [17] and robust subspace learning [18], which helps uncover the multiple structure of the data. Recent advances in low-rank learning has shown

1550-4786/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDM.2014.29

excellent performance by handling the noise in various applications. The block structural coefficient matrix, learned in low-rank methods, discovers the multiple subspace structure in source and target domain. As a result, data from source and target domains can be well aligned, so the knowledge in source domain can be transferred to that of the target. From this, LRR can handle multi-view data problem by aligning the samples with the same label but different views, and integrating the discrepancy of multiple views into the error part.

In this paper, we come up with a more general algorithm for multi-view analysis, named as Low-Rank Common Subspace (LRCS), which is designed to handle the situations that the probe and gallery data are both multi-view, but the view-information of probe data is inaccessible ahead. Along the lines of multi-view learning, we also adopt the view-specific projections for multiple views to align the data into a latent shared space. Considering that the multiple projections all uncover the information of the same class, but from different views, there should be lots of consistent information shared amongst them across different views. So we deploy a low-rank common subspace to capture the most compatible structure from the view-specific ones. Furthermore, with the low-rank constraint on the view-specific transformed data and the common projected data, samples from the same class but with different views would be coupled in the low-rank common subspace, by integrating the unique parts into the sparse error term. As far as we know, it is the first time to address the problem with no prior view information of the testing data available. We highlight our main contributions as follows:

- A low-rank common subspace is learned from the multiple view-specific projections in order to discover more shared information across views of the same class. With a low-rank constraint on the projected data, our method uncovers more intrinsic information with the common subspace by grouping the samples from different views with the same class label into a cluster.
- A weak-supervised approach is employed in our model, which means only view information is used in the training stage. Our model tackles the problem with prior view information of probe and gallery data being unknown. In this way, our algorithm is more flexible for dealing with realistic problems.
- Our proposed method is a more general algorithm, that can be easily extended to different scenarios (e.g. feature extraction, transfer learning), by just changing the input of the framework. Different scenarios of experiments were conducted to demonstrate the effectiveness of our algorithm. In many cases, our method is even more superior to existing supervised algorithms.

II. Related Work

In this section, we discuss the related work: low-rank representation, subspace learning and transfer learning, and highlight their differences to our work.

Low-rank representation has caught a lot of attention in recent years, since it has been widely applied in many areas, (e.g. machine learning, data mining and computer vision). A representative one is named as Robust PCA [19], which assumes that the data are lying in one single space. Real-world data, however, are usually coming from a set of multiple subspaces, e.g. multi-class images. To address this problem, low-rank representation (LRR) [14], [15] is presented to discover the global multiple subspace structures of data by detecting sparse noise. The objective function of LRR can be formulated as follows:

$$\begin{aligned} \min_{Z} \quad & \mathrm{rank}(Z) + \lambda \|E\|_{l_p}, \\ \text{s.t.} \quad & X = AZ + E, \end{aligned} \tag{1}$$

where X is the data matrix and A is the basis data matrix. $\mathrm{rank}(Z)$ is the rank of coefficients matrix Z, which can be solved by substituting as an equivalent convex optimization problem through nuclear norm $\|\cdot\|_*$. $\|\cdot\|_{l_p}$ are different kinds of sparse terms on the error part E (e.g. l_1, l_2-norms). When observed data is insufficient for recovering the underlying structure, however, LatLRR [20] was proposed by mining the unobserved data from the limited observed data.

Recently, Low-rank representation was integrated with subspace learning, in attempt to find a more robust subspace with low-rank constraint [18], [16], [17]. When the ratio of dimensionality to the sample size is extremely high, it becomes infeasible to learn from data directly due to the *curse of dimensionality* [21]. Subspace learning methods are able to preserve the intrinsic information while keeping a relatively low dimensionality for the data. Typical subspace learning methods are usually split into two categories, unsupervised and supervised fashion. For example, PCA [22] and LPP [23] are two well-known unsupervised subspace learning methods; LDA [24] and MFA [25] are two popular supervised methods. However, those conventional subspace learning methods are very sensitive to noise and outliers [18], therefore, those methods often result in a poor recognition performance when dealing real-world data.

To cope with the above problem, Low-rank Transfer Subspace Learning (LTSL) [16], Supervised Regularization based Robust Subspace (SRRS) [18], and Latent Low-rank Transfer Subspace Learning (L^2TSL) [17] are the most representative methods, which joint low-rank representation and subspace learning in a unified framework. These methods can leverage the advantages of both low-rank representation and subspace learning. In detail, SRRS incorporates class-label information as supervised regularization to improve the classification performance. SRRS deploys low-rank constraint in original feature space, where it considers more structure information could be kept. LTSL and L^2TSL are both developed for transfer learning by seeking a shared subspace for both source and target domains.

Transfer learning has proved with its appealing performance in many real-world scenarios, (e.g. image classification [26], [16], text categorization [27], collaborative recommendation [28] and sentiment analysis [29]). Transfer learning usually can be classified by "tasks" and"domains". An excellent survey with a more detailed discussion can be referred to [6].

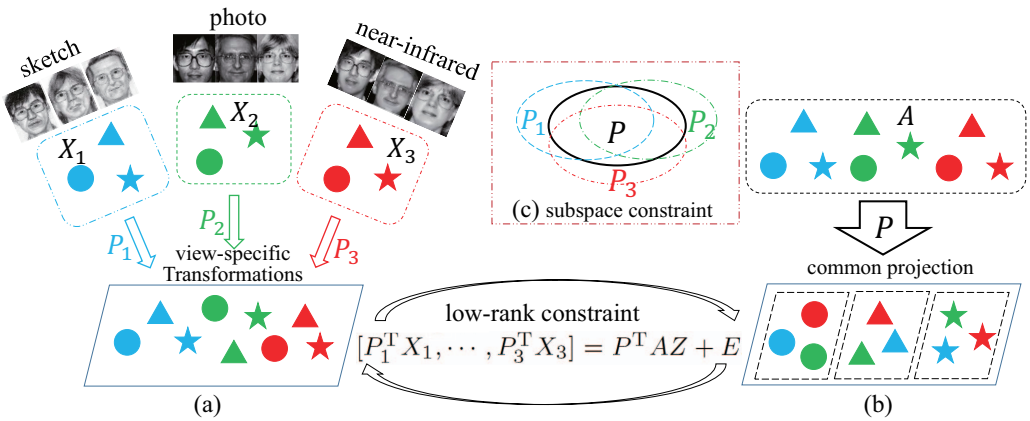


Fig. 1. General framework for our low-rank common subspace (LRCS). Here are three views (sketch, photo and near infrared), each color representing one different view, made up by three categories (different shapes represent different labels). (a) shows the view-specific transformations $\{P_1, P_2, P_3\}$ for three views $\{X_1, X_2, X_3\}$. (b) presents the common projection P for data A also with three views. Subspace constraint (c) indicates the relationship amongst the common P and those view-specific ones ($P_i = P + E_i$, where E_i is the sparse unique part split from the i-th view projection). The data projected from view-specific transformations intends to be reconstructed by that of common subspace in low-rank constraint $[P_1^{\mathrm{T}}X_1, \cdots, P_3^{\mathrm{T}}X_3] = P^{\mathrm{T}}AZ + E$, with an error term E, that uncovers the multiple subspace structures of the data, with each subspace correlated to one class. Therefore, our proposed LRCS can better preserve the identity information with the extra view information.

