Journal of Electronic Imaging

JElectronicImaging.org

Iterative landmark selection and subspace alignment for unsupervised domain adaptation

Ting Xiao
Peng Liu
Wei Zhao
Xianglong Tang

Ting Xiao, Peng Liu, Wei Zhao, Xianglong Tang, "Iterative landmark selection and subspace alignment for unsupervised domain adaptation," *J. Electron. Imaging* **27**(3), 033037 (2018), doi: 10.1117/1.JEI.27.3.033037.

Journal of Electronic Imaging 27(3), 033037 (May/Jun 2018)

Iterative landmark selection and subspace alignment for unsupervised domain adaptation

Ting Xiao, Peng Liu, Wei Zhao,* and Xianglong Tang
Harbin Institute of Technology, Pattern Recognition and Intelligent System Research Center, School of Computer Science and Technology, Harbin China

Abstract. Domain adaptation (DA) solves a learning problem in a target domain by utilizing the training data in a different but related source domain, when the two domains have the same feature space and label space but different distributions. An unsupervised DA approach based on iterative landmark selection and subspace alignment (SA) is proposed. The proposed method automatically selects source landmarks from the source domain and iteratively selects target landmarks from the target domain. These well-selected landmarks accurately reflect the similarity between the two domains and are applied to kernel projection of both source and target samples onto a common subspace, where SA is performed. In each iteration, target labels are updated by a classifier that is retrained with the source samples aligned with the target domain. Thus, the distribution of the selected target landmarks gradually approximates the distribution of the source domain. During landmark selection, the quadratic optimization functions are constrained such that the proportions of selected samples per class remain the same as in the original domain, which makes the problem easy to solve and avoids setting hyperparameters. Comprehensive experimental results show that the proposed method is effective and outperforms state-of-the-art adaptation methods. *© 2018 SPIE and IS&T* [DOI: 10.1117/1.JEI.27.3.033037]

Keywords: unsupervised domain adaptation; transfer learning; subspace alignment; object classification.

Paper 180237 received Mar. 16, 2018; accepted for publication May 30, 2018; published online Jun. 18, 2018.

1 Introduction

Machine learning has been widely used in many knowledge engineering areas, including classification, regression, and clustering. However, most existing approaches are based on a common assumption that the training data and testing data are from the same feature space and follow the same data distribution.[1] When the distribution changes, the performance of the original learning system will degrade. Therefore, many models require being rebuilt from scratch with an immense number of training samples. However, in real-world applications, recollecting the training data is prohibitive owing to the considerable human effort involved.[2,3] Moreover, retraining the models without applying the knowledge learned from previous domains or tasks is wasteful.[4] To address these issues, the learner must consider the distribution shifts between the two domains, which is the motivation of domain adaptation (DA).

As a subfield of transfer learning (TL), DA assumes that the learning system has the same tasks but different domains. It is intended to employ information from both source and target domains during the learning process and automatic adapting.[5] There are two main categories of DA methods. They differ in terms of the labeled samples considered for the target domain. When a small set of labeled data is available in the target domain, the problem is semisupervised DA.[6–9] When no labeled data are available in the target domain, the problem is unsupervised DA.[10–13] This paper focuses on the more challenging problem of unsupervised DA in visual object recognition. This is because, in real-world applications, unlabeled target data are often much more abundant and difficult to annotate.

Two types of methods have proven successful for unsupervised DA. One type is the instance reweighting approach,[14–19] which minimizes the source and target distributions by reweighting the most appropriate source samples for the target data. It then trains a classifier on the reweighted source data. The second type is the feature transformation approach,[20–36] which is intended to find or construct a common space wherein the distributions of the two domains are similar. The feature transformation methods are further divided into two main categories: data-centric methods and subspace-based methods.[37]

In recent years, subspace-based DA methods have attracted considerable research interest. These methods share the same principles. First, two domain-specific d-dimensional subspaces for the source data and target data are computed. Then, source and target data are projected into intermediate subspaces, and the distribution shift is modeled by seeking the best intermediate subspaces, such as the method of subspaces by sampling geodesic flow (SGF)[12] and geodesic flow kernel (GFK[29]). In both SGF[12] and GFK,[29] a set of intermediate subspaces is used to model the shift between the two distributions. This can be a costly tuning procedure. Fernando et al.[10] proposed the alignment of the two subspaces directly in the original space (SA). However, these two subspaces have no semantic link or similarity in the original space. In having no link (similarity), learning an optimal projection of one onto the other does not make sense. In the field of DA, the major issue is

*Address all correspondence to: Wei Zhao, E-mail: zhaowei@hit.edu.cn

how to find, express, and utilize similarities between the two domains. Especially for unsupervised DA, where target labels are not available, a means of achieving DA is a challenging problem. An intuitive idea is that some well-selected landmarks in the source and target domains contain the common knowledge between the two domains. Thus, landmarks can serve as bridges connecting the source and target domains.

In this work, a subspace-based feature transformation method named iterative landmark selection and subspace alignment (ILSSA) is proposed for the unsupervised DA problem. Source landmarks are selected from the source domain based on the maximum mean discrepancy (MMD) criterion that the selected source landmarks should have the most similar distribution as the target domain. Target landmarks are iteratively selected from the target domain based on the MMD criterion that the selected target landmarks should have the most similar distribution as the source domain. When selecting target landmarks, target pseudolabels (predicted by the source domain) are used. The role of these labels is to identify target landmarks so that distribution similarities of the selected target landmarks and source domain can be calculated by the MMD criterion.

Source landmarks together with target landmarks are used to construct a common space that contains the shared knowledge of the source and target domains. The distribution shift of the two domains can be reduced by mapping all the source and target samples into this common subspace. Thus, subspace alignment (SA) can further reduce the shift between the two domains in that common subspace, and target pseudolabels will be updated by a classifier trained with the aligned source samples. In the iteration, with the updated target-pseudolabels, targeted landmarks will be selected. The common subspace is also updated by the selected target landmarks. Ultimately, the common knowledge of the two domains in that common subspace will be maximized. Then, all the samples of the two domains are mapped onto this common subspace and SA is performed, which causes the distribution shifts between the two domains to be minimized. Thus, the traditional machine learning method can be performed later to classify the target domain. In this approach, our method can not only avoid the cost-tuning procedure involved by a set of intermediate subspaces but it also allows the source and target domains to be linked by landmarks.

To summarize, our paper provides major contributions to the unsupervised DA problem by the proposed ILS method. In this paper, it is shown how to automatically select landmarks from the source domain, and how to iteratively select landmarks from the target domain when target labels are unavailable. Moreover, when selecting target landmarks, target pseudolabels combined with substitution variables are integrated into a constraint to identify the target landmarks so that the distribution similarities of the selected target landmarks and source domain can be calculated by the MMD criterion. Furthermore, results of comprehensive experiments conducted on standard benchmark datasets for object recognition show that the proposed method outperforms the state-of-the-art algorithms by a significant margin.

The remainder of this paper is organized as follows. Section 2 reviews related work. In Sec. 3, the proposed method based on ILS and SA is introduced. Section 4 provides experimental details and comparisons with other unsupervised DA methods for visual object recognition. The paper is concluded in Sec. 5.

