Structure-Preserved Unsupervised Domain Adaptation

Hongfu Liu, *Student Member, IEEE*, Ming Shao, *Member, IEEE*,
Zhengming Ding, *Student Member, IEEE* and Yun Fu, *Senior Member, IEEE*

Abstract—Domain adaptation has been a primal approach to addressing the issues by lack of labels in many data mining tasks. Although considerable efforts have been devoted to domain adaptation with promising results, most existing work learns a classifier on a source domain and then predicts the labels for target data, where only the instances near the boundary determine the hyperplane and the whole structure information is ignored. Moreover, little work has been done regarding to multi-source domain adaptation. To that end, we develop a novel unsupervised domain adaptation framework, which ensures the whole structure of source domains is preserved to guide the target structure learning in a semi-supervised clustering fashion. To our knowledge, this is the first time when the domain adaptation problem is re-formulated as a semi-supervised clustering problem with target labels as missing values. Furthermore, by introducing an augmented matrix, a non-trivial solution is designed, which can be exactly mapped into a K-means-like optimization problem with modified distance function and update rule for centroids in an efficient way. Extensive experiments on several widely-used databases show the substantial improvements of our proposed approach over the state-of-the-art methods.

Index Terms—Transfer Learning; Multi-Domain Adaptation; Constrained Clustering; Utility Function

1 Introduction

Domain adaptation, as a branch of transfer learning, has recently attracted lots of attention [1], where adapts features in different domains with same or similar tasks. A good instance would be adapting the object classifier trained from low-resolution webcam images for the image recognition of the same category captured by high-resolution digital cameras. The challenge lies in the significantly different distributions between webcam and digital camera images due to image resolutions.

In domain adaptation, we denote domains with well-labeled data as source domains while the domain being classified as the target domain. Most domain adaptation algorithms manage to align them so that the well-established knowledge can be transferred from source to target domain. Briefly, these algorithms are characterized by the following two groups: (1) feature space adaptation, (2) classifier adaptation. Research work regarding to feature space adaptation seeks for a common subspace where the feature space divergence between source and target domains is minimized [3], [4], [5], [6], [7], [8], [9], [10]. However, as fewer target labels are available in the training, they may not be able to achieve conditional distribution alignment, which is more challenging for multiple source data. On the other hand, classifier adaptation usually adapts the classifier learned in the source to the target data [11], [12], [13]. Apparently, such techniques require target labels for classifier adaptation, and therefore are inappropriate for unsupervised domain adaptation. While considerable endeavor has been made to domain adaptation, it concentrates more on the single source domain adaptation [6], [9], [9], [14]. Even worse, for classifier adaptation, only the knowledge derived from the hyperplane is transferred to the target domain and the global structure information of the source domain is ignored. In fact, the performance of existing multi-source domain adaptation methods is far from satisfactory (See Table 4) and is even worse than those single source domain adaptation methods.

In this paper, we target at the challenging unsupervised domain adaptation problem, given the unavailable target labels and complex composition of single or multiple source domains. To that end, a novel semi-supervised clustering framework is proposed to preserve the intrinsic structures of source and target domains and predict the labels of target domain, as shown in Figure 1. We employ semi-supervised clustering in two source domains together with the target domain, while ensuring the label consistency at the partition level for the unknown target data. Specifically, we put the source and target data together for clustering with the source structure preserved. In this way, we cast the original single or multiple source domain adaptation to a joint semi-supervised clustering with common unknown target labels and known multiple source labels. To the best of our knowledge, this is the first work to formulate unsupervised domain adaptation into a semi-supervised clustering framework. Then we derive the algorithm by taking the derivatives and give its corresponding solution. Furthermore, a K-means-like optimization solution is further designed to the proposed method in a neat mathematical and highly efficient way. Extensive experiments on several popular domain adaptation databases demonstrate the effectiveness of our method against the most recent state-of-the-art single

Manuscript received XXX; revised XXX. *Hongfu Liu is with the Department of Electrical & Computer Engineering, Northeastern University, Boston, USA. Email: liu.hongf@husky.neu.edu. Ming Shao is with Department of Computer and Information Science, University of Massachusetts Dartmouth, Dartmouth, USA. Email:mshao@umassd.edu. Zhengming Ding is with the Department of Electrical & Computer Engineering, Northeastern University, Boston, USA. Email: allanding@ece.neu.edu. Yun Fu is with the Department of Electrical & Computer Engineering and College of Computer and Information Science, Northeastern University, Boston, USA. Email: yunfu@ece.neu.edu*

This is the author's manuscript of the article published in final edited form as:
Liu, H., Shao, M., Ding, Z., & Fu, Y. (2019). Structure-Preserved Unsupervised Domain Adaptation. IEEE Transactions on Knowledge and Data Engineering, 31(4), 799–812. https://doi.org/10.1109/TKDE.2018.2843342

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

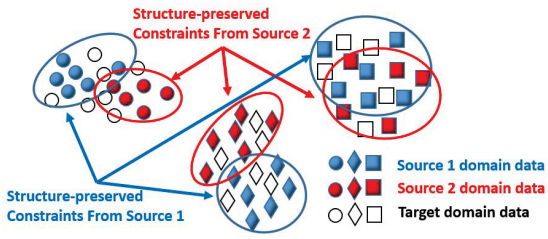


Fig. 1. Illustration of the proposed semi-supervised clustering for unsupervised multi-source domain adaptation. Different shapes mean different classes and different colors denote different domains. While solid symbols with red and blue are labeled source data, hollow shape is the unknown target data. The big ellipses are the enforced partition-level label constraints for the structure-preserved source domains.

source or multi-source domain adaptation methods by a large margin. We highlight our main contributions as follows.

- We propose a novel constrained clustering algorithm for single or multiple source domain adaptation. Specially we put the source and target data together for cluster analysis, where the structures of source domains are regularized consistent with the label information as much as possible.
- By introducing an augmented matrix, a K-means-like optimization is nontrivially designed with modified distance function and update rule for centroids in an efficient way.
- Extensive experiments on several popular domain adaptation databases demonstrate the advantages of our proposed methods over most recent state-of-the-art methods verify the effectiveness of structure-preserved clustering for unsupervised domain adaptation.

This paper is an extension of our conference paper [15], where we add the following contents to improve this paper in both methodological and experimental parts. (1) A Structure-Preserved Unsupervised Domain Adaptation (SP-UDA) framework is proposed with a partition-level constraint; (2) We present the single source domain adaptation method to enrich the SP-UDA; (3) Non-trivially, a K-means-like optimization is designed for the neat formulation and efficient solution with theoretical supports; and (4) We add a subsection on objective recognition with deep features and give the detailed analyses.

The rest of this paper is organized as follows. Section 2 gives the introduction of the literature in domain adaptation and constrained clustering area. We illustrate the SP-UDA framework in Section 3 with two methods for single source and multi-source domain adaptation in Section 3.3 and 3.4, respectively. Experimental results on several widely used databases are demonstrated in Section 4. Finally, Section 5 concludes this paper.

Notation. We denote a vector (matrix) with a lowercase (uppercase) letter. $A^\top$, A^{-1} and $\mathrm{tr}(A)$ stand for the transpose, the inverse and the trace of a matrix $A = \{a_{ij}\}$, respectively. $||A||_\mathrm{F} = \sqrt{\sum_i \sum_j a_{ij}^2}$ denotes the Frobenius norm of A.

2 Related Work

In this section, we provide the related work in terms of unsupervised domain adaptation and constrained clustering, and highlight the different between the existing work and ours.

2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation aims to classify the target data given the related source data with labels [16], [17]. Feature adaption is one of the typical methods to address the domain shit, include searching intermediate subspaces that smoothly transfers from one domain to another [7], [8], [18] and learning common feature space [3], [5], [6], [10], [19]. Among them, LSTL [10] and JDA [9] are two typical subspace based domain adaption algorithms. Hou et al. even proposed by involving the pseudo target labels optimization to further consider the conditional distribution alignment under the common subspace [20]. On the other hand, classifier adaption on source data is also widely used for target data prediction; however, few target data labels are needed [11], [12], [13], [21], which is beyond our paper scope.

The multi-source scenarios make this problem more challenging, where beyond the alignment between source and target domain, the differences among the source domains should be taken into consideration [22], [23], [24]. Some native methods mix all source data and treat all of them equally [7], [8], [25], and ignore the relationship within different source domains. In such a way, they fail to explore the underlying structure of each domain, and introduce negative transfer due to the complex composition of multiple domains. To cope this, a few methods have been recently proposed to reshape the multiple sources by discovering latent domains. For example, the modality information is deeply mined with re-organized data, followed by a constrained clustering method [26] to discover latent domains. Another representative method is to explore the latent domains according to domain integrity and separability [27]. RDALR [28] and SDDL [25] are two typical multi-source domain adaptation models, which aim to transform sources into a new space with a reconstruction formulation in a low-rank or sparse constraint. Although there are studies on multi-source domain adaptation [29], most of them still require target labels for classifier adaptation, which is different from unsupervised domain adaptation problem setting here.

Most recently, deep transfer learning algorithms attract increasing attention by generalizing deep structure to increase the feature representation ability for the transfer learning scenario [19], [30], [31], [32], [33]. The main idea is to enhance the feature transferability in the task-specific layers of the deep neural networks by explicitly reducing the domain discrepancy. By this means, the obtained feed-forward networks can be applicable to the target domain without being hindered by the domain shift. For example, Long et al. explored multi-layer adaptation on the fully-connected layers for soruce and target networks, and therefore, the new designed loss would help solve the domain mismatch during network learning [31]. However, those

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

algorithms all focus on how to reduce the marginal distribution divergence across two domains, which fails to preserve the intrinsic class-wise structure of two domains.

