Domain Adaptive Subspace Clustering

Mahdi Abavisani
mahdi.abavisani@rutgers.edu
Vishal M. Patel
http://www.rci.rutgers.edu/~vmp93

Electrical and Computer Engineering Department
Rutgers, The State University of New Jersey
New Brunswick, NJ, USA

Abstract

We propose domain adaptive extensions of the recently introduced sparse subspace clustering and low-rank representation-based subspace clustering algorithms for clustering data lying in a union of subspaces. We propose a general method that learns the projections of data in a space where the sparsity or low-rankness of data is maintained. We propose an efficient iterative procedure for solving the proposed optimization problems. Various experiments on face, object and handwritten digits datasets show that the proposed methods can perform better than many competitive subspace clustering methods.

1 Introduction

Many practical applications in image processing and computer vision require one to analyze and process high-dimensional data. It has been observed that these high-dimensional data can be represented by a low-dimensional subspace. For instance, it is well known in computer vision and graphics that a set of face images under all possible illumination conditions can be well approximated by a 9-dimensional linear subspace [1]. Similarly, handwritten digits with different variations as well as trajectories of a rigidly moving object in a video can be represented by low-dimensional subspaces [3, 11]. As a result, the collection of data from different classes can be viewed as samples from a union of low-dimensional subspaces. In subspace clustering, given the data from a union of subspaces, the objective is to find the number of subspaces, their dimensions, the segmentation of the data and a basis for each subspace [24].

Various subspace clustering algorithms have been developed in the literature [24]. These methods can be categorized into four main groups - algebraic methods [2, 25], iterative methods [12, 31], statistical methods [5, 20, 29], and the methods based on spectral clustering [7, 8, 9, 17, 28]. In particular, methods based on sparse and low-rank representations have gained a lot of attraction in recent years [7, 8, 17]. These methods find a sparse or low-rank representation of the data and build a similarity graph whose weights depend on the sparse or low-rank coefficient matrix for segmenting the data. Some of these approaches do not require the knowledge of the dimensions and the number of subspaces and produce state-of-the-art results on many publicly available datasets such as the Hopkins155 benchmark motion segmentation dataset [23] and Extended YaleB face dataset [16].

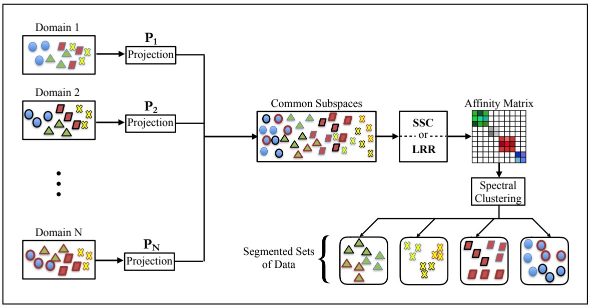


Figure 1: An overview of the proposed domain adaptive subspace clustering framework.

In many applications, one has to deal with heterogeneous [1] data. For example, when clustering digits, one may have to process both computer generated as well as handwritten digits. Similarly, when clustering face images collected in the wild, one may have to cluster images of the same individual collected using different cameras and possibly under different resolution and lighting conditions. Clustering of heterogeneous data is difficult because it is not meaningful to directly compare the heterogeneous samples with different distributions which may span different feature spaces. In recent years, various domain adaptation methods have been developed to deal with the distributional changes that occur after learning a classifier for supervised and semi-supervised learning [19]. However, to the best of our knowledge, these methods have not been developed for clustering heterogeneous data that lie in a union of low-dimensional subspaces.

In this paper, we present domain adaptive versions of the sparse and low-rank subspace clustering methods. Figure 1 gives an overview of the proposed method. Given data from K different domains, we simultaneously learn the projections and find the sparse or low-rank representation in the projected common subspace. Once the projection matrices and the sparse or low-rank coefficient matrix is found, it can be used for subspace clustering.

This paper is organized as follows. Section 2 gives a brief background on sparse and low-rank representation-based subspace clustering. Details of the proposed domain adaptive subspace clustering methods are given in Section 3. Optimization procedure for solving the proposed problems is described in Section 4. Experimental results are presented in Section 5. Finally, Section 6 concludes the paper with a brief summary and discussion.