2 Related Work

According to the literature, unsupervised DA methods can be broadly organized into two types: instance reweighting methods and feature transformation approaches. As mentioned in Sec. 1, feature transformation approaches can be further divided into two categories: data-centric methods and subspace-based methods. The methods discussed in this paper are summarized in Table 1.

Instance reweighting methods[14–19] aim to find the most appropriate source samples and reduce the distribution shift by reweighting the source samples based on their relation to the target samples. Dai et al.[14] proposed the TrAdaBoost method, which enables users to employ a small amount of labeled data to leverage the old data. A high-quality classification model is therefore constructed for the data. Since TrAdaBoost[14] transfers knowledge from one source, its performance heavily relies on the relationship between the source and target. Yao and Doretto[15] extended the boosting framework for transferring knowledge from multiple sources, and they proposed the MS-TrAdaBoost method. Sun et al.[16] proposed a two-stage domain adaptation (abbreviated as 2SW-MDA) methodology that combines with the target domain weighted data from multiple sources based on marginal probability differences (first stage) as well as conditional probability differences (second stage). Gong et al.[17] proposed a method (named CDL) that selects samples from the source domain to create a group of auxiliary tasks, whereas landmarks explicitly bridge the source and target domains.[17] However, that method only selects samples from the source domain and leverages them in a semisupervised manner. Instance reweighting methods are simple to implement. However, when the domain difference is substantially large, there will always be some source instances that are not relevant to the target instances, even in the feature-matching subspace.[38]

Data-centric methods are intended to find one unified or two different transformation matrices that project both the source and target data onto a domain-invariant space. Examples include transfer component analysis (TCA[23]) and TL via dimensionality reduction (MMDE[22]), which project source and target data onto a reproducing kernel Hilbert space (RKHS). Accordingly, the marginal distribution of the two domains with respect to the MMD[39] is reduced. Similar to MMDE[22] and TCA,[23] in joint distribution analysis (JDA[24]), not only the marginal distribution but also the conditional distribution is considered to reduce the joint distribution in the RKHS. Transfer joint matching (TJM[25]) improves upon TCA[23] by jointly reweighting instances, and it finds the common subspace in a principled dimensionality reduction procedure to reduce the domain difference. Meanwhile, scatter component analysis[20] finds a representation by taking the between-class and within-class scatter of the source domain into consideration. Unlike seeking one transformation matrix, joint geometrical and statistical alignment (JGSA[26]) learns two coupled projections that project the source domain and target domain data onto a low-dimensional subspace, whereas the geometrical shift

Table 1 DA methods in Sec. 2 listing different characteristics of each method.

Approach	Adaptation category	Target data	Applications
TrAdaBoost[14]	Instance reweighting	Limited labels	Text classification
MS-TrAdaBoost[15]	Instance reweighting	Limited labels	Object recognition
2SW-MDA[16]	Instance reweighting	Unlabeled	Text classification
CDL[17]	Instance reweighting	Unlabeled	Object recognition
TCA[23]	Data-centric methods	Unlabeled	WiFi localization/Text classification
MMDE[22]	Data-centric methods	Unlabeled	WiFi localization/Text classification
JDA[24]	Data-centric methods	Unlabeled	Object recognition
TJM[25]	Data-centric methods	Unlabeled	Object recognition
Scatter component analysis [20]	Data-centric methods	Unlabeled	Object recognition/Synthetic data
JGSA[26]	Data-centric methods	Unlabeled	Object recognition
SGF[28]	Subspace-based	Unlabeled	Object recognition
GFK[29]	Subspace-based	Unlabeled/Limited labels	Object recognition
SA[10]	Subspace-based	Unlabeled/limited labels	Object recognition
SDA[30]	Subspace-based	Unlabeled	Object recognition
LSA[31]	Subspace-based	Unlabeled	Object recognition

and distribution shift are simultaneously reduced. However, when the two domains have a large discrepancy, the appropriate domain-invariant space is extremely difficult to achieve.

Subspace-based methods reduce the distribution shift by moving the source and target subspaces closer so that the subspace of each individual domain contributes to the final mapping. Some subspace-based methods[28,29] project the source and target domains onto a Grassmann manifold, where the two domains are viewed as two points. Thus, the distance between the two domains is the geodesic. Specifically, Gopalan et al.[28] proposed a method (SGF) by creating intermediate representations (points along the geodesic on the Grassmann manifold) of data between the two domains. The intermediate representations are obtained from sampling points along the geodesic. GFK[29] extends and improves on SGF by using a kernel-based method that eliminates the limitation of tuning many parameters required in SGF. Unlike SGF and GFK, SA[10] suggests directly reducing the discrepancy between the two domains by optimizing a linear mapping function that transforms the source subspace into the target one.

Subsequently, the subspace distribution alignment (SDA[30]) method was proposed based on the concept of aligning the distribution as well as the two subspaces. In most cases, only a subset of the source data has a similar distribution as the target domain, and vice versa, as verified by the authors of Ref. 31. Furthermore, landmark-based kernelized subspace alignment (LSA[31]) is a method of selecting landmarks from both the source and target domains to construct a common space for the SA of the two domains. However, LSA faces three limitations in selecting landmarks. First, each landmark is computed independently from others when considering the distribution distance between the source samples and the target samples. The result is that the overall distribution of the selected landmarks may not be close to either the source domain or the target domain. Second, the distance distribution between a landmark and the source (or target) domain is approximately assumed to be a normal distribution. Third, when deciding whether to select a sample as a landmark, the threshold is a hyperparameter that is set according to experience.

In this paper, an ILS method based on the subspace method for the unsupervised DA problem is proposed. These well-selected landmarks accurately reflect the similarity between the target and source domains. Consequently, the common subspace constructed by these landmarks maximizes the common knowledge of the source and target domains, which can therefore avoid negative transfer.[1] Prior to the iteration, the target labels are initially estimated by the classifier trained with the original source samples. During the iteration, the target labels are updated by the classifier trained with the source samples that are aligned to the target domain. Thus, the distribution of selected target landmarks gradually approximates the source domain distribution. A constraint is added such that the proportions per class of landmarks remain the same as in the original data domain. This renders the DA optimization problem easy to solve and avoids the setting of hyperparameters. Comprehensive experiments on standard benchmark datasets

for object recognition demonstrate that our method significantly outperforms the state-of-the-art algorithms.

3 Proposed Approach

In this section, the proposed ILSSA method for unsupervised DA is presented.

3.1 *Problem Statement and Overview*

Let $S = \{x_i^s\}_{i=1}^n$, $x_i^s \in \mathbb{R}^D$ denote n samples in the source domain, let $Y_S = \{y_i^s\}_{i=1}^n$ denote their labels, and let $T = \{x_j^t\}_{j=1}^m$, $x_j^t \in \mathbb{R}^D$ represent m samples without labels in the target domain, where D is the dimension of data samples. The source and target data are assumed to draw the distributions $P(S)$ and $P(T)$, respectively. Unsupervised DA provides the means to reduce the distribution shift between the two domains when the target domain labels are unavailable and when the source and target domains have the same feature space and label space but different distributions $[P(S) \neq P(T)]$.

