Journal of Machine Learning Research 16 (2015) 187-225 Submitted 9/13; Revised 6/14; Published 2/15

Learning Transformations for Clustering and Classification

Qiang Qiu QIANG.QIU@DUKE.EDU
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708, USA

Guillermo Sapiro GUILLERMO.SAPIRO@DUKE.EDU
Department of Electrical and Computer Engineering
Department of Computer Science
Department of Biomedical Engineering
Duke University
Durham, NC 27708, USA

Editor: Ben Recht

Abstract

A low-rank transformation learning framework for subspace clustering and classification is proposed here. Many high-dimensional data, such as face images and motion sequences, approximately lie in a union of low-dimensional subspaces. The corresponding subspace clustering problem has been extensively studied in the literature to partition such high-dimensional data into clusters corresponding to their underlying low-dimensional subspaces. Low-dimensional intrinsic structures are often violated for real-world observations, as they can be corrupted by errors or deviate from ideal models. We propose to address this by learning a linear transformation on subspaces using nuclear norm as the modeling and optimization criteria. The learned linear transformation restores a low-rank structure for data from the same subspace, and, at the same time, forces a maximally separated structure for data from different subspaces. In this way, we reduce variations within the subspaces, and increase separation between the subspaces for a more robust subspace clustering. This proposed learned robust subspace clustering framework significantly enhances the performance of existing subspace clustering methods. Basic theoretical results presented here help to further support the underlying framework. To exploit the low-rank structures of the transformed subspaces, we further introduce a fast subspace clustering technique, which efficiently combines robust PCA with sparse modeling. When class labels are present at the training stage, we show this low-rank transformation framework also significantly enhances classification performance. Extensive experiments using public data sets are presented, showing that the proposed approach significantly outperforms state-of-the-art methods for subspace clustering and classification. The learned low cost transform is also applicable to other classification frameworks.

Keywords: subspace clustering, classification, low-rank transformation, nuclear norm, feature learning

1. Introduction

High-dimensional data often have a small intrinsic dimension. For example, in the area of computer vision, face images of a subject (Basri and Jacobs, 2003; Wright et al., 2009),

handwritten images of a digit (Hastie and Simard, 1998), and trajectories of a moving object (Tomasi and Kanade, 1992) can all be well-approximated by a low-dimensional subspace of the high-dimensional ambient space. Thus, multiple class data often lie in a union of low-dimensional subspaces. The ubiquitous subspace clustering problem is to partition high-dimensional data into clusters corresponding to their underlying subspaces.

Standard clustering methods such as k-means in general are not applicable to subspace clustering. Various methods have been recently suggested for subspace clustering, such as Sparse Subspace Clustering (SSC) (Elhamifar and Vidal, 2013), and its extensions (Liu et al., 2010; Soltanolkotabi and Candes, 2012; Soltanolkotabi et al., 2013; Wang and Xu, 2013), Local Subspace Affinity (LSA) (Yan and Pollefeys, 2006), Local Best-fit Flats (LBF) (Zhang et al., 2012), Generalized Principal Component Analysis (Vidal et al., 2003), Agglomerative Lossy Compression (Ma et al., 2007), Locally Linear Manifold Clustering (Goh and Vidal, 2007), and Spectral Curvature Clustering (Chen and Lerman, 2009). A recent survey on subspace clustering can be found in Vidal (2011).

Low-dimensional intrinsic structures, which enable subspace clustering, are often violated for real-world data. For example, under the assumption of Lambertian reflectance, Basri and Jacobs (2003) show that face images of a subject obtained under a wide variety of lighting conditions can be accurately approximated with a 9-dimensional linear subspace. However, real-world face images are often captured under pose variations; in addition, faces are not perfectly Lambertian, and exhibit cast shadows and specularities (Candès et al., 2011). Therefore, it is critical for subspace clustering to handle corrupted underlying structures of realistic data, and as such, deviations from ideal subspaces.

When data from the same low-dimensional subspace are arranged as columns of a single matrix, the matrix should be approximately low-rank. Thus, a promising way to handle corrupted data for subspace clustering is to restore such low-rank structure. Recent efforts have been invested in seeking transformations such that the transformed data can be decomposed as the sum of a low-rank matrix component and a sparse error one (Peng et al., 2010; Shen and Wu, 2012; Zhang et al., 2011). Peng et al. (2010) and Zhang et al. (2011) are proposed for image alignment, Kuybeda et al. (2013) for the extension to multiple-classes with applications in cryo-tomograhy, and Shen and Wu (2012) is discussed in the context of salient object detection. All these methods build on recent theoretical and computational advances in rank minimization.

In this paper, we propose to improve subspace clustering and classification by learning a linear transformation on subspaces using matrix rank, via its nuclear norm convex surrogate, as the optimization criteria. The learned linear transformation recovers a low-rank structure for data from the same subspace, and, at the same time, forces a maximally separated structure for data from different subspaces (actually high nuclear norm, which as discussed later, improves the separation between the subspaces). In this way, we reduce variations within the subspaces, and increase separations between the subspaces for more accurate subspace clustering and classification.

For example, as shown in Figure 1, after faces are detected and aligned, e.g., using Zhu and Ramanan (2012), our approach learns linear transformations for face images to restore for the same subject a low-dimensional structure. By comparing the last row to the first row in Figure 1, we can easily notice that faces from the same subject across different poses

are more visually similar in the new transformed space, enabling better face clustering and classification across pose.

This paper makes the following main contributions:

- Subspace low-rank transformation (LRT) is introduced and analyzed in the context of subspace clustering and classification;
- A Learned Robust Subspace Clustering framework (LRSC) is proposed to enhance existing subspace clustering methods;
- A discriminative low-rank (nuclear norm) transformation approach is proposed to reduce the variation within the classes and increase separations between the classes for improved classification;
- We propose a specific fast subspace clustering technique, called Robust Sparse Subspace Clustering (R-SSC), by exploiting low-rank structures of the learned transformed subspaces;
- We discuss online learning of subspace low-rank transformation for big data;
- We demonstrate through extensive experiments that the proposed approach significantly outperforms state-of-the-art methods for subspace clustering and classification.

The proposed approach can be considered as a way of learning data features, with such features learned in order to reduce within-class rank (nuclear norm), increase between class separation, and encourage robust subspace clustering. As such, the framework and criteria introduced here can be incorporated into other data classification and clustering problems.

In Section 2, we formulate and analyze the low-rank transformation learning problem. In Sections 3 and 4, we discuss the low-rank transformation for subspace clustering and classification respectively. Experimental evaluations are given in Section 5 on public data sets commonly used for subspace clustering evaluation. Finally, Section 6 concludes the paper.

2. Learning Low-rank Transformations (LRT)

Let $\{\mathcal{S}_c\}_{c=1}^{C}$ be C m-dimensional subspaces of $\mathbb{R}^d$ (not all subspaces are necessarily of the same dimension, this is only assumed here to simplify notation). A data set is denoted as $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$, with each data point $\mathbf{y}_i$ in one of the C subspaces and arranged as a column of $\mathbf{Y}$. $\mathbf{Y}_c$ denotes the set of points in the c-th subspace $\mathcal{S}_c$, points arranged as columns of the matrix $\mathbf{Y}_c$.

As data points in $\mathbf{Y}_c$ lie in a low-dimensional subspace, the matrix $\mathbf{Y}_c$ is expected to be *low-rank*, and such low-rank structure is critical for accurate subspace clustering. However, as discussed above, this low-rank structure is often violated for real data.

Our proposed approach is to learn a global linear transformation on subspaces. Such linear transformation restores a low-rank structure for data from the same subspace, and, at the same time, encourages a maximally separated structure for data from different subspaces. In this way, we reduce the variation within the subspaces and introduce separations between the subspaces for more robust subspace clustering or classification.

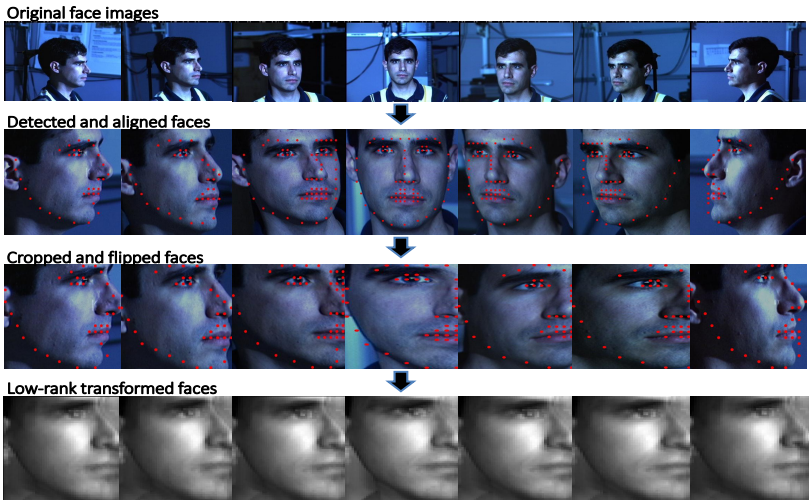


Figure 1: Learned low-rank transformation on faces across pose. In the second row, the input faces are first detected and aligned, e.g., using the method in Zhu and Ramanan (2012). Pose models defined in Zhu and Ramanan (2012) enable an optional crop-and-flip step to retain the more informative side of a face in the third row. Our proposed approach learns linear transformations for face images to restore for the same subject a low-dimensional structure as shown in the last row. By comparing the last row to the first row, we can easily notice that faces from the same subject across different poses are more visually similar in the new transformed space, enabling better face clustering or recognition across pose (note that the goal is clustering/recognition and not reconstruction).

2.1 Preliminary Pedagogical Formulation using Rank

We first assume the data cluster labels are known beforehand, and this assumption is removed when discussing the full clustering approach in Section 3. We adopt matrix rank as the key learning criterion (presented here first for pedagogical reasons, to be later replaced by the nuclear norm), and compute one global linear transformation on all subspaces as

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} rank(\mathbf{T}\mathbf{Y}_c) - rank(\mathbf{T}\mathbf{Y}), \quad \text{s.t.} ||\mathbf{T}||_2 = 1, \tag{1}$$

where $\mathbf{T} \in \mathbb{R}^{d \times d}$ is one global linear transformation on all data points (we will later discuss then $\mathbf{T}$'s dimension is less than d), $||\cdot||_2$ denotes the matrix induced 2-norm, and γ is a positive constant. Intuitively, minimizing the first *representation* term $\sum_{c=1}^{C} rank(\mathbf{T}\mathbf{Y}_c)$ encourages a consistent representation for the transformed data from the same subspace; and minimizing the second *discrimination* term $-rank(\mathbf{T}\mathbf{Y})$ encourages a diverse representation for transformed data from different subspaces (we will later formally discuss that the convex surrogate nuclear norm actually has this desired effect). The normalization condition $||\mathbf{T}||_2 = 1$ prevents the trivial solution $\mathbf{T} = 0$.

We now explain that the pedagogical formulation in (1) using rank is however not optimal to simultaneously reduce the variation within the same class subspaces and introduce separations between the different class subspaces, motivating the use of the nuclear norm not only for optimization reasons but for modeling ones as well. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices of the same dimensions (standing for two classes $\mathbf{Y}_1$ and $\mathbf{Y}_2$ respectively), and $[\mathbf{A}, \mathbf{B}]$ (standing for $\mathbf{Y}$) be the concatenation of $\mathbf{A}$ and $\mathbf{B}$, we have (Marsaglia and Styan, 1972)

$$rank([\mathbf{A}, \mathbf{B}]) \leq rank(\mathbf{A}) + rank(\mathbf{B}), \tag{2}$$

with equality if and only if $\mathbf{A}$ and $\mathbf{B}$ are disjoint, i.e., they intersect only at the origin (often the analysis of subspace clustering algorithms considers disjoint spaces, e.g., Elhamifar and Vidal (2013)).

It is easy to show that (2) can be extended for the concatenation of multiple matrices,

$$\begin{aligned} rank([\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \cdots, \mathbf{Y}_C]) &\leq rank(\mathbf{Y}_1) + rank([\mathbf{Y}_2, \mathbf{Y}_3, \cdots, \mathbf{Y}_C]) \\ &\leq rank(\mathbf{Y}_1) + rank(\mathbf{Y}_2) + rank([\mathbf{Y}_3, \cdots, \mathbf{Y}_C]) \\ &\cdots \\ &\leq \sum_{c=1}^{C} rank(\mathbf{Y}_c), \end{aligned} \tag{3}$$

with equality if matrices are independent. Thus, for (1), we have

$$\sum_{c=1}^{C} rank(\mathbf{T}\mathbf{Y}_c) - rank(\mathbf{T}\mathbf{Y}) \geq 0, \tag{4}$$

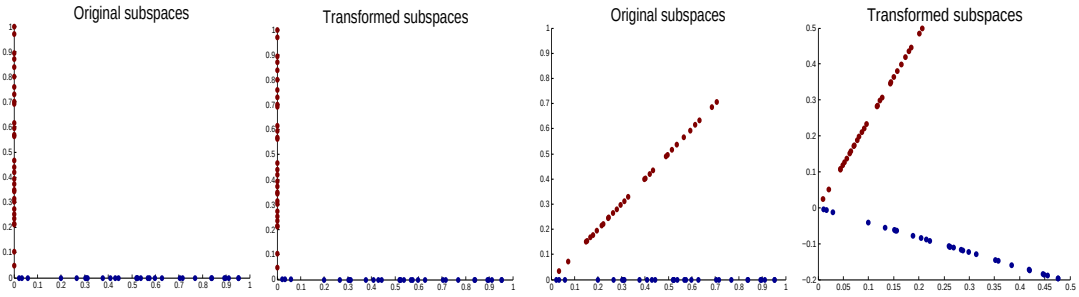
and the objective function (1) reaches the minimum 0 if matrices are independent after applying the learned transformation $\mathbf{T}$. However, independence does not infer maximal separation, an important goal for robust clustering and classification. For example, two lines intersecting only at the origin are independent regardless of the angle in between, and they are maximally separated only when the angle becomes $\frac{\pi}{2}$. With this intuition in mind, we now proceed to describe our proposed formulation based on the nuclear norm.