2.2 Constrained Clustering

Constrained clustering aims to employ the auxiliary side information to enhance the clustering performance. Due to the non-order property of clustering, Must-Link and Cannot-Link are proposed as the pairwise constraints by K. Wagstaff and C. Cardie [34], which are involved into a clustering algorithm. COP-K-means was put forward by assigning each instances to the nearest centroid with all the constraints guaranteed [35]. Later, Shental et al. incorporated the pairwise constraints into Gaussian Mixture Model and provided a closed-form EM procedure and generalized EM procedure for Must-Link and Cannot-Link, respectively [36]. The above methods belong to hard constrained clustering, since no violation of the constraints is allowed.

However, satisfying all the constraints might make the clustering intractable and no solution can be achieved. To overcome such shortcomings, soft constrained clustering algorithms have been developed to employ the side information with the minimum violated constraints. Constrained Vector Quantization Error (CVQE) put the cost of violating constraints into consideration, and optimized the cost based on K-means clustering [37]. Furthermore, Linear-time CVQE modified CVQE with different computation of violating constraints in a fast way [38]. Metric Pairwise Constrained K-means (MPCK-means) learns a best Mahalanobis distance metric to meet all the constraints for clustering [39]. Different from the pairwise constraints, the partition level side information with partial labels is also made full use to enhance the clustering performance, such as NMF-based method [40] and K-means-based method [41], [42].

Another representative category of constrained clustering is built on spectral clustering with constraints by modifying the Laplacian graph or Eigenspace. Some methods directly set the value in the Laplacian graph with 1 or 0 according to Must-link and Cannot-link constraints, and applied the classical spectral clustering or random walk to obtain the final solution [43], [44]. Or the constraints are propagated in the affinity matrix as a regularizer to modify the Laplacian graph [45], [46], [47]. For modifying the Eigenspace, Li et al. enforced constraints by regularizing the spectral embedding [49], while Wang et al. proposed a flexible constrained spectral clustering to encode the constraints as part of a constrained optimization problem.

The most exiting work treats the unsupervised domain adaptation task as a classification problem, where a classifier is learnt on the source data and adapted to predict the labels of target data. In such a case, only the instances near the classification boundary play the key role in representing the knowledge from source domain. Unfortunately, the whole structures of source and target data are ignored. In light of this, we aim to preserve the source structure, which is further employed to guide the target structure mining. Moreover, our model is also different from the pair-wise constrained clustering, where a partition-level constraint is utilized to preserve the source structure as a whole to guide the target data clustering.

3 SP-UDA Framework

Typically, domain adaptation aims to borrow some well-defined knowledge in the source domain and apply it to the task on the target domain [14]. Here source domain and target domain are different but related. The goal of domain adaptation is to make use of the data and labels in the source domains to predict the labels for the target domain.

Since the distributions of data from source and target domains have large divergences, the alignment of two distributions is regarded as the key problem in domain adaptation area. In light of this, tremendous efforts have been taken to seek a common space. After that, a classifier learnt with the source data and the corresponding labels can be adapted to the task on target data. Admittedly, the alignment is crucial to the success of domain adaptation. However, how to effectively transfer the knowledge from source domain to the target domain is another key factor, which is unfortunately usually being ignored.

Most of existing work trains a classifier in the common space with the source data and applies it for target domain. In such a way, only several points in the source domain play the determined role for the hyper-plain of the classifier and other points are not utilized effectively. To cope with this challenge, we focus on the way of knowledge transfer for domain adaptation. Specifically, a partition-level constraint is employed to preserve and transfer the whole source structure and then the source and target data are put together as a constrained clustering problem.

3.1 Problem Definition

The alignment and transfer are two key challenges in domain adaptation, and we focus on the second one. Previous work formulates the domain adaptation as the classification problem, where a hyperplane learnt from the source data is adapted to predict the labels for target data. In such a way, only several instances near the boundary determine the hyperplane, while other instances and the complete structure are not fully utilized for knowledge transfer. To avoid the knowledge waste, we aim to make use of the complete source and target structure for domain adaptation, where the source structure is preserved to guide the target structure mining. Moreover, although many efforts have been taken in this field and some reasonable performance has been achieved, most existing work pays more attention to the single source domain adaptation [6], [9], [14]. For the methods, which can handle multi-source domain adaptation, the performance is far from satisfactory (See Table 4), or even worse than the single source domain adaptation.

Without loss of generality, suppose we have the source data with label information and target data without label information, our task is to assign labels for the target data. Let X_S denote the data matrix of the source domain with n_S instances and m features, Y_S is 1-of-K coding label matrix of source data, where K is the number of classes; X_T represents the data matrix of target domain with n_T instances and m features. Since our goal is to effectively transfer the knowledge from source domain to the target domain, rather than align the distributions of different domains, here we assume that the alignment projection P (either linear or non-linear) is pre-known or pre-learned:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
Notations

Notation	Description
X_S	Source domain data matrix in the original feature space
Y_S	Source domain indicator matrix
X_T	Target domain data matrix in the original feature space
K	Number of clusters
P	Projection from original space to common space
Z_S	Source domain data matrix in the aligned feature space
Z_T	Target domain data matrix in the aligned feature space
H_S	Learnt source domain indicator matrix
H_T	Learnt target domain indicator matrix

$Z_S = X_S P$, $Z_T = X_T P$. With the projection P, the source data and target data are aligned into the common space. Since the source data have labels, we can formulate the domain adaptation problem as a semi-supervised clustering problem.

In light of this, we also take the single or multi-source domain adaptation into consideration in a unified framework. Therefore, we formalize the problems addressed in this paper as follows:

- How to preserve the structure of source domain to predict the labels of target domain?
- How to conduct multi-source domain adaptation in a unified framework?
- How to provide a neat formulation and its corresponding solution?

3.2 Framework

In order to explore the structure of different domains, we formulate the problem as a clustering problem. Generally speaking, we put the source and target data together for clustering, where the learnt partition of source data is regularized to agree with the labels of source data. To achieve this, here we propose the framework of Structure-Preserved Unsupervised Domain Adaptation (SP-UDA). Table 1 provides the key variables used along this paper. Given the pre-learnt alignment projection P, we have the new representation in the common space of source and target data as Z_S and Z_T. Our goal is to utilize the whole structure of source domain for the recognition of target data. To achieve this, the source and target data are put together for clustering with the partition-level constraint from the source data label, which preserves the whole source structure and further guides the clustering process. The SP-UDA can be summarized as follows:

$$\min_{H_S, H_T} \mathcal{J}(Z_S, Z_T; K) - \lambda U_c(H_S, Y_S), \tag{1}$$

where $\mathcal{J}$ is the objective function of certain clustering algorithm, which takes Z_S and Z_T as the input, partitions the data into K clusters and returns the assignment matrices H_S and H_T; U_c is the well-known categorical utility function [51], which treats the similarity of two partitions. By introducing the contingence table in Table 2, U_c can be calculated as follows:

$$U_c(H_S, Y_S) = \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K} (\frac{p_{kj}}{p_{k+}})^2 - \sum_{j=1}^{K} (p_{+j})^2, \tag{2}$$

TABLE 2
Contingency Matrix

		Y_S				
		$C_1^{(Y)}$	$C_2^{(Y)}$	$\cdots$	$C_K^{(Y)}$	$\sum$
H_S	C_1	n_{11}	n_{12}	$\cdots$	n_{1K}	n_{1+}
	C_2	n_{21}	n_{22}	$\cdots$	n_{2K_i}	n_{2+}
	$\cdot$	$\cdot$	$\cdot$	$\cdots$	$\cdot$	$\cdot$
	C_K	n_{K1}	n_{K2}	$\cdots$	n_{KK}	n_{K+}
	$\sum$	n_{+1}	n_{+2}	$\cdots$	n_{+K}	n

where H_S and Y_S are two partitions, p_{kj} means the joint probability of one instance simultaneously belonging to the k-th cluster in H_S and the j-th cluster in Y_S, and p_{k+} and p_{+j} are the cluster portion of H_S and Y_S, respectively. λ is the trade-off parameter, which balances the clustering term and the partition-level constraint.

Categorical utility function measures the difference between the information to prediction H_S with Y_S and without Y_S, which can be regarded as the metric for the partition level similarity. And it recently has been widely applied to constrained clustering and consensus clustering to calculate the similarity between two partitions. For example, partition level constrained clustering employs the partial labels as the side information and applies the categorial utility function to make the learnt partition close to the partial labels [41], [42]; consensus clustering fuses several basic partitions into an integrated one by maximizing the categorial utility function between basic partitions and consensus one [52], [53], [54], [55].

The benefits of the SP-UDA framework in Eq. (1) lie in that (1) we employ the constrained clustering approach instead of classification for the recognition of target data, so that these target data without labels are involved during the training process, (2) the categorical utility function plays as the partition-level constraint, which not only preserves and transfers the whole source structure to target data, but also guides the target data clustering and (3) the framework can be efficiently solved via a K-means-like solution, if we choose K-means as the core clustering algorithm in $\mathcal{J}$, which will be further discussed in Section 3.5.

Note that in our SP-UDA framework, we assume that the projection P from the original feature space to the common space is known, and the inputs are $Z_S = X_S P$, $Z_T = X_T P$, the source and target data matrix after the projection P. Actually, there are tremendous efforts to address the projection problem, such as Geodesic Flow Kernel (GFK) [8], Transfer Component Analysis (TCA) [14], Transfer Subspace Learning (TSL) [6] and Joint Domain Adaptation (JDA) [9], where the projection P learnt from these algorithms plays a role in aligning the data from source and target domain into a common space and it preserves the cluster structure to some extent. Although we can involve the projection learning within our SP-UDA framework, combining some mature techniques is not our selling point and this also leads our model complex and loses the neat formulation. In this paper, we focus on the structure-preserved learning to enhance the domain adaptation performance. Therefore, we directly start from source and target data matrix after the projection. In the following, we introduce how to apply the SP-UDA framework for single and multi-source domain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

adaptation.