The key task for DA is to reduce the distribution shift between the two domains.[40] Additionally, the essence of subspace-based methods for DA is to enable the source subspace to align with the target one in a common subspace. Thus, the common subspace plays the critical role of a bridge that connects the source and target domains. Our key insight is that some well-selected samples (landmarks) from both the source and target domains can be considered as the bridge. These well-selected landmarks have the most similar distribution as both the domains. Hence, as long as these landmarks are selected, the common subspace will be determined.

The successive steps in our approach are illustrated in Fig. 1. First, source landmarks are automatically selected from the source domain under the condition that source landmarks have distributions most similar to the target domain. Because the target samples have no labels, the source classifier is used to predict the initial target pseudolabels (denoted by the dotted lines in "T"). Second, target landmarks are selected based on the MMD criterion under the condition that target landmarks have the most similar distribution to the source domain. Both source and target domains have the constraint that the proportions of selected samples per class must remain the same as in the original domain. The union set of both source and target landmarks is then projected to construct the common subspace via the Gaussian kernel. Third, in the common subspace, the source and target subspaces are obtained using the PCA method. Then, the source subspace is aligned to the target subspace using a transformation matrix. After the SA, the representations of the source and target samples are obtained.

Finally, the original target labels can be updated by the target labels predicted by the classifier that was trained on all the aligned source samples. All steps from the second step to the final step are repeated until the last two target landmark selections are the same. During the iteration, common knowledge of both the source and target domains is integrated into the common subspace constructed by the selected landmarks. Thus, the distributions of the source and target domains become increasingly similar.

3.2 *Selection Landmarks*

A subset of samples selected from both the source and target domains is a good set of landmarks. These landmarks construct a common kernel subspace, which contains common knowledge of both the source and target domains. The sample selection criterion is the distribution similarity measurement.

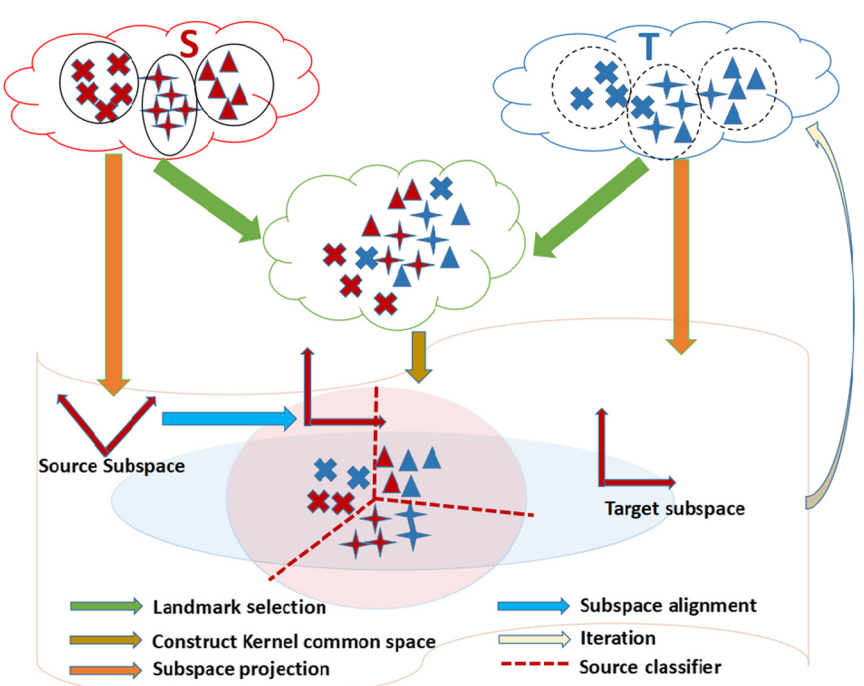

Fig. 1 Steps in the proposed approach (in color online).

3.2.1 *Distribution similarity measurement*

In the field of TL and DA, MMD is a widely used and effective nonparametric metric for comparing the distribution shift based on two sets of data.[41] Given the source and target data S, T, their distributions are $P(S)$ and $P(T)$. By mapping the data to a RKHS using function $\phi(\cdot)$, the MMD (or distribution distance) between $P(S)$ and $P(T)$ is defined as in Eq. (1):

$$\mathrm{MMD}^2(P(S),P(T)) = \sup_{\|\phi\|_H \le 1} \|E_{x^s\sim P(S)}[\phi(x^s)] - E_{x^t\sim P(T)}[\phi(x^t)]\|_{\mathrm{H}}^2, \tag{1}$$

where $E_{x^s\sim P(S)}[\cdot]$ denotes the expectation with regard to the distribution $P(S)$ and $\|\phi\|_H \le 1$ defines a set of functions in the unit ball of an RKHS, H. Based on the statistical tests defined by MMD, it has $\mathrm{MMD}(P(S),P(T)) = 0$ if $P(S) = P(T)$.

To measure the similarity of the distributions between S and T, the empirical MMD is given by Eq. (2):

$$\begin{aligned}\mathrm{MMD}^2(P(S),P(T)) &= \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(x_i^s) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_j^t)\right\|_{\mathrm{H}}^2 \\ &= \left(\sum_{i,j=1}^{n}\frac{1}{n^2}\phi(x_i^s)*\phi(x_j^s) + \sum_{i,j=1}^{m}\frac{1}{m^2}\phi(x_i^t) \right. \\ &\quad \left. *\phi(x_j^t) - \sum_{i,j=1}^{n,m}\frac{2}{nm}\phi(x_i^s)*\phi(x_j^t)\right)^{\frac{1}{2}} \\ &= \left(\sum_{i,j=1}^{n}\frac{1}{n^2}k(x_i^s,x_j^s) + \sum_{i,j=1}^{m}\frac{1}{m^2}k(x_i^t,x_j^t) - \sum_{i,j=1}^{n,m}\frac{2}{nm}k(x_i^s,x_j^t)\right)^{\frac{1}{2}},\end{aligned} \tag{2}$$

where s and t represent the source domain and target domain, respectively, and $\phi(\cdot)$ is the feature map associated with the kernel map, $k(x_1,x_2) = \phi(x_1)*\phi(x_2)$. In addition, $k(x_1,x_2)$ is the kernel function that maps the source and target data to the RKHS.