2 Background

In this section, we give a brief background on sparse and low-rank subspace clustering. Let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{D \times N}$ be a collection of N signals $\{\mathbf{y}_i \in \mathbb{R}^D\}_{i=1}^N$ drawn from a union of n linear subspaces $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$ of dimensions $\{d_\ell\}_{\ell=1}^n$ in $\mathbb{R}^D$. Let $\mathbf{Y}_\ell \in \mathbb{R}^{D \times N_\ell}$ be a sub-matrix of $\mathbf{Y}$ of rank d_ℓ with $N_\ell > d_\ell$ points that lie in $\mathcal{S}_\ell$ with $N_1 + N_2 + \cdots + N_n = N$. Given $\mathbf{Y}$, the task of subspace clustering is to cluster the signals according to their subspaces.

Sparse Subspace Clustering (SSC). The SSC algorithm [7], which exploits the fact that noiseless data in a union of subspaces are *self-expressive*, i.e. each data point can be expressed as a *sparse* linear combination of other data points. Hence, SSC aims to find a

[1] Data with different sizes

sparse matrix $\mathbf{C}$ such that $\mathbf{Y} = \mathbf{YC}$ and $\mathrm{diag}(\mathbf{C}) = 0$, where the constraint prevents the trivial solution $\mathbf{C} = \mathbf{I}$. Since this problem is combinatorial and to deal with the presence of noise, SSC solves the following optimization problem instead

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 + \frac{\tau}{2}\|\mathbf{Y} - \mathbf{YC}\|_F^2, \quad \text{s. t. } \mathrm{diag}(\mathbf{C}) = \mathbf{0}, \tag{1}$$

where $\|\mathbf{C}\|_1 = \sum_{i,j} |C_{i,j}|$ is the ℓ_1-norm of $\mathbf{C}$ and $\tau > 0$ is a parameter.

Low-Rank Representation-Based Subspace Clustering (LRR). The LRR algorithm [17] for subspace clustering is very similar to the SSC algorithm except that a low-rank representation is found instead of a sparse representation. In particular, the following problem is solved

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* + \frac{\tau}{2}\|\mathbf{Y} - \mathbf{YC}\|_F^2, \tag{2}$$

where $\|\mathbf{C}\|_*$ is the nuclear-norm of $\mathbf{C}$ which is defined as the sum of its singular values. In SSC and LRR, once $\mathbf{C}$ is found, spectral clustering methods [18] are applied on the affinity matrix $|\mathbf{C}| + |\mathbf{C}|^T$ to obtain the segmentation of the data $\mathbf{Y}$.

3 Domain Adaptive Subspace Clustering

Suppose that we are given N_s samples, $\{\mathbf{y}_i^{d_s}\}_{i=1}^{N_s}$, from domain D_s, and N_t samples, $\{\mathbf{y}_i^{d_t}\}_{i=1}^{N_t}$, from domain D_t. Assuming that each sample from domain D_s has the dimension of M_s, let $\mathbf{Y}_s = [\mathbf{y}_1^{d_s}, ..., \mathbf{y}_{N_s}^{d_s}] \in \mathbb{R}^{M_s \times N_s}$ denote the matrix of samples from domain D_s. Similarly, let $\mathbf{Y}_t \in \mathbb{R}^{M_t \times N_t}$ denote the matrix containing N_t samples each of dimension M_t from domain D_t. Note that the dimensions of features in D_s and D_t are not required to be the same, i.e., $M_s \neq M_t$. The task of domain adaptive subspace clustering is to cluster the data according to their original subspaces even though they might lie in different domains.

Domain Adaptive Sparse Subspace Clustering (DA-SSC). Let $\mathbf{P}_s \in \mathbb{R}^{m \times N_s}$ and $\mathbf{P}_t \in \mathbb{R}^{m \times N_t}$ be mappings represented as matrices that project the data from D_s and D_t to a latent m-dimensional space, respectively. As a result, $\mathbf{P}_s\mathbf{Y}_s$ and $\mathbf{P}_t\mathbf{Y}_t$ lie on an m-dimensional space. Let $\mathbf{G} = [\mathbf{P}_s\mathbf{Y}_s, \mathbf{P}_t\mathbf{Y}_t] = [\mathbf{g}_1, \cdots, \mathbf{g}_{N_s+N_t}] \in \mathbb{R}^{m \times (N_s+N_t)}$ denote the concatenation of the projected samples in the m-dimensional space from both source and target domains. The proposed method takes advantage of the *self-expressiveness property* of the data in the low-dimensional space as discussed in the previous section. That is, we want to write each sample as a sparse linear combination of the other samples in the projected space. Assuming the presence of noise in the projected samples, the sparse coefficients can be found by solving the following optimization problem