Along the line of transfer learning, LTSL and L^{2}TSL both employ low-rank constraints in the low-dimensional subspace in order to transfer knowledge from source domain to target one. As a result, data from source and target domains can be aligned well, so that the knowledge in source domain can be transferred to the target one. In our paper, we also adopt the low-rank constraint in the low-dimensional subspace. However, our method aims to find multiple view-specific projections and a common low-rank one, which is opposed to the existing ones by only learning one projection. With the view-specific projections in low-rank constraint, our method can uncover more extra structure of the data in a weakly supervised way. The extra structure has proved to be helpful in low-rank representation. Regardless, our algorithm is a more general method, which can also be applied to robust subspace learning [18] and transfer learning [16], [17].

III. The Proposed Algorithm

In this section, we brief on the motivation, then propose our low-rank common subspace (LRCS) before presenting the solution and complexity analysis. Finally, we compare LRCS with two of the most relevant algorithms.

A. Motivation

In general, the real-world data is multi-view, as observations are often made at different viewpoints, or even captured with different sensors. Therefore, multi-view data from the same class could definitely be different, leading to a big challenge in multi-view data analysis. Let's take cross-pose face recognition tasks [3] for example. The variance of different poses from one individual could be greater than that of the same pose from different subjects. Suppose we have data with k views, $X = [X_1, \cdots, X_k]$, and each view X_i contains the same c classes. Due to the high similarity among samples from one single view, we consider that the data from the same view lie in individual view-subspace. That is, the whole dataset contains k view-subspaces, which degrades its identity recognition performance, especially with a large within-class view-variance.

Most previous work [1], [2], [3] set out to seek a common subspace by introducing multiple view-specific linear projections, which aim to bridge the gap between the observed image space and the view-free representation space. Most of the recent ones [2], [3] are done in a supervised fashion by employing the Fisher discriminative framework to couple the different views from the same class. Meanwhile, the algorithms are designed to learn multiple view-specific projections, as in their setting, the view information of probe and gallery data is already known in the testing stage. In reality, however, we cannot obtain the prior view information ahead of time. Thus, it is difficult to extend the previous multi-view algorithms into the testing phase, since they only learn multiple view-specific projections. Moreover, samples of the same class are captured from different views, therefore, it is reasonable to assume that information is shared across all views (Fig. 1). Our goal is to learn a low-rank common subspace that captures most shared intrinsic information across different views, such that it best aligns data from the same subject across views, which ultimately leading to better performance in recognition tasks. More specifically, each view-specific projection is divided into two parts, a low-rank common part and a unique sparse part.

Low-rank representation (LRR) has demonstrated its ability to discover the data's global structure when data lie in multiple independent subspaces. Some recent works [18], [16], [17] were designed to learn robust subspace and low-rank representation in a unified framework. The point here is to achieve better results by leveraging the merits from both. Therefore, a low-rank constraint is introduced to couple the view-specific low-dimensional features with the common low-dimensional features. This low-rank constraint can be used to help the common low-rank projection find the whole structure of the dataset under the condition that extra prior view information is

available. To elaborate further, the view-specific transformed data from the same class intend to be reconstructed by the common projected data with the same class label (Fig. 1). The common projection is easily extendable to the testing stage, when the view information of testing data is unknown. Our method can be considered a weakly supervised approach, which only has access to the view information in advance at the training stage. Next, we will introduce our novel low-rank common subspace method, which can be applied to robust subspace learning [18] and transfer learning [16], [17].

B. Low-Rank Common Subspace Learning

Suppose the i-th view X_i corresponds to the view-specific projection P_i and each P_i is the same size. After the projection, the data from different views would lie in a common space, so each view can span from one another. As mentioned, each view has the same class so that they should share lots of similar information within-class across different views. We assume that a low-rank common projection P can preserve this shared information, which could make the same class from different views align into the common subspace. Specifically, each P_i consists of a shared low-rank P and their unique sparse information E_i (Fig. 1 (c)). Therefore, we expect the common part P to be low-rank and the error E to be sparse, so more of this shared information is recoverable. We define the objective function as follows:

$$\begin{aligned} &\min_{P,E_i,P_i} \operatorname{rank}(P) + \lambda_0 \sum_{i=1}^{k} \|E_i\|_1 \\ &\text{s.t.} \quad P_i = P + E_i, \quad i = 1, \cdots, k, \end{aligned} \tag{2}$$

where $\operatorname{rank}(P)$ is the rank of matrix P, and $\|\cdot\|_1$ is l_1-norm, which is simply the maximum absolute column sum of the matrix. λ_0 is the balanced parameter between the common low-rank part and sparse ones. It is hard to directly address the rank minimization problem in Eq. (2). However, we are fortunately to find a good surrogate, *nuclear norm*, for the rank minimization problem [14], [15]. Therefore, Eq. (2) becomes:

$$\begin{aligned} &\min_{P,E_i,P_i} \|P\|_* + \lambda_0 \sum_{i=1}^{k} \|E_i\|_1 \\ &\text{s.t.} \quad P_i = P + E_i, \quad i = 1, \cdots, k, \end{aligned} \tag{3}$$

where the nuclear norm $\|\cdot\|_*$ of a matrix can be calculated by the sum of singular values of the matrix. This common low-rank P can uncover most of the shared information amongst different view-specific transformations.