3.2.2 *Landmark selection in the source domain*

To identify the samples that should be selected as landmarks, each sample in the source domain corresponds to a binary indicator variable, α_i^s, where $\alpha_i^s = 1$ means that the i'th sample is selected as a landmark, and $\alpha_i^s = 0$ means that it is not selected as a landmark. Thus, for n samples in the source domain, there are n indicator variables, $\{\alpha_i^s\}_{i=1}^n$, which can be represented as $\alpha_{\mathbf{s}} = [\alpha_1^s, \alpha_2^s, \cdots \alpha_n^s]$. The goal is to choose among all possible configurations of $\boldsymbol{\alpha}_{\mathbf{s}}$, such that the distribution of the selected landmarks is maximally similar to that of the target domain. The most appropriate $\boldsymbol{\alpha}_{\mathbf{s}}$ will be chosen such that the distribution difference of the target data and the selected source subset is minimized. It can be denoted as shown in Eq. (3):

$$\boldsymbol{\alpha}_{\mathbf{s}} = \arg\min_{\boldsymbol{\alpha}_{\mathbf{s}}} \left\| \frac{1}{\sum_i^n \alpha_i^s}\sum_{i=1}^{n}\alpha_i^s\phi(x_i^s) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_j^t)\right\|^2, \tag{3}$$

where the first term is the distribution expectation of selected source landmarks and the second term is the distribution expectation of the target domain. Furthermore, the constraint that labels are balanced in the selected landmarks[17] is imposed. That is, the proportions of source samples per class are enforced to remain the same as in the original domain. For example, suppose there are 100 samples in the source domain, 60 in class A and 40 in class B. Among the selected source domain landmarks, class A accounts for 60% and class B accounts for 40%. For the source domain, this constraint can be written as in Eq. (4):

$$\begin{aligned}&\frac{1}{\sum_i^n \alpha_i^s}\sum_{i}^{n}\alpha_i^s y_{ic} = \frac{1}{n}\sum_{i=1}^{n} y_{ic}, \\ &\{y_{ic} = 1, (0) | y_i = c, (y_i \ne c),\ \forall\ 1 \le c \le C\}.\end{aligned} \tag{4}$$

Here, c is the index of the class, C denotes the total number of classes, and y_{ic} is a binary variable indicating whether the i'th sample belongs to class c (e.g., if $y_i = c$, then $y_{ic} = 1$; otherwise, $y_i \ne c$, $y_{ic} = 0$). Note that the optimization of Eq. (3) requires no labels of target data; only source sample labels are required.

Owing to the binary constraint on $\boldsymbol{\alpha}_{\mathbf{S}}$, the optimization problem is intractable. Instead, to solve the relaxed problem by including another variable, $\beta_i^s = \alpha_i^s / \sum_i^n \alpha_i^s$, all these variables can be represented by vector $\boldsymbol{\beta}_{\mathbf{s}} = [\beta_1^s, \beta_2^s, \cdots \beta_n^s]^{\mathrm{T}}$. Obviously, for β_i^s, it has

$$\left\{\beta_i^s : 0 \le \beta_i^s \le 1, \sum_i^n \beta_i^s = 1,\ \forall\ 1 \le i \le n\right\}. \tag{5}$$

Substituting β_i^s into Eq. (3), the source optimization problem can be expressed as follows:

$$\begin{aligned}&\arg\min_{\alpha_S} \left\| \frac{1}{\sum_i^n \alpha_i^s}\sum_{i=1}^{n}\alpha_i^s\phi(x_i^s) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_j^t)\right\|_{\mathrm{H}}^2 \\ &= \arg\min_{\boldsymbol{\beta}_S} \left\| \sum_{i=1}^{n}\beta_i^s\phi(x_i^s) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_j^t)\right\|^2 \\ &= \sum_{i,j=1}^{n}(\beta_i^s)^{\mathrm{T}}\phi(x_i^s)*\phi(x_j^s)\beta_i^s - \frac{2}{m}\sum_{i,j=1}^{n,m}(\beta_i^s)^{\mathrm{T}}\phi(x_i^s) \\ &\quad *\phi(x_j^t) + \frac{1}{m^2}\sum_{i,j=1}^{m}\phi(x_i^t)*\phi(x_j^t) \\ &= \sum_{i,j=1}^{n}(\beta_i^s)^{\mathrm{T}}k(x_i^s,x_j^s)\beta_i^s - \frac{2}{m}\sum_{i,j=1}^{n,m}(\beta_i^s)^{\mathrm{T}}k(x_i^s,x_j^t) \\ &\quad + \frac{1}{m^2}\sum_{i,j=1}^{m}k(x_i^t,x_j^t) \\ &= \boldsymbol{\beta}_s^{\mathrm{T}}\mathbf{A}_{ss}\boldsymbol{\beta}_s - \frac{2}{m}\boldsymbol{\beta}_s^{\mathrm{T}}\mathbf{B}_{st}\mathbf{1}_{m\times 1} + \frac{1}{m^2}\mathbf{1}^{\mathrm{T}}\mathbf{C}_{tt}\mathbf{1},\end{aligned} \tag{6}$$

where $\mathbf{A}_{ss} \in \mathbb{R}^{n\times n}$ is the kernel matrix computed over the source domain, $\mathbf{B}_{st} \in \mathbb{R}^{n\times m}$ denotes the kernel matrix computed between the source and target domains, $\mathbf{1}_{m\times 1}$ represents an all-one column matrix, and $\mathbf{C}_{tt}$ is the kernel matrix computed over the target domain. The selection of landmarks depends on the kernel mapping $\phi(\cdot)$ and its parameters. When computing the kernel matrix, the

Gaussian kernel is used. Take an item, $\forall\, b_{i,j} \in \mathbf{B}_{st}$, as example. It can be computed by Eq. (7):

$$b_{i,j} = \exp\{-(x_i^{(s)} - x_j^{(t)})^T \mathbf{K}(x_i^{(s)} - x_j^{(t)})/\sigma^2\}, \tag{7}$$

where $\mathbf{K}$ is a positive semidefinite matrix, the value of $(\cdot)$ is s or t, and σ denotes the scaling factor for measuring distances and similarities between data. For the sake of comparison, for both $\mathbf{K}$ and σ, they follow the typical setup as in GFK[29] in our experiments.

Then, combining Eqs. (3)–(7), the optimization problem is transformed into a quadratic programming problem, as shown in Eq. (8), and the optimization goal is also transformed from solving α_s to solving $\boldsymbol{\beta}_s$:

$$\begin{cases} \boldsymbol{\beta}_s = \arg\min\limits_{\boldsymbol{\beta}_s} \left(\boldsymbol{\beta}_s^{\mathrm{T}} \mathbf{A}_{ss} \boldsymbol{\beta}_s - \frac{2}{m} \boldsymbol{\beta}_s^{\mathrm{T}} \mathbf{B}_{st} + \frac{1}{m^2} \mathbf{C}_{tt} \right) \\ \text{states that } \sum_i^n \beta_i^s y_{ic} = \frac{1}{n} \sum_i^n y_{ic}, 1 \le c \le C \\ \sum_i^n \beta_i^s = 1, 0 \le \beta_i^s \le 1 \end{cases}. \tag{8}$$

After obtaining the solution of β_i^s, the binary weights α_i^s can be obtained by thresholding β_i^s as in Eq. (9):

$$\alpha_i^s = \begin{cases} 1, & \beta_i^s > th \\ 0, & \beta_i^s \le th \end{cases}, \tag{9}$$

where th is a very small positive real number. Hence, the source landmarks are selected and denoted as set $L_s = \{x_i^s | \text{if}, \alpha_i^s = 1\}_{i=1}^n$.