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 + \frac{\tau}{2}\|\mathbf{G} - \mathbf{GC}\|_F^2, \quad \text{s. t. } \mathrm{diag}(\mathbf{C}) = \mathbf{0}, \tag{3}$$

where the ith column of $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{N_s+N_t}] \in \mathbb{R}^{N_s+N_t \times N_s+N_t}$ is the sparse coefficient for $\mathbf{g}_i$ and $\mathrm{diag}(\mathbf{C})$ is the vector of the diagonal elements of $\mathbf{C}$. We propose to learn projections $\mathbf{P}_s$ and $\mathbf{P}_t$ and the sparse coefficient matrix $\mathbf{C}$ simultaneously by solving the following optimization problem

$$\min_{\mathbf{P},\mathbf{C}} \|\mathbf{C}\|_1 + \frac{\tau}{2}\|\mathbf{PY}-\mathbf{PYC}\|_F^2 \text{ s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0},\ \mathbf{P}_s\mathbf{P}_s^T = \mathbf{P}_t\mathbf{P}_t^T = \mathbf{I}, \tag{4}$$

where $\tau > 0$ is a parameter and

$$\mathbf{P} = [\mathbf{P}_s, \mathbf{P}_t], \text{ and } \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_s & \mathbf{0}_{N_s \times N_t} \\ \mathbf{0}_{N_t \times N_s} & \mathbf{Y}_t \end{bmatrix}. \tag{5}$$

The constrain, $\mathbf{P}_s\mathbf{P}_s^T = \mathbf{P}_t\mathbf{P}_t^T = \mathbf{I}$, is added to avoid degenerate solutions.

Domain Adaptive Low-Rank Representation-based Subspace Clustering (DA-LRR). Similar to the DA-SSC method, once the data are projected onto the latent space, rather than finding sparse representation, we seek the lowest rank representation. In particular, the following optimization is proposed for obtaining the domain adaptive low-rank representation

$$\min_{\mathbf{P},\mathbf{C}} \|\mathbf{C}\|_* + \frac{\tau}{2}\|\mathbf{PY}-\mathbf{PYC}\|_F^2 \text{ s.t. } \mathbf{P}_s\mathbf{P}_s^T = \mathbf{P}_t\mathbf{P}_t^T = \mathbf{I}. \tag{6}$$

Note in the case when there is significant distributional change between the source and the target data, the proposed methods tend to produce over segmentation. That is, even though data from the same class should be segmented into one cluster, they are segmented into two clusters - one corresponding to the target domain and the other corresponding to the source domain. To avoid this, we force $\mathbf{C}$ to pick samples from both domains for each class. To this end, we modify (4) and (6) by solving two separate problems - one for intra-domain coefficients and the other for inter-domain coefficients. From the formulation of (5), one can see that $\mathbf{C}$ consists of four blocks $\mathbf{C}_{11}, \mathbf{C}_{12}, \mathbf{C}_{21}$ and $\mathbf{C}_{22}$ as follows

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} = \mathbf{C}_1 + \mathbf{C}_2, \text{ with } \mathbf{C}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{0} \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix}, \tag{7}$$

where $\mathbf{C}_{11} \in \mathbb{R}^{N_s \times N_s}$ and $\mathbf{C}_{22} \in \mathbb{R}^{N_t \times N_t}$ correspond to the inter-domain similarity between samples, while $\mathbf{C}_{12} \in \mathbb{R}^{N_s \times N_t}$ and $\mathbf{C}_{21} \in \mathbb{R}^{N_t \times N_s}$ are responsible for connecting samples between the two domains. Thus, by replacing $\mathbf{C}$ in (4) with $\mathbf{C}_1$ [2] and $\mathbf{C}_2$, matrices for selecting intra-domain and inter-domain parts of $\mathbf{C}$, we can solve the problem in two steps - one for $\mathbf{C}_1$ and the other for $\mathbf{C}_2$. Once the coefficient matrix $\mathbf{C}$ is found by adding the estimated $\mathbf{C}_1$ and $\mathbf{C}_2$ matrices, spectral clustering can be applied on the affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ to obtain the segmentation of the heterogeneous data. The proposed domain adaptive sparse and low-rank subspace clustering methods are summarized in Algorithm 1.