Besides, low-rank representation is well-known for discovering the structure information of the data [14], [15]. Also there are several methods [18], [16], [17] that learn the low-rank structure simultaneously a robust low-dimensional subspace. Previous works have demonstrated that the low-rank structures are best uncovered jointly learning a robust low-dimensional subspace. With this idea, we aim to apply low-rank constraint to couple the view-specific transformed data and the common subspace projected data (Fig. 1). In detail, the data in each subspaces projected by view-specific transformations can be well reconstructed by the data lying the common subspace with low-rank constraint. Furthermore, in real-world applications, data often include large amount of noise. Therefore, we introduce an error term to deal with noise data and get the objective function by integrating Eq. (3) and low-rank constraint together as:

$$\begin{aligned} &\min_{P,Z,E,E_i,P_i} \|Z\|_* + \|P\|_* + \lambda_0 \sum_{i=1}^{k} \|E_i\|_1 + \lambda_1 \|E\|_{2,1} \\ &\text{s.t.} \quad \widetilde{X} = P^{\mathrm{T}} A Z + E, \quad P^{\mathrm{T}} P = \mathrm{I}, \\ &\qquad P_i = P + E_i, \quad i = 1, \cdots, k. \end{aligned} \tag{4}$$

where $\widetilde{X} = [P_1^{\mathrm{T}} X_1, \cdots, P_k^{\mathrm{T}} X_k]$. $\|\cdot\|_{2,1}$ is the $L_{2,1}$ norm, defined as $\|E\|_{2,1} = \sum_{k=1}^{p} \sqrt{\sum_{j=1}^{n} ([E]_{kj})^2}$, which makes it sample specific, so the outliers can be detected. And λ_1 is the balanced parameter between the error part and the low-rank part. The orthogonal constraint $P^{\mathrm{T}} P = \mathrm{I}$ is imposed to ensure the obtained P is a basis transformation matrix. Matrix A represents the data with multiple views, and is defined according to different scenarios. For representation or subspace learning, A is generally defined as the dictionary, and always replaced by the data X itself. In our experiment, we use the data itself as A for simplicity. For transfer learning, A is the target domain, while X is the source domain. It is now easier to understand, i.e. the view-information of source domain is well-learned, while the target has sparse labeled data and amount of unlabeled data. And its view information is unknown. Therefore, this formula is able to transfer the view information from the source domain to the target domain. With that common projection, we can directly extract feature from the testing data no matter what view the data are.

Up to now, we have proposed a joint learning framework by seeking the common subspace from the view-specific transformations and low-rank representation from data, simultaneously. Next, we will introduce the solution to the objective function (4). Following the previous multiple projections learning methods, we first transform the objective function (4) into the following one:

$$\begin{aligned} &\min_{P_{\mathcal{T}},P_{\mathcal{S}},Z,E,E_{\mathcal{P}}} \|Z\|_* + \|P_{\mathcal{T}}\|_* + \lambda_0 \|E_{\mathcal{P}}\|_1 + \lambda_1 \|E\|_{2,1} \\ &\text{s.t.} \quad P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}} = P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}} Z + E, \\ &\qquad P_{\mathcal{S}} = P_{\mathcal{T}} + E_{\mathcal{P}}, \; P_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T}} = k \cdot \mathrm{I}. \end{aligned} \tag{5}$$

where

$$P_{\mathcal{S}} = \begin{bmatrix} P_1 \\ \vdots \\ P_k \end{bmatrix}, \quad X_{\mathcal{S}} = \begin{bmatrix} X_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & X_k \end{bmatrix},$$

$$P_{\mathcal{T}} = \begin{bmatrix} P \\ \vdots \\ P \end{bmatrix}, \quad X_{\mathcal{T}} = \frac{1}{k} \begin{bmatrix} A \\ \vdots \\ A \end{bmatrix}, \text{ and } E_{\mathcal{P}} = \begin{bmatrix} E_1 \\ \vdots \\ E_k \end{bmatrix}.$$

Therefore, we get $\widetilde{X} = P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}}$ and $P^{\mathrm{T}} A = P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}}$. It is clear that $\mathrm{rank}(P) = \mathrm{rank}(P_{\mathcal{T}})$ and $\sum_{i=1}^{k} \|E_i\|_1 = \|E_{\mathcal{P}}\|_1$, so the final objective function of (5) can achieve the same result with (4). For classification tasks, we split the learned $P_{\mathcal{T}}$ into k small matrixes, then average them to achieve the common low-rank P to the final feature extraction for both gallery and probe data in the testing stage.

Discussion: The more recent multi-view learning algorithms [2], [3] were designed to achieve multiple view-specific projections, that couple multi-view data into a common space. In this common space, view variations have less influence on label recognition. Those multiple view-specific projections are shared by the same class across different views, so lots of common information that is independent of view-variations are expected to be shared. In our model, we assume the multiple projections share a common low-rank part, which discovers the most highly correlated information from different view projection P_i. This is done by integrating the unique parts into the sparse term E_i. With the low-rank constraint between view-specific projected data and common projected data, the samples from the same class with different views would integrate into a common subspace, such that all the data from different categories intend to be clustered into their own subspace. Our algorithm works in a weakly supervised way, where only view information is accessible in the training stage, together with unacquainted label information. Experimental results show that our weak-supervised model outperforms the existing supervised ones. This indicates that our learned low-rank common projection possesses discriminative information to some extent in a weakly-supervised manner.

C. Solving Objective Function

In this section, we deploy Augmented Lagrangian Multiplier [30], [31] to solve the introduced above problem. First, we transform Eq. (5) into the following equivalent minimization problem by introducing two relaxation variables J and $Q_{\mathcal{T}}$,

$$\begin{aligned} \min_{P_{\mathcal{T}}, P_{\mathrm{S}}, Z, E, E_{\mathcal{P}}, J, Q_{\mathcal{T}}} & \|J\|_* + \|Q_{\mathcal{T}}\|_* + \lambda_1 \|E\|_{2,1} + \lambda_0 \|E_{\mathcal{P}}\|_1 \\ \text{s.t.} \quad & P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}} = P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}} Z + E, \quad P_{\mathcal{S}} = P_{\mathcal{T}} + E_{\mathcal{P}}, \\ & Q_{\mathcal{T}} = P_{\mathcal{T}}, \quad Z = J, \quad P_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T}} = k \cdot \mathrm{I}, \end{aligned}$$

whose augmented Lagrangian function is

$$\begin{aligned} & \|J\|_* + \|Q_{\mathcal{T}}\|_* + \lambda_1 \|E\|_{2,1} + \lambda_0 \|E_{\mathcal{P}}\|_1 + \\ & \langle Y_1^T, P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}} - P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}} Z - E \rangle + \langle Y_2, P_{\mathcal{S}} - P_{\mathcal{T}} - E_{\mathcal{P}} \rangle \\ & \langle Y_3, Z - J \rangle + \langle Y_4, P_{\mathcal{T}} - Q_{\mathcal{T}} \rangle + \frac{\mu}{2} (\|P_{\mathcal{S}} - P_{\mathcal{T}} - E_{\mathcal{P}}\|_{\mathrm{F}}^2 + \\ & \|Z - J\|_{\mathrm{F}}^2 + \|P_{\mathcal{T}} - Q_{\mathcal{T}}\|_{\mathrm{F}}^2 + \|P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}} - P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}} Z - E\|_{\mathrm{F}}^2), \end{aligned} \tag{6}$$

where Y_1, Y_2, Y_3, and Y_4 are four lagrange multipliers and $\mu > 0$ is the penalty parameter. $\|\cdot\|_{\mathrm{F}}^2$ is the matrix Frobenius norm. $\langle , \rangle$ is the inner product of two matrixes, that is, $\langle A, B \rangle = \mathrm{tr}(A^{\mathrm{T}} B)$. However, it is hard to optimize the variables in the Eq. (6) jointly, since a few variables are needed to be updated. Fortunately, we can obtain the optimization result through an iterative manner. We propose using Alternating Direction Method of Multipliers (ADMM) [32] to deal with this problem, as it converges well even some of them are not smooth.