3.2.3 *Iterative landmark selection in target domain and subspace alignment*

To enable the source and target domain to share as much common knowledge as possible, landmarks in the target domain that have a similar distribution as the source domain should also be selected.

Similar to the source landmark selection, the samples that can be selected as landmarks from the target domain should have their binary indicator variables $\boldsymbol{\gamma}_t = [\gamma_j^t]_{j=1}^m$, where m is the number of target samples. If $\gamma_j^t = 1$, it indicates that the j'th sample is selected as a target landmark; otherwise, the sample is rejected. Thus, the target optimization function is shown in Eq. (10):

$$\boldsymbol{\gamma}_t = \arg\min_{\boldsymbol{\gamma}_t} \left\| \frac{1}{n} \sum_{i=1}^n \phi(x_i^s) - \frac{1}{\sum_j^m \gamma_j^t} \sum_{j=1}^m \gamma_j^t \phi(x_j^t) \right\|. \tag{10}$$

Here, the first term is the distribution expectation of the source domain and the second term is the distribution expectation of the selected target landmarks. Observe that the target optimization function has a similar form as the source optimization function. However, there are obvious differences and additional changes required for target landmark selection because target labels are not available in the unsupervised DA setting. It cannot be solved in the manner of the source optimization function. We observe that there are distribution shifts between the two domains. Nevertheless, the target pseudolabels predicted by a classifier trained on source samples can also reflect the real category of the target domain to some extent. Thus, target pseudolabels are used, and the target optimization function can be solved in the same manner as solving the source optimization function under the constraint condition in Eq. (11):

$$\frac{1}{\sum_j^m \gamma_j^t} \sum_j^m \gamma_j^t y_{jc}^t = \frac{1}{m} \sum_{j=1}^m y_{jc}^t, \{y_{jc}^t | y_j^t = c, \ \forall\ 1 \le c \le C\}, \tag{11}$$

where y_{jc}^t is a binary variable and is determined by the target pseudolabels. If $y_{jc}^t = 1$, it indicates that the label of the j'th sample belongs to the c'th class; otherwise, $y_{jc}^t = 0$.

Once target landmarks are selected, source landmarks are added to kernelized mapping of all points onto the common subspace, where SA is effectively performed to reduce the distribution shift between the two domains. Because the common subspace contains shared knowledge between the two domains, the source and target domains are associated. Therefore, SA can further reduce the distribution shift. Then, target pseudolabels can be updated by a classifier trained with the aligned source samples. Additionally, target pseudolabels are increasingly approximated to the true labels of the target in the iteration. Next, processes of iterative target landmark selection and SA are detailed.

In the initial step, a base classifier is trained using the labeled samples from the source domain. The base classifier can be any standard learner. In this study, a support vector machine (SVM), SVM_l, is used to predict target labels for the l'th iteration. The initial target pseudolabels are predicted by SVM_0. Inevitably, some prediction errors occur because of the distribution shift between the source and target domains. Then, with the target labels, the classification indicator variable y_{jc}^t is substituted by the initial prediction $y_{jc}^{t(l)}$, $l = 0$. Equation (10) is optimized to obtain the initial landmarks of the target domain under the constraint condition of Eq. (11). This landmark set $L_t^{(l)}, l = 0$ of the target domain is composed of the samples with the indicator variable $\gamma_j^t = 1$.

Both the source landmark set L_s and target landmark set $L_t^{(l)}, l = 0$ comprise the landmarks set $L^{(l)} = L_s \cup L_t, l = 0$. Subsequently, to link the source and target domain as much as possible, a kernel trick is used to nonlinearly map all the source samples and target samples onto a common subspace constructed by these landmarks. In this paper, the Gaussian kernel is applied, and its standard deviation σ is set to the median distance between all the source data and target data.[31] Each point x_i^s from the source data and each target sample x_j^t are projected onto each landmark $p \in L^{(l)}$, as shown in Eq. (12):

$$\begin{aligned} K_S(i,p) &= \exp\left(\frac{-\|x_i^s - p\|^2}{2\sigma^2}\right); \\ K_T(j,p) &= \exp\left(\frac{-\|x_j^t - p\|^2}{2\sigma^2}\right). \end{aligned} \tag{12}$$

Thus, in the common subspace constructed by these selected landmarks, the representations of the source and target samples are obtained by Eq. (12) and denoted by $\mathbf{K}_S$ and $\mathbf{K}_T$, respectively.

Even though both the source samples S and target samples T are mapped onto the same subspace and are linked to each other, their distributions remain different owing to the shift in their subspaces. It is necessary to perform an SA to further reduce the distribution shift. SA involves finding a linear

transformation matrix $\mathbf{M}$ that best aligns the source subspace coordinate system to the target one. PCA is separately applied on each domain, and the largest d eigenvectors $\mathbf{X}_S$ and $\mathbf{X}_T$ are extracted as the source and target subspace. The transformation matrix $\mathbf{M}$ is learned by minimizing the following Frobenius norm shown in Eq. (13):

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \|\mathbf{X}_S\mathbf{M} - \mathbf{X}_T\|_f^2. \tag{13}$$

Note that the bases of the source (target) subspace are orthogonal, $\mathbf{X}_S\mathbf{X}_S^{\mathrm{T}} = \mathbf{I}$; hence, the solution of Eq. (13) is $\mathbf{M}^* = \mathbf{X}_S^{\mathrm{T}}\mathbf{X}_T$.

The source and target data can be projected onto their respective subspaces by the operations $\mathbf{K}_S\mathbf{X}_S$ and $\mathbf{K}_T\mathbf{X}_T$, respectively. Then, the representations of source and target data in the aligned subspace are obtained by using the following respective equations:

$$\mathbf{P_s} = \mathbf{K}_S\mathbf{X}_S\mathbf{M}*; \qquad \mathbf{P}_T = \mathbf{K}_T\mathbf{X}_T, \tag{14}$$

where $\mathbf{P}_S$ and $\mathbf{P}_T$ are the transformations of representations of the source and target domains in the aligned target subspace. In other words, all samples of both domains are represented in the same subspace. The classifier is trained using the transformed samples, $\mathbf{P}_S$. This classifier then predicts the target pseudolabels. With the updated target pseudolabels, the class indicator variables y_{jc}^t in Eq. (11) are updated and the sample selection indicator variables are obtained by optimizing Eq. (10) under the constraint condition of Eq. (11).

Subsequently, the target landmarks $L_t^{(l)}$, $l \leftarrow l+1$ are reselected according to the sample selection indicator variables, γ_j^t. The union set of the common subspace is reconstructed as $L^{(l)} = L_s \cup L_t^{(l)}$, $l \leftarrow l+1$. The processes of target landmark selection, common subspace projection, SA, and target label updating are iteratively performed until $L_t^{(l+1)} = L_t^{(l)}$.