3.3 SP-UDA for Single Source Domain

Here we illustrate how to apply the SP-UDA framework for single source domain adaptation. For similarity, we choose K-means as the core clustering algorithm in $\mathcal{J}$, which leads the following objective function:

$$\min \left|\left| \begin{bmatrix} Z_S \\ Z_T \end{bmatrix} - \begin{bmatrix} H_S \\ H_T \end{bmatrix} G \right|\right|_{\mathrm{F}}^2 - \lambda U_c(H_S, Y_S), \tag{3}$$

where Z_S, Z_T, Y_S are input variables, H_S and H_T are the unknown assignment matrices for source and target data, respectively, and G is the corresponding centroids matrix.

We can see that Eq. (3) has two components. The first one explores the combined source and target data with a standard K-means clustering, while the second one is a regularizer to make the learnt source partition consistent with the source labels for preserving the source structure. After the projection, the source and target data Z_S and Z_t are aligned in the common space. Data points with the same label, no matter from the source domain or target domain form a cluster and they have the same cluster centroid. Therefore, we employ K centroids G to represent all the data points in the aligned space, where H_S and H_T are the indicator matrices to indicate the data point belonging to the nearest centroid in G. The two terms in Eq. (3) share different functions. The K-means term aims to explore the combined source and target data structure, while the categorical utility function is expected to make the learnt source structure be similar to the source labels as much as possible in order to preserve the source structure, where it plays a role in uncovering the target structure with the guidance of source structure.

Here we aim to find a solution containing H_S and H_T, where the source and target structures are simultaneously explored with the source one consistent with the source labels. Different from the exiting work, which focuses on the classification with hyperplane, we pay attention to the structure and formulate the domain adaptation problem into a clustering task. If we take a close look at the optimization problem in Eq. (3), the difficulties result from that there exist both continuous and discrete variables and that the objective function has both matrix-wise and element-wise formulation. Thanks to the following Lemma 1, it gives a new insight of the second term in SP-UDA framework.

Lemma 1. *Given one fixed partition Y and any partition H, we have*

$$nU_c(H, Y) + ||Y - HM||_{\mathrm{F}}^2 = \text{constant}, \tag{4}$$

where $M_k = (\frac{p_{k1}}{p_{k+}}, \cdots, \frac{p_{kj}}{p_{k+}}, \cdots, \frac{p_{kK}}{p_{k+}})$ is the k-th row of $M, \forall k, k = 1, \cdots, K$.

The proof of Lemma 1 can be found in our previous work [52], [53], which is omitted here due to page limit. We can see that the categorical utility function U_c measures the similarity of two partitions, while $||Y - HM||_{\mathrm{F}}^2$ is used to calculate the divergence of two partitions with the alignment matrix M. Based on Lemma 1, we have a new formulation of the problem in Eq. (3) as follows:

$$\min \left|\left| \begin{bmatrix} Z_S \\ Z_T \end{bmatrix} - \begin{bmatrix} H_S \\ H_T \end{bmatrix} G \right|\right|_{\mathrm{F}}^2 + \lambda ||Y_S - H_S M||_{\mathrm{F}}^2. \tag{5}$$

Algorithm 1 The algorithm of SP-UDA for single source domain.

Input: Z_S, Z_T: data matrix;
Y_S: the labels of source domains;
K: number of clusters;
λ: trade-off parameter.
Output: optimal H_S, H_T;
1: Initialize H_S and H_T;
2: **repeat**
3: Update G by Eq. (7);
4: Update M by Eq. (9);
5: Update H_S and H_T by Eq. (10) and (11), respectively;
6: **until** the objective value in Eq. (3) remains unchanged.

In Eq. (5), M plays a role in shuffling the order of clusters in Y_S. Since the partition is orderless in terms of cluster labels, it is necessary to align two partitions before measuring their similarity. Although one variable M is involved in Eq. (5), we can seek the solution by iteratively updating each unknown continuous variable by taking derivation and greedy search for the discrete variables.

Fixing others, Update G. Let $Z = [Z_S; Z_T]$ and $H = [H_S; H_T]$, then the term related to G is $J_1 = ||Z - HG||_{\mathrm{F}}^2$. By taking the derivative of J over G, we have

$$\frac{\partial J_1}{\partial G} = -2H^\top Z + 2H^\top HG = 0. \tag{6}$$

The solution leads to the update rule of G_1 as follows.

$$G = (H^\top H)^{-1} H^\top Z. \tag{7}$$

Fixing others, Update M. Let $J_2 = ||Y_S - H_S M||_{\mathrm{F}}^2$ and minimize J_2 over M by taking the derivative, we have

$$\frac{\partial J_2}{\partial M} = -2H_S^\top Y_S + 2H_S^\top H_S M = 0. \tag{8}$$

Thus, we have the following update rule for M as:

$$M = (H_S^\top H_S)^{-1} H_S^\top Y_S. \tag{9}$$

Fixing others, Update H_S. The rules of updating H_S is slightly different from the above rules. Due to the discrete variable, here we use an exhaustive search for the optimal assignment to find the solutions for each data point in H_S as follows:

$$k = \arg\min_j ||Z_{S,i} - G_j||_2^2 + \lambda ||Y_{S,i} - b_j M||_2^2, \tag{10}$$

where $Z_{S,i}$ and $Y_{S,i}$ denote the i-th row in Z_{S_1} and Y_{S_1}, G_j is the j-th centroid or row of G and b_j is a $1 \times K$ vector with j-th position 1 and others 0.

Fixing others, Update H_T. For H_T, similarly we apply an exhaustive search for each data point in H_T,

$$k = \arg\min_j ||Z_{T,i} - G_j||_2^2, \tag{11}$$

where $Z_{T,i}$ denotes the i-th row in Z_T and G_j is the j-th centroid or row in G.

The algorithm by derivation is given in Algorithm 1. We decompose the problem into several sub-problems, which have the closed-form solutions. Therefore, the final solution can be guaranteed to converge to the local minimum. In essence, Algorithm 1 is a constrained clustering method.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

Different from the traditional constrained clustering algorithms, which employs the pair-wise cannot-link or must-link constraints to shape the cluster structure, here a novel partition-level constraint [41] is applied here to treat the source structure as a whole and preserve the whole structure during the clustering process. This further guides the target data clustering. Although the update rule in Eq. (11) seems not to include Y_S, the source structure affects the assignment matrix H_S and further conducts on the centroid matrix G in the common space. This indicates that Y_S helps to seek the better cluster centers in the common space, which facilitates the target data clustering.

3.4 SP-UDA for Multiple Source Domains

Next we continue to apply the SP-UDA framework for single source domain adaptation. Without loss of generality, suppose we have the two source domains and one target domain. With some alignment projections P_1 and P_2, we have the common features $Z_{S_1} = X_{S_1}P_1$, $Z_{T_1} = X_TP_1$, $Z_{S_2} = X_{S_2}P_2$ and $Z_{T_2} = X_TP_2$. Our goal is to fuse the information from multi-source domain to provide better performance on target domain. Here suppose that the alignment projects P_1 and P_2 are given, we start from the $Z_{S_1}, Z_{S_2}, Z_{T_1}$ and Z_{T_2} to predict the labels H_T for target domain. In the following, we first give the objective function for two source domains in the SP-UDA and provide the corresponding solution.

Based on Lemma 1, we directly give the following objective function for two source domains scenario.

$$\begin{aligned}\min &\left|\left|\begin{bmatrix}Z_{S_1}\\ Z_{T_1}\end{bmatrix} - \begin{bmatrix}H_{S_1}\\ H_T\end{bmatrix} G_1\right|\right|_{\mathrm{F}}^2 + \lambda||Y_{S_1} - H_{S_1}M_1||_{\mathrm{F}}^2 \\ &+ \left|\left|\begin{bmatrix}Z_{S_2}\\ Z_{T_2}\end{bmatrix} - \begin{bmatrix}H_{S_2}\\ H_T\end{bmatrix} G_2\right|\right|_{\mathrm{F}}^2 + \lambda||Y_{S_2} - H_{S_2}M_2||_{\mathrm{F}}^2,\end{aligned} \tag{12}$$

where $Z_{S_1}, Z_{S_2}, Z_{T_1}, Z_{T_2}, Y_{S_1}$ and Y_{S_2} are input variables, the rest are unknown. H_{S_1}, H_{S_2} and H_T are the indicator matrices for two source domains and the target domain respectively, G_1 and G_2 are the corresponding centroids matrices, M_1 and M_2 are two alignment matrices to match Y_{S_1} and Y_{S_2}, respectively.

Since the problem in Eq. (12) is not jointly convex to all the variables, here we iteratively update each unknown variable by taking derivation.

Fixing others, Update G_1, G_2. Let $Z_1 = [Z_{S_1}; Z_{T_1}]$ and $H_1 = [H_{S_1}; H_T]$, then the term related to G_1 is $J_1 = ||Z_1 - H_1G_1||_{\mathrm{F}}^2$. By taking the derivative of J_1 over G_1, we have

$$\frac{\partial J_1}{\partial G_1} = -2H_1^\top Z_1 + 2H_1^\top H_1G_1 = 0. \tag{13}$$

The solution leads to the update rule of G_1 as follows.

$$G_1 = (H_1^\top H_1)^{-1}H_1^\top Z_1. \tag{14}$$

Similarly, $Z_2 = [Z_{S_2}; Z_{T_2}]$ and $H_2 = [H_{S_2}; H_T]$, we have the following rule to update G_2.

$$G_2 = (H_2^\top H_2)^{-1}H_2^\top Z_2. \tag{15}$$

Fixing others, Update M_1, M_2. Let $J_2 = ||Y_{S_1} - H_{S_1}M_1||_{\mathrm{F}}^2$ and minimize J_2 over M_1 by taking the derivative, we have

$$\frac{\partial J_2}{\partial M_1} = -2H_{S_1}^\top Y_{S_1} + 2H_{S_1}^\top H_{S_1}M_1 = 0. \tag{16}$$

Algorithm 2 The algorithm of SP-UDA for multiple source domains.