2.2 Problem Formulation using Nuclear Norm

Let $||\mathbf{A}||_*$ denote the nuclear norm of the matrix $\mathbf{A}$, i.e., the sum of the singular values of $\mathbf{A}$. The nuclear norm $||\mathbf{A}||_*$ is the convex envelop of $rank(\mathbf{A})$ over the unit ball of matrices Fazel (2002). As the nuclear norm can be optimized efficiently, it is often adopted as the best convex approximation of the rank function in the literature on rank optimization, e.g., Candès et al. (2011) and Recht et al. (2010).

One factor that fundamentally affects the performance of subspace clustering and classification algorithms is the distance between subspaces. An important notion to quantify the distance (separation) between two subspaces $\mathcal{S}_i$ and $\mathcal{S}_j$ is the smallest principal angle θ_{ij} (Miao and Ben-Israel, 1992; Elhamifar and Vidal, 2013), which is defined as

$$\theta_{ij} = \min_{\mathbf{u} \in \mathcal{S}_i, \mathbf{v} \in \mathcal{S}_j} \arccos \frac{\mathbf{u}'\mathbf{v}}{||\mathbf{u}||_2 ||\mathbf{v}||_2}, \tag{5}$$

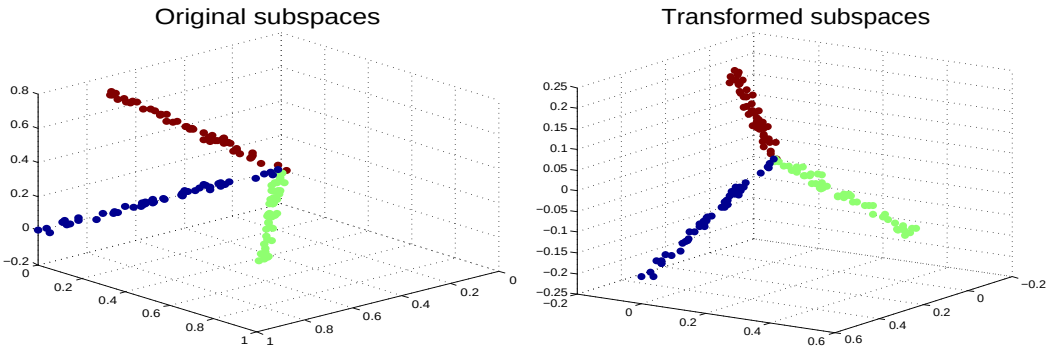


(a) $\theta_{AB} = \frac{\pi}{2} = 1.57$.

(b) $\mathbf{T} = \begin{bmatrix} 1.00 & 0 \\ 0 & 1.00 \end{bmatrix}$; $\theta_{AB} = 1.57$.

(c) $\theta_{AB} = \frac{\pi}{4} = 0.79$, $|\mathbf{A}|_* = 1, |\mathbf{B}|_* = 1$, $|[\mathbf{A}, \mathbf{B}]|_* = 1.41$

(d) $\mathbf{T} = \begin{bmatrix} 0.50 & -0.21 \\ -0.21 & 0.91 \end{bmatrix}$; $\theta_{AB} = 1.57$, $|\mathbf{A}|_* = 1, |\mathbf{B}|_* = 1$, $|[\mathbf{A}, \mathbf{B}]|_* = 1.95$



(e) $\begin{bmatrix} \theta_{AB} = 0.79, & \theta_{AC} = 0.79, & \theta_{BC} = 1.05 \\ \epsilon_A = 0.0141, & \epsilon_B = 0.0131, & \epsilon_C = 0.0148 \end{bmatrix}$, $|\mathbf{A}|_* = 4.06$, $|\mathbf{B}|_* = 4.08$, $|\mathbf{C}|_* = 4.16$.

(f) $\mathbf{T} = \begin{bmatrix} 0.39 & -0.16 & -0.16 \\ -0.13 & 0.90 & 0.11 \\ -0.23 & 0.11 & 0.57 \end{bmatrix}$; $\begin{bmatrix} \theta_{AB} = 1.51, & \theta_{AC} = 1.49, & \theta_{BC} = 1.57 \\ \epsilon_A = 0.0091, & \epsilon_B = 0.0085, & \epsilon_C = 0.0114 \end{bmatrix}$, $|\mathbf{A}|_* = 1.93$, $|\mathbf{B}|_* = 2.37$, $|\mathbf{C}|_* = 1.20$.

Figure 2: The learned transformation $\mathbf{T}$ using (6) with the nuclear norm as the key criterion. Three subspaces in $\mathbb{R}^3$ are denoted as $\mathbf{A}$(red), $\mathbf{B}$(blue), $\mathbf{C}$(green). We denote the angle between subspaces $\mathbf{A}$ and $\mathbf{B}$ as θ_{AB} (and analogous for the other pairs of subspaces). Using (6), we transform $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ in (a),(c),(e) to (b),(d),(f) respectively (in the first row the subspace C is empty, being this basically a two dimensional example). Data points in (e) are associated with random noises $\sim \mathcal{N}(0, 0.01)$. We denote the root mean square deviation of points in $\mathbf{A}$ from the true subspace as ϵ_A (and analogous for the other subspaces). We observe that the learned transformation $\mathbf{T}$ maximizes the distance between every pair of subspaces towards $\frac{\pi}{2}$, and reduces the deviation of points from the true subspace when noise is present, note how the individual subspaces nuclear norm is significantly reduced. Note that, in (c) and (d), we have the same rank values $rank(\mathbf{A}) = 1, rank(\mathbf{B}) = 1, rank([\mathbf{A}, \mathbf{B}]) = 2$, but different nuclear norm values, manifesting the improved between-subspaces separation.

Note that $\theta_{ij} \in [0, \frac{\pi}{2}]$. We replace the rank function in (1) with the nuclear norm,

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_*, \quad \text{s.t.} ||\mathbf{T}||_2 = 1. \tag{6}$$

The normalization condition $||\mathbf{T}||_2 = 1$ prevents the trivial solution $\mathbf{T} = 0$. However, understanding the effects of adopting a different normalization norm here is interesting and is the subject of future research.

It is important to note that (6) is not simply a relaxation of (1). Not only the replacement of the rank by the nuclear norm is critical for optimization considerations in reducing the variation within same class subspaces, but as we show next, the learned transformation $\mathbf{T}$ using the objective function (6) also maximizes the separation between different class subspaces (a missing property in (1)), leading to improved clustering and classification performance.

We start by presenting some basic norm relationships for matrices and their corresponding concatenations.

Theorem 1 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_* \leq ||\mathbf{A}||_* + ||\mathbf{B}||_*.$$

Proof: See Appendix A. ■

Theorem 2 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_* = ||\mathbf{A}||_* + ||\mathbf{B}||_*.$$

when the column spaces of $\mathbf{A}$ *and* $\mathbf{B}$ *are orthogonal.*

Proof: See Appendix B. ■

It is easy to see that theorems 1 and 2 can be extended for the concatenation of multiple matrices. Thus, for (6), we have,

$$\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_* \geq 0. \tag{7}$$

Based on (7) and Theorem 2, the proposed objective function (6) reaches the minimum 0 if the column spaces of every pair of matrices are orthogonal after applying the learned transformation $\mathbf{T}$; or equivalently, (6) reaches the minimum 0 when the separation between every pair of subspaces is maximized after transformation, i.e., the smallest principal angle between subspaces equals $\frac{\pi}{2}$. Note that such improved separation is not obtained if the rank is used in the second term in (6), thereby further justifying the use of the nuclear norm instead.

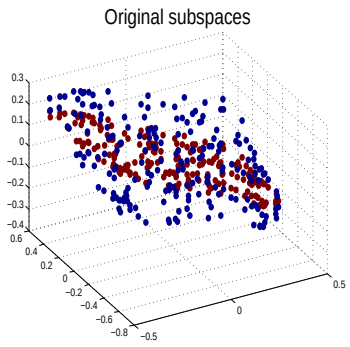
We have then, both intuitively and theoretically, justified the selection of the criteria (6) for learning the transform $\mathbf{T}$. We now illustrate the properties of the learned transformation $\mathbf{T}$ using synthetic examples in Figure 2 (real examples are presented in Section 5). Here we adopt a projected subgradient method described in Appendix C (though other modern nuclear norm optimization techniques could be considered, including recent real-time formulations Sprechmann et al. (2012)) to search for the transformation matrix T that minimizes (6). As shown in Figure 2, the learned transformation $\mathbf{T}$ via (6) maximizes the separation between every pair of subspaces towards $\frac{\pi}{2}$, and reduces the deviation of the data points to the true subspace when noise is present. Note that, comparing Figure 2c to Figure2d, the learned transformation using (6) maximizes the angle between subspaces, and the nuclear norm changes from $|[\mathbf{A},\mathbf{B}]|_* = 1.41$ to $|[\mathbf{A},\mathbf{B}]|_* = 1.95$ to make $|\mathbf{A}|_* + |\mathbf{B}|_* - |[\mathbf{A},\mathbf{B}]|_* \approx 0$; However, in both cases, where subspaces are independent, $rank([\mathbf{A},\mathbf{B}]) = 2$, and $rank(\mathbf{A}) + rank(\mathbf{B}) - rank([\mathbf{A},\mathbf{B}]) = 0$.

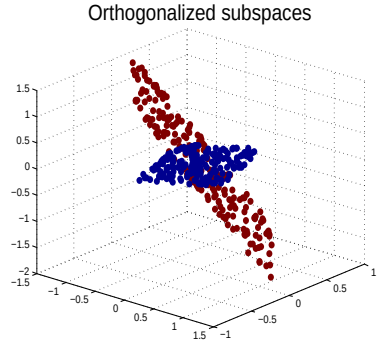
2.3 Comparisons with other Transformations

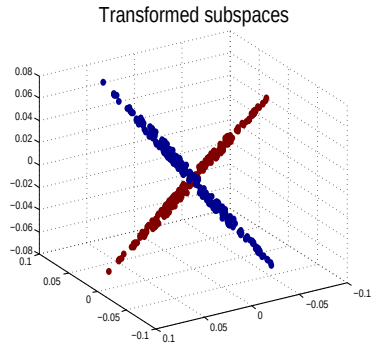
For independent subspaces, a transformation that renders them pairwise orthogonal can be obtained in a closed-form as follows: we take a basis $\mathbf{U}_c$ for the column space of $\mathbf{Y}_c$ for each subspace, form a matrix $\mathbf{U} = [\mathbf{U}_1, ..., \mathbf{U}_C]$, and then obtain the orthogonalizing transformation as $\mathbf{T} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'$. To further elaborate the properties of our learned transformation, using synthetic examples, we compare with the closed-form orthogonalizing transformation in Figure 3 and with linear discriminant analysis (LDA) in Figure 4.

Two intersecting planes are shown in Figure 3a. Though subspaces here are neither independent nor disjoint, the closed-form orthogonalizing transformation still significantly increases the angle between the two planes towards $\frac{\pi}{2}$ in Figure 3b (note that the angle for the common line here is always 0). Note also that the closed-form orthogonalizing transformation is of size $r \times d$, where r is the sum of the dimension of each subspace, and we plot just the first 3 dimensions for visualization. Comparing to the orthogonalizing transformation, our leaned transformation in Figure 3c introduces similar subspace separation, but enables significantly reduced within subspace variations, indicated by the decreased nuclear norm values (close to 1). The same set of experiments with different samples per subspace are shown in the second row of Figure 3. Our formulation in (6) not only maximizes the separations between the different classes subspaces, but also simultaneously reduces the variations within the same class subspaces.

Our learned transformation shares a similar methodology with LDA, i.e., minimizing intra-class variation and maximizing inter-class separation. Two classes $\mathbf{Y}_+$ and $\mathbf{Y}_-$ are shown in Figure 4a, each class consisting of two lines. Our learned transformation in Figure 4c shows smaller intra-class variation than LDA in Figure 4b by merging two lines in each class, and simultaneously maximizes the angle between two classes towards $\frac{\pi}{2}$ (such two-class clustering and classification is critical for example for trees-based techniques Qiu and Sapiro (2014)). Note that we usually use LDA to reduce the data dimension to the number of classes minus 1; however, to better emphasize the distinction, we learn a $(d-1) \times d$ sized transformation matrix using both methods. The closed-form orthogonalizing transformation discussed above also gives higher intra-class variations as $|\mathbf{Y}_+|_* = 1.45$ and $|\mathbf{Y}_+|_* = 1.68$. Figure 4d shows an example of two non-linearly separable classes, i.e., two



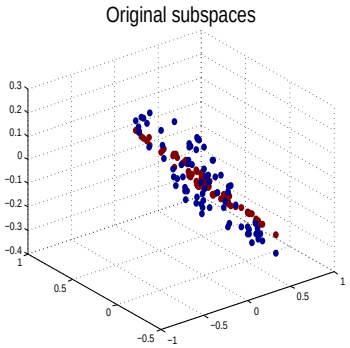


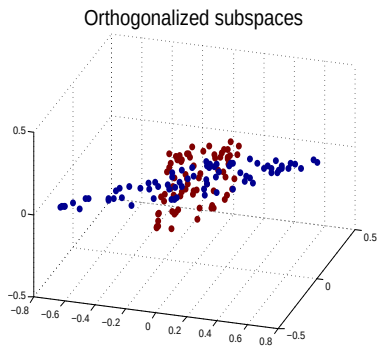


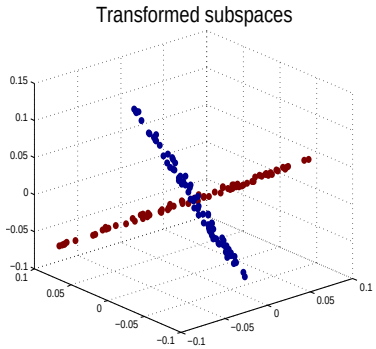
(a) 200 samples per plane, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.91, |\mathbf{B}|_* = 1.88$.

(b) $\mathbf{T} = \begin{bmatrix} -0.36 & 0 & 0.10 \\ -1.4 & 0.09 & -2.98 \\ -0.92 & 0 & 1.54 \\ 1.06 & -1.12 & 2.63 \end{bmatrix}$; $\theta_{AB} = 1.41$, $|\mathbf{A}|_* = 1.61, \quad |\mathbf{B}|_* = 1.62$.