Specifically, we alternately optimize the following variables $J, Z, E, P_{\mathcal{T}}$, $P_{\mathcal{S}}, E_{\mathcal{P}}$, and $Q_{\mathcal{T}}$ one by one. We define $J_t, Z_t, E_t, P_{\mathcal{T},t}, P_{\mathcal{S},t}, E_{\mathcal{P},t}, Q_{\mathcal{T},t}, Y_{1,t}, Y_{2,t}, Y_{3,t}, Y_{4,t}$, and μ_t as the variables updated in the t-th iteration. Then those variables are optimized in the t+1 iteration as follows:

for J:

$$J_{t+1} = \arg\min_{J} \frac{1}{\mu_t} \|J\|_* + \frac{1}{2} \|J - (Z_t + Y_{1,t}/\mu_t)\|_{\mathrm{F}}^2; \tag{7}$$

for Z:

$$Z_{t+1} = (X_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T},t} P_{\mathcal{T},t}^{\mathrm{T}} X_{\mathcal{T}} + \mathrm{I})^{-1} \mathcal{Z}_t, \tag{8}$$

where $\mathcal{Z}_t = X_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T},t} (P_{\mathcal{S},t}^{\mathrm{T}} X_{\mathcal{S}} - E_t) + J_t + (X_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T},t} Y_{1,t} - Y_{3,t})/\mu_t$;

for E:

$$\begin{aligned} E_{t+1} = \quad & \arg\min_{E} \frac{\lambda_1}{\mu} \|E\|_{2,1} \\ & + \frac{1}{2} \|E - P_{\mathcal{S},t}^{\mathrm{T}} X_{\mathcal{S}} - P_{\mathcal{T},t}^{\mathrm{T}} X_{\mathcal{T}} Z_t + Y_{1,t}/\mu_t\|_{\mathrm{F}}^2; \end{aligned} \tag{9}$$

for $P_{\mathcal{S}}$:

$$P_{\mathcal{S},t+1} = (X_{\mathcal{S}} X_{\mathcal{S}}^{\mathrm{T}} + \mathrm{I})^{-1} \mathcal{S}_t, \tag{10}$$

where $\mathcal{S}_t = X_{\mathcal{S}} (Z_t^{\mathrm{T}} X_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T},t} + E_t^{\mathrm{T}}) + P_{\mathcal{T},t} + E_{\mathcal{P},t} - (X_{\mathcal{S}} Y_{1,t}^T + Y_{2,t})/\mu_t$;

for $P_{\mathcal{T}}$:

$$P_{\mathcal{T},t+1} = (X_{\mathcal{T}} Z_t Z_t^{\mathrm{T}} X_{\mathcal{T}} + 2\mathrm{I})^{-1} \mathcal{T}_t, \tag{11}$$

where $\mathcal{T}_t = X_{\mathcal{T}} Z_t (X_{\mathcal{S}}^{\mathrm{T}} P_{\mathcal{S},t} - E_t^{\mathrm{T}}) + P_{\mathcal{S},t} - E_{\mathcal{P},t} + Q_{\mathcal{T},t} - Y_{4,t} + (X_{\mathcal{T}} Z Y_{1,t}^T + Y_{2,t})/\mu_t$;

for $E_{\mathcal{P}}$:

$$\begin{aligned} E_{\mathcal{P},t+1} = \quad & \arg\min_{E_{\mathcal{P}}} \frac{\lambda_0}{\mu} \|E_{\mathcal{P}}\|_1 \\ & + \frac{1}{2} \|E_{\mathcal{P}} - (P_{\mathcal{S},t} - P_{\mathcal{T},t} + Y_{4,t}/\mu_t)\|_{\mathrm{F}}^2; \end{aligned} \tag{12}$$

for $Q_{\mathcal{T}}$:

$$\begin{aligned} Q_{\mathcal{T},t+1} = \quad & \arg\min_{Q_{\mathcal{T}}} \frac{1}{\mu_t} \|Q_{\mathcal{T}}\|_* \\ & + \frac{1}{2} \|Q_{\mathcal{T}} - (P_{\mathcal{T},t} + Y_{3,t}/\mu_t)\|_{\mathrm{F}}^2. \end{aligned} \tag{13}$$

Eqs. (7)(13) are solvable using Singular Value Thresholding (SVT) [33]. Eqs. (9)(12) are solved by the shrinkage operator [34]. The detailed steps of the solution are presented in **Algorithm 1**. We set the parameters μ_0, ρ, ϵ, and $\max_\mu$ empirically, while tune the two balanced parameters λ_0 and λ_1 throughout the experiment, which is shown in Section IV. Also $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$ get initialized at random. We have evaluated on different initializations with traditional subspace learning methods to find that the final classification performance is very similar when the optimization converges.

Algorithm 1 Solving Problem by ADMM

Input: $X_{\mathcal{T}}, X_{\mathcal{S}}, \lambda_0, \lambda_1$

Initialize: $Z_0 = J_0 = 0, E_0 = 0, Y_{1,0} = 0, Y_{2,0} = 0, Y_{3,0} = 0,$ $Y_{4,0} = 0, \mu_0 = 10^{-4}, \rho = 1.3, \max_\mu = 10^8, \epsilon = 10^{-5}, t = 0.$