Input: $Z_{S_1}, Z_{T_1}, Z_{S_2}, Z_{T_2}$: data matrix;
Y_{S_1}, Y_{S_2}: the labels of source domains;
K: number of clusters;
λ: trade-off parameter.
Output: optimal H_{S_1}, H_{S_2}, H_T;
1: Initialize H_{S_1}, H_{S_2} and H_T;
2: **repeat**
3: Update G_1 and G_2 by Eq. (14) and (15);
4: Update M_1 and M_2 by Eq. (17);
5: Update H_{S_1}, H_{S_2} and H_T by Eq. (18), (19) and (20), respectively;
6: **until** the objective value in Eq. (12) remains unchanged.

The update rule of M_2 is similar to the one of M_1, so we have the following update rules.

$$\begin{aligned}M_1 &= (H_{S_1}^\top H_{S_1})^{-1}H_{S_1}^\top Y_{S_1}, \\ M_2 &= (H_{S_2}^\top H_{S_2})^{-1}H_{S_2}^\top Y_{S_2}.\end{aligned} \tag{17}$$

Fixing others, Update H_{S_1}, H_{S_2}. The rules of updating H_{S_1} and H_{S_2} are slightly different from the above rules, since they are not continuous variables. Here we use a exhaustive search for the optimal assignment to find the solutions.

For H_{S_1}, we have

$$k = \arg\min_j ||Z_{S_1,i} - G_{1,j}||_2^2 + \lambda||Y_{S_1,i} - b_jM_1||_2^2, \tag{18}$$

where $Z_{S_1,i}$ and $Y_{S_1,i}$ denote the i-th row in Z_{S_1} and H_{S_1}, $G_{1,j}$ is the j-th centroid of G_1 and b_j is a $1 \times K$ vector with j-th position 1 and others 0.

For H_{S_2}, we have

$$k = \arg\min_j ||Z_{S_2,i} - G_{2,j}||_2^2 + \lambda||Y_{S_2,i} - b_jM_2||_2^2, \tag{19}$$

where $Z_{S_2,i}$ and $Y_{S_2,i}$ denote the i-th row in Z_{S_2} and H_{S_2}, $G_{2,j}$ is the j-th centroid of G_2 and b_j is a $1 \times K$ vector with j-th position 1 and others 0.

Fixing others, Update H_T. For H_T, we still use an exhaustive search for the solution,

$$k = \arg\min_j ||Z_{T_1,i} - G_{1,j}||_2^2 + ||Z_{T_2,i} - G_{2,j}||_2^2, \tag{20}$$

where $Z_{T_1,i}$ and $Z_{T_1,i}$ denote the i-th row in Z_{T_1} and Z_{T_2}, and $G_{1,j}, G_{2,j}$ are the j-th centroid of G_1, G_2.

The algorithm by derivation is provided in Algorithm 2. Similar to Algorithm 1, we decompose the whole problem into several sub-problems, each of which has the closed-form solutions. Therefore, the final solution can be guaranteed to converge to the local minimum. Although we can take the derivative of each unknown variable to obtain the solution, it is not neat and efficient due to the matrix product and inverse. Moreover, there are 7 unknown variables to update; for several source domains, more unknown variables are involved, which prevents the algorithm from practical use. In light of this, we propose another equivalent solution in a neat mathematical way with high efficiency. In the following, we provide a K-means-like optimization problem for the multi-source domain adaptaion via an augmented matrix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

3.5 K-means-like Optimization

In the above two sections, we apply the derivatives and greedy search for the solution. However, we find that when the number of source domains increases, the solution requests many variables to be updated, which makes the model fragmented and inefficient. To cope with this challenge, we equivalently transfer the problem into a K-means like optimization problem in a neat and efficient way. Generally speaking, a K-means-like solution is designed with neat mathematical formulation by introducing an augmented matrix and the convergence of the new solution is guaranteed. The discussion on the time complexity is also provided for fully understanding the solution.

Before giving the K-means-like optimization, we first introduce the augmented matrix D as follows:

$$D = \begin{pmatrix} Z_{S_1} & Y_{S_1} & 0 & 0 \\ 0 & 0 & Z_{S_2} & Y_{S_2} \\ Z_{T_1} & 0 & Z_{T_2} & 0 \end{pmatrix}, \tag{21}$$

where d_i is the i-th row of D, which consists of four parts. The first one is the features $d_i^{(1)} = (d_{i,1}, \cdots, d_{i,m})$ after projection P_1, the next K columns $d_i^{(2)} = (d_{i,m+1}, \cdots, d_{i,m+K})$ denotes the label information of the first source domain, while the third and fourth parts denote the features and labels of the second domain. From Eq. (21), we can see that each row denotes each domain and the first and third columns represent the common spaces between two source domains and target domain, respectively, while the second and fourth columns represent the label information of each domain. Zeros are used to fill up the other parts of the augmented matrix.

By these means, we equivalently transfer the problem addressed here into a semi-supervised clustering with missing values. Due to the artificial zeros in the augmented matrix, we cannot directly apply the standard K-means on the matrix D. To achieve this, a modified K-means with incomplete centroid updating and conditional distance function is proposed. Generally speaking, these artificial zeros do not contribute to the utility when measuring the similarities, which are not involved into the centroid updating either. That is to say, the centroids of the modified K-means are updated by the incomplete data. Here we give the new updating rules for the centroids. Let $m_k = (m_k^{(1)}, m_k^{(2)}, m_k^{(3)}, m_k^{(4)})$ be the k-th centroid $\mathbf{C}_k$, which $m_k^{(1)} = (m_{k,1}, \cdots, m_{k,m})$, $m_k^{(2)} = (m_{k,m+1}, \cdots, m_{k,m+K})$, $m_k^{(3)} = (m_{k,m+K+1}, \cdots, m_{k,2m+K})$ and $m_k^{(4)} = (m_{k,2m+K+1}, \cdots, m_{k,2m+2K})$. Let $Z_1 = Z_{S_1} \cup Z_{T_1}$ and $Z_2 = Z_{S_2} \cup Z_{T_2}$, we modify the computation of the centroids as follows,

$$\begin{aligned} m_k^{(1)} &= \frac{\sum_{x_i \in \mathbf{C}_k \cap Z_1} d_i^{(1)}}{|\mathbf{C}_k \cap Z_1|}, \; m_k^{(2)} = \frac{\sum_{x_i \in \mathbf{C}_k \cap Y_{S_1}} d_i^{(2)}}{|\mathbf{C}_k \cap Y_{S_1}|}. \\ m_k^{(3)} &= \frac{\sum_{x_i \in \mathbf{C}_k \cap Z_2} d_i^{(3)}}{|\mathbf{C}_k \cap Z_2|}, \; m_k^{(4)} = \frac{\sum_{x_i \in \mathbf{C}_k \cap Y_{S_2}} d_i^{(4)}}{|\mathbf{C}_k \cap Y_{S_2}|}. \end{aligned} \tag{22}$$

Compared with the standard K-means, where the centroids are computed by arithmetic means with the denominator being the number of instances in its corresponding cluster, here we only put the "real" instances into the computation of centroids. With the new centroid updating rule, we have the following Theorem 1.

Theorem 1. *Given the data matrix Z_{S_1}, Z_{T_1}, Z_{S_2}, Z_{T_2} and the label information from two source domains Y_{S_1} and Y_{S_2} and augmented matrix D, we have the following equivalence*

$$\begin{aligned} &\min || \begin{bmatrix} Z_{S_1} \\ Z_{T_1} \end{bmatrix} - \begin{bmatrix} H_{S_1} \\ H_T \end{bmatrix} G_1 ||_{\mathrm{F}}^2 + \lambda ||Y_{S_1} - H_{S_1} M_1||_{\mathrm{F}}^2 \\ &+ || \begin{bmatrix} Z_{S_2} \\ Z_{T_2} \end{bmatrix} - \begin{bmatrix} H_{S_2} \\ H_T \end{bmatrix} G_2 ||_{\mathrm{F}}^2 + \lambda ||Y_{S_2} - H_{S_2} M_2||_{\mathrm{F}}^2, \\ &\Leftrightarrow \min \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} f(d_i, m_k), \end{aligned} \tag{23}$$

where the centroids are calculated by Eq. 22 and the distance function f can be computed by

$$\begin{aligned} &f(d_i, m_k) \\ &= \mathbf{1}(d_i \in Z_1) ||d_i^{(1)} - m_k^{(1)}||_2^2 + \lambda \mathbf{1}(d_i \in Y_{S_1}) ||d_i^{(2)} - m_k^{(2)}||_2^2 \\ &+ \mathbf{1}(d_i \in Z_2) ||d_i^{(3)} - m_k^{(3)}||_2^2 + \lambda \mathbf{1}(d_i \in Y_{S_2}) ||d_i^{(4)} - m_k^{(4)}||_2^2, \end{aligned} \tag{24}$$

where $\mathbf{1}(\cdot)$ returns 1 when it meets the condition, otherwise returns 0 .