(c) $\mathbf{T} = \begin{bmatrix} 0.14 & 0.03 & 0.33 \\ -0.01 & 0.14 & -0.16 \\ 0.29 & -0.16 & 0.86 \end{bmatrix}$; $\theta_{AB} = 1.55$, $|\mathbf{A}|_* = 1.06, \quad |\mathbf{B}|_* = 1.06$.





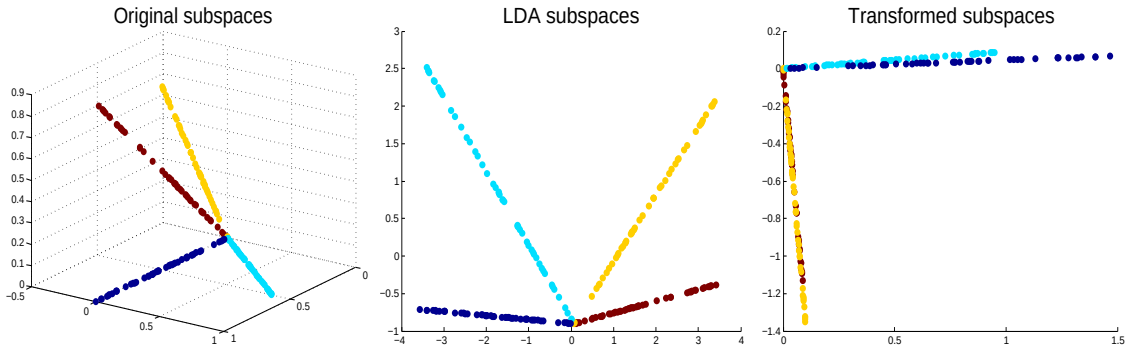


(d) 75 samples per plane, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.92, |\mathbf{B}|_* = 1.81$.

(e) $\mathbf{T} = \begin{bmatrix} -1.62 & 0 & -1.94 \\ -0.50 & 0 & -2.38 \\ 0.25 & -0.50 & 1.81 \\ 1.12 & -1.50 & 2.37 \end{bmatrix}$; $\theta_{AB} = 1.04$, $|\mathbf{A}|_* = 1.75, \quad |\mathbf{B}|_* = 1.71$.

(f) $\mathbf{T} = \begin{bmatrix} 0.06 & -0.12 & 0.30 \\ -0.01 & 0.13 & -0.15 \\ 0.33 & -0.12 & 0.86 \end{bmatrix}$; $\theta_{AB} = 1.55$, $|\mathbf{A}|_* = 1.08, \quad |\mathbf{B}|_* = 1.07$.

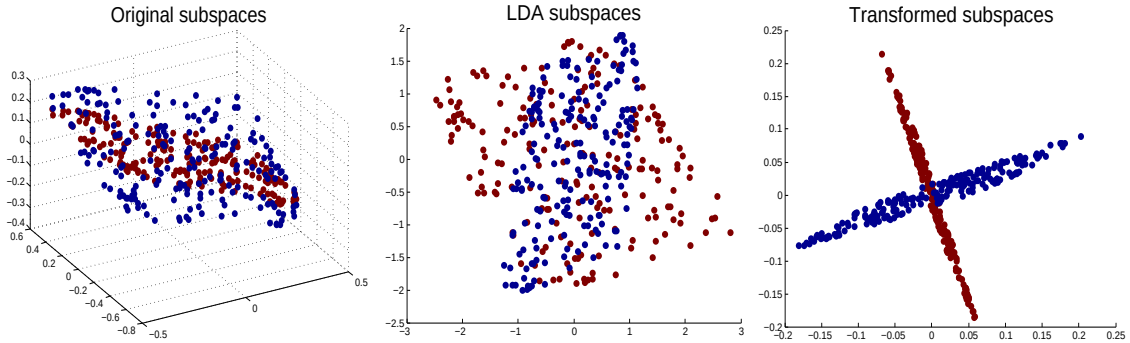
Figure 3: Comparisons with the closed-form orthogonalizing transformation. Two intersecting planes are shown in (a), and each plane contains 200 points. The closed-form orthogonalizing transformation significantly increase the angle between the two planes towards $\frac{\pi}{2}$ in (b). Our leaned transformation in (c) introduces similar subspace separation, but simultaneously enables significantly reduced within subspace variation, indicated by the smaller nuclear norm values (close to 1). The same set of experiments with 75 points per subspace are shown in the second row.



(a) Two classes $\{\mathbf{Y}_+, \mathbf{Y}_-\}$, $\mathbf{Y}_+ = \{\mathbf{A}(blue), \mathbf{B}(cyan)\}$, $\mathbf{Y}_- = \{\mathbf{C}(yellow), \mathbf{D}(red)\}$, $\begin{bmatrix}\theta_{AB} = 1.1, \theta_{AC} = 1.1, \theta_{AD} = 1.1,\\ \theta_{BC} = 1.3, \theta_{BD} = 1.4, \theta_{CD} = 0.5\end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.58, \quad |\mathbf{Y}_-|_* = 1.27.$

(b) $\mathbf{T} = \begin{bmatrix}-3.64 & -1.95 & 5.98\\ 0.19 & 3.87 & 3.35\end{bmatrix}$; $\begin{bmatrix}\theta_{AB} = 0.7, \theta_{AC} = 0.78, \theta_{AD} = 0.2,\\ \theta_{BC} = 1.5, \theta_{BD} = 0.9, \theta_{CD} = 0.57\end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.35, \quad |\mathbf{Y}_-|_* = 1.27.$

(c) $\mathbf{T} = \begin{bmatrix}1.47 & 0.26 & -0.73\\ 0.07 & 0.06 & -1.62\end{bmatrix}$; $\begin{bmatrix}\theta_{AB} = 0.04, \theta_{AC} = 1.54, \theta_{AD} = 1.54,\\ \theta_{BC} = 1.55, \theta_{BD} = 1.56, \theta_{CD} = 0.01\end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.02, \quad |\mathbf{Y}_-|_* = 1.00.$



(d) Two classes $\{\mathbf{A}(\text{blue}), \mathbf{B}(\text{red})\}$, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.91, |\mathbf{B}|_* = 1.88.$

(e) $\mathbf{T} = \begin{bmatrix}-0.54 & 2.60 & -9.51\\ 0.56 & -3.21 & -1.02\end{bmatrix}$; $\theta_{AB} = 0$, $|\mathbf{A}|_* = 1.52, \quad |\mathbf{B}|_* = 1.69.$

(f) $\mathbf{T} = \begin{bmatrix}0.49 & -0.11 & 1.27\\ -0.09 & 0.29 & -0.59\end{bmatrix}$; $\theta_{AB} = 1.57$, $|\mathbf{A}|_* = 1.08, \quad |\mathbf{B}|_* = 1.03.$

Figure 4: Comparisons with the linear discriminant analysis (LDA). Two classes $\mathbf{Y}_+$ and $\mathbf{Y}_-$ are shown in (a), each class consisting of two lines. We notice that our learned transformation (c) shows smaller intra-class variation than LDA in (b) by merging two lines in each class, and simultaneously maximizes the angle between two classes towards $\frac{\pi}{2}$ (such two-class clustering and classification is critical for example for trees-based techniques Qiu and Sapiro (2014)). (d) shows an example of two non-linearly separable classes, i.e., two intersecting planes, which cannot be improved by LDA in (e). However, our learned transformation in (f) prepares data to be separable using subspace clustering.

intersecting planes, which cannot be improved by LDA, as shown in Figure 4e. However, our learned transformation in Figure 4f prepares the data to be separable using subspace clustering. As shown in Qiu and Sapiro (2014), the property demonstrated above makes our learned transformation a better learner than LDA in a binary classification tree.

Lastly, we generated an interesting disjoint case: we consider three lines A, B and C on the same plane that intersect at the origin; the angles between them are $\theta_{AB} = 0.08$, $\theta_{BC} = 0.08$, and $\theta_{AC} = 0.17$. As the closed-form orthogonalizing approach is valid for independent subspaces, it fails by producing $\theta_{AB} = 0.005$, $\theta_{BC} = 0.005$, $\theta_{BC} = 0.01$. Our framework is not limited to that, even if additional theoretical foundations are yet to come. After our learned transformation, we have $\theta_{AB} = 1.20$, $\theta_{BC} = 1.20$, and $\theta_{AC} = 0.75$. We can make two immediate observations: First, all angles are significantly increased within the valid range of $[0, \frac{\pi}{2}]$. Second, $\theta_{AB}+\theta_{BC}+\theta_{AC} = \pi$ (we made the same two observations while repeating the experiments with different subspace angles). Though at this point we have no clean interpretation about how those angles are balanced when pair-wise orthogonality is not possible, we strongly believe that some theories are behind the above persistent observations and we are currently exploring this.

2.4 Discussions about Other Matrix Norms

We now discuss the advantages of replacing the rank function in (1) with the nuclear norm over other (popular) matrix norms, e.g., the induced 2-norm and the Frobenius norm.

Proposition 3 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_2 \leq ||\mathbf{A}||_2 + ||\mathbf{B}||_2,$$

with equality if at least one of the two matrices is zero.

Proposition 4 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

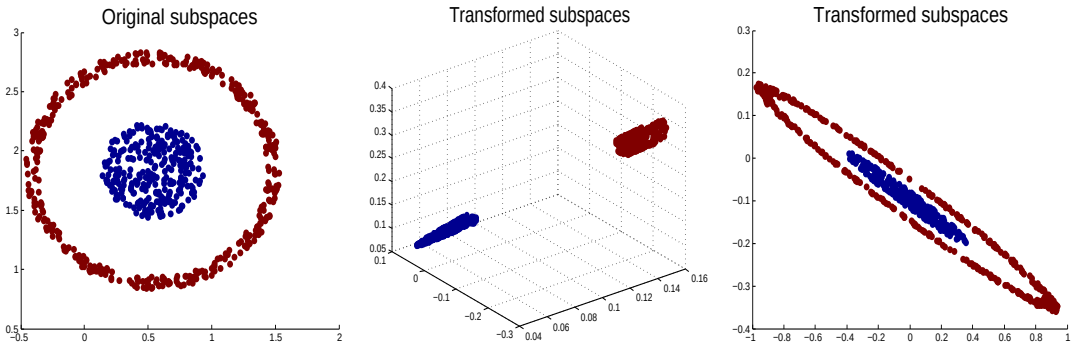
$$||[\mathbf{A}, \mathbf{B}]||_F \leq ||\mathbf{A}||_F + ||\mathbf{B}||_F,$$

with equality if and only if at least one of the two matrices is zero.

We choose the nuclear norm in (6) for two major advantages that are not so favorable in other (popular) matrix norms:

- The nuclear norm is the best convex approximation of the rank function Fazel (2002), which helps to reduce the variation within the subspaces (first term in (6));
- The objective function (6) is optimized when the distance between every pair of subspaces is maximized after transformation, which helps to introduce separations between the subspaces.

Note that (1), which is based on the rank, reaches the minimum when subspaces are independent but not necessarily maximally distant. Propositions 3 and 4 show that the property of the nuclear norm in Theorem 1 holds for the induced 2-norm and the Frobenius norm. However, if we replace the rank function in (1) with the induced 2-norm norm or the Frobenius norm, the objective function is minimized at the trivial solution $\mathbf{T} = 0$, which is prevented by the normalization condition $||\mathbf{T}||_2 = 1$.



(a) (b) (c)

Figure 5: A synthetic example illustrating the kernelized transformation learning. (a) is transformed to (b) with an RBF kernel applied, and to (c) without kernel.

2.5 Online Learning Low-rank Transformations

When data $\mathbf{Y}$ is big, we use an online algorithm to learn the low-rank transformation $\mathbf{T}$:

- We first randomly partition the data set $\mathbf{Y}$ into B mini-batches;
- Using mini-batch subgradient descent, a variant of stochastic subgradient descent, the subgradient in Appendix C is approximated by a sum of subgradients obtained from each mini-batch of samples,

$$\mathbf{T}^{(t+1)} = \mathbf{T}^{(t)} - \nu \sum_{b=1}^{B} \Delta \mathbf{T}_b, \tag{8}$$

  where $\Delta \mathbf{T}_b$ is obtained using only data points in the b-th mini-batch;
- Starting with the first mini-batch, we learn the subspace transformation $\mathbf{T}_b$ using data only in the b-th mini-batch, with $\mathbf{T}_{b-1}$ as warm restart.

2.6 Subspace Transformation with Compression

Given data $\mathbf{Y} \subseteq \mathbb{R}^d$, so far, we considered a square linear transformation $\mathbf{T}$ of size $d \times d$. If we devise a "fat" linear transformation $\mathbf{T}$ of size $r \times d$, where $(r < d)$, we enable dimension reduction along with transformation. This connects the proposed framework with the literature on compressed sensing, though the goal here is to learn a "sensing" matrix $\mathbf{T}$ for subspace classification and not for reconstruction Carson et al. (2012). The nuclear-norm minimization provides a new metric for such compressed sensing design (or compressed feature learning) paradigm. Results with this reduced dimensionality will be presented in Section 5.

2.7 Kernelized Transformation

The linear transformation suggested above shows effective when data approximately lie in linear subspaces. To improve the ability in handling more generic data, we can further map

data points into an inner product space prior to learning the transformation. Given a data point $\mathbf{y}$, we create a nonlinear map $\mathcal{K}(\mathbf{y}) = (\kappa(\mathbf{y}, \mathbf{y}_1); ...; \kappa(\mathbf{y}, \mathbf{y}_n))$ by computing the inner product between $\mathbf{y}$ and a fixed set of n points $\{\mathbf{y}_1, ..., \mathbf{y}_n\}$ randomly drawn from the training set. The inner products are computed via the kernel function, $\kappa(\mathbf{y}, \mathbf{y}_i) = \varphi(\mathbf{y})'\varphi(\mathbf{y}_i)$, which has to satisfy the Mercer conditions; note that no explicit representation for φ is required. Examples of kernel functions include polynomial kernels $\kappa(\mathbf{y}, \mathbf{y}_i) = (\mathbf{y}'\mathbf{y}_i + p)^q$ (with p and q being constants), and radial basis function (RBF) kernels $\kappa(\mathbf{y}, \mathbf{y}_i) = \exp(-\frac{||\mathbf{y}-\mathbf{y}_i||_2^2}{2\sigma^2})$ with variance σ^2. Given a set of data points $\mathbf{Y}$, the set of mapped data is denoted as $\mathcal{K}(\mathbf{Y}) \subseteq \mathbb{R}^n$. We now learn an $n \times n$ kernelized transformation $\mathbf{T}$ minimizing

$$\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathcal{K}(\mathbf{Y}_c)||_* - ||\mathbf{T}\mathcal{K}(\mathbf{Y})||_*, \quad \text{s.t. } ||\mathbf{T}||_2 = 1. \tag{9}$$

Figure 5 shows a synthetic example illustrating the kernelized transformation learning, where a 256-dimensional RBF kernel is applied.