while not converged **do**

1. Update J_{t+1} using Eq. (7) by fixing other variables.
2. Update Z_{t+1} using Eq. (8) by fixing other variables.
3. Update E_{t+1} using Eq. (9) by fixing other variables.
4. Update $P_{\mathcal{S},t+1}$ using Eq. (10) by fixing other variables.
5. Update $P_{\mathcal{T},t+1}$ using Eq. (11) by fixing other variables.
6. Update $E_{\mathcal{P},t+1}$ using Eq. (12) by fixing other variables.
7. update $Q_{\mathcal{T},t+1}$ using Eq. (13) by fixing other variables.
8. Update the multipliers $Y_{1,t+1}, Y_{2,t+1}, Y_{3,t+1}, Y_{4,t+1}$ using
   $Y_{1,t+1} = Y_{1,t} + \mu_t(P^{\mathrm{T}}_{\mathcal{S},t+1}X_{\mathcal{S}} - P^{\mathrm{T}}_{\mathcal{T},t+1}X_{\mathcal{T}}Z_{t+1} - E_{t+1})$;
   $Y_{2,t+1} = Y_{2,t} + \mu_t(P_{\mathcal{S},t+1} - P_{\mathcal{T},t+1} - E_{\mathcal{P},t+1})$;
   $Y_{3,t+1} = Y_{3,t} + \mu_t(Z_{t+1} - J_{t+1})$.
   $Y_{4,t+1} = Y_{4,t} + \mu_t(P_{\mathcal{T},t+1} - Q_{\mathcal{T},t+1})$.
9. Update the penalty parameter μ_{t+1} using $\mu_{t+1} = \min(\rho\mu_t, \max_\mu)$
10. Check the convergence conditions
    $\|P^{\mathrm{T}}_{\mathcal{S},t+1}X_{\mathcal{S}} - P^{\mathrm{T}}_{\mathcal{T},t+1}X_{\mathcal{T}}Z_{t+1} - E_{t+1}\|_\infty < \epsilon,$
    $\|P_{\mathcal{S},t+1} - P_{\mathcal{T},t+1} - E_{\mathcal{P},t+1}\|_\infty < \epsilon,$
    $\|Z_{t+1} - J_{t+1}\|_\infty < \epsilon,\ \|P_{\mathcal{T},t+1} - Q_{\mathcal{T},t+1}\|_\infty < \epsilon.$
11. $t = t + 1$.

end while

output: $Z, J, E, E_{\mathcal{P}}, P_{\mathcal{T}}, P_{\mathcal{S}}, Q_{\mathcal{T}}$.

D. Complexity Analysis

For simplicity, we assume $X_{\mathcal{S}}$ and $X_{\mathcal{T}}$ are both $n_k \times m$ matrices, $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$ are $n_k \times p$ matrices. n_k is the k times of the original data dimension, p is the low dimension, and m is the size of dataset. The main time-consuming components of **Algorithm 1** are the following steps:

- Nuclear norm calculation in Step 1 and 7.
- Matrix multiplication and inverse in Step 2, 4, and 5.

We now discuss each part in detail. The general matrix multiplication takes $O(m^3)$, since there are α multiplications; therefore, the total cost is $\alpha O(m^3)$. The inverse of a $m \times m$ matrix costs $O(m^3)$. Therefore, Step 2, 4, and 5 will each cost $(\alpha+1)O(m^3)$. The SVD computation in Step 1 takes $O(m^3)$, which calculates the singular values of matrix $m \times m$. Step 7 costs $O(n_k p^2)$, which calculates the singular values of matrix $n_k \times p$. This indicates that the time cost will increase when the number of views is large.

E. Model Comparison

The most relevant works to ours are SRRS [18] and LTSL [16]. SRRS aims to learn a robust subspace simultaneous low-rank representation by introducing class information into their framework. The objective function of SRRS is

$$\begin{aligned} &\min_{P,Z,E} \|Z\|_* + \lambda_1\|E\|_{2,1} + \lambda_2 f(Z,P) \\ &\text{s.t. } X = AZ + E, \quad P^{\mathrm{T}}P = \mathrm{I}, \end{aligned} \tag{14}$$

where $f(Z,P)$ is the supervised regularization. The low-rank constraint is employed in the original dimensional space.

LTSL is designed to transfer well-learned source data into the target one in low-rank constraint. LTSL introduces general subspace learning methods into the low-rank constraint, which is defined as follows:

$$\begin{aligned} &\min_{P,Z,E} \|Z\|_* + \lambda_1\|E\|_{2,1} + \lambda_2 F(P, X_s) \\ &\text{s.t. } P^{\mathrm{T}}X_s = P^{\mathrm{T}}X_t Z + E, \end{aligned} \tag{15}$$

where X_s and X_t are the source and target domains, and $F(P, X_s)$ is the subspace term on source domain.

Comparing with these algorithms, we first replace P_i with $P + E_i$ to the low-rank constraint in Eq. (4) and achieve $P_i^{\mathrm{T}}X_i = (P + E_i)^{\mathrm{T}}X_i = P^{\mathrm{T}}X_i + E_i^{\mathrm{T}}X_i$. Therefore, the low-rank constraint is rewritten as

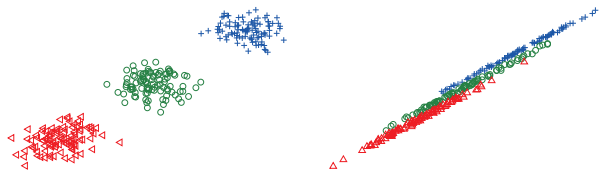
$$\begin{aligned} & [P_1^{\mathrm{T}}X_1, \cdots, P_k^{\mathrm{T}}X_k] \\ = \ & P^{\mathrm{T}}X + [E_1^{\mathrm{T}}X_1, \cdots, E_k^{\mathrm{T}}X_k] \\ = \ & P^{\mathrm{T}}AZ + E. \end{aligned}$$

The low-rank constraint now becomes $P^{\mathrm{T}}X = P^{\mathrm{T}}AZ + \widetilde{E}$, where $\widetilde{E}$ is the new error term that includes the outlier detect term E and unique sparse view-specific projected data $[E_1^{\mathrm{T}}X_1, \cdots, E_k^{\mathrm{T}}X_k]$. Therefore, our model degrades to the previous LTSL. Compared with SRRS, both our algorithm and LTSL adopt the low-rank constraint in the low-dimensional subspace; therefore, the learned subspaces can mitigate the discrepancy of the data from two different domains. Different from these previous two, the low-rank constraint of our algorithm has two low-rank terms P and Z. This property is able to preserve two kinds of low-rank structures, one from feature direction and the other from the data structure. There are already some works [20], [35] considering low-rank constraint on both directions, but they learn the two low-rank terms in the original high dimensional space. Another thing is our multi-view low-rank constraint could serve as if the extra information of the data is available. This property benefits by uncovering a more robust and better low-rank structure.

Our method is designed to tackle the new problem with prior view information of the testing data being inaccessible. This problem always happens, especially when we are going to recognize real-time coming data, or newly collected data from different domain. In this problem, traditional multi-view learning algorithms would fail, as they only seek multiple view-specific projections. When the view information of the probe data is unknown, the learned view-specific projections cannot work. This motivates us to learn a common projection across different views, which aims to capture most shared information amongst those multiple view-specific ones.