Proof. We start from the objective function of K-means.

$$\begin{aligned} &\sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} f(d_i, m_k) \\ &= \sum_{k=1}^{K} (\sum_{d_i \in \mathbf{C}_k \cap Z_1} ||d_i^{(1)} - m_k^{(1)}||_2^2 + \sum_{d_i \in \mathbf{C}_k \cap Y_{S_1}} ||d_i^{(2)} - m_k^{(2)}||_2^2 \\ &+ \sum_{d_i \in \mathbf{C}_k \cap Z_2} ||d_i^{(3)} - m_k^{(3)}||_2^2 + \sum_{d_i \in \mathbf{C}_k \cap Y_{S_2}} ||d_i^{(4)} - m_k^{(4)}||_2^2) \\ &= ||Z_{S_1} - H_{S_1} G_1||_{\mathrm{F}}^2 + ||Z_{T_1} - H_T G_1||_{\mathrm{F}}^2 + \lambda ||Y_{S_1} - H_{S_1} M_1||_{\mathrm{F}}^2 \\ &+ ||Z_{S_2} - H_{S_2} G_2||_{\mathrm{F}}^2 + ||Z_{T_2} - H_T G_2||_{\mathrm{F}}^2 + \lambda ||Y_{S_2} - H_{S_2} M_2||_{\mathrm{F}}^2. \end{aligned}$$

According to the definition of D, Z_{S_1}, Z_{S_S}, Z_{T_1}, Z_{T_2}, H_{S_1}, H_{S_2}, H_T and Eq. 12, we finish the proof. □

Remark 1. *Theorem 1 gives a way to handle the problem in Eq. 12 via a K-means-like optimization problem, which has a neat mathematical way and can be solved with high efficiency. After changing the update rule for centroids and the computation for the distance function, we can still use two-phase iterative optimization with data assignment and centroid update successively.*

Remark 2. *With a close look at the augmented matrix D, the label information can be regarded as new features with more weights, which is controlled by λ. It is worthy to note that the label information is a kind of categorial features, which should not treat the same with the numeric features. Here Theorem 1 provides a way to cluster with both numeric and categorical features together with a utility interpretation.*

By Theorem 1, we transfer the problem into a K-means-like clustering problem. Although there are 10 unknown variables in a two-source domain scenario, the benefits of this solution are that not only the problem can be solved in a neat mathematical and efficient way, but also the model can be easily extended from two source domains to several source domains. Since the update rule and distance function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

Algorithm 3 The algorithm of SP-UDA for multiple source domains via K-means-like optimization

Input: $Z_{S_1}, Z_{T_1}, Z_{S_2}, Z_{T_2}$: data matrix;
Y_{S_1}, Y_{S_2}: the labels of source domains;
K: number of clusters;
λ: trade-off parameter.
Output: optimal H_{S_1}, H_{S_2}, H_T;
1: Build the concatenating matrix D;
2: Randomly select K instances as centroids;
3: **repeat**
4: Assign each instance to its closest centroid by the distance function in Eq. 24;
5: Update centroids by Eq. 22;
6: **until** the objective value in Eq. 12 remains unchanged.

have changed, it is necessary to verify the convergence of the K-means-like algorithm.

Theorem 2. *For the objective function in Theorem 1, the optimization problem is guaranteed to converge in finite two-phase iterations of K-means-like optimization problem.*

Proof. K-means includes two iterations of assign phase and update phase. In the assign phase, each instance is assigned to the nearest centroid so that the objective function decreases. Thus, we focus on the change of objective function during update phase with side information. Next we prove that the computation of centroid by Eq. 22 is optimal. For any centroid $y_k = (y_k^{(1)}, y_k^{(2)}, y_k^{(3)}, y_k^{(4)})$ with $y_k^{(1)}$, $y_k^{(3)}$ containing m elements, $y_k^{(2)}$, $y_k^{(4)}$ containing K elements, and $y_k \neq m_k$, we have

$$\begin{aligned}\Delta &= \sum_{k=1}^{K}\sum_{d_i\in\mathbf{C}_k}||d_i - y_k||_2^2 - \sum_{k=1}^{K}\sum_{d_i\in\mathbf{C}_k}||d_i - m_k||_2^2 \\ &= \sum_{k=1}^{K}\Big(\sum_{d_i\in\mathbf{C}_k\cap Z_1}(||d_i^{(1)} - y_k^{(1)}||_2^2 - ||d_i^{(1)} - m_k^{(1)}||_2^2) \\ &+ \sum_{d_i\in\mathbf{C}_k\cap Y_{S_1}}\lambda(||d_i^{(2)} - y_k^{(2)}||_2^2 - ||d_i^{(2)} - m_k^{(2)}||_2^2) \\ &+ \sum_{d_i\in\mathbf{C}_k\cap Z_2}(||d_i^{(3)} - y_k^{(3)}||_2^2 - ||d_i^{(3)} - m_k^{(3)}||_2^2) \\ &+ \sum_{d_i\in\mathbf{C}_k\cap Y_{S_2}}\lambda(||d_i^{(4)} - y_k^{(4)}||_2^2 - ||d_i^{(4)} - m_k^{(4)}||_2^2)\Big).\end{aligned}$$

According to the Bergman divergence [56], $f(a,b) = ||a - b||_2^2 = \phi(a) - \phi(b) - (a-b)^\top\nabla\phi(b)$, where $\phi(a) = ||a||_2^2$, Here for simplicity, let $\mathbf{S} = \mathbf{C}_k \cap (Z_{S_1} \cup Z_{T_1})$, we rewrite the first term in Eq. 25 as follows:

$$\begin{aligned}&\sum_{d_i\in\mathbf{S}}||d_i^{(1)} - y_k^{(1)}||_2^2 - ||d_i^{(1)} - m_k^{(1)}||_2^2 \\ &= \sum_{d_i\in\mathbf{S}}\big((\phi(d_i^{(1)}) - \phi(y_k^{(1)}) - (d_i^{(1)} - y_k^{(1)})^\top\nabla\phi(y_k^{(1)})) \\ &- \phi(d_i^{(1)}) + \phi(m_k^{(1)}) - (d_i^{(1)} - m_k^{(1)})^\top\nabla\phi(m_k^{(1)})\big).\end{aligned} \tag{25}$$

According to the computation of $m_k^{(1)}$, we have $\sum_{d_i\in S}(d_i^{(1)} - m_k^{(1)}) = 0$ and $\sum_{d_i\in\mathbf{S}}||d_i^{(1)} - y_k^{(1)}||_2^2 - ||d_i^{(1)} - m_k^{(1)}||_2^2 = |\mathbf{S}| \cdot ||m_k^{(1)} - y_k^{(1)}||_2^2$.

Similarly, we have

$$\begin{aligned}\Delta = \sum_{k=1}^{K}\Big(&\sum_{d_i\in\mathbf{C}_k\cap Z_1}||m_k^{(1)} - y_k^{(1)}||_2^2 \\ &+ \sum_{d_i\in\mathbf{C}_k\cap Y_{S_1}}\lambda||m_k^{(2)} - y_k^{(2)}||_2^2 + \sum_{d_i\in\mathbf{C}_k\cap Z_2}||m_k^{(3)} - y_k^{(3)}||_2^2 \\ &+ \sum_{d_i\in\mathbf{C}_k\cap Y_{S_2}}\lambda||m_k^{(4)} - y_k^{(4)}||_2^2\Big).\end{aligned}$$

Therefore, $\Delta > 0$ and the objective function value will decrease during the update phase as well. And due to the finite solution space, the iteration will converge within finite steps. We complete the proof. □

Note that the K-means-like optimization also suits for the single source domain adaptation in Eq. (5). Next, we analyze the time complexity. Since we equivalently transfer the problem into a K-means-like optimization problem, the time complexity of the proposed method enjoys the same time complexity with K-means, $O(tndK)$, where t is the number of iteration, n is the number of data instances including source and target domains, d is the dimension of the concatenating matrix matrix, which equals to $2m + 2K$ and m is the dimension of the common space of source and target domain. We summarize the algorithm in Algorithm 3. The process is similar to K-means clustering. The major differences are the distance function and update rule for centroids.

4 Experimental Results

In this part, we testify the performance of our proposed model in terms of two scenarios, i.e., object classification and face identification.

4.1 Experimental Settings

Databases. *Office+Caltech* is a very popular cross-domain benchmark, which includes three real-world object domains, i.e., Amazon (images downloaded from online merchants), Webcam (low-resolution images by a web camera), and DSLR (high-resolution images by a digital SLR camera). While Caltech-256 is a standard database for object recognition. Here we use the public Office+Caltech datasets [8], which has four domains with 10 shared categories. We adopt two kinds of features, i.e, SURF features [8] and deep features [57].

USPS+MNIST is a digital image database in gray pixel values, which are rescaled to 16×16. Following the setting in [9], we employ 1,800 images in USPS and 2,000 images in MNIST.

COIL is the object image database with different directions. Here COIL1 contains 720 images in the directions of $[0°, 85°] \cup [180°, 265°]$, and COIL2 contains the rest 720 images in the directions of $[90°, 175°] \cup [270°, 355°]$.