3. Subspace Clustering using Low-rank Transformations

We now move from classification, where we learned the transform from training labeled data, to clustering, where no training data is available. In particular, we address the *subspace clustering* problem, meaning to partition the data set $\mathbf{Y}$ into C clusters corresponding to their underlying subspaces. We first present a general procedure to enhance the performance of existing subspace clustering methods in the literature. Then we further propose a specific fast subspace clustering technique to fully exploit the low-rank structure of (learned) transformed subspaces.

3.1 A Learned Robust Subspace Clustering (LRSC) Framework

In clustering tasks, the data labeling is of course not known beforehand in practice. The proposed algorithm, Algorithm 1, iterates between two stages: In the first assignment stage, we obtain clusters using any subspace clustering methods, e.g., SSC (Elhamifar and Vidal, 2013), LSA (Yan and Pollefeys, 2006), LBF (Zhang et al., 2012). In particular, in this paper we often use the new improved technique introduced in Section 3.2. In the second update stage, based on the current clustering result, we compute the optimal subspace transformation that minimizes (6). The algorithm is repeated until the clustering assignments stop changing.

The LRSC algorithm is a general procedure to enhance the performance of any subspace clustering methods, and part of the beauty of the proposed model is that it can be applied to any such algorithm, and even beyond (Qiu and Sapiro, 2014). We don't enforce an overall objective function at the present form for such versatility purpose.

To study convergence, one way is to adopt the subspace clustering method for the LRSC assignment step by optimizing the same LRSC update criterion (6): given the cluster assignment and the transformation $\mathbf{T}$ at the current LRSC iteration, we take a point $\mathbf{y}_i$ out of its current cluster (keep the rest assignments no change) and place it into a cluster $\mathbf{Y}_c$ that minimize $\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_*$. We iteratively perform this for all points, and then update

$\mathbf{T}$ using current $\mathbf{T}$ as warm restart. In this way, we decrease (or keep) the overall objective function (6) after each LRSC iteration.

However, the above approach is computational expensive and only allow one specific subspace clustering method. Thus, in the present implementation, an overall objective function of the type that the LRSC algorithm optimizes can take a form such as

$$\arg\min_{\mathbf{T},\{\mathcal{S}_c\}_{c=1}^{C}} \sum_{c=1}^{C} \sum_{\mathbf{y}_i \in \mathcal{S}_c} ||\mathbf{T}\mathbf{y}_i - P_{\mathbf{T}\mathbf{Y}_c}\mathbf{T}\mathbf{y}_i||_2^2 + \lambda[\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_*], \quad \text{s.t.} ||\mathbf{T}||_2 = 1, \tag{10}$$

where $\mathbf{Y}_c$ denotes the set of points $\mathbf{y}_i$ in the c-th subspace $\mathcal{S}_c$, and $P_{\mathbf{T}\mathbf{Y}_c}$ denotes the projection onto $\mathbf{T}\mathbf{Y}_c$. The LRSC iterative algorithm optimize (10) through alternative minimization (with a similar form as the popular k-means, but with a different data model and with the learned transform). While formally studying its convergence is the subject of future research, the experimental validation presented already demonstrates excellent performance, with LRSC just one of the possible applications of the proposed learned transform.

In all our experiments, we observe significant clustering error reduction in the first few LRSC iterations, and the proposed LRSC iterations enable significantly cleaner subspaces for all subspace clustering benchmark data in the literature. The intuition behinds the observed empirical convergence is that the update step in each LRSC iteration decreases the second term in (10) to a small value close to 0 as discussed in Section 2; at the same time, the updated transformation tends to reduce the intra-subspace variation, which further reduces the first cluster deviation term in (10) even with assignments derived from various subspace clustering methods.

Input: A set of data points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$ in a union of C subspaces.
Output: A partition of $\mathbf{Y}$ into C disjoint clusters $\{\mathbf{Y}_c\}_{c=1}^{C}$ based on underlying subspaces.
begin
1. Initial a transformation matrix $\mathbf{T}$ as the identity matrix ;
repeat
Assignment stage:
2. Assign points in $\mathbf{TY}$ to clusters with any subspace clustering methods, e.g., the proposed R-SSC;
Update stage:
3. Obtain transformation $\mathbf{T}$ by minimizing (6) based on the current clustering result ;
until *assignment convergence*;
4. Return the current clustering result $\{\mathbf{Y}_c\}_{c=1}^{C}$;
end

Algorithm 1: Learning a robust subspace clustering (LRSC) framework.

3.2 Robust Sparse Subspace Clustering (R-SSC)

Though Algorithm 1 can adopt any subspace clustering methods, to fully exploit the low-rank structure of the learned transformed subspaces, we further propose the following specific technique for the clustering step in the LRSC framework, called Robust Sparse Subspace Clustering (R-SSC):

1. For the transformed subspaces, we first recover their low-rank representation $\mathbf{L}$ by performing a low-rank decomposition (11), e.g., using RPCA (Candès et al., 2011),[1]

$$\arg\min_{\mathbf{L},\mathbf{S}} ||\mathbf{L}||_* + \beta||\mathbf{S}||_1 \quad \text{s.t. } \mathbf{TY} = \mathbf{L} + \mathbf{S}. \tag{11}$$

2. Each transformed point $\mathbf{Ty}_i$ is then sparsely decomposed over $\mathbf{L}$,

$$\arg\min_{\mathbf{x}_i} \|\mathbf{Ty}_i - \mathbf{L}\mathbf{x}_i\|_2^2 \quad \text{s.t. } \|\mathbf{x}_i\|_0 \leq K, \tag{12}$$

   where K is a predefined sparsity value ($K > d$). As explained in Elhamifar and Vidal (2013), a data point in a linear or affine subspace of dimension d can be written as a linear or affine combination of d or $d+1$ points in the same subspace. Thus, if we represent a point as a linear or affine combination of all other points, a sparse linear or affine combination can be obtained by choosing d or $d+1$ nonzero coefficients.
3. As the optimization process for (12) is computationally demanding, we further simplify (12) using Local Linear Embedding (Roweis and Saul, 2000; Wang et al., 2010). Each transformed point $\mathbf{Ty}_i$ is represented using its K Nearest Neighbors (NN) in $\mathbf{L}$, which are denoted as $\mathbf{L}_i$,

$$\arg\min_{\mathbf{x}_i} \|\mathbf{Ty}_i - \mathbf{L}_i\mathbf{x}_i\|_2^2 \quad \text{s.t. } \mathbf{1}'\mathbf{x}_i = 1. \tag{13}$$

   Let $\bar{\mathbf{L}}_i = \mathbf{L}_i - \mathbf{1}\mathbf{Ty}_i^T$. $\mathbf{x}_i$ can then be efficiently obtained in closed form (Saul and Roweis, 2000),

$$\mathbf{x}_i = \bar{\mathbf{L}}_i\bar{\mathbf{L}}_i^T \setminus \mathbf{1},$$

   where $\mathbf{x} = \mathbf{A} \setminus \mathbf{B}$ solves the system of linear equations $\mathbf{Ax} = \mathbf{B}$, and then we rescale $\mathbf{x}_i$ so that $\mathbf{1}'\mathbf{x}_i = 1$. As suggested in Roweis and Saul (2000), if the correlation matrix $\bar{\mathbf{L}}_i\bar{\mathbf{L}}_i^T$ is nearly singular, it can be conditioned by adding a small multiple of the identity matrix. From experiments, we observe this simplification step dramatically reduces the running time, without sacrificing the accuracy.
4. Given the sparse representation $\mathbf{x}_i$ of each transformed data point $\mathbf{Ty}_i$, we denote the sparse representation matrix as $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]$. It is noted that $\mathbf{x}_i$ is written as an N-sized vector with no more than $K << N$ non-zero values (N being the total number of data points). The pairwise affinity matrix is now defined as $\mathbf{W} = |\mathbf{X}| + |\mathbf{X}^T|$, and the subspace clustering is obtained using spectral clustering (Luxburg, 2007).

Based on experimental results presented in Section 5, the proposed R-SSC outperforms state-of-the-art subspace clustering techniques, in both accuracy and running time, e.g., about 500 times faster than the original SSC using the implementation provided in Elhamifar and Vidal (2013). Performance is further enhanced when R-SCC is used as an internal step of LRSC in Algorithm 1.

1. Note that while the learned transform $\mathbf{T}$ encourages low-rank in each sub-space, outliers might still exist. Moreover, during the iterations in Algorithm 1, the intermediate learned $\mathbf{T}$ is not yet the desired one. This justifies the incorporation of this further low-rank decomposition.

4. Classification using Single or Multiple Low-rank Transformations

In Section 2, learning one global transformation over all classes has been discussed, and then incorporated into a clustering framework in Section 3. The availability of data labels for training enables us to consider instead learning individual class-based linear transformation. The problem of class-based linear transformation learning can be formulated as

$$\arg\min_{\{\mathbf{T}_c\}_{c=1}^{C}} \sum_{c=1}^{C} [||\mathbf{T}_c\mathbf{Y}_c||_* - \lambda||\mathbf{T}_c\mathbf{Y}_{\neg c}||_*], \tag{14}$$

where $\mathbf{T}_c \in \mathbb{R}^{d\times d}$ denotes the transformation for the c-th class, $\mathbf{Y}_{\neg c} = \mathbf{Y} \setminus \mathbf{Y}_c$ denotes all data except the c-th class, and λ is a positive balance parameter.

When a global transformation matrix $\mathbf{T}$ is learned, we can perform classification in the transformed space by simply considering the transformed data $\mathbf{TY}$ as the new features. For example, when a Nearest Neighbor (NN) classifier is used, a testing sample $\mathbf{y}$ uses $\mathbf{Ty}$ as the feature and searches for nearest neighbors among $\mathbf{TY}$.

To fully exploit the low-rank structure of the transformed data, we propose to perform classification through the following procedure:

- For the c-th class, we first recover its low-rank representation $\mathbf{L}_c$ by performing low-rank decomposition (15), e.g., using RPCA (Candès et al., 2011):[2]

$$\arg\min_{\mathbf{L}_c,\mathbf{S}_c} ||\mathbf{L}_c||_* + \beta||\mathbf{S}_c||_1 \quad \text{s.t. } \mathbf{TY}_c = \mathbf{L}_c + \mathbf{S}_c. \tag{15}$$

- Each testing image $\mathbf{y}$ will then be assigned to the low-rank subspace $\mathbf{L}_c$ that gives the minimal reconstruction error through sparse decomposition, e.g., using OMP (Pati et al., Nov. 1993):

$$\arg\min_{\mathbf{x}} \|\mathbf{Ty} - \mathbf{L}_i\mathbf{x}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_0 \leq T, \tag{16}$$

  where T is a predefined sparsity value.

When class-based transformations $\{\mathbf{T}_c\}_{c=1}^{C}$ are learned, we perform recognition in a similar way. However, now we apply all the learned transforms $\mathbf{T}_c$ to each testing data point and then pick the best one using the same criterion of minimal reconstruction error through sparse decomposition (16).

5. Experimental Evaluation

This section first presents experimental evaluations on subspace clustering using three public data sets (standard benchmarks): the MNIST handwritten digit data set, the Extended YaleB face data set (Georghiades et al., 2001) and the Hopkins 155 database of motion segmentation. The MNIST data set consists of 8-bit gray scale handwritten digit images of "0" through "9" and 7000 examples for each class. The Extended YaleB face data set

2. Note that this is done only once and can be considered part of the training stage. As before, this further low-rank decomposition helps to handle outliers not addressed by the learned transform.

contains 38 subjects with near frontal pose under 64 lighting conditions. All the images are resized to 16×16. The classical Hopkins 155 database of motion segmentation, which is available at `http://www.vision.jhu.edu/data/hopkins155`, contains 155 video sequences along with extracted feature trajectories, where 120 of the videos have two motions and 35 of the videos have three motions.

Subspace clustering methods compared are SSC (Elhamifar and Vidal, 2013), LSA (Yan and Pollefeys, 2006), and LBF (Zhang et al., 2012). Based on the studies in Elhamifar and Vidal (2013), Vidal (2011) and Zhang et al. (2012), these three methods exhibit state-of-the-art subspace clustering performance. We adopt the LSA and SSC implementations provided in Elhamifar and Vidal (2013) from `http://www.vision.jhu.edu/code/`, and the LBF implementation provided in Zhang et al. (2012) from `http://www.ima.umn.edu/~zhang620/lbf/`. We adopt similar setups as described in Zhang et al. (2012) for experiments on subspace clustering.

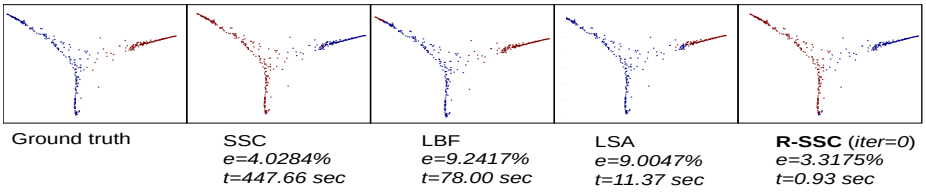
This section then presents experimental evaluations on classification using two public face data sets: the CMU PIE data set (Sim et al., 2003) and the Extended YaleB data set. The PIE data set consists of 68 subjects imaged simultaneously under 13 different poses and 21 lighting conditions. All the face images are resized to 20×20. We adopt a NN classifier unless otherwise specified.