3.3 *Algorithm*

In the proposed method, source landmarks are selected according to the distribution similarity with the target domain, as described in Sec. 3.2.2. The initial target pseudolabels predicted by a classifier trained on all the original source samples are used to select the target landmarks under the MMD criterion. Both source and target quadratic optimization equations have the constraint that the proportions of selected samples per class must remain the same as in the original domain. Then, both the source and target landmarks are used to construct the common subspace by kernel projection. Subsequently, all source samples are projected onto the common subspace and aligned with the target subspace. Then, the original target pseudolabels are updated by the target pseudolabels predicted by a classifier trained on all the transformed source samples. By repeating the above process of target landmark selection, source and target sample projection, and SA, the target pseudolabels can be iteratively updated. In the iteration, common knowledge of both the source and target domains is integrated into the common subspace. Thus, the distribution shift is decreased at each step.

The complete pseudocode of our ILSSA method is described in Algorithm 1.

Algorithm 1 Iterated landmark selection-based SA and classification

Input: Source data S and its labels Y_s, target data T, subspace dimension d

Output: Predicted target labels: $\hat{Y}_t$

1 Obtain $\boldsymbol{\beta}_S$ by solving Eq. (8) and obtain α_S by solving Eq. (9). Thus, obtain the source landmarks L_s;

2 $\hat{Y}_t^{(l)} \leftarrow \mathrm{SVM}_0(S, Y_s, T)$, $l = 0$; //Use the classifier trained on all the source samples to predict initial target pseudo-labels.

3 Repeat;

4 Obtain $\boldsymbol{\gamma}_t$ by solving Eq. (10), and obtain the target landmarks $L_t^{(l)}$. All the landmarks $L^{(l)} \leftarrow L_s \cup L_t^{(l)}$;

Compute $\mathbf{K}_S$, $\mathbf{K}_T$ by Eq. (12); compute $\mathbf{X}_S$ and $\mathbf{X}_T$ by $\mathbf{X}_S \leftarrow PCA(\mathbf{K}_S, d)$ and $\mathbf{X}_T \leftarrow PCA(\mathbf{K}_T, d)$; compute $\mathbf{M}^*$ by Eq. (13);

Compute source and target data representations $\mathbf{P}_S$, $\mathbf{P}_T$ by Eq. (14);

Predict the new target labels: $\hat{Y}_t^{(l+1)} \leftarrow \mathrm{SVM}(\mathbf{P}_S, Y_S, \mathbf{P}_T)$; //Use the classifier trained by all the transformed source samples in the aligned target subspace to predict the new target labels;

If $L_t^{(l+1)} == L_t^{(l)}$ go to End; else, Repeat;

5 End

4 Experiments

This section describes the evaluation of the proposed method. The evaluation was set in the context of object recognition using standard datasets and protocols for evaluating the visual DA method, as in Refs. 10, 28, 29, and 42. Additionally, several state-of-the-art methods were compared with our ILSSA method: TCA,[23] GFK,[29] SA,[10] JDA,[24] connecting the dots with landmarks (CDL[17]), TJM,[25] SDA,[30] LSA[31], return of frustratingly easy (CORAL[32]), and JGSA.[26] The parameters used in the experiments were recommended by its original papers for all the baseline methods.

4.1 *Datasets and Data Preparation*

To evaluate all DA methods, our experiments were conducted on the standard datasets—Office and Caltech10—which contain four domains. The Office dataset consists of three different types of real-world object images from Amazon (denoted by A; images downloaded from online merchants), Webcam (denoted by W; low-resolution images obtained from a web camera), and Dslr (denoted by D; high-resolution images obtained from a digital SLR camera). Caltech256[43] contains 256 object classes downloaded from Google Images, and the Caltech10 dataset contains 10 classes selected from Caltech256.[43] These classes are common to the three domains of the Office dataset. Each dataset was treated as a separate domain.

The number of images per class in the four domains ranged from 8 to 151, and the total number of images in

the four domains was $|A| = 958$, $|C| = 1123$, $|D| = 157$, and $|W| = 295$. Owing to its small number of images, D was not used as a source domain. Thus, by randomly selecting two different domains as the source and the other as the target, nine possible domain pairs were constructed, such as A→D, C→A, and W→C. All experiments followed the standard procedures for feature extraction and experiment protocols of Refs. 10, 28, 29, and 42. The SURF features were quantized into an 800-bin histogram with codebooks computed via K-means on a subset of images from Amazon.com. Then, the histograms were normalized and the z-score was applied such that there was a zero mean and unit standard deviation in each dimension within each domain.

4.2 *Experimental Setup*

For subspace-based unsupervised DA methods, optimal subspace dimensions are important and should be automatically selected. In this study, the subspace disagreement measure (SDM)[29] based on selected landmarks was used to automatically find optimal dimensions. In the experiments, a selection of dimensions from two ways was established: the optimal dimensions found by landmark-based SDM (ISLSA-AdaPCA) and dimensions (10–20), which are widely used by other (SA,[10] LSA,[31] SDA[30]) subspace-based DA methods (ISLSA-PCA).

For the method of ISLSA-AdaPCA, landmarks selected from the source and target domain contained common knowledge of the two domains. Thus, the common subspace (PCA_L) constructed by theses landmarks had similarity with both the source subspace (PCA_S) and target subspace (PCA_T). Intuitively, if two datasets have similar landmarks, then all three subspaces should not be too distant from each other. SDM captures this notion and is defined in terms of the principal angles, $D(d) = 0.5(\sin\alpha_d + \sin\beta_d)$, where α_d denotes the d'th principal angle between the PCA_S and PCA_L, β_d denotes the d'th principal angle between the PCA_T and PCA_L. In addition, $\sin\alpha_d$ or $\sin\beta_d$ is called the minimum correlation distance.[44]

Note that $D(d)$ is at most one. A small value indicates that both α_d and β_d are small; thus, PCA_S and PCA_T are aligned at the d'th dimension. If $D(d) = 1(\alpha_d = \beta_d = \pi/2)$, the two subspaces have orthogonal directions. In this case, PCA_L has almost no similarity with PCA_S and PCA_T. Hence, DA will become difficult because variances captured in one subspace will be unable to transfer to the other subspace. To identify the optimal dimension, d, a greedy strategy is adopted:

$$d^* = \min\{d | D(d) = 1 - \varepsilon, \varepsilon > 0\},$$

where ε is a very small positive real number. Intuitively, the optimal d^* should be as high as possible to preserve variances in the source domain for the purpose of building good classifiers. Nonetheless, it should not be so high that the two subspaces start to have orthogonal directions.

To confirm the validity of our landmark-selection-based method, three baselines methods were conducted:

(1) Selecting landmarks randomly (RD): Randomly select 300 landmarks from the source and target domains (150 for each domain) to construct the common kernel space and repeat the selection task five times to obtain the average behavior.

(2) Selecting all the source and target samples (ALL): In this setting, all samples are used to construct the common kernel space.

(3) Our method without iteration (ILSSA-0): The initial target labels are used to solve Eq. (9), only L_s and $L_t^{(0)}$ are used to construct the common space, and SA is performed only once.

(4) ILSSA: Our proposed method for ILSSA.

Distribution similarity is central to the landmark-based method and SA. Therefore, verification experiments on distribution similarities were conducted. In addition to the baselines, our method was compared with a series of state-of-the-art methods that were proposed recent years. All results were obtained under the published procedures with parameters given in the respective papers.