IV. Experiment

In this section, we first show the effectiveness of our algorithm on synthesis data via visualization. Then, we introduce the real datasets and experimental settings, and compare the existing algorithms in two different scenarios, feature representation and transfer learning.



(a) (b)

Fig. 2. Synthesized experiment. (a) The synthesized data of three views (different colors means different views); (b) The projected data with the learned common projection. Notice that the three views of projected data are aligned well. The figure is better viewed with color.

Fig. 3. Samples of different poses (C02, C05, C07, C14, C22, C25, C27, C29, C31, C34) from the same subject in CMU-PIE face database. This highlights the dissimilarity between different views of the same subject.

A. Synthesis Data

Here, we aim to synthesize samples of toy data to demonstrate that our learned common subspace can well align different views of data into a closer space. Three subsets of Gaussian distributed data is synthesized, which can be seen as three views of a single class. Each view has 50 data points. Three views are separated in the original two dimensional space, then we learn a two-dimensional common projection. The visualized result is shown in Fig. 2. We can observe that the common projection aligns the three views into a closer space. Therefore, we consider it captures most of shared information across views of the same subject.

B. Real Database Description

CMU-PIE Face database [36] contains 68 subjects in total. Samples of each subject are with 21 variations in lighting. In the experiments, we use these different poses (C02, C05, C07, C14, C22, C27, C29, C31, C34), which have large variances between the same subject at different poses (Fig. 3). We do several rounds of experiments by varying the numbers of poses (ranging from 2 to 5), to construct different cases. We crop images into size of 64×64 and use the raw feature.

BUAA VIS-NIR face database [37] includes 150 subjects, each having two modalities, visible images (VIS) and near infrared images (NIR). Each modality has 9 samples per subject. This database can be treated as two views, one view per modality. The size of the cropped image is 200×200. The raw feature is applied.

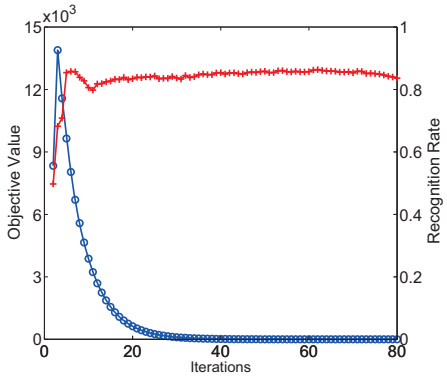


Fig. 4. Convergence curve (blue 'o') and recognition curve (red '+') of our algorithm for Case 2 (C02&C27) of CMU-PIE face database. The dimension was set to 100. We show the results with 80 iterations (two balanced parameters are set as $\lambda_0 = 0.1, \lambda_1 = 0.1$).

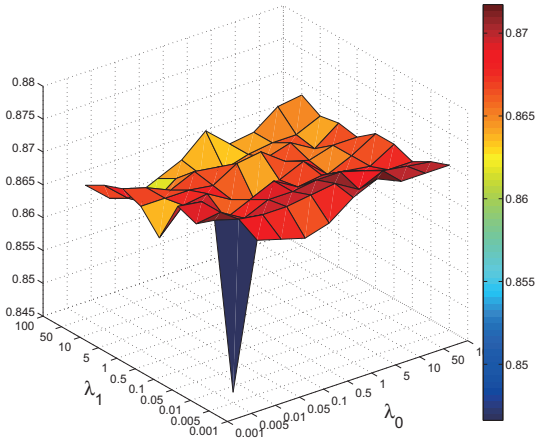


Fig. 5. Recognition rate of our algorithm with different parameter values $\{\lambda_0, \lambda_1\}$ on Case 2 (C02&C27) of CMU-PIE face database.

C. Feature Representation Scenario

Due to the different settings of previous multi-view learning methods, we only consider that the view information in testing data is unknown in the experiment, so the previous multi-view algorithms [2], [3] would fail. In this stage, we mainly compare feature extraction algorithms, listed as follows: PCA [22], LDA [24], LPP [23], Rotated Sparse Regression (RSR) [38], TFRR [35] and SRRS [18]. RSR aims to learn a sparse projection by pre-learning the LDA features. TFRR is the feature extraction version of FRR, which uses matrix factorization to replace the nuclear norm part. Among them, LDA, RSR, and SRRS are supervised algorithms; while PCA, LPP, and TFRR are unsupervised. In this scenario, we use the CMU-PIE face dataset. The nearest neighbor classifier is applied to calculate the recognition performance. We randomly select 10 samples per subject, each view as the training data, while the left samples as the testing data. We do five random selections, and then average the results. The experimental results on original images and the noisy images with 10% corruptions are listed in Table I and Table II. Moreover, we

TABLE I

AVERAGE RECOGNITION RESULTS (%) OF **ORIGINAL IMAGES** ON CMU-PIE FACE DATABASE, WHERE CASE 1: {C02, C14}, CASE 2: {C02, C27}, CASE 3: {C14, C27}, CASE 4: {C05, C07, C29}, CASE 5: {C05, C14, C29, C34}, CASE 6: {C02, C05, C14, C29, C31}

Methods	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
PCA[22]	69.03±0.08	69.21±0.08	68.52±0.12	52.65±0.04	34.94±0.08	29.09±0.01
LDA[24]	70.46±0.05	71.32±0.02	63.51±0.75	56.53±0.02	24.07±0.25	7.06±0.01
LPP[23]	57.25±0.06	58.83±0.07	59.25±0.56	43.56±0.08	19.67±0.05	13.11±0.01
RSR[38]	77.51±0.01	74.74±0.17	71.10±0.04	67.57±0.01	29.72±0.01	9.44±0.02
TFRR[35]	77.92±0.03	76.24±0.12	75.29±0.07	69.74±0.05	33.91±0.12	28.36±0.04
SRRS[18]	78.27±0.04	78.74±0.23	77.45±0.02	71.44±0.03	38.86±0.02	30.16±0.02
Ours	**87.78±0.02**	**86.67±0.01**	**87.38±0.99**	**74.84±0.04**	**44.48±0.03**	**36.17±0.01**