5.1 Subspace Clustering with Illustrative Examples

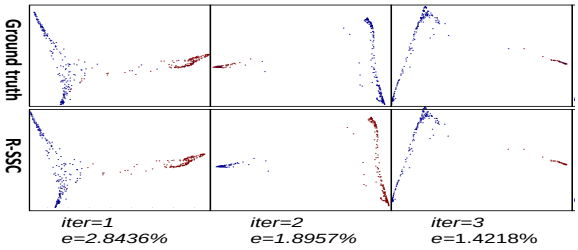
For illustration purposes, we conduct the first set of experiments on a subset of the MNIST data set. We adopt a similar setup as described in Zhang et al. (2012), using the same sets of 2 or 3 digits, and randomly choose 200 images for each digit. We set the sparsity value $K = 6$ for R-SSC, and perform 100 iterations for the subgradient updates while learning the transformation on subspaces. The subgradient update step was $\nu = 0.02$ (see Appendix C for details on the projected subgradient optimization algorithm).

Unless otherwise stated, we do not perform dimension reduction, such as PCA or random projections, to preprocess the data, thereby further saving computations (please note that the learned transform can itself reduce dimensions if so desired, see Section 5.8). In the literature, e.g., Elhamifar and Vidal (2013), Vidal (2011) and Zhang et al. (2012), projection to a very low dimension is usually performed to enhance the clustering performance. However, it is often not obvious how to determine the correct projection dimension for real data, and many subspace clustering methods show sensitive to the choice of the projection dimension. This dimension reduction step is not needed in the framework proposed here.

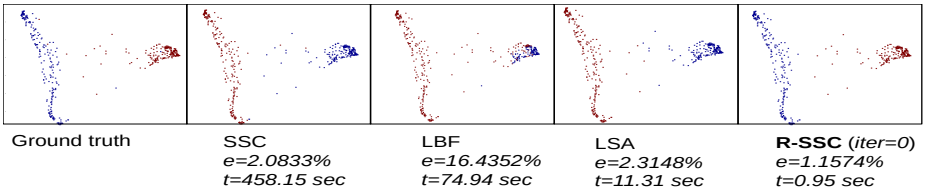
Figure 6 shows the misclassification rate (e) and running time (t) on clustering subspaces of two digits. The misclassification rate is the ratio of misclassified points to the total number of points, i.e., the ratio of points that were assigned to the wrong cluster. For visualization purposes, the data are plotted with the dimension reduced to 2 using Laplacian Eigenmaps Belkin and Niyogi (2003). Different clusters are represented by different colors and the ground truth is plotted using the true cluster labels. The proposed R-SSC outperforms state-of-the-art methods, both in terms of clustering accuracy and running time. The clustering error of R-SSC is further reduced using the proposed LRSC framework in Algorithm 1 through the learned low-rank subspace transformation. The clustering converges after about 3 LRSC iterations. The learned transformation not only recovers a low-rank



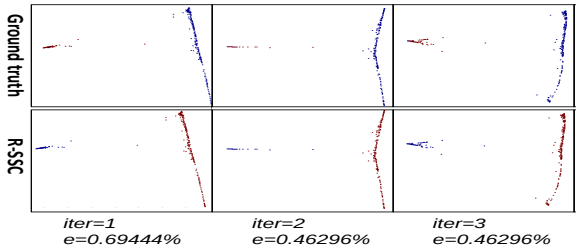
(a) Original subspaces for digits {1, 2}.



(b) Transformed subspaces for digits {1, 2}.

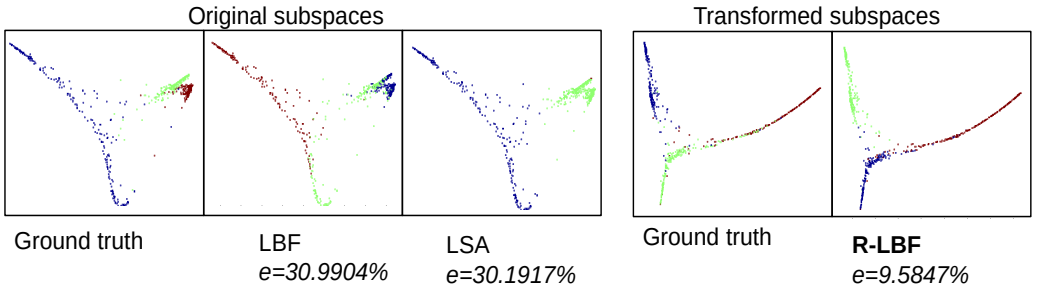


(c) Original subspaces for digits {1, 7}.

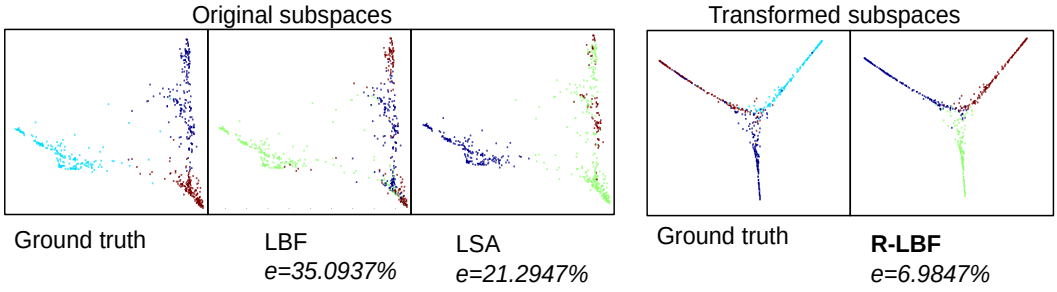


(d) Transformed subspaces for digits {1, 7}.

Figure 6: Misclassification rate (e) and running time (t) on clustering 2 digits. Methods compared are SSC Elhamifar and Vidal (2013), LSA Yan and Pollefeys (2006), and LBF Zhang et al. (2012). For visualization, the data are plotted with the dimension reduced to 2 using Laplacian Eigenmaps Belkin and Niyogi (2003). Different clusters are represented by different colors and the *ground truth* is plotted with the true cluster labels. *iter* indicates the number of LRSC iterations in Algorithm 1. The proposed R-SSC outperforms state-of-the-art methods in terms of both clustering accuracy and running time, e.g., about 500 times faster than SSC. The clustering performance of R-SSC is further improved using the proposed LRSC framework. Note how the data is clearly clustered in clean subspaces in the transformed domain (best viewed zooming on screen).



(a) Digits {1, 2, 3}.



(b) Digits {2, 4, 8}.

Figure 7: Misclassification rate (e) on clustering 3 digits. Methods compared are LSA Yan and Pollefeys (2006) and LBF Zhang et al. (2012). LBF is adopted in the proposed LRSC framework and denoted as R-LBF. After convergence, R-LBF significantly outperforms state-of-the-art methods.

Subsets	[0:1]	[0:2]	[0:3]	[0:4]	[0:5]	[0:6]	[0:7]	[0:8]
C	2	3	4	5	6	7	8	9
LSA	0.47	47.57	36.73	30.90	40.46	48.13	39.87	44.03
LBF	0.47	23.62	29.19	51.37	48.99	53.01	39.87	38.79
LRSC	**0**	**3.88**	**3.89**	**5.31**	**14.04**	**13.79**	**14.50**	**16.05**

Table 1: Misclassification rate ($e\%$) on clustering different numbers of digits in the MNIST data set, $[0 : c]$ denotes the subset of $c + 1$ digits from digit 0 to c. We randomly pick 100 samples per digit. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods.

structure for data from the same subspace, but also increases the separations between the subspaces for more accurate clustering.

Figure 7 shows misclassification rate (e) on clustering subspaces of three digits. Here we adopt LBF in our LRSC framework, denoted as Robust LBF (R-LBF), to illustrate that the performance of existing subspace clustering methods can be enhanced using the proposed LRSC algorithm. After convergence, R-LBF, which uses the proposed learned subspace transformation, significantly outperforms state-of-the-art methods.

Table 1 shows the misclassification rate on clustering different number of digits, $[0 : c]$ denotes the subset of $c+1$ digits from digit 0 to c. We randomly pick 100 samples per digit to compare the performance when a fewer number of data points per class are present. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods.

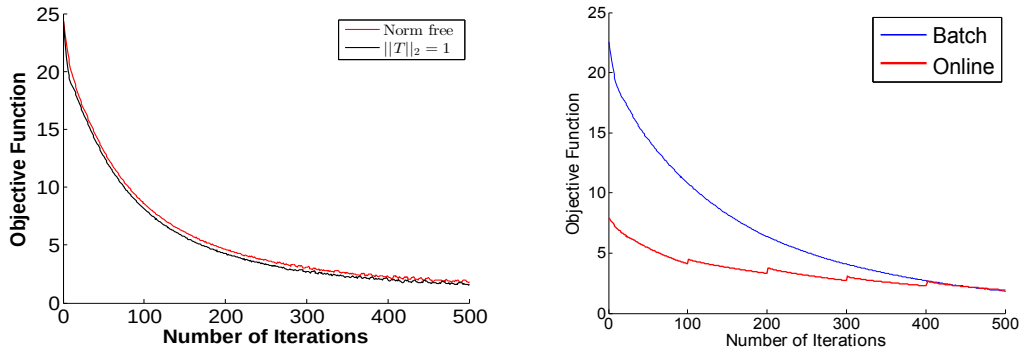
5.1.1 Online vs. Batch Learning

In this set of experiments, we use digits $\{1, 2\}$ from the MNIST data set. We select 1000 images for each digit, and randomly partition them into 5 mini-batches. We first perform one iteration of LRSC in Algorithm 1 over all selected data with and without the norm constraint. As shown in Figure 8a, we both observe empirical convergence for subspace transformation learning via (6) using the projected subgradient method presented in Appendix C.

Starting with the first mini-batch, we then perform one iteration of LRSC over one mini-batch a time, with the subspace transformation learned from the previous mini-batch as warm restart. We adopt here 100 iterations for the subgradient descent updates. As shown in Figure 8b, we observe similar empirical convergence for online transformation learning. To converge to the same objective function value, it takes 131.76 sec. for online learning and 700.27 sec. for batch learning.

5.2 Application to Face Clustering

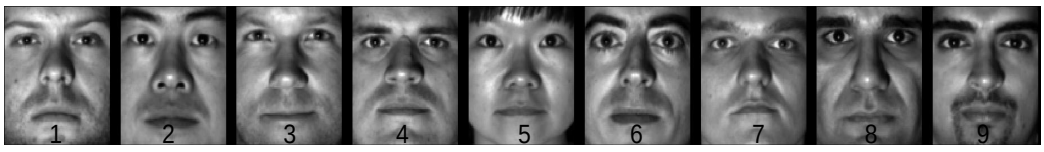
In the Extended YaleB data set, each of the 38 subjects is imaged under 64 lighting conditions, shown in Figure 9a. Under the assumption of Lambertian reflectance, face images of each subject under different lighting conditions can be accurately approximated with a 9-dimensional linear subspace (Basri and Jacobs, 2003). We conduct the face clustering experiments on the first 9 subjects shown in Figure 9b. We set the sparsity value $K = 10$



(a) Batch learning. (b) Online vs. batch learning ($\gamma = 1$).

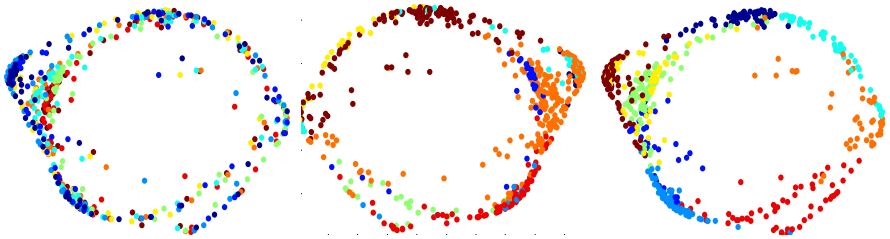
Figure 8: Convergence of the objective function (6) using online and batch learning for subspace transformation. We always observe empirical convergence for both online and batch learning. In (a), we learn with and without the norm constraint respectively. More discussions on convergence can be found in Appendix C. In (b), to converge to the same objective function value, it takes 131.76 sec. for online learning and 700.27 sec. for batch learning.

(a) Example illumination conditions.

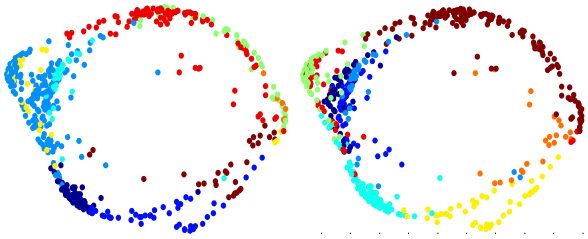


(b) Example subjects.

Figure 9: The extended YaleB face data set.