For a fair comparison, all experiments followed the same evaluation protocols.[17–31] SVM with a linear kernel was trained on the labeled source data and tested on the unlabeled target data. In the experiments, when selecting landmarks, for both $\mathbf{K}$ and σ in Eq. (6), the CDL[17] experimental setup was followed. Kernel matrix $\mathbf{K}$ for computing the distances was chosen as the kernel from the GFK method[29] using all instances, and σ were chosen as $\sigma_q = 2^q\sigma_0$, where $q \in \{-6, -5, \cdots, 5, 6\}$. The σ_0 is the median distance computed over all pairwise data in Eq. (6). When using the landmarks to construct the Gaussian kernel common space, the standard deviation σ was set to the median distance between all the source data and target data.

4.3 *Experimental Results and Analysis*

4.3.1 *Optimal dimension selection and comparison with baselines*

The experimental results of using the optimal dimensions found by landmark-based SDM (ISLSA-AdaPCA) and dimensions widely used by other (SA,[10] LSA,[31] SDA[30]) subspace-based DA methods (ISLSA-PCA) are reported in Table 2. In the table, it is observed that the results of ISLSA-AdaPCA are better than those of ISLSA-PCA in almost all of the subproblems.

Furthermore, the correlation of SDM and the accuracy with respect to dimensions is shown in Fig. 2 (the left

Table 2 Accuracy (%) compared with different subspace dimensions.

Method	A→D	A→W	A→C	W→A	W→D	W→C	C→A	C→D	C→W	Average
ISLSA-PCA	43.31	43.4	44.1	37.27	87.9	32.59	56.05	53.5	49.49	49.71
ISLSA-AdaPCA	45.10	44.41	44.43	37.58	87.9	33.04	56.68	54.91	51.59	50.63

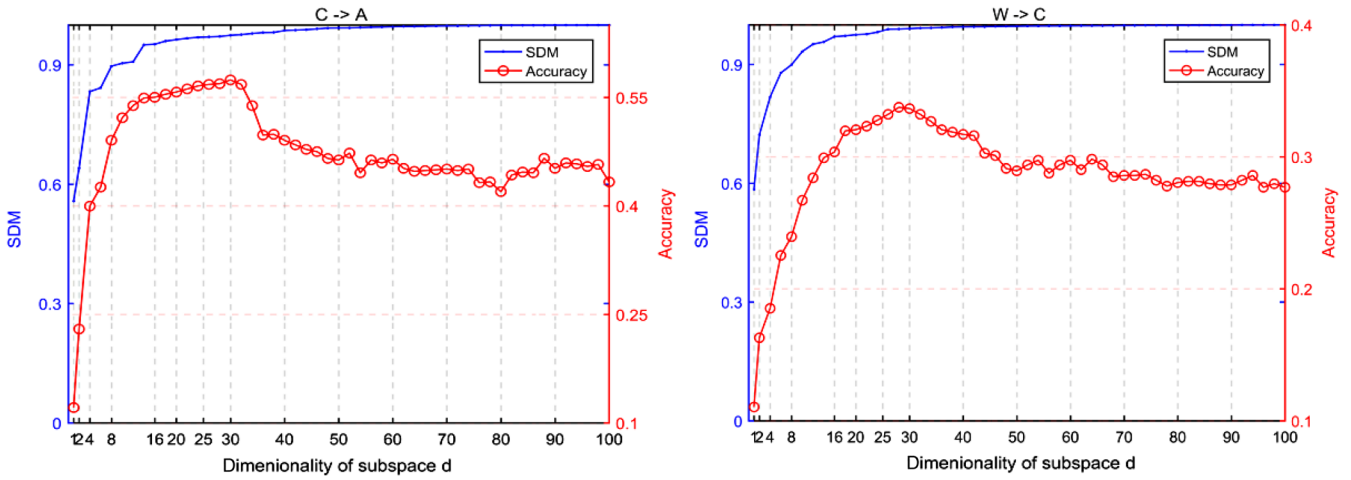

Fig. 2 Selecting the optimal dimensionality d with SDM and the accuracy (in color online).

side is the subproblem of C→A; the right is W→C). In Fig. 2, the horizontal axis is the subspace dimensions; the right vertical axis reports accuracies; and the left vertical axis reports the SDM values. As the dimension increases, SDM rises quickly and eventually reaches its maximum value of one owing to the geometric structures of bases. Meanwhile, as the dimension increases, the DA accuracy gradually increases until the SDM is very close to one. Then, the accuracy begins to decline until it converges to a smaller value. Similar trends are observed on other subproblems. Thus, the optimal dimension should be a point, where SDM is no more than one and the dimension may not be too small to preserve variances in the source data.

Note that the optimal dimension d^* selected by SDM is usually in the range of [25,35]. It is larger than dimensions (10 to 20) used on other subspace-based DA methods (SA,[10] LSA,[31] and SDA[30]). Therefore, to better highlight the effectiveness of our method, and for the sake of a fair comparison, the later experiments will be conducted by using the dimension ($d = [10, 20]$) as SA[10] and LSA.[31]

The classification results of our ILSSA method and the three baselines are reported in Table 3. From the results, it can be observed that the ILSSA method significantly outperforms the other baselines. The average classification accuracy of ILSSA is 49.74%. Among the nine DA tasks, ILSSA achieves the highest accuracy in seven subproblems.

Note that the baselines "RD" and "ALL" perform no adaptation, because they only randomly select some samples, or they roughly use all the samples to define the common subspace. They do not aim at specifically moving the distributions close to each other. Thus, the accuracy of RD is lower than those of both ILSSA-0 and ILSSA, which indicates the importance of selecting good landmarks. Moreover, our method outperforms the noniteration method ILSSA-0 (target landmarks are selected with the target initial pseudolabels; kernel mapping and SA are performed only once) by a large margin. It justifies the effectiveness in iteratively selecting better target landmarks and refining the target pseudolabels.

The convergence property of ILSSA is also evaluated in Fig. 3. It is shown that, with each iteration, the classification accuracy gradually increases until it converges to an optimum value. This result shows that, in the iteration, better target landmarks are selected, which enables the common subspace to contain more shared knowledge. Then, after SA, the distribution of source and target domains becomes smaller and smaller. Therefore, the accuracy is increased in each iteration.

4.3.2 *Comparison with landmark selection methods*

We compared our proposed method with two state-of-the-art landmark selection methods (LSA[31] and CDL[17]). In LSA,[31] each landmark is computed independently from others when considering the distribution similarity with the source data and the target data. For CDL,[17] landmarks are selected only from the source domain to create a group of auxiliary tasks, where landmarks explicitly bridge the source and target domains in a semisupervised manner.

Table 4 illustrates the results. For better interpretation, the results are also visualized in Fig. 4. In the figure, it is worth noting that ILSSA significantly outperforms CDL and LSA

Table 3 Accuracy (%) on Office and Caltech datasets compared with three baselines.