Multiple Domains. The above formulations can be extended from two domains to multiple domains. For K domain problem, we have data $\{\mathbf{Y}_i \in \mathbb{R}^{M_i \times N_i}\}_{i=1}^K$ from K different domains $\{D_i\}_{i=1}^K$. By simply constructing $\mathbf{P}$ and $\mathbf{Y}$ as

$$\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_K] \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_2 & & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \mathbf{Y}_K \end{bmatrix}, \tag{8}$$

where $\{\mathbf{P}_i \in \mathbb{R}^{m \times N_i}\}_{i=1}^K$ are the projection matrices that map the data from their corresponding domains to an m-dimensional latent space, one can extend (4) and (6) to multiple domains

[2] Intra-domain matrix can mathematically be written as: $\mathbf{C}_1 = \begin{bmatrix} \mathbf{I}_{N_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \times \mathbf{C} \times \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N_t} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N_t} \end{bmatrix} \times \mathbf{C} \times \begin{bmatrix} \mathbf{I}_{N_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$

Algorithm 1: The DA-SSC and DA-LRR Algorithms.

```
Input: $\mathbf{Y}_s, \mathbf{Y}_t, \tau$
0: Initialize $\mathbf{P}$; $\mathbf{Y} = [\mathbf{Y}_s, \mathbf{0}; \mathbf{0}, \mathbf{Y}_t]$.
1a: DA-SSC: Find $\mathbf{C}$ by solving the DA-SSC optimization problem (4).
1b: DA-LRR: Find $\mathbf{C}$ by solving the DA-LRR optimization problem (6).
2: Normalize the columns of $\mathbf{C}$ as $\mathbf{c}_i \leftarrow \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_\infty}$.
3: Form a similarity graph with $N$ nodes and set the weights on the edges between the
   nodes by $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}^T|$.
4: Apply spectral clustering to the similarity graph.
Output: Segmented data: $\{\mathbf{Y}_D^l\}_{l=1,\ldots,n,\, D=\{s,t\}}$.
```

with the constraints that $\mathbf{P}_i\mathbf{P}_i^T = \mathbf{I}$. Similarly, the intra-domain and inter-domain coefficient matrices can be defined as

$$\mathbf{C}_{Inter} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} & & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \mathbf{C}_{nn} \end{bmatrix}, \mathbf{C}_{Intra} = \begin{bmatrix} \mathbf{0} & \mathbf{C}_{12} & \ldots & \mathbf{C}_{1n} \\ \mathbf{C}_{21} & \mathbf{0} & & \mathbf{C}_{2n} \\ \vdots & & \ddots & \\ \mathbf{C}_{n1} & \mathbf{C}_{n2} & & \mathbf{0} \end{bmatrix}. \tag{9}$$

4 Optimization

We solve the optimization problems (4) and (6) by optimizing over $\mathbf{P}$ and $\mathbf{C}$, iteratively. Note that although problems (4) and (6) are non-convex, numerical results show that they typically converge to a local minimum in a few iterations.

Update step for C. Fixing $\mathbf{P}$ turns (4) and (6) into SSC and LRR problems, respectively as follows

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 + \frac{\tau}{2}\|\mathbf{G} - \mathbf{G}\mathbf{C}\|_F^2 \text{ s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0}, \text{and} \tag{10}$$

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* + \frac{\tau}{2}\|\mathbf{G} - \mathbf{G}\mathbf{C}\|_F^2, \tag{11}$$

where $\mathbf{G}$ is the projected data. These problems can be efficiently solved using the the alternating direction method of multipliers (ADMM) [7], [17].