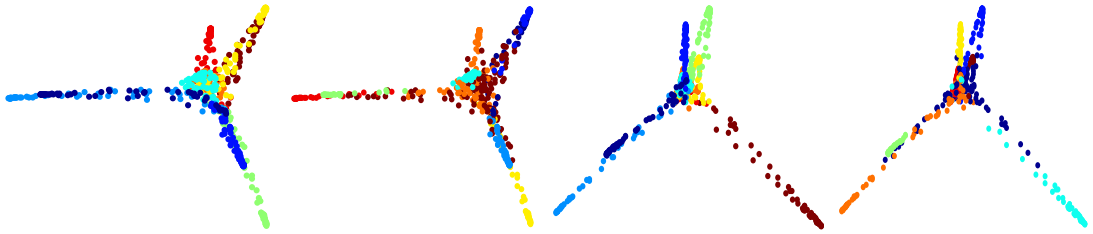


(a) Ground truth. (b) SSC, $e = 71.25\%$, $t = 714.99$ *sec.* (c) LBF, $e = 76.37\%$, $t = 460.76$ *sec.*

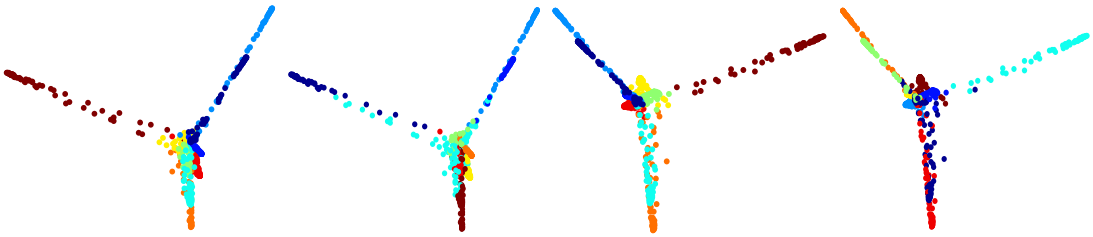


(d) LSA, $e = 71.96\%$, $t = 22.57$ *sec.* (e) R-SSC, $e = \mathbf{67.37}\%$, $t = \mathbf{1.83}$ *sec.*

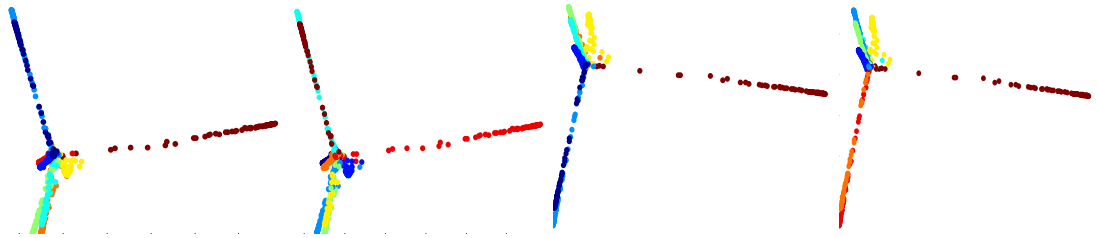
Figure 10: Misclassification rate (e) and running time (t) on clustering 9 subjects using different subspace clustering methods. The proposed R-SSC outperforms state-of-the-art methods both in accuracy and running time. This is further improved using the learned transform, LRSC reduces the error to 4.94%, see Figure 11.



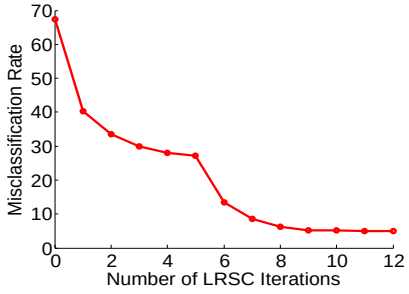
(a) Ground truth (iter=1).(b) $e = 40.39\%$ (iter=1).(c) Ground truth (iter=2).(d) $e = 33.51\%$ (iter=2).



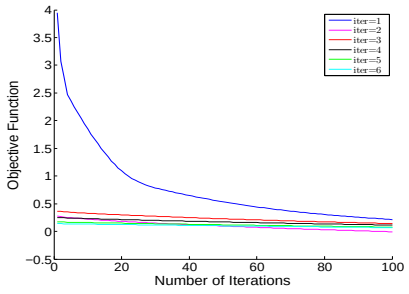
(e) Ground truth (iter=3).(f) $e = 29.98\%$ (iter=3).(g) Ground truth (iter=6).(h) $e = 13.40\%$ (iter=6).



(i) Ground truth (iter=8). (j) $e = 6.17\%$ (iter=8). (k) Ground truth (iter=12). (l) $e = \mathbf{4.94}\%$ (iter=12).

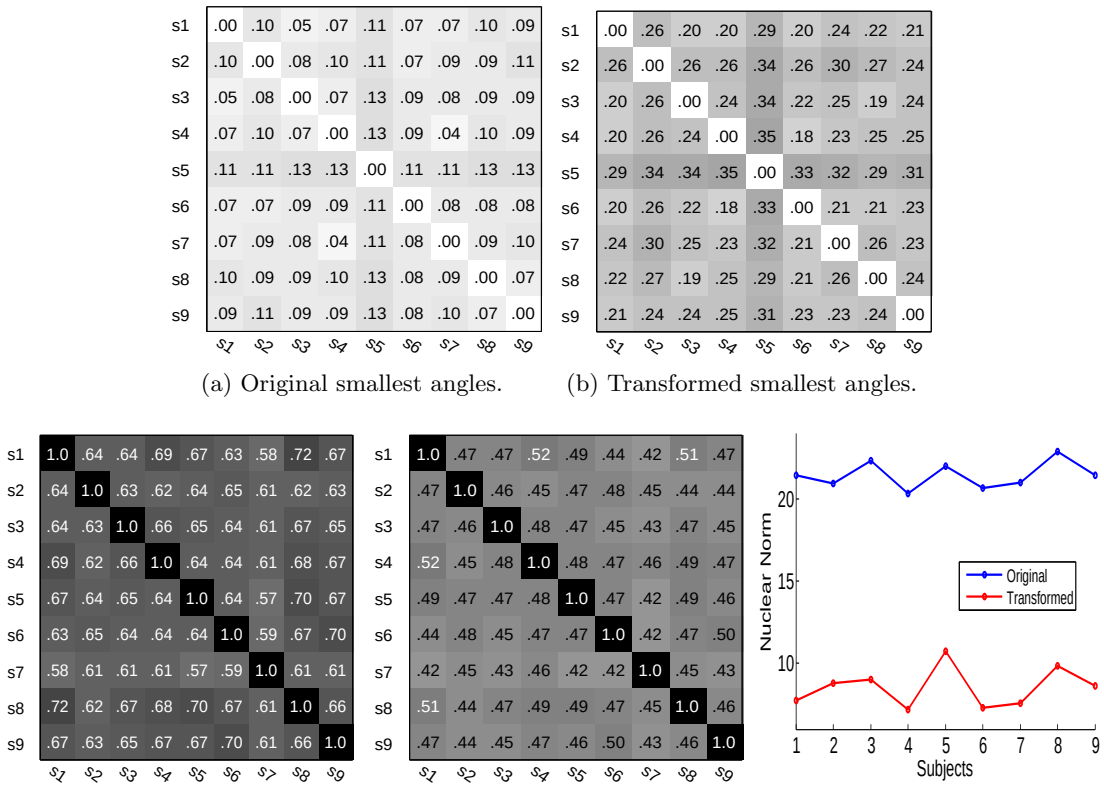


(m) Misclassification rate.



(n) Convergence of $\mathbf{T}$ updating

Figure 11: Misclassification rate (e) on clustering 9 subjects using the proposed LRSC framework. We adopt the proposed R-SSC technique for the clustering step. With the proposed LRSC framework, the clustering error of R-SSC is further reduced significantly, e.g., from 67.37% to 4.94% for the 9-subject case. Note how the classes are clustered in clean subspaces in the transformed domain.



(a) Original smallest angles. (b) Transformed smallest angles.

(c) Original mean cosine angles. (d) Transformed mean cosine angles. (e) Subspace nuclear norm.

Figure 12: The smallest and mean principal angles between pairs of 9 subject subspaces and the nuclear norms of 9 subject subspaces before and after transformation. Note that each entry in (a) and (b) denotes the smallest principal angle, and each entry in (c) and (d) denotes the average cosine over all principal angles. We observe that the learned subspace transformation increases the angles between subspaces and also reduces the nuclear norms of subspaces. Overall, the average smallest principal angles between subspaces increased from 0.09 to 0.26, and the average subspace nuclear norm decreased from 21.43 to 8.53.

Subsets	[1:10]	[1:15]	[1:20]	[1:25]	[1:30]	[1:38]
C	10	15	20	25	30	38
LSA	78.25	82.11	84.92	82.98	82.32	84.79
LBF	78.88	74.92	77.14	78.09	78.73	79.53
LRSC	**5.39**	**4.76**	**9.36**	**8.44**	**8.14**	**11.02**

Table 2: Misclassification rate ($e\%$) on clustering different number of subjects in the Extended YaleB face data set, $[1:c]$ denotes the first c subjects in the data set. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods.

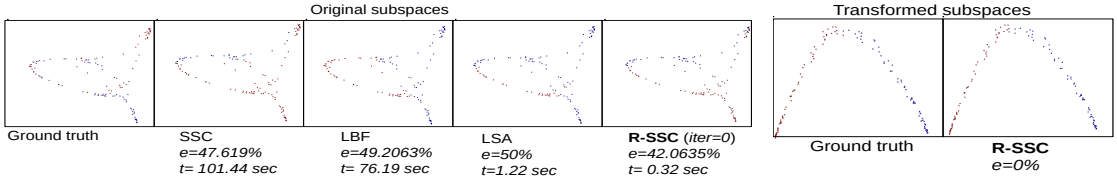
Methods	Misclassification (%)
orthogonalizing	61.36
LDA	9.77
Proposed	**5.47**

Table 3: Misclassification rate ($e\%$) on clustering 38 subjects in the Extended YaleB data set using supervised transformation learning. The proposed transformation learning outperforms both the closed-form orthogonalizing transformation and LDA on clustering the transformed data.

for R-SSC, and perform 100 iterations for the subgradient descent updates while learning the transformation.

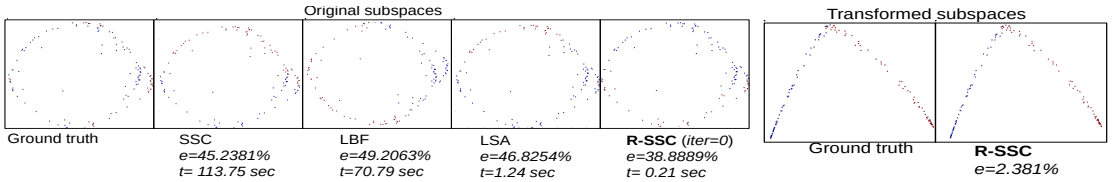
Figure 10 shows error rate (e) and running time (t) on clustering subspaces of 9 subjects using different subspace clustering methods. The proposed R-SSC techniques outperforms state-of-the-art methods both in accuracy and running time. As shown in Figure 11, using the proposed LRSC algorithm (that is, learning the transform), the misclassification errors of R-SSC are further reduced significantly, for example, from 67.37% to 4.94% for the 9 subjects. Figure 11n shows the convergence of the $\mathbf{T}$ updating step in the first few LRSC iterations. The dramatic performance improvement can be explained in Figure 12. We observe, as expected from the theory presented before, that the learned subspace transformation increases the distance (the smallest principal angle) between subspaces and, at the same time, reduces the nuclear norms of subspaces. More results on clustering subspaces of 2 and 3 subjects are shown in Figure 13.

Table 2 shows misclassification rate (e) on clustering subspaces of different number of subjects, $[1:c]$ denotes the first c subjects in the extended YaleB data set. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods. Note that without the low-rank decomposition step in (11), we obtain a misclassification rate 18.38% for clustering all 38 subjects in the Extended YaleB data set, which is slightly lower than the 11.02% reported in Table 2. Thus, pushing the subspaces apart through our learned transformation plays a major role here; and the robustness in the low-rank decomposition enhances the performance even further.



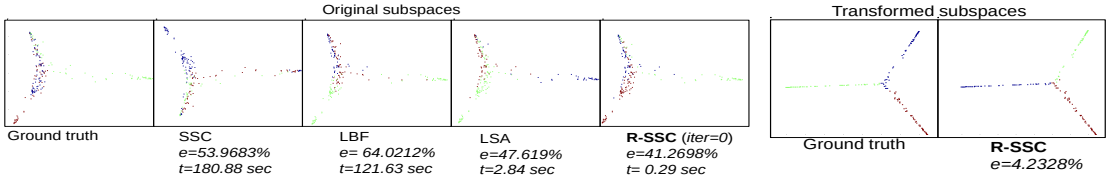
(a) Subjects {1, 2}.

(b) Subjects {1, 2}.



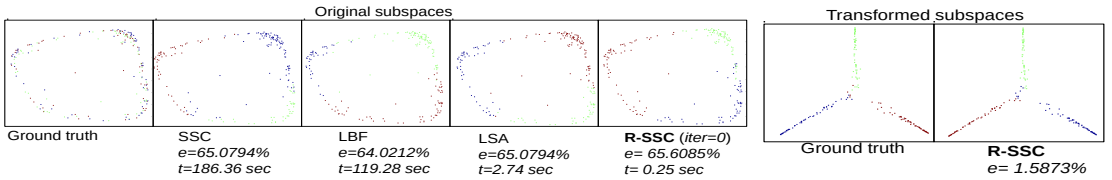
(c) Subjects {2, 3}.

(d) Subjects {2, 3}.



(e) Subjects {4, 5, 6}.

(f) Subjects {4, 5, 6}.



(g) Subjects {7, 8, 9}.

(h) Subjects {7, 8, 9}.

Figure 13: Misclassification rate (e) and running time (t) on clustering 2 and 3 subjects. The proposed R-SSC outperforms state-of-the-art methods both in accuracy and running time. With the proposed LRSC framework, the clustering error of R-SSC is further reduced significantly. Note how the classes are clustered in clean subspaces in the transformed domain (best viewed zooming on screen).

	Check		Traffic		Articulated		All	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
2-motion								
LSA	2.57	0.27	5.43	1.48	4.10	1.22	3.45	0.59
LBF	1.59	0	0.20	0	0.80	0	1.16	0
SSC	1.12	0	0.02	0	0.62	0	**0.82**	0
LRSC	1.19	0	0.23	0	0.88	0	0.92	0
3-motion								
LSA	5.80	1.77	25.07	23.79	7.25	7.25	9.73	2.33
LBF	4.57	0.94	0.38	0	2.66	2.66	3.63	0.64
SSC	2.97	0.27	0.58	0	1.42	0	2.45	0.2
LRSC	1.59	0	0.32	0	1.60	1.60	**1.34**	0

Table 4: Misclassification rate ($e\%$) on two motions and three motions segmentation in the Hopkins 155 data set. As shown in Vidal (2011); Zhang et al. (2012), the SSC method significantly outperforms all previous state-of-the-art methods on this data set. The proposed LRSC shows comparable results to SSC for two motions and outperforms SSC for three motions. Note that our method is orders of magnitude faster than SSC.

In Figure 3 and Figure 4, using synthetic examples, we previously compared our learned transformation with the closed-form orthogonalizing transformation and LDA. In Table 3, we further compare three transformations using real data. We perform supervised transformation learning on all 38 subjects in the Extended YaleB data set using three different transformation learning algorithms, and then perform subspace clustering on the transformed data. The proposed transformation learning significantly outperforms the other two methods.