Method	A →D	A→W	A→C	W→A	W→D	W→C	C→A	C→D	C→W	Average
RD	38.8	40.3	42.3	32.9	84.0	28.4	47.5	41.2	40.6	44.00
ALL	39.4	41.0	44.7	33.0	85.3	33.0	49.6	41.4	41.6	45.44
ILSSA-0	40.8	41.7	40.3	34.1	84.0	29.9	46.1	39.4	45.8	44.68
ILSSA	43.3	43.4	44.1	37.3	87.9	32.6	56.1	49.5	53.5	49.74

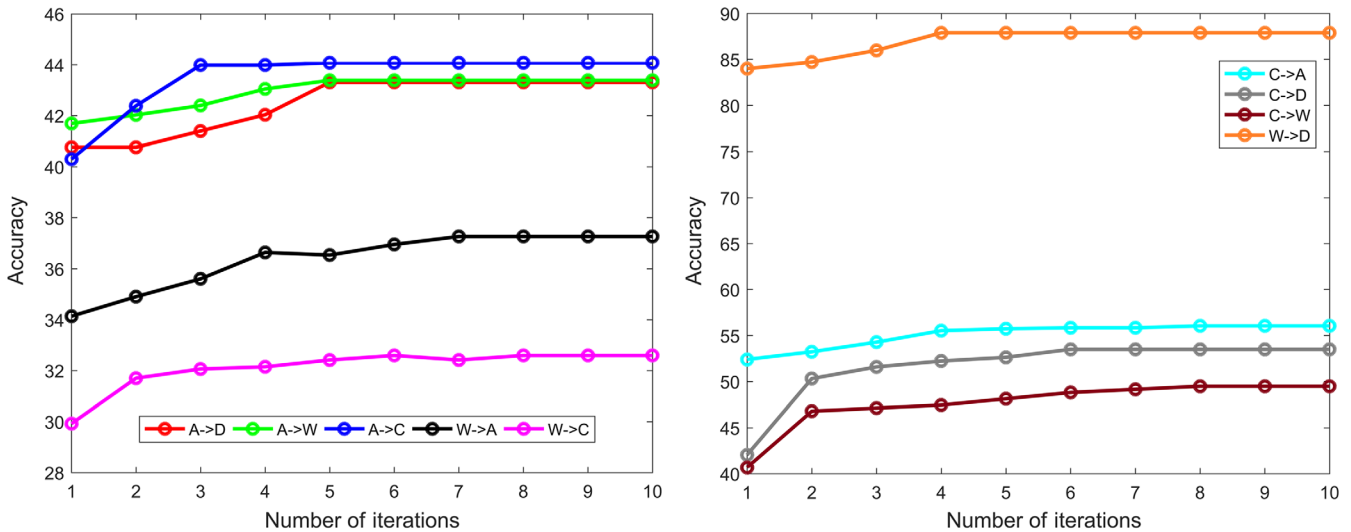

Fig. 3 Accuracy with respect to iterations.

Table 4 Accuracy (%) compared with the LSA method.

Method	A→D	A→W	A→C	W→A	W→D	W→C	C→A	C→D	C→W	Average
CDL[17]	42	41	43.8	34.9	73.3	27.6	56.4	46.5	46.8	45.81
LSA[31]	38.2	42.7	44.2	36.0	86.0	30.5	52.3	49.7	42.0	46.84
ILSSA	43.3	43.4	44.1	37.3	87.9	32.6	56.1	49.5	53.5	49.74

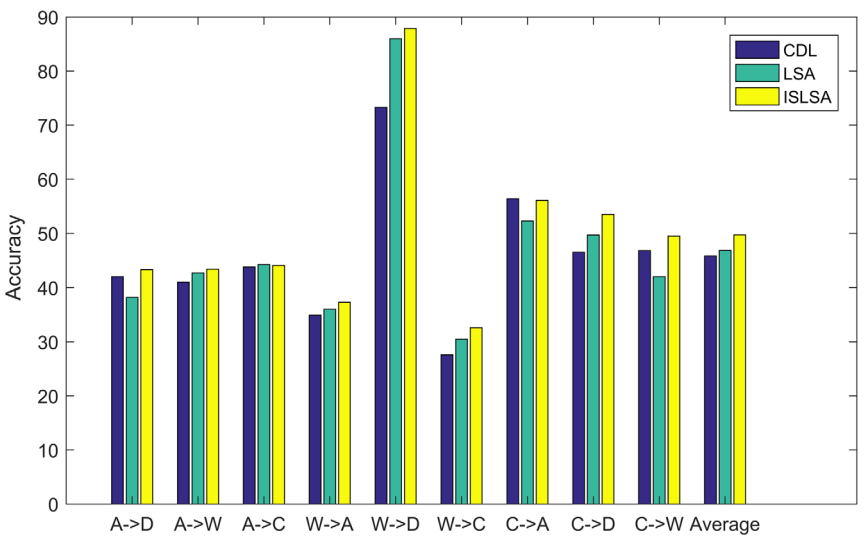

Fig. 4 Recognition accuracy of LSA, CDL, and ILSSA.

in seven out of nine DA subproblems. For the other two subproblems (A→C and C→A), the ILSSA results are very close to the best results. Moreover, ILSSA gains a significant performance improvement of 3.93% compared to CDL and 2.9% compared to LSA. This performance can be attributed to its advantages, which are outlined as follows. (1) For the unsupervised DA problem, labeled data are only available in the source domain. Therefore, CDL only selects landmarks from the source domain for DA. Meanwhile, ILSSA uses source domain classifiers to predict target pseudolabels and updates target pseudolabels during iterations. Landmarks are used to construct the common subspace emerging between the source domain and the target domain. Thus, the two domains are associated by that common subspace. (2) When selecting landmarks, ILSSA takes the overall distribution similarity of selected landmarks into account, whereas LSA independently computes each landmark's similarity. This verifies that ILSSA can identify more effective and adaptable landmarks for the unsupervised DA problem.

Furthermore, the effectiveness of ILSSA by inspecting the distribution distance was verified. For each DA pair, the MMD distance between the selected landmarks and the source (target) domain was computed. Note that a smaller distribution distance implies better generalization performance of the feature representation across domains in the common subspace. The results are shown in Fig. 4. Red represents our ILSSA method; blue represents LSA. The symbols of a circle (o) and star (*) indicate the MMD distance between the selected landmarks and the source (and the target) data, respectively, while the plus sign (+) indicates the average MMD distance of both the domains and the selected data.

Figure 5 shows that, for each identical symbol, almost all of the red is below the blue except for W→D. As these two domains have the least amount of data, there is an inclination to use all samples for adaptation (the total number of W and D is 452: LSA uses 448, and our method uses 274). This demonstrates that our iterative method selects better landmarks that have closer distributions to both the source and target data than LSA. This is the primary reason that our approach achieves better performance than LSA.

4.3.3 *Comparison with state-of-the-art methods*

Table 5 reports the results of the experimental comparison between state-of-the-art methods based on subspace. For better visualization, the results are shown in Fig. 6, where red symbols indicate the methods for achieving the best performance for each subproblem, and blue symbols indicate the methods for achieving the worst performance in each subproblem. It is observed in Fig. 6 that ILSSA achieves significantly better performance than the state-of-the-art methods. In terms of the best and worst results, ILSSA achieves