Update step for P. For a fixed $\mathbf{C}$, we can rewrite (4) and (6) as

$$\min_{\mathbf{P}} \|\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{C}\|_F^2 \text{ s.t. } \mathbf{P}_s\mathbf{P}_s^T = \mathbf{P}_t\mathbf{P}_t^T = \mathbf{I}, \tag{12}$$

which can be simplified as

$$\min_{\mathbf{P}} \operatorname{Trace}\left(\mathbf{P}[\mathbf{Y}\mathbf{Y}^T - \mathbf{Y}\mathbf{C}^T\mathbf{Y}^T - \mathbf{Y}\mathbf{C}\mathbf{Y}^T + \mathbf{Y}\mathbf{C}\mathbf{C}^T\mathbf{Y}^T]\mathbf{P}^T\right) \text{ s.t. } \mathbf{P}_s\mathbf{P}_s^T = \mathbf{P}_t\mathbf{P}_t^T = \mathbf{I}. \tag{13}$$

This problem is not a convex problem because of the orthonormality constraints on $\mathbf{P}_i$. Specifically, it involves optimization on Stiefel manifold, hence, we solve it using the manifold optimization technique described in [27].

5 Experimental Results

We evaluate the performance of our domain adaptive subspace clustering methods on three publicly available datasets - UMD-AA01 face dataset [30], Amazon/DLSR/Webcam dataset [21] and USPS/MNIST/Alphadigits handwritten digits datasets [13, 15]. Sample images from these datasets are shown in Figure 2. We compare the performance of our methods with that of several recent domain adaptation methods including frustratingly easy domain adaptation (ED) method [4], Correlation Alignment (CORAL) [22], and a Grassmann manifold (GM) based method [10]. Note that these methods were not necessarily developed for domain adaptive subspace clustering but we first use them to extract the domain adaptive features and then simply feed them into the SSC and LRR algorithms. We denote these methods as EA-SSC/EA-LRR, CO-SSC/CO-LRR, and GM-SSC/GM-LRR. We also compare the performance of our method with that of the traditional SSC and LRR methods where we simply apply these algorithms on the original data without any domain adaptive normalization. These methods essentially serve as baseline for comparisons.

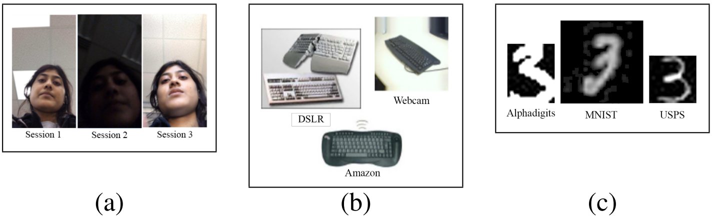


Figure 2: Samples images from (a) UMD dataset [30], (b) Amazon/DLSR/Webcam dataset [21], and (c) USPS/MNIST/Alphadigit datasets [13, 15]. One can clearly see the domain changes among the samples in these datasets.

Note that the CORAL method [22] requires the source and target domains to have the same number of samples. Thus, in order to make this method capable of handling the tests, the same number of samples as the number of target samples are randomly chosen. The colored target samples along with the source samples are given to the SSC and LRR methods. Regarding the Grassmann manifold method, as suggested in the original paper [10], 10 sample points are selected in the geodesic path between the source and the target domains. Samples corresponding to all of these 10 subspaces are concatenated to form the domain invariant features. These features are then fed into the SSC and LRR algorithms. For the EA-SSC/EA-LRR methods, we first map the data according to the mapping introduced in [4] and feed the resulting mapped data into SSC and LRR for clustering. The regulation parameters in our methods are selected by cross validation. Parameters for the other domain adaptation methods were optimized according to the discussion provided in the corresponding papers. Subspace clustering error is used to measure the performance of different algorithms. It is defined as

$$\text{Subspace clustering error} = \frac{\text{\# of misclassified points}}{\text{total \# of points}} \times 100.$$

Face Clustering. The UMD-AA01 dataset is collected on mobile devices for the original purpose of active authentication, but as it contains various ambient conditions, we use it for experiments in this paper. This dataset contains facial images of 50 users over 3 sessions corresponding to different illumination conditions. In each session more than 750 sample