5.3 Application to Motion Segmentation

The Hopkins 155 data set consists of three types of videos: checker, traffic and articulated, and 120 of the videos have two motions and 35 of the videos have three motions. The main task is to segment a video sequence of multiple rigidly moving objects into multiple spatiotemporal regions that correspond to different motions in the scene. This motion data set contains much cleaner subspace data than the digits and faces data evaluated above. To enable a fair comparison, we project the data into a lower dimensional subspace using PCA as explained in Vidal (2011); Zhang et al. (2012). Results on other comparing methods are taken from Vidal (2011). As shown in Vidal (2011); Zhang et al. (2012), the SSC method significantly outperforms all previous state-of-the-art methods on this data set. From Table 4, we can see that our method shows comparable results to SSC for two motions and outperforms SSC for three motions. Note that our method is orders of magnitude faster than SSC as discussed earlier.

Method	Accuracy (%)
D-KSVD Zhang and Li (2010)	94.10
LC-KSVD Jiang et al. (2011)	96.70
SRC Wright et al. (2009)	97.20
Original+NN	91.77
Class LRT+NN	97.86
Class LRT+OMP	92.43
Global LRT+NN	99.10
Global LRT+OMP	**99.51**

Table 5: Recognition accuracies (%) under illumination variations for the Extended YaleB data set. The recognition accuracy is increased from 91.77% to 99.10% by simply applying the learned low-rank transformation (LRT) matrix to the original face images.

5.4 Application to Face Recognition across Illumination

For the Extended YaleB data set, we adopt a similar setup as described in Jiang et al. (2011); Zhang and Li (2010). We split the data set into two halves by randomly selecting 32 lighting conditions for training, and the other half for testing. We learn a global low-rank transformation matrix from the training data.

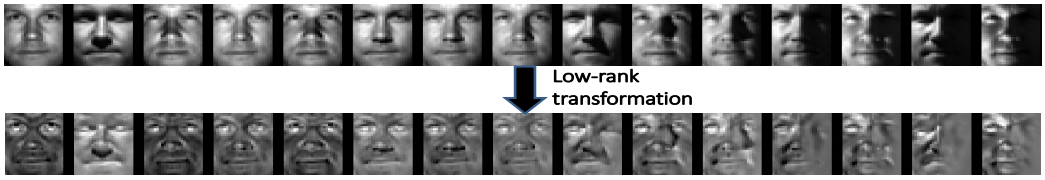
We report recognition accuracies in Table 5. We make the following observations. First, the recognition accuracy is increased from 91.77% to 99.10% by simply applying the learned transformation matrix to the original face images. Second, the best accuracy is obtained by first recovering the low-rank subspace for each subject, e.g., the third row in Figure 14a. Then, each transformed testing face, e.g., the second row in Figure 14b, is sparsely decomposed over the low-rank subspace of each subject through OMP, and classified to the subject with the minimal reconstruction error. A sparsity value 10 is used here for OMP. As shown in Figure 14c, the low-rank representation for each subject shows reduced variations caused by illumination. Third, the global transformation performs better here than class-based transformations, which can be due to the fact that illumination in this data set varies in a globally coordinated way across subjects. Last but not least, our method outperforms state-of-the-art sparse representation based face recognition methods.

5.5 Application to Face Recognition across Pose

We adopt the similar setup as described in Castillo and Jacobs (2009) to enable the comparison. In this experiment, we classify 68 subjects in three poses, frontal (c27), side (c05), and profile (c22), under lighting condition 12. We use the remaining poses as the training data.

For this example, we learn a class-based low-rank transformation matrix per subject from the training data. It is noted that the goal is to learn a transformation matrix to help in the classification, which may not necessarily correspond to the real geometric transform. Table 6 shows the face recognition accuracies under pose variations for the CMU PIE

(a) Low-rank decomposition of globally transformed training samples



(b) Globally transformed testing samples

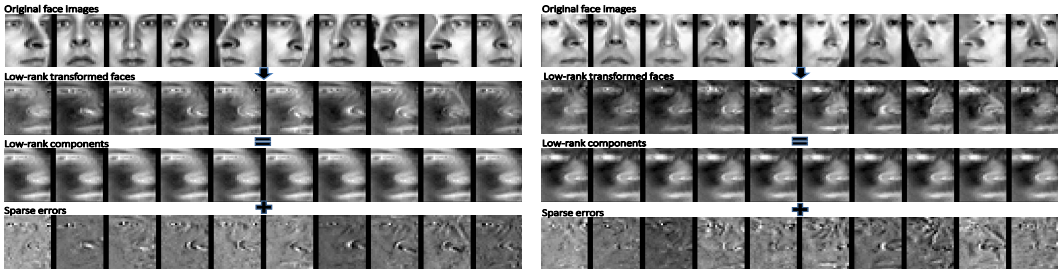
(c) Mean low-rank components for subjects in the training data

Figure 14: Face recognition across illumination using global low-rank transformation.

Method	Frontal (c27)	Side (c05)	Profile (c22)
SMD Castillo and Jacobs (2009)	83	82	57
Original+NN	39.85	37.65	17.06
Original(crop+flip)+NN	44.12	45.88	22.94
Class LRT+NN	98.97	96.91	67.65
Class LRT+OMP	**100**	**100**	**67.65**
Global LRT+NN	97.06	95.58	50
Global LRT+OMP	100	98.53	57.35

Table 6: Recognition accuracies (%) under pose variations for the CMU PIE data set.

data set (we applied the crop-and-flip step discussed in Figure 1.). We make the following observations. First, the recognition accuracy is dramatically increased after applying the learned transformations. Second, the best accuracy is obtained by recovering the low-rank subspace for each subject, e.g., the third row in Figure 15a and Figure 15b. Then, each transformed testing face, e.g., Figure 15c and Figure 15d, is sparsely decomposed over the low-rank subspace of each subject through OMP, and classified to the subject with the minimal reconstruction error, Section 4. Third, the class-based transformation performs better than the global transformation in this case. The choice between these two settings



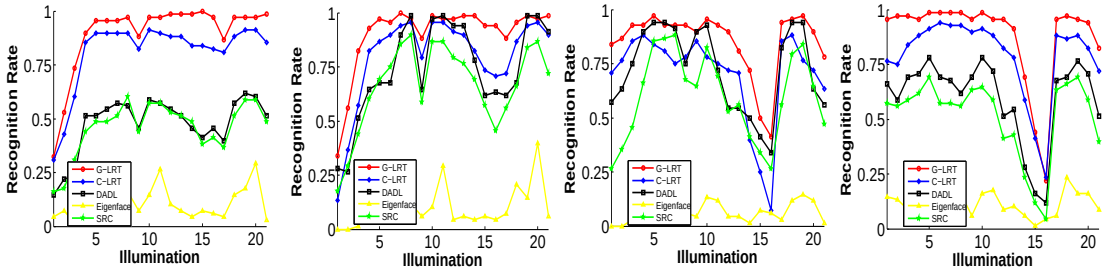
(a) Low-rank decomposition of class-based transformed training samples for *subject3*

(b) Low-rank decomposition of class-based transformed training samples for *subject1*

(c) class-based transformed testing samples for *subject3*

(d) class-based transformed testing samples for *subject1*

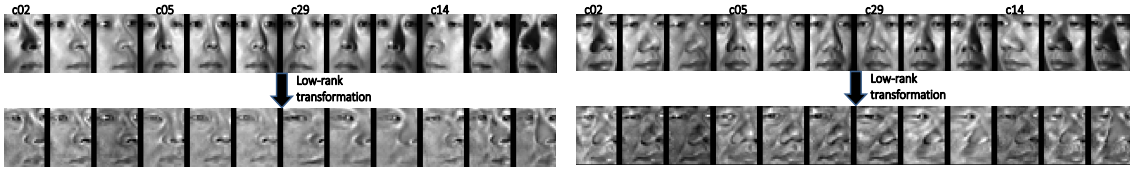
Figure 15: Face recognition across pose using class-based low-rank transformation. Note, for example in (c) and (d), how the learned transform reduces the pose-variability.



(a) Pose c02 (b) Pose c05 (c) Pose c29 (d) Pose c14

Figure 16: Face recognition accuracy under combined pose and illumination variations on the CMU PIE data set. The proposed methods are denoted as *G-LRT* in color red and *C-LRT* in color blue. The proposed methods significantly outperform the comparing methods, especially for extreme poses c02 and c14.

is data dependent. Last but not least, our method outperforms SMD, which the best of our knowledge, reported the best recognition performance in such experimental setup. However, SMD is an unsupervised method, and the proposed method requires training, still illustrating how a simple learned transform (note that applying it to the data at testing time if virtually free of cost), can significantly improve performance.



(a) Globally transformed testing samples for *subject1* (b) Globally transformed testing samples for *subject2*

Figure 17: Face recognition under combined pose and illumination variations using global low-rank transformation.

5.6 Application to Face Recognition across Illumination and Pose

To enable the comparison with Qiu et al. (Oct. 2012), we adopt their setup for face recognition under combined pose and illumination variations for the CMU PIE data set. We use 68 subjects in 5 poses, c22, c37, c27, c11 and c34, under 21 illumination conditions for training; and classify 68 subjects in 4 poses, c02, c05, c29 and c14, under 21 illumination conditions.

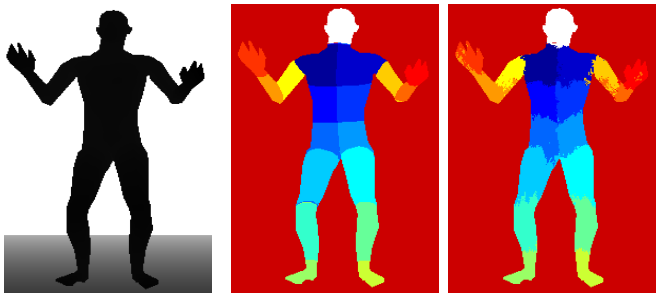
Three face recognition methods are adopted for comparisons: Eigenfaces Turk and Pentland (1991), SRC Wright et al. (2009), and DADL Qiu et al. (Oct. 2012). SRC and DADL are both state-of-the-art sparse representation methods for face recognition, and DADL adapts sparse dictionaries to the actual visual domains. As shown in Figure 16, the proposed methods, both the global LRT (G-LRT) and class-based LRT (C-LRT), significantly outperform the comparing methods, especially for extreme poses c02 and c14. Some testing examples using a global transformation are shown in Figure 17. We notice that the transformed faces for each subject exhibit reduced variations caused by pose and illumination.

5.7 Transformation Forest

In order to further illustrate the power of the framework here proposed, we briefly describe its use in combination with random forests, as discussed in detail in Qiu and Sapiro (2014). In this work we introduced a transformation-based learner model for random forest, further stressing how the proposed transformation learning can be combined with other successful classification techniques beyond subspace techniques. The weak learner at each split node plays a crucial role in a classification tree. We optimized the splitting by learning a two-class transformation $\mathbf{T}$ at each split node, and observed significantly performance improvements in various real-world applications, such as scene classification and 3D pose estimation (Figure 18). In particular, we experimentally demonstrated how learning such transform at each node reduces by 1-2 orders of magnitude the number of trees in the random forest.

5.8 Discussion on the Size of the Transformation Matrix T

In the experiments presented above, we learned a square linear transformation. For example, if images are resized to 16×16, the learned subspace transformation $\mathbf{T}$ is of size 256×256. If we learn a transformation of size $r \times 256$ with $r < 256$, we enable dimension reduction while performing subspace transformation (feature learning). Through experiments, we



(a) Depth. (b) Ground truth. (c) Prediction.

Figure 18: Body parts prediction from a depth image using transformation forests (Qiu and Sapiro, 2014). With the learned transform we classify 20 regions (19 body parts and one background) with 55.5% correct for a single tree (only about 40% with standard trees), and achieve already 73.12% with just 30 trees (hundreds are normally used with standard trees).

notice that the peak clustering accuracy is usually obtained when r is smaller than the dimension of the ambient space. For example, in Figure 13, through exhaustive search for the optimal r, we observe the misclassification rate reduced from 2.38% to 0% for subjects $\{2, 3\}$ at $r = 96$, and from 4.23% to 0% for subjects $\{4, 5, 6\}$ at $r = 40$. As discussed before, this provides a framework to sense for clustering and classification, connecting the work presented here with the extensive literature on compressed sensing, and in particular for sensing design, e.g., Carson et al. (2012). We plan to study in detail the optimal size of the learned transformation matrix for subspace clustering and classification, including its potential connection with the number of subspaces in the data, and further investigate such connections with compressive sensing.

6. Conclusion

We introduced a subspace low-rank transformation approach for subspace clustering and classification. Using nuclear norm as the optimization criteria, we learn a subspace transformation that reduces variations within the subspaces, and increases separations between the subspaces. We demonstrated that the proposed approach significantly outperforms state-of-the-art methods for subspace clustering and classification, and provided some theoretical support to these experimental results.

Numerous venues of research are opened by the framework introduced here. At the theoretical level, extending the analysis to the noisy case is needed. Furthermore, understanding the virtues of the global vs the class-dependent transform is both important and interesting, as it is the study of the framework in its compressed dimensionality form. Beyond this, considering the proposed approach as a feature extraction technique, its combination with other successful clustering and classification techniques is the subject of current research.

Acknowledgments

Work partially supported by ONR, NGA, ARO, AFOSR (NSSEFF), and NSF. We thank Dr. Pablo Sprechmann, Dr. Ehsan Elhamifar, Ching-Hui Chen, and Dr. Mariano Tepper for important feedback on this work. The AE and reviewers did an outstanding job in helping us improve this paper.