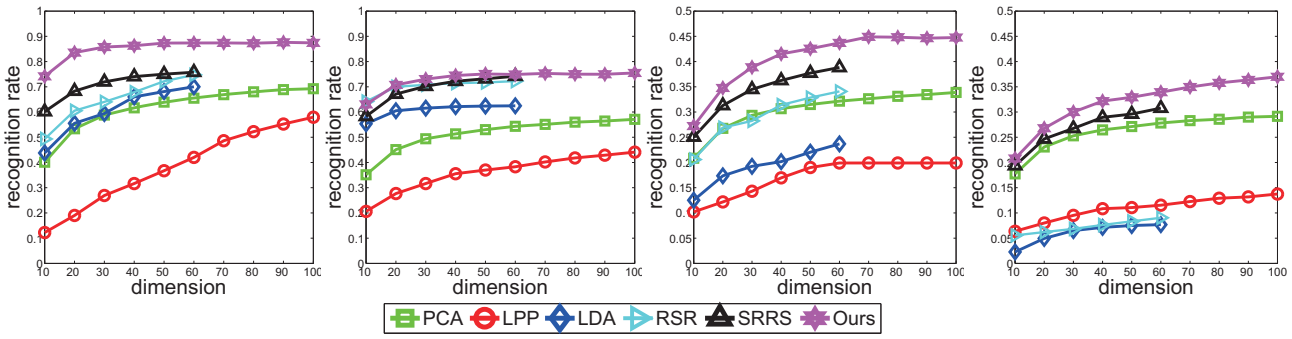


Fig. 6. Recognition rate over different dimensions for six algorithms on **original images** of CMU-PIE face database. The figures from left to right show the results of Case 2, Case 4, Case 5 and case 6. The dimensionality of LDA-based algorithms (LDA, RSR and SRRS) can only achieve at most 67. Here we only show 60 dimensions.

TABLE II

AVERAGE RECOGNITION RESULTS OF **CORRUPTED IMAGES** (%) ON CMU-PIE FACE DATABASE, WHERE CASE 1: {C02, C14}, CASE 2: {C02, C27}, CASE 3: {C14, C27}, CASE 4: {C05, C07, C29}, CASE 5: {C05, C14, C29, C34}, CASE 6: {C02, C05, C14, C29, C31}

Methods	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
PCA[22]	64.87±0.32	66.04±0.08	65.21±0.04	50.16±0.04	31.74±0.08	27.21±0.01
LDA[24]	26.71±0.20	23.19±0.35	20.34±0.75	46.72±0.02	6.67±0.25	4.06±0.01
LPP[23]	31.26±0.26	30.98±0.18	32.21±0.36	27.66±0.05	14.34±0.04	12.02±0.01
RSR[38]	37.02±0.03	34.34±0.15	31.69±0.09	52.45±0.01	10.02±0.01	4.95±0.02
TFRR[35]	68.10±0.07	68.24±0.32	67.85±0.12	50.94±0.09	29.26±0.12	28.12±0.03
SRRS[18]	72.27±0.05	72.74±0.18	71.45±0.08	54.32±0.03	32.34±0.02	29.03±0.02
Ours	**78.98±0.03**	**78.67±0.05**	**78.38±0.26**	**65.84±0.04**	**39.48±0.03**	**32.57±0.01**

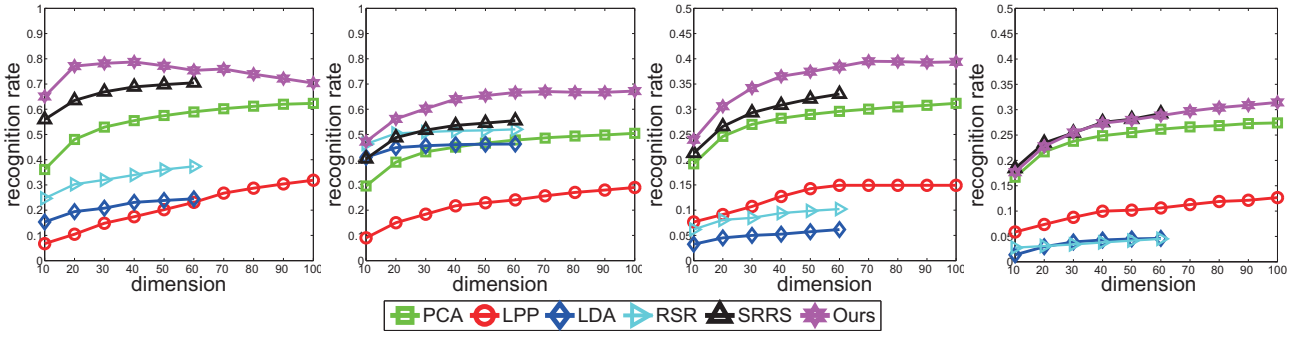


Fig. 7. Recognition rate over different dimensions for six algorithms on 10% **corrupted images** of CMU-PIE face database. The figures from left to right show the results of Case 2, Case 4, Case 5 and case 6. The dimensionality of LDA-based algorithms (LDA, RSR and SRRS) can only achieve at most 67. Here we only show 60 dimensions.

show the recognition results of these algorithms with different dimensions for a few cases in Fig. 6, 7. Furthermore, the convergence of our algorithm is shown in Fig. 4. The influence of the balanced parameters $\{\lambda_0, \lambda_1\}$ is also evaluated, shown in Fig. 5.

Discussion: From the results (Fig. 4), we can observe our method converges to a stable point quickly. Also the recognition performance reaches its highest very fast and stays relatively constant. Therefore, in the following experiment, we only do 50 iterations to achieve the optimized common projection P.

Table I and II show our method outperforming other algorithms, even the supervised ones, in any case. With the number of poses increasing, all the algorithms suffer from a lower recognition performance, but we observe that LDA and RSR degrade in a much faster way. This means that the supervised information is not always useful, especially for traditional subspace learning methods. For two-pose cases (Case 1-3), the algorithms achieve the similar performance for each case, meaning the similarity between any two poses is almost equivalent. These three cases effectively indicate the effectiveness of our algorithm. This means our common projection, learned from two view-specific projections of two poses with large variance, captures the most intrinsic information from the data. For Case 4, where three poses are relatively close (nearly frontal faces), our algorithm does not show the large margin of improvement, as in the previous cases. For Case 5 and 6, our method can still achieve better results with more poses, but the time cost also increases fast as we simultaneously learn multiple view-specific projections. Besides, low-rank based methods (TFRR, SRRS and ours) outperform the others, especially in corrupted cases, as they can uncover the global structure of the data by detecting the errors. Another observation is that the samples in each view have 21 different illuminations, some even invisible. This phenomenon also results in the similar performance of PCA in the original images and the corrupted case.

Fig. 5 shows the influence of different parameter values. The recognition performance is insensitive to the variation of parameters. In other cases, we also tune the two parameters mostly between 0.01 to 1.

D. Transfer Learning Scenario

In this section, we conduct two groups of experiments to measure our method for transfer learning. We compare with LTSL [16], GFK [39], TSL [40] in PCA subspace setting. Also the nearest neighbor classifier is applied to evaluate the recognition performance.