	Method	$\{1\} \rightarrow \{2\}$	$\{1\} \rightarrow \{3\}$	$\{2\} \rightarrow \{1\}$	$\{2\} \rightarrow \{3\}$	$\{3\} \rightarrow \{1\}$	$\{3\} \rightarrow \{2\}$	Avg. ± std.
50-subjects	SSC	56.93	57.14	58.16	57.14	54.69	54.08	56.36 ± 1.60
	CO-SSC	55.91	57.14	58.16	57.75	**52.65**	53.87	55.91 ± 2.22
	DA-SSC	**52.86**	**54.29**	**55.71**	**57.55**	53.67	**50.81**	**54.15** ± 2.33
	ED-SSC	57.75	58.75	59.59	60.61	54.08	53.67	57.40 ± 2.90
	GM-SSC	54.69	54.69	59.39	58.57	58.78	57.76	57.31 ± 2.09
	LRR	52.04	48.57	53.26	56.53	44.28	43.26	49.66 ± 5.23
	CO-LRR	46.73	47.35	47.96	52.45	54.49	53.27	50.37 ± 3.4
	DA-LRR	**36.76**	**36.12**	**35.51**	**34.69**	**37.55**	**36.12**	**36.13** ± 0.99
	ED-LRR	42.45	44.29	42.04	49.39	41.43	42.45	43.67 ± 2.96
	GM-LRR	47.76	45.10	47.14	37.96	47.96	49.59	45.92 ± 4.16

Table 1: Average clustering errors on the UMD-AA01 face dataset. The top performing method in each experiment is shown in boldface. Note that $\{1\},\{2\}$ and $\{3\}$ correspond to session 1, session 2 and session 3, respectively.

images are taken from each face. Some of the sample images from this dataset are shown in Figure 2 (a), where one can clearly see the differences in the ambient lighting conditions. Following the standard experimental protocol for testing domain adaptation methods [19], we randomly select seven samples per class from the source domain and three samples per class from the target domain. We repeat this process 10 times and report the average clustering errors. After extracting face regions, we normalize the images using the method introduced in [26]. Then, the layer “fc7" features from the Alexnet convolutional neural network [14] are extracted from each image. These features are then used for testing the performance of different domain adaptive subspace clustering methods.

Table 1 compares the performance of our method with that of different subspace clustering methods on this dataset. Here, $\{a\} \rightarrow \{b\}$ means that data in domain a is used as the source domain data in b is used as the target domain data. First, it is apparent from this table that compared to the SSC-based methods, LRR-based methods generally perform better. An explanation for this could that since most of the images are somewhat aligned, the resulting coefficient matrix $\mathbf{C}$ is more low-rank than sparse. Second, it can be seen from the table that our domain adaptive SSC and LRR methods perform significantly better than original SSC and LRR methods. The table also reveals that the CORAL method, Easy Domain adaptation method and Grassmann manifold-based method can improve the performance of the original SSC and LRR methods. However, these methods on average do not provide the improvements which our methods bring. Also, since these methods find domain invariant features, one can improve the performance of our method by using these features in our method as the input features.

Object Clustering. For clustering objects in different domains, we use the Amazon, DLSR, Webcam dataset introduced in [21]. The dataset consists of 31 objects and the images are from the following three sources: Amazon (consumer images from online merchant sites), DSLR (images by DSLR camera) and Webcam (low quality images from webcams). Figure 2 (b) shows sample images from these datasets, and clearly highlights the differences between the domains. This dataset is more challenging as the images in this dataset contain various illumination, resolution and pose variations. We used the DeCAF features provided by [6] for this dataset [3]. As before, we sample seven samples per class from the source do-

[3] These features are available online at:
`https://www.eecs.berkeley.edu/~jhoffman/domainadapt/`