Appendix A. Proof of Theorem 1

Proof:

$$||\mathbf{A}||_* + ||\mathbf{B}||_* = |||[\mathbf{A}\ \mathbf{0}]|||_* + |||[\mathbf{0}\ \mathbf{B}]|||_* \geq |||[\mathbf{A}\ \mathbf{0}] + [\mathbf{0}\ \mathbf{B}]|||_* = |||[\mathbf{A}, \mathbf{B}]|||_*$$

■

Appendix B. Proof of Theorem 2

Proof: We perform the singular value decomposition of $\mathbf{A}$ and $\mathbf{B}$ as

$$\mathbf{A} = [\mathbf{U_{A1}}\mathbf{U_{A2}}] \begin{bmatrix} \mathbf{\Sigma_A} & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{V_{A1}}\mathbf{V_{A2}}]', \quad \mathbf{B} = [\mathbf{U_{B1}}\mathbf{U_{B2}}] \begin{bmatrix} \mathbf{\Sigma_B} & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{V_{B1}}\mathbf{V_{B2}}]',$$

where the diagonal entries of $\mathbf{\Sigma_A}$ and $\mathbf{\Sigma_B}$ contain non-zero singular values. We have

$$\mathbf{AA'} = [\mathbf{U_{A1}}\mathbf{U_{A2}}] \begin{bmatrix} \mathbf{\Sigma_A}^2 & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{A2}}]', \quad \mathbf{BB'} = [\mathbf{U_{B1}}\mathbf{U_{B2}}] \begin{bmatrix} \mathbf{\Sigma_B}^2 & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{U_{B1}}\mathbf{U_{B2}}]'.$$

The column spaces of $\mathbf{A}$ and $\mathbf{B}$ are considered to be orthogonal, i.e., $\mathbf{U_{A1}}'\mathbf{U_{B1}} = 0$. The above can be written as

$$\mathbf{AA'} = [\mathbf{U_{A1}}\mathbf{U_{B1}}] \begin{bmatrix} \mathbf{\Sigma_A}^2 & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{B1}}]', \quad \mathbf{BB'} = [\mathbf{U_{A1}}\mathbf{U_{B1}}] \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{\Sigma_B}^2 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{B1}}]'.$$

Then, we have

$$[\mathbf{A}, \mathbf{B}][\mathbf{A}, \mathbf{B}]' = \mathbf{AA'} + \mathbf{BB'} = [\mathbf{U_{A1}}\mathbf{U_{B1}}] \begin{bmatrix} \mathbf{\Sigma_A}^2 & 0 \\ 0 & \mathbf{\Sigma_B}^2 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{B1}}]'.$$

The nuclear norm $||\mathbf{A}||_*$ is the sum of the square root of the singular values of $\mathbf{AA'}$. Thus, $|||[\mathbf{A}, \mathbf{B}]|||_* = ||\mathbf{A}||_* + ||\mathbf{B}||_*$. ■

Appendix C. The Concave-Convex Procedure

We use a simple projected subgradient method to search for the transformation matrix $\mathbf{T}$ that minimizes (6). Before describing it, we should note that the problem is non-differentiable and non-convex, and it deserves in its own right a proper study of efficient optimization techniques, which is of course not the focus of this paper. The development of more advanced optimization techniques will further improve the performance of the proposed framework. We selected a simple subgradient-based approach since the goal of this

paper is to present the framework, and already this simple optimization leads to very fast convergence and excellent performance as detailed in Section 5, with significant improvements in performance when compared to prior state-of-the-art.

The objective function (6) is a D.C. (difference of convex functions) program, and the concave-convex procedure (CCCP) is a majorization-minimization algorithm often adopted to solve D.C. programs as a sequence of convex programs (Yuille and Rangarajan, 2003; Sriperumbudur and Lanckriet, 2012; Dinh and An, 1997). CCCP is used in many machine learning algorithms such as transductive SVMs (Collobert et al., 2006), sparse PCA (Sriperumbudur et al., 2007), and SVM feature selection (Neumann et al., 2005).

Initialize $\mathbf{T}^{(0)}$ with the identity matrix ;
repeat

$$\begin{aligned}\mathbf{T}^{(t+1)} =& \arg\min_{\mathbf{T}} \mathcal{J}_{vex}(\mathbf{T}) + \partial\mathcal{J}_{cav}(\mathbf{T}^{(t)})\mathbf{T} \\ =& \arg\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'\mathbf{T}').\end{aligned} \tag{17}$$

until *convergence or stopping criteria*;

Algorithm 2: The Concave-Convex Procedure (CCCP).

Input: An $m \times n$ matrix $\mathbf{A}$, a small threshold value δ
Output: A subgradient of the nuclear norm $\partial||\mathbf{A}||_*$.
begin

1. Perform singular value decomposition:
$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$;
2. $s \leftarrow$ the number of singular values smaller than δ ,
3. Partition $\mathbf{U}$ and $\mathbf{V}$ as
$\mathbf{U} = [\mathbf{U}^{(1)}, \mathbf{U}^{(2)}]$, $\mathbf{V} = [\mathbf{V}^{(1)}, \mathbf{V}^{(2)}]$;
where $\mathbf{U}^{(1)}$ and $\mathbf{V}^{(1)}$ have $(n-s)$ columns.
4. Generate a random matrix $\mathbf{B}$ of the size $(m-n+s) \times s$,
$\mathbf{B} \leftarrow \frac{\mathbf{B}}{||\mathbf{B}||}$;
5. $\partial||\mathbf{A}||_* \leftarrow \mathbf{U}^{(1)}\mathbf{V}^{(1)'} + \mathbf{U}^{(2)}\mathbf{B}\mathbf{V}^{(2)'}$;
6. Return $\partial||\mathbf{A}||_*$;

end

Algorithm 3: An approach to evaluate a subgradient of matrix nuclear norm.

Our D.C. cost function $\mathcal{J}(\mathbf{T})$ can be rewritten as the sum of a convex part $\mathcal{J}_{vex}(\mathbf{T})$ and a concave part $\mathcal{J}_{cav}(\mathbf{T})$, i.e.,

$$\begin{aligned}\mathcal{J}(\mathbf{T}) =& \mathcal{J}_{vex}(\mathbf{T}) + \mathcal{J}_{cav}(\mathbf{T}) \\ =& [\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_*] + [-||\mathbf{T}\mathbf{Y}||_*].\end{aligned}$$

In each iteration of the CCCP procedure, Algorithm 2, we approximate the concave part using its subgradient $\partial\mathcal{J}_{cav}$, and minimize the resulting convex sub-problem. Note that the

first term in (17) is the convex term in (6), and the added second term is a linear term on $\mathbf{T}$ using a subgradient of the concave term in (6) evaluated at the current iteration. $\partial||\cdot||_*$ is a subgradient of the nuclear norm $||\cdot||_*$, which can be evaluated using the simple approach shown in Algorithm 3 (Watson, 1992). Formal convergence analysis of CCCP for differentiable cases can be found in Yuille and Rangarajan (2003) and Sriperumbudur and Lanckriet (2012). Though the objective function (6) is non-differentiable, we still observe empirical convergence in all experiments, see Figure8 and Figure11n.

We provide here more details about Algorithm 2. During each CCCP iteration, we solve the convex sub-objective (17) using the subgradient method, i.e., using a constant step size ν ($\nu > 0$), we iteratively take a step in the negative direction of subgradient, and the subgradient is evaluated as

$$\sum_{c=1}^{C} \partial||\mathbf{T}\mathbf{Y}_c||_*\mathbf{Y}_c' - \partial||\mathbf{T}^{(\mathbf{t})}\mathbf{Y}||_*\mathbf{Y}'. \tag{18}$$

Using a constant step size, the subgradient method is guaranteed to converge to within some range of the optimal value for a convex problem (convergence to the optimal value is guaranteed by using a diminishing step size with an infinite travel condition) (Boyd et al., 2003). Therefore, given $\mathbf{T}^{(t+1)}$ as the minimizer found for the convex sub-problem (17) using the subgradient method, we have for (17),

$$\begin{aligned}\sum_{c=1}^{C} ||\mathbf{T}^{(t+1)}\mathbf{Y}_c||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'\mathbf{T}^{(t+1)'}) \\ \leq \sum_{c=1}^{C} ||\mathbf{T}^{(t)}\mathbf{Y}_c||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'\mathbf{T}^{(t)'}),\end{aligned} \tag{19}$$

and from the concavity of the second term in (6), we have

$$-||\mathbf{T}^{(t+1)}\mathbf{Y}||_* \leq -||\mathbf{T}^{(t)}\mathbf{Y}||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'(\mathbf{T}^{(t+1)} - \mathbf{T}^{(t)})'). \tag{20}$$

By summing (19) and (20), we obtain

$$\sum_{c=1}^{C} ||\mathbf{T}^{(t+1)}\mathbf{Y}_c||_* - ||\mathbf{T}^{(t+1)}\mathbf{Y}||_* \leq \sum_{c=1}^{C} ||\mathbf{T}^{(t)}\mathbf{Y}_c||_* - ||\mathbf{T}^{(t)}\mathbf{Y}||_*. \tag{21}$$

Thus, the objective (6) is non-increasing after each CCCP iteration, and is bounded from below by 0 (shown in Section 2) for our non-differentiable case. For efficiency considerations, while solving the convex sub-objective function (17), we perform only one iteration of the subgradient method to obtain a simplified method, and still observe empirical convergence in all experiments, as shown in Figure8 and Figure11n.

The norm constraint $||\mathbf{T}||_2 = 1$ is adopted in our formulation to prevent the trivial solution $\mathbf{T} = 0$. By initializing $\mathbf{T}^{(0)}$ with the identity matrix, we observed no trivial solution convergence in all experiments, such as the normalization free case in Figure8.

As shown in Douglas et al. (2000), the norm constraint $||\mathbf{T}||_2 = 1$ can be incorporated to a gradient-based algorithm using various alternatives, e.g., Lagrange multipliers, coefficient

normalization, and gradients in the tangent space. We implement the coefficient normalization method, i.e., after obtaining $\mathbf{T}^{(t+1)}$ from (17), we normalize $\mathbf{T}^{(t+1)}$ via $\frac{\mathbf{T}^{(t+1)}}{||\mathbf{T}^{(t+1)}||}$. In other words, we normalize the length of $\mathbf{T}^{(t+1)}$ without changing its direction. As discussed in Douglas et al. (2000), the problem of minimizing a cost function subject to a norm constraint forms the basis for many important tasks, and gradient-based algorithms are often used along with the norm constraint. Though it is expected that a norm constraint does not change the convergence behavior of a gradient algorithm (Douglas et al., 2000; Fuhrmann and Liu, 1984), Figure8, to the best of our knowledge, a formal analysis of these issues is still missing.

References

R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(2):218–233, 2003.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

S. Boyd, L. Xiao, and A. Mutapcic. Subgradient method. *Notes for EE392o, Stanford University*, 2003.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

W. R. Carson, M. Chen, M. R. D. Rodrigues, R. Calderbank, and L. Carin. Communications-inspired projection design with application to compressive sensing. *SIAM J. Imaging Sci.*, 5(4):1185–1212, 2012.

C. Castillo and D. Jacobs. Using stereo matching for 2-D face recognition across pose. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31:2298–2304, 2009.

G. Chen and G. Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, 2009.

R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *J. Mach. Learn. Res.*, 7:1687–1712, December 2006.

T. P. Dinh and L. T. H. An. Convex analysis approach to d.c. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289355, 1997.

S. C. Douglas, S. Amari, and S. Y. Kung. On gradient adaptation with unit-norm constraints. *IEEE Trans. on Signal Processing*, 48(6):1843–1847, 2000.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 2013. To appear.

M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

D. R. Fuhrmann and B. Liu. An iterative algorithm for locating the minimal eigenvector of a symmetric matrix. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dallas, TX, 1984.

A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(6):643–660, June 2001.

A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Minneapolis, Minnesota, 2007.

T. Hastie and P. Y. Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 13(1):54–65, 1998.

Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Colorado springs, CO, 2011.

O. Kuybeda, G. A. Frank, A. Bartesaghi, M. Borgnia, S. Subramaniam, and G. Sapiro. A collaborative framework for 3D alignment and classification of heterogeneous subvolumes in cryo-electron tomography. *Journal of Structural Biology*, 181:116–127, 2013.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, Haifa, Israel, 2010.

U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 29 (9):1546–1562, 2007.

G. Marsaglia and G. P. H. Styan. When does rank (a + b) = rank(a)+rank(b)? *Canad. Math. Bull.*, 15(3), 1972.

J. Miao and A. Ben-Israel. On principal angles between subspaces in R_n. *Linear Algebra and its Applications*, 171(0):81 – 98, 1992.

J. Neumann, C. Schnörr, and G. Steidl. Combined SVM-based feature selection and classification. *Mach. Learn.*, 61(1-3):129–150, November 2005.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, Nov. 1993.

Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, USA, 2010.

Q. Qiu and G. Sapiro. Learning transformations for classification forests. In *International Conference on Learning Representations*, Banff, Canada, 2014.

Q. Qiu, V. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *Proc. European Conference on Computer Vision*, Florence, Italy, Oct. 2012.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

L. K. Saul and S. T. Roweis. An introduction to locally linear embedding. 2000. URL `http://www.cs.nyu.edu/~roweis/lle/publications.html`.

X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Rhode Island, USA, 2012.

T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(12):1615 –1618, Dec. 2003.

M. Soltanolkotabi and E. J. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. Robust subspace clustering. *CoRR*, abs/1301.2603, 2013. URL `http://arxiv.org/abs/1301.2603`.

P. Sprechmann, A. M. Bronstein, and G. Sapiro. Learning efficient sparse and low rank models. *CoRR*, abs/1212.3631, 2012. URL `http://arxiv.org/abs/1212.3631`.

B. K. Sriperumbudur and G. R. G. Lanckriet. A proof of convergence of the concave-convex procedure using zangwill's theory. *Neural Computation*, 24(6):1391–1407, 2012.

B. K. Sriperumbudur, D. A. Torres, and G. R. G. Lanckriet. Sparse eigen methods by d.c. programming. In *International Conference on Machine Learning*, 2007.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Maui, Hawaii, 1991.

R. Vidal. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, 2011.

R. Vidal, Yi Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Madison, Wisconsin, 2003.

J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, USA, 2010.

Y. Wang and H. Xu. Noisy sparse subspace clustering. In *International Conference on Machine Learning*, Atlanta, USA, 2013.

G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and Applications*, 170:1039–1053, 1992.

J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31(2):210–227, 2009.

J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proc. European Conference on Computer Vision*, Graz, Austria, 2006.

A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 4: 915–936, 2003.

Q. Zhang and B. Li. Discriminative k-SVD for dictionary learning in face recognition. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, CA, 2010.

T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012.

Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. TILT: transform invariant low-rank textures. In *Proc. Asian conference on Computer vision*, Queenstown, New Zealand, 2011.

X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Providence, Rhode Island, 2012.