PIE is a cross-pose face benchmark, which contains 68 individuals with 41,368 samples. Here we adopt five poses, i.e., PIE05 (left pose), PIE07 (upward pose), PIE09 (downward pose), PIE27 (frontal pose), PIE29 (right pose). Each pose is treated as one domain.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

TABLE 4
Performance (%) comparison on three multiple sources domain benchmarks using SURF features

Source	Target	NC	A-SVM	LTSL-PCA	LTSL-LDA	SFC-C	SFC-J	RDALR	FDDL	SDDL	Ours
A,D	W	20.6	30.4	55.5	30.2	52.0	64.5	36.9	41.0	57.8	**76.3**
A,W	D	16.4	25.3	57.4	43.0	39.0	51.3	31.2	38.4	56.7	**73.9**
D,W	A	16.9	17.3	20.0	17.1	29.0	38.4	20.9	19.0	24.1	**43.8**

TABLE 3
Performance (%) comparison on *Office+Caltech* with one source using SURF features

Dataset	PCA	GFK	TCA	TSL	JDA	Ours	CDDA
C→ A	37.0	41.0	38.2	44.5	44.8	45.6	52.1
C→ W	32.5	40.7	38.6	34.2	37.3	53.9	47.1
C→ D	38.2	38.9	41.4	43.3	43.3	47.8	45.9
A→ C	34.7	40.3	37.8	37.6	36.8	30.7	41.3
A→ W	35.6	39.0	37.6	33.9	38.0	39.7	38.3
A→ D	27.4	36.3	33.1	26.1	28.7	40.8	38.2
W→ C	26.4	30.7	29.3	29.8	29.7	30.5	33.3
W→ A	31.0	29.8	30.1	30.3	35.9	43.5	41.8
W→ D	77.1	80.9	87.3	87.3	85.4	72.6	89.8
D→ C	29.7	30.3	31.7	28.5	31.3	29.9	33.7
D→ A	32.1	32.1	32.2	27.6	30.2	44.8	33.6
D→ W	75.9	75.6	86.1	85.4	84.8	61.7	93.2
UPSP → MNIST	45.0	46.5	51.1	53.8	59.7	60.9	70.8
MNIST → UPSP	66.2	67.2	56.3	66.1	67.3	77.3	82.3
COIL1 → COIL2	84.7	72.5	88.5	88.1	89.3	87.2	99.6
COIL2 → COIL1	84.0	74.2	85.8	87.9	88.5	83.8	99.7
Average	47.3	48.5	50.3	50.3	51.94	53.2	58.8

Note: Since our method is based on JDA, our goal is to show the improvement over JDA. To our best knowledge, we also report the best performance of CDDA [58] for complete understanding.

Comparisons and implementation details. We testify our proposed model in terms of single-source and multiple-source domain adaptation. Five comparisons are explored in the single source setting, including Principal Component Analysis (PCA), Geodesic Flow Kernel (GFK) [8], Transfer Component Analysis (TCA) [14], Transfer Subspace Learning (TSL) [6] and Joint Domain Adaptation (JDA) [9]. GFK [8] models domain shift by integrating an infinite number of subspaces from the source to the target domain. TCA [3], TSL [6], JDA [9] ARRLS [59] and LSC [20] are four subspace based algorithms, which manages to seek a common shared subspace to mitigate the domain shift. The last two further incorporates the pseudo labels the target data to fight off the conditional distribution divergence across two domains. For subspace-based methods (except LSC), we use the conventional SVM to do classification. Moreover, some deep learning methods are also involved for comparisons. CNN is a powerful network for image classification, which also has been proved that it is effective for learning transferable features [60]. LapCNN, a variant of CNN is proposed based on Laplacian graph regularization. Similarly, DDC is a domain adaptation variant of CNN that adds an adaptation layer between the $fc7$ and $fc8$ layers. DAN embeds the hidden representations of all task-specific layers in a reproducing kernel Hilbert space to address the domain discrepancy [31]. Note that CNN, LapCNN, DDC, and DAN are based on the Caffe [61] implementation of AlexNet [62] trained on the ImageNet dataset.

In the multiple-source scenario, we evaluate on Naive Combination (NC), Adaptive-SVM (A-SVM) [63]; LTSL [10]; SGF [64], [65], SGF-C and SFG-J denote the conference and journal version, respectively; RDALR [28]; FDDL [66] and SDDL [67]. We set the dimension of common space as 100 and λ also as 100 for all algorithms except PCA.

TABLE 5
Performance (%) of our algorithm on *Office+Caltech* of our method with two source domains using SURF features

Dataset	Ours	Dataset	Ours	Dataset	Ours
C,W→ A	54.8	C,D→ A	54.4	D,W→ A	43.8
C,A→ W	52.5	C,D→ W	80.0	A,D→ W	76.3
C,W→ D	80.3	C,A→ D	51.0	A,W→ D	73.9
A,W→ C	40.8	A,D→ C	43.5	D,W→ C	35.1
Average: **57.2**					

For our method, we focus on better exploring the knowledge from the source domain, rather than to seek a better domain-invariant feature space and hence, we adopt the projection P from JDA as the input of our method. Accuracy is adopted to evaluate the performance of all algorithms. Since our method is a clustering based method, the best alignment is applied first, then the accuracy is calculated.

$$Accuracy = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}, \tag{26}$$

where $\delta(x, y)$ equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the ground truth label.

4.2 Object Recognition with SURF Features

Results of single source. Here we verify the effectiveness of our model with single-source scenario. From Table 3, we can notice that our model obtains better results in 9 out of 12 cases over JDA. Specifically, around 10% improvements over the second best algorithm are achieved in $C \rightarrow W$, $A \rightarrow D$ and $D \rightarrow A$. However, our model works much worse on $D \rightarrow W$ and $W \rightarrow D$. Furthermore, we explore multi-source data to enhance the performance.

In the single-source case, the performance drops heavily when we select another source, although some models can obtain very high accuracy, e.g., $D \rightarrow W$ and $A \rightarrow D$, which denotes that different sources play a crucial role in the tasks on target learning. In practice, we don't know which the optimal source domain is in advance, and hence, we always need to build a robust method when we have multiple sources. Even though different source domains have large divergences, such as $A, D \rightarrow W$, we could still achieve a satisfactory output. Since our method is based on JDA, our goal is to show the improvement over JDA. To our best knowledge, we also report the best performance on *Office+Caltech* CDDA [58] for complete understanding.

Results of multiple sources. Here we verify the performance of our model in the multiple-source case. From Table 5, the performance with multiple sources generally performs better than that with single source. This demonstrates that our model can fuse different feature spaces in an effective scheme. Considering the average result, we could obtain 12% improvements over the best result in the single-source case. Although we adopt more source data to obtain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

TABLE 6
Performance (%) on *Office+Caltech* with one source domain using deep features or deep models

Dataset		C→ A	C→ W	C→ D	A→ C	A→ W	A→ D	W→ C	W→ A	W→ D	D→ C	D→ A	D→ W	Average
Deep model	CNN	91.1	83.1	89.0	83.8	61.6	63.8	76.1	49.8	95.4	80.8	51.1	95.4	76.8
	LapCNN	92.1	81.6	87.8	83.6	60.4	63.1	77.8	48.2	94.7	80.6	51.6	94.7	76.4
	DAN	91.3	85.5	89.1	84.3	61.8	64.4	76.9	52.2	95.0	80.5	52.1	95.0	77.3
	DDC	92.0	92.0	90.5	86.0	68.5	67.0	81.5	53.1	96.0	82.0	54.0	96.0	79.9
Deep features	Direct	91.9	79.7	86.5	82.6	74.6	81.5	64.6	74.6	99.4	60.2	72.1	96.6	80.4
	GFK	87.7	75.1	83.1	79.1	79.4	76.7	73.3	84.3	99.3	80.4	85.0	79.7	81.9
	TCA	90.2	81.0	87.3	85.0	82.2	76.9	77.4	82.7	98.2	79.7	87.7	97.0	85.4
	JDA	92.0	85.1	90.4	86.3	88.5	83.8	83.6	87.0	100	83.9	90.3	98.0	88.9
	LSC	94.3	91.2	95.3	87.9	88.8	94.9	88.0	93.3	100	86.2	92.4	99.3	92.6
	ARRLS	93.4	91.5	91.1	88.9	91.2	89.8	87.5	92.4	100	86.6	92.2	99.0	92.2
	Ours	99.0	89.5	91.7	89.8	89.2	91.1	88.3	94.0	99.4	88.2	94.0	98.0	92.7

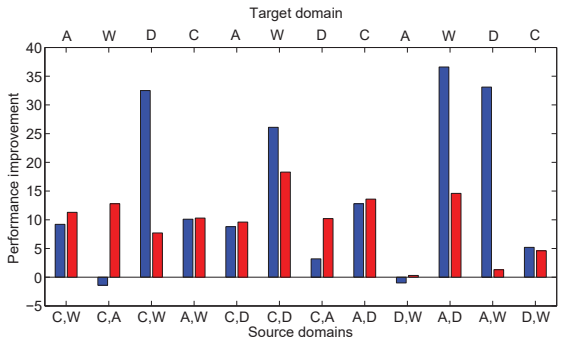


Fig. 2. Performance (%) improvement of our algorithm in the multi-source setting compared to single source setting with SURF features. The blue and red bars denote two source domains, respectively. For example, in the first bar C,W→A, the blue bar shows the improvement of our method with two source domains C and W over the one only with the source domain C.

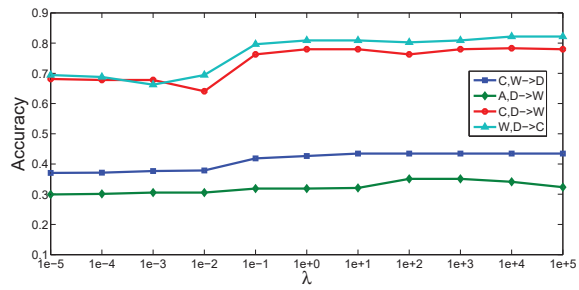


Fig. 3. Parameter analysis of λ with SURF feature on *Office+Caltech*.

TABLE 7
Performance (%) comparison on *Office+Caltech* with multi-source domains using deep features

Source / Target	A,C,D W	A,C,W D	C,D,W A	A,D,W C	Average
Direct	81.7	96.2	82.9	78.0	84.7
A-SVM	81.4	94.9	85.9	78.4	85.2
GFK	79.8	84.9	84.9	79.7	82.3
TCA	86.1	97.5	92.3	84.4	90.1
JDA	92.9	97.5	92.7	88.3	92.9
LSC	93.2	98.7	94.0	88.8	93.6
Ours	94.9	96.2	94.5	88.7	93.6

better performance, it is still very promising. In practice, it is common to get many auxiliary well-labeled datasets. Table 4 lists the performance of different algorithms in the multi-source scenario. Our proposed model renders obvious advantages over the others by over 20% improvements. These comparisons perform even worse than that in single source scenario, which means the competitors learn the deformed common space and degrade the performance when dealing with complex multi-source scenario. Differently, our proposed model preserves all the source structures and adapts well to the target domain learning.