	Method	$a \rightarrow d$	$a \rightarrow w$	$d \rightarrow a$	$d \rightarrow w$	$w \rightarrow a$	$w \rightarrow d$	Avg. ± std.
31-subjects	SSC	59.67	57.41	56.45	53.54	52.90	47.74	54.61 ± 4.20
	CO-SSC	55.16	54.51	55.80	51.93	42.90	48.38	51.44 ± 4.99
	DA-SSC	**54.19**	**53.55**	**50.00**	**51.29**	**40.65**	**46.77**	**49.41** ± 5.05
	ED-SSC	60.00	59.67	51.22	52.90	45.16	50.96	53.32 ± 5.69
	GM-SSC	58.06	58.39	55.81	58.06	44.52	48.71	53.92 ± 5.89
	LRR	40.32	**30.96**	41.61	36.12	36.77	28.70	35.75 ± 5.08
	CO-LRR	46.73	47.35	47.96	52.45	54.49	53.27	50.37 ±3.40
	DA-LRR	**37.09**	32.25	**41.29**	**35.48**	**36.45**	**27.74**	**35.05** ± 4.62
	ED-LRR	40.00	32.58	49.03	40.32	46.12	43.54	41.93 ± 5.73
	GM-LRR	44.90	45.31	48.98	52.86	49.80	46.53	48.06 ± 3.06

Table 2: Average clustering errors on the Amazon/DSLR/Webcam dataset. The top performing method in each experiment is shown in boldface. Note that $\{a\}, \{d\}$ and $\{w\}$ correspond to Amazon, DSLR and Webcam datasets, respectively.

main and three samples per class from the target domain. We repeat this process 10 times and report the average clustering errors in Table 5. As can be seen from this table, our LRR-based methods perform better than the SSC-based methods. The CORAL-based methods, easy domain adaptation-based methods and Grassmann-based methods provide some improvements to original SSC and LRR, but the proposed methods outperform them in terms of clustering error.

Heterogeneous and Multi-Domain Clustering of Digits. In the final set of experiments, we use three publicly available handwritten digits datasets - USPS [13], MNIST [15] and Alphadigits[4] for conducting heterogeneous as well as multi-domain adaptation experiments. There exist 1100, 16×16 images in the USPS dataset, $7K$, 28×28 images in the MNIST dataset, and 39 binary 20×16 images in the Alphadigits dataset for each digit. Figure 2 (c) shows sample images from these datasets. For heterogeneous domain adaptation experiments, we follow the same protocol as defined before. For multi-source domain adaptation, we follow a similar protocol but now we have more than one domains in the source domain and a single target domain. We sample 39 images per class from each of the source domains, and 19 samples per class from the target domain. We repeat this procedure 10 times and report average clustering errors. Note that the ED-based methods, the Grassmann manifold-based methods and the classical SSC and LRR methods require the data in the source and target domains to be of same dimension. Thus, we resize the larger size images of Alphadigits and MNIST datasets to 16×16 so that it matches with size of images in the USPS dataset. The results obtained by different methods in various combinations of source/target pairs are summarized in Table 3.

As before, it can be seen from this table that the LRR-based methods outperform the SSC-based methods. Another interesting finding from this table is that GM-LRR method performs comparably to DA-LRR, especially in the multi-source domain adaptation cases. However, in multi-source domain adaptation, GM-based methods sample the geodesic path 20 times. As a result the final concatenated feature is extremely high-dimensional which makes the processing of multi-source data inefficient and expensive. Also, from this table, one can see that there is a larger gap between our method and the other methods compared to the previous experiments. This is mainly due to the fact that in previous experiments we

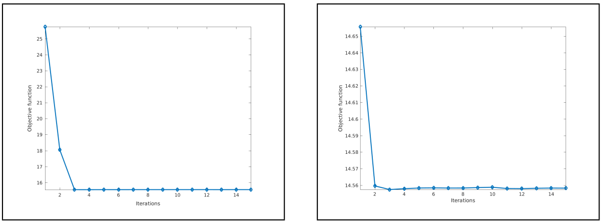
[4] Available at http://www.cs.toronto.edu/~roweis/data.html