Group 1: we separate CMU-PIE into 2 subsets, each with 34 different subjects, to construct two domains, one is resolution (HR) and the other is low resolution (LR). For HR, we use the original 64×64 images. For LR, we first resize the 64×64 data (HR) into 16×16, then resize it into 64×64. We use the default method of the *imresize()* function in matlab. We apply the LR as the source domain while HR as the target domain, and choose different poses as the previous setting. For testing, we randomly select 2 samples per pose from the target domain as the reference, while other as the testing data. There is no overlap between reference and testing data. We do ten random selections, then average the results.

TABLE III
AVERAGE RECOGNITION RESULTS (%) ON **GROUP 1**, WHERE CASE 1: {C02, C14}, CASE 2: {C05, C07, C29}, CASE 3: {C05, C14, C29, C34}, CASE 4: {C02, C05, C14, C29, C31}

Methods	Case 1	Case 2	Case 3	Case 4
TSL[40]	49.43±0.09	29.28±0.05	19.34±0.08	16.76±0.14
GFK[39]	56.21±0.03	34.87±0.05	23.29±0.08	19.45±0.12
LTSL[16]	58.29±0.01	37.66±0.06	28.11±0.05	22.54±0.03
Ours	**61.23±0.12**	**49.09± 0.09**	**33.05±0.04**	**28.05±0.05**

TABLE IV
AVERAGE RECOGNITION RESULTS (%) OF **GROUP 2** ON BUAA NIR-VIS FACE DATABASE

Methods	TSL [40]	GFK [39]	LTSL [16]	Ours
Case 1	56.02±0.42	65.82±0.26	68.07±0.15	**70.56±0.20**
Case 2	57.32±0.24	67.56±0.30	69.17±0.19	**71.63±0.22**

Group 2: the BUAA NIR-VIS database is split into two parts, source domain and target domain. We do two cases by selecting: (1) 50 subjects as the source domain, the remaining 100 subjects as the target in Case 1; (2) 75 subjects as the source domain, the remaining 75 subjects as the target in Case 2. Each subject includes two modalities, that is, 9 VIS images and 9 NIR images. There is no ID overlap between the two domains. To make two domains different, we also apply low resolution procession as we did with Group 1 in the source domain. We randomly choose two samples, one VIS and the other NIR, per subject as the reference from the target domain. The remaining samples in the target domain is used as the testing data. We run 9 rounds of random selection, then average the recognition results.

Discussion: From the results (Table III, IV), we can see our algorithm performs better than others. With the multiple view-specific projections learning from the source, our method is capable of transfering such knowledge to the target domain, and aligning the different views properly with the common projection. Besides, low-rank based methods (LTSL and ours) achieve better results than the other two. In group 1, the superiority of our algorithm is obvious, but relatively not in group 2. We consider that the two modalities have a large discrepancy. So the common information seems not to be enough for aligning them well. For pose variances of a single subject, our method performs better and couples them with the common projection.

V. CONCLUSION

In this paper, we propose a novel Low-Rank Common Subspace (LRCS) method for multi-view data analysis. This is done by learning a common projection from multiple view-specific projections. With a low-rank constraint in the low-dimensional subspaces, our method obtains a discriminative

subspace by capturing the cross-view information of the same class in a weakly supervised way. Different from previous multi-view learning methods, our method works in the case that the view information of testing data is unknown. Furthermore, the proposed is a general low-rank subspace learning method, which can be applied to robust feature representation [18] and transfer learning [16]. Experiments on several databases demonstrate that our weakly supervised method achieves better results in multi-view learning field, especially when view variance is very large, resulting in the failure of supervised algorithms.

ACKNOWLEDGMENT

This research is supported in part by the NSF CNS award 1314484 and 1449266, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, U.S. Army Research Offi ce Young Investigator Award W911NF-14-1-0218, and Air Force Office of Scientific Research award FA9550-12-1-0201.

REFERENCES

[1] H. Hotelling, "Relations between two sets of variates," *Biometrika*, pp. 321–377, 1936.

[2] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," in *Proceedings of European Conference on Computer Vision*. Springer, 2012, pp. 808–821.

[3] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Regularized latent least square regression for cross pose face recognition," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 2013, pp. 1247–1253.

[4] S. Liu, D. Yi, Z. Lei, and S. Z. Li, "Heterogeneous face image matching using multi-scale features," in *Fifth IAPR International Conference on Biometrics (ICB),*. IEEE, 2012, pp. 79–84.

[5] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Conference on Data Mining and Data Warehouses*, 2010, pp. 1–4.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[7] I. Kuzborskij, F. Orabona, and B. Caputo, "From n to n+1: Multiclass transfer incremental learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *IEEE International Conference on Computer Vision*, 2013.

[9] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

[10] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars *et al.*, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, 2013.

[11] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[12] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *International Journal of Computer Vision*, pp. 1–18, 2014.

[13] N. Patricia and B. Caputo, "Learning to learn, from transfer learning to domain adaptation: A unifying perspective," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[14] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, 2010, pp. 663–670.

[15] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 171–184, 2013.

[16] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *International Journal of Computer Vision*, pp. 1–20, 2014.

[17] Z. Ding, M. Shao, and Y. Fu, "Latent low-rank transfer subspace learning for missing modality recognition," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[18] S. Li and Y. Fu, "Robust subspace discovery through supervised low-rank constraints," in *Proceedings of SIAM International Conference on Data Mining*, 2014, pp. 163–171.

[19] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[20] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *IEEE International Conference on Computer Vision*, 2011, pp. 1615–1622.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[22] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[23] X. He and P. Niyogi, "Locality preserving projections," in *Neural information processing systems*, vol. 16, 2004, p. 153.

[24] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[25] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[26] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2168–2175.

[27] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, "Transfer learning with graph co-regularization," in *the Twenty-Fifth AAAI Conference on Artificial*, 2012.

[28] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang, "Active transfer learning for cross-system recommendation," in *the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[29] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 751–760.

[30] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," *Optimization*, pp. 283–298, 1969.

[31] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[32] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[33] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[34] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 569–592, 2009.

[35] R. Liu, Z. Lin, F. De la Torre, and Z. Su, "Fixed-rank representation for unsupervised visual learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 598–605.

[36] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database of human faces," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-01-02, January 2001.

[37] H. Di, S. Jia, and W. Yunhong, "The buaa-visnir face database instructions," in *IRIP-TR-12-FR-001*, 2012.

[38] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3025–3032.

[39] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[40] S. Si, D. Tao, and B. Geng, "Bregman divergence -based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2010.