If we take a close look at Figure 2, nearly in all the cases our method in the multi-source setting has substantial improvement over the one in the single source case. This verifies that structure-preserved information from multi-source domains can help to boost the performance.

Parameter analysis. In our algorithm, there is only one parameter λ, which controls the similarity between the learned indicator matrix and ground-truth source labels. We aim to preserve the structure of source domains and adapt to the target domain. We vary λ from 10^{-5} to 10^{5} to evaluate the performance. Figure 3, shows that the performance goes up with the increasing of λ on these 4 datasets, and when λ reaches some certain value, the performance keeps stable. Generally, the performance is good enough with $\lambda = 100$.

4.3 Object Recognition with Deep Features

Deep learning attracts more and more attention in recent years due to the dramatic improvement over the traditional methods. In essence, the features are extracted layer-by-layer for more effective information. In this subsection, we continue to work on the object recognition scenario and evaluate the performance of different unsupervised domain adaptation methods with deep features [57].

First we compare our method with K-means on the target data to demonstrate the benefit of our SP-UDA framework, which is exactly the first part of our framework. Figure 4 shows the performance improvement of our algorithm in the single source setting over K-means with deep features. We can see that our method has nearly 6%-30% improvements over K-means on different datasets, which results from the second structure-preserved term. The categorical utility function U_c is usually to measure the submiliary between two partitions, while we apply U_c to preserve the whole source structure. Different from the traditional pairwise constraints, the source labels are treated as a whole to guide the target data clustering.

Table 6 shows the performance of several unsupervised domain adaptation methods in the single source domain setting. Compared with the results with SURF features in Table 3, the performance has significant improvements with deep features or deep models. This indicates that deep features or deep models are effective to learn the transferable

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

TABLE 8
Performance (%) on *PIE* with one or multi-source and one target setting

Dataset	PCA	GFK	TCA	TSL	JDA	Ours	Dataset	PCA	GFK	TCA	TSL	JDA	Ours	Dataset	PCA	GFK	TCA	TSL	JDA	Ours
7→ 5	24.2	25.2	41.8	46.8	49.2	**57.4**	7→ 5	24.2	25.2	41.8	46.8	49.2	**67.5**	7→ 5	24.2	25.2	41.8	46.8	49.2	**58.3**
9→ 5	21.0	21.8	34.7	37.0	47.8		27→ 5	32.0	34.2	55.6	63.7	64.2		29→ 5	18.9	20.4	27.0	33.3	47.2	
9→ 5	21.0	21.8	34.7	37.0	**47.8**	45.4	9→ 5	21.0	21.8	34.7	37.0	47.8	**58.2**	27→ 5	32.0	34.2	55.6	63.7	64.2	**66.8**
27→ 5	32.0	34.2	55.6	63.7	**64.2**		29→ 5	18.9	20.4	27.0	33.3	47.2		29→ 5	18.9	20.4	27.0	33.3	47.2	
5→ 7	24.8	26.2	40.8	44.1	40.0	**43.8**	5→ 7	24.8	26.2	40.8	44.1	40.0	**60.0**	5→ 7	24.8	26.2	40.8	44.1	40.0	**44.9**
9 → 7	40.1	43.2	47.7	**47.0**	32.0		27→ 7	61.0	62.9	67.8	**72.7**	48.5		29→ 7	23.4	24.6	29.9	34.1	27.3	
9 → 7	40.1	43.2	47.7	47.0	32.0	**51.0**	9 → 7	40.1	43.2	**47.7**	47.0	32.0	**40.1**	27→ 7	61.0	62.9	67.8	**72.7**	48.0	**51.0**
27→ 7	61.0	62.9	67.8	**72.7**	48.5		29→ 7	23.4	24.6	29.9	34.1	27.3		29→ 7	23.4	24.6	29.9	34.1	27.3	
5 → 9	25.2	27.3	41.8	47.5	43.4	**53.1**	5 → 9	25.2	27.3	41.8	47.5	43.4	**51.8**	5 → 9	25.2	27.3	41.8	47.5	43.4	**55.7**
7 → 9	45.5	47.4	51.5	**57.6**	37.87		27→ 9	72.2	73.4	75.9	**83.5**	43.4		29→ 9	27.2	28.5	29.9	36.6	38.5	
7 → 9	45.5	47.4	51.5	**57.6**	37.9	46.7	7 → 9	45.5	47.4	51.5	**57.6**	37.9	**51.0**	27→ 9	72.2	73.4	75.9	**83.5**	43.4	**58.0**
27→ 9	72.2	73.4	75.9	**83.5**	43.4		29→ 9	27.2	28.5	29.9	36.6	38.5		29→ 9	27.2	28.5	29.9	36.6	38.5	
5 → 27	16.3	17.6	29.4	36.2	67.0	**71.6**	5 → 27	16.3	17.6	29.4	36.2	67.0	**71.0**	5 → 27	16.3	17.6	29.4	36.2	67.0	**71.7**
7 → 27	53.4	54.3	64.7	71.4	37.9		9 → 27	46.1	46.4	56.2	59.5	30.9		29→ 27	30.3	31.3	33.6	38.8	46.4	
7 → 27	53.4	54.3	64.7	**71.4**	37.9	50.1	7 → 27	53.4	54.3	64.7	**71.4**	37.9	**58.4**	7 → 27	53.4	54.3	64.7	**71.4**	37.9	**57.1**
9 → 27	46.1	46.4	56.2	**59.5**	30.9		29→ 27	30.3	31.3	33.6	38.8	46.4		29→ 27	30.3	31.3	33.6	38.8	46.4	
5 → 29	16.3	17.6	29.4	36.2	47.1	**50.3**	5 → 29	16.3	17.6	29.4	36.2	47.1	**51.5**	5 → 29	16.3	17.6	29.4	36.2	47.1	**55.3**
7 → 29	25.4	27.1	33.7	35.7	26.2		9 → 29	25.3	26.8	33.2	36.3	29.8		27→ 29	35.1	38.4	40.3	44.8	39.7	
7 → 29	25.4	27.1	33.7	35.7	26.2	**43.1**	7 → 29	25.4	27.1	33.7	35.7	26.2	**38.5**	9 → 29	25.3	26.8	33.2	36.3	29.8	**47.2**
9 → 29	25.3	26.8	33.2	36.3	29.8		27→ 29	35.1	38.4	40.3	**44.8**	39.7		27→ 29	35.1	38.4	40.3	44.8	39.7	
Average: PCA(33.2) GFK(34.7) TCA(43.2) TSL(48.1) 42.2(JDA) **Ours(54.2)**																				

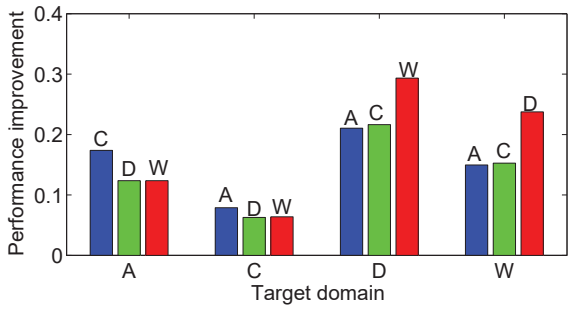


Fig. 4. Performance (%) improvement of our algorithm in the single source setting over K-means with deep features. The letter on each bar denotes the source domain.

features. With deep features, the domain adaptation methods can further boost the performance with positive transfer. One might notice that Direct with deep features gets better performance than deep models, due to that the pre-trained models for deep features might see the images of interest beforehand with extra datasets. Recall that our method is based on the common space learnt by JDA. It is exciting to see that our method has 3.6% improvement over JDA on average. Most existing domain adaptation methods employ the classification for the target data recognition, where only several key data points determine the hyperplane, and the target data are not involved to contribute the decision boundary. Differently, in the SP-UDA framework the whole source structure is utilized for transfer. Moreover, the target data and source data are put together to mutually determine the decision boundary. This indicates that the partition-level constraint can preserve the whole source structure for the guidance of target data clustering, which demonstrates the effectiveness of SP-UDA framework. Even with the simple K-means as the core clustering method, our method can achieve the competitive performance with the state-of-the-art methods.

Next we evaluate the performance in the multi-source setting. Table 7 shows the results with deep features. On the average, the multi-source setting gains slight improvement over the result in single source setting in Table 6 and our method achieves competitive performance compared with rivals. In the last subsection, our model achieves lots of gains with multiple source domains using SURF features; however, less than 1% improvement has been obtained with deep features. If we compare the results in Table 6 and 7, it comes to the same conclusion that it is difficult to boost the result of domain adaptation with deep features. This makes sense since the deep structure exacts discriminative but similar representation. Although this kind of features is promising for recognition, different source domains have too little complementary information for further improvement.

4.4 Face Identification

Domain adaptation results. In the following, we demonstrate our method in face identification. Table 8 lists the results with single or multiple sources and one target setting. Similar phenomenon is achieved. (1) Our model for multi-source domains achieves the best results in most cases; (2) it is difficult to differentiate which source is the best. For instance, the result of $27 \rightarrow 29$ only gets about 40%, although one source case achieves very good results on some datasets, e.g., $27 \rightarrow 9$ and $27 \rightarrow 7$. Our model on multi-source domains benefits the robustness and gets the satisfactory results. Generally, our average performance exceeds others with a large margin.