	$U \to M$	$U \to A$	$M \to U$	$M \to A$	$A \to U$	$A \to M$	$U,M \to A$	$U,A \to M$	$M,A \to U$	Avg. ± std.
SSC	49.65	45.86	62.34	61.55	59.48	50.51	67.32	68.35	68.55	59.29 ± 16.59
CO-SSC	58.62	48.79	60.86	62.76	**53.79**	52.24	63.12	63.12	68.05	59.04 ± 14.20
DA-SSC	**43.62**	**44.82**	**54.83**	**57.59**	59.48	**46.55**	**54.83**	**61.68**	**67.92**	**54.59** ± 19.73
ED-SSC	50.69	51.9	58.79	62.24	55.86	52.07	62.08	62.08	70.39	58.46 ± 21.51
GM-SSC	51.55	53.28	63.28	59.83	57.32	52.76	64.03	67.14	69.35	59.84 ± 19.08
LRR	23.28	30.17	27.76	18.28	29.31	23.28	27.27	19.74	21.3	24.49 ± 13.02
CO-LRR	18.45	22.59	29.31	23.97	26.72	31.21	25.84	23.25	22.73	24.90 ± 20.56
DA-LRR	**14.14**	**17.28**	18.16	**13.78**	**11.90**	**22.59**	20.13	**14.65**	**20.78**	**17.05** ± 24.43
ED-LRR	28.62	29.66	**17.93**	20.86	26.9	37.24	27.4	19.48	23.51	25.73 ± 24.66
GM-LRR	19.83	18.1	23.97	25.69	17.07	24.14	**19.48**	15.37	22.21	20.65 ± 3.54

Table 3: Subspace clustering performance of different methods for on the handwritten digits datasets. Note that $\{U\},\{M\}$ and $\{A\}$ correspond to USPS, MNIST and Alphadigits datasets, respectively.

used deep features while in this experiment we use pixel intensities as features. It has been shown that deep features are less dependent on different domains [6].

Convergence. As discussed earlier, our method is non-convex and often converges to a local minima in a few iterations. To empirically show the convergence of our methods, in Figure 3 (a) and (b), we show the objective function vs iteration plots for solving (4) and (6), respectively in the case of single-source digits clustering experiment. As can be seen from these figures, our algorithms do converge in a few iterations.



(a) (b)

Figure 3: Objective function versus number of iterations of the proposed optimization problems. (a) Convergence plot corresponding to the DA-SSC problem (4). (b) Convergence plot corresponding to the DA-LRR problem (6).

6 Conclusion

We introduced domain adaptive extensions of the classical SSC and LRR methods for subspace clustering. The proposed DA-SSC and DA-LRR algorithms are applicable to single-source domain, multi-source and heterogeneous domain adaptive clustering problems. We proposed an iterative method for solving the proposed optimization problems. Extensive experiments on face, object and digit clustering showed that the proposed methods can perform better than many state-of-the-art domain adaptive subspace clustering methods.

Acknowledgment

This work was supported by the NSF grant 1618677.

References

[1] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[2] Terrance E Boult and Lisa Gottesfeld Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, pages 179–186, 1991.

[3] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[4] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[5] Harm Derksen, Yi Ma, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy coding and compression. In *SPIE Visual Communications and Image Processing*, volume 6508, 2007.

[6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[7] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2765–2781, 2013.

[8] Paolo Favaro, René Vidal, and Avinash Ravichandran. A closed form solution to robust subspace estimation and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[9] Alvina Goh and René Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[10] Raghavan Gopalan, Ruonan Li, and Rama Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2288–2302, 2014.

[11] Trevor Hastie and Patrice Y. Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 13(1):54–65, 1998.

[12] Jason Ho, Ming-Hsuan Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[13] Jonathan J Hull. A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):550–554, 1994.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.

[15] Yann LeCun and Corinna Cortes. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[16] Kuang Chih Lee, Jeffrey Ho, and David J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, May 2005. ISSN 0162-8828.

[17] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[18] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems*, volume 2, pages 849–856, 2002.

[19] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *Signal Processing Magazine, IEEE*, 32(3): 53–69, 2015.

[20] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

[21] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010*, pages 213–226. Springer, 2010.

[22] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2016.

[23] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[24] René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011. ISSN 1053-5888. doi: 10.1109/MSP.2010.939739.

[25] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.

[26] Haitao Wang, Stan Z Li, and Yangsheng Wang. Face recognition under varying lighting conditions using self quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 819–824, 2004.

[27] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[28] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision*, 2006.

[29] Allen Y Yang, Shankar R Rao, and Yi Ma. Robust statistical estimation and segmentation of multiple subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–99, 2006.

[30] Heng Zhang, Vishal M Patel, Sumit Shekhar, and Rama Chellappa. Domain adaptive sparse representation-based classification. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015.

[31] Teng Zhang, Arthur Szlam, and Gilad Lerman. Median k-flats for hybrid linear modeling with many outliers. In *Workshop on Subspace Methods*, 2009.