5 Conclusions

In this paper, we proposed a novel framework for unsupervised domain adaptation named structure-preserved unsupervised domain adaptation (SP-UDA). Different from the current research, which learned a classifier on a source domain and then predicted target labels, we attempted to preserve the whole source structures to facilitate the target learning. Generally speaking, both source and target domains were put together for clustering, which jointly uncovered the structures of both domains. In addition, the well-preserved structure knowledge from the source domain

facilitated and guided the adaptation for the target domain in a semi-supervised clustering fashion. To the best of our knowledge, we were the first to consider domain adaptation problem in a semi-supervised clustering issue with target labels as missing elements. Moreover, we solved the problem by a K-means-like optimization problem in an efficient way. Extensive experiments on some popularly used cross-domains benchmarks verified the large improvements of our proposed model over several state-of-the-art algorithms. In the future, we will analyze the learning bound error in our semi-supervised context.

Acknowledgment

This research is supported in part by the NSF IIS award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.

References

[1] V. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, 2015.

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

[3] S. Pan, J. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction.," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2008.

[4] Y. Zhu, Y. Chen, Z. Lu, S. Pan, G. Xue, Y. Yu, and Q. Yang, "Heterogeneous transfer learning for image classification.," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2011.

[5] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proceedings of European Conference on Computer Vision*, 2010.

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

[7] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proceedings of International Conference on Computer Vision*, 2011.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

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

[10] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 74–93, 2014.

[11] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

[12] L. Duan, D. Xu, I. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1667–1680, 2012.

[13] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *Proceedings of European Conference on Computer Vision*, 2014.

[14] S. Pan, I. Tsang, J. Kwok, and Q.Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[15] H. Liu, M. Shao, and Y. Fu, "Structure-preserved multi-source domain adaptation," in *Proceedings of International Conference on Data Mining*, 2016.

[16] M. Shao, Z. Ding, H. Zhao, and Y. Fu, "Spectral bisection tree guided deep adaptive exemplar autoencoder for unsupervised domain adaptation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[17] D. Kumar, C. Kumar, and M. Shao, "Cross-database mammographic image analysis through unsupervised domain adaptation," in *2017 IEEE International Conference on Big Data*, 2017.

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

[19] Y. Ganin and L. Victor, "Unsupervised domain adaptation by back-propagation," in *Proceedings of International Conference on Machine Learning*, 2015.

[20] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5552–5562, 2016.

[21] Y. Kong, M. Shao, K. Li, and Y. Fu, "Probabilistic low-rank multi-task learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 670–680, 2018.

[22] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proceedings of Advances in Neural Information Processing Systems*, 2009.

[23] Z. Ding and Y. Fu, "Low-rank common subspace for multi-view learning," in *IEEE International Conference on Data Mining*, 2014.

[24] Z. Ding, M. Shao, and Y. Fu, "Incomplete multisource transfer learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 2, pp. 310–323, 2018.

[25] Z. Cui, H. Chang, S. Shan, and X. Chen, "Generalized unsupervised manifold alignment," in *Proceedings of Advances in Neural Information Processing Systems*, 2014.

[26] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *Proceedings of European Conference on Computer Vision*, 2012.

[27] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Proceedings of Advances in Neural Information Processing Systems*, 2013.

[28] I. Jhuo, D. Liu, D. Lee, and S. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[29] L. Duan, I. Tsang, D. Xu, and T. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proceedings of International Conference on Machine Learning*, 2009.

[30] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[31] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of International Conference on Machine Learning*, 2015.

[32] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2016.

[33] M. Long, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of International Conference on Machine Learning*, 2017.

[34] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *AAAI/IAAI*, p. 109, 2000.

[35] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrdl, "Constrained k-means clustering with background knowledge," in *Proceedings of International Conference on Machine Learning*, pp. 577–584, 2001.

[36] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing gaussian mixture models with em using equivalence constraints," in *Advances in Neural Information Processing Systems*, pp. 465–472, 2004.

[37] I. Davidson and S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proceedings of SIAM International Conference on Data Mining*, pp. 201–211, 2005.

[38] D. Pelleg and D.Baras, "K-means with large and noisy constraint sets," in *Proceedings of European Conference on Machine Learning*, pp. 674–682, 2007.

[39] M. Bilenko, S. Basu, and R. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of International Conference on Machine Learning*, pp. 201–211, 2004.

[40] H. Wu and Z. Liu., "Non-negative matrix factorization with constraints," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2010.

[41] H. Liu and Y. Fu, "Clustering with partition-level side information," in *Proceedings of International Conference on Data Mining*, 2015.

[42] H. Liu, Z. Tao, and Y. Fu, "Partition level constrained clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[43] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher, "Spectral learning," in *Proceedings of International Joint Conference of Artificial Intelligence*, 2003.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2843342, IEEE Transactions on Knowledge and Data Engineering

[44] Q. Xu, M. Desjardins, and K. Wagstaff, "Constrained spectral clustering under a local proximity structure assumption," in *Proceedings of International Florida Artificial Inte Research Society Conference*, 2005.

[45] Z. Lu and M. Carreira-Perpinan, "Constrained spectral clustering through affinity propagation," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2008.

[46] X. Ji and W. Xu, "Document clustering with prior knowledge," in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[47] F. Wang, C. Ding, and T. Li, "Integrated kl (k-means-laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations," in *Proceedings of SIAM International Conference on Data Mining*, 2009.

[48] T. Coleman, J. Saunderson, and A. Wirth, "Spectral clustering with inconsistent advice," in *Proceedings of International Conference on Machine learning*, 2008.

[49] Z. Li, J. Liu, and X. Tang, "Constrained clustering via spectral regularization," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2009.

[50] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 1–30, 2014.

[51] B. Mirkin, "Reinterpreting the category utility function," *Machine Learning*, vol. 45, no. 2, pp. 219–228, 2001.

[52] J. Wu, H. Liu, H. Xiong, and J. Cao, "A theoretic framework of k-means-based consensus clustering," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2013.

[53] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, "K-means-based consensus clustering: A unified view," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 155–169, 2015.

[54] H. Liu, T. Liu, J. Wu, and D. T. amd Y. Fu, "Spectral ensemble clustering," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 715–724, 2015.

[55] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1129–1143, 2017.

[56] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[57] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of International Conference on Machine Learning*, pp. 647–655, 2014.

[58] L. Luo, X. Wang, S. Hu, C. Wang, Y. Tang, and L. Chen, "Close yet distinctive domain adaptation," *ArXiv e-prints*, 2017.

[59] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076–1089, 2014.

[60] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proceedings of Advances in Neural Information Processing Systems*, 2014.

[61] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "How transferable are features in deep neural networks?," in *Proceedings of ACM international conference on Multimedia*, 2014.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2012.

[63] J. Yang, R. Yan, and A. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proceedings of International Conference on Multimedia*, 2007.

[64] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proceedings of International Conference on Computer Vision*, 2011.

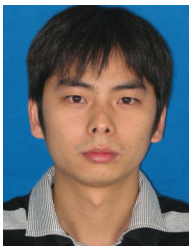
[65] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2288–2302, 2014.

[66] M. . Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proceedings of International Conference on Computer Vision*, 2011.

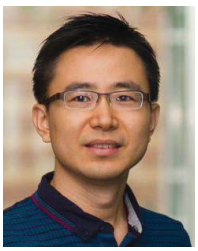
[67] S. Shekhar, V. Patel, H. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

Hongfu Liu received his bachelor and master degree in Management Information Systems from the School of Economics and Management, Beihang University, in 2011 and 2014 respectively. He is currently pursuing the Ph.D. degree in Northeastern University, Boston. His research interests generally focus on data mining and machine learning, with special interests in ensemble learning.

Ming Shao (S'11-M'16) received the B.E. degree in computer science, the B.S. degree in applied mathematics, and the M.E. degree in computer science from Beihang University, Beijing, China, in 2006, 2007, and 2010, respectively. He received the Ph.D. degree in computer engineering from Northeastern University, Boston MA, 2016. He is a tenure-track Assistant Professor affiliated with College of Engineering at University of Massachusetts Dartmouth since 2016 Fall. His current research interests include sparse modeling, low-rank matrix analysis, deep learning, and applied machine learning on social media analytics. He was the recipient of the Presidential Fellowship of State University of New York at Buffalo from 2010 to 2012.



Zhengming Ding (S'14) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from University of Electronic Science and Technology of China (UESTC), China, in 2010 and 2013, respectively. He is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, Northeastern University, USA. His research interests include machine learning and computer vision. Specifically, he devotes himself to develop scalable algorithms for challenging problems in transfer learning scenario. He was the recipient of the Student Travel Grant of ACM MM 14, ICDM 14, AAAI 16 an IJCAI 16. He received the National Institute of Justice Fellowship. He was the recipient of the best paper award (SPIE). He has served as the reviewers for IEEE journals: IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, etc. He is an IEEE student member and AAAI student member.



Yun Fu (S07-M08-SM11) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xian Jiaotong University, China, respectively, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana- Champaign, respectively. He is an interdisciplinary faculty member affiliated with College of Engineer- ing and the College of Computer and Information Science at Northeastern University since 2012. His research interests are Machine Learning, Computational Intelligence, Big Data Mining, Computer Vision, Pattern Recognition, and Cyber-Physical Systems. He has extensive publications in leading journals, books/book chapters and international conferences/workshops. He serves as associate editor, chairs, PC member and reviewer of many top journals and international conferences/workshops. He received seven Prestigious Young Investigator Awards from NAE, ONR, ARO, IEEE, INNS, UIUC, Grainger Foundation; nine Best Paper Awards from IEEE, IAPR, SPIE, SIAM; many major Industrial Research Awards from Google, Samsung, and Adobe, etc. He is currently an Associate Editor of the IEEE Transactions on Neural Networks and Leaning Systems (TNNLS). He is fellow of IAPR and SPIE, a Lifetime Senior Member of ACM, Lifetime Member of AAAI, OSA, and Institute of Mathematical Statistics, member of Global Young Academy (GYA), INNS and Beckman Graduate Fellow during 2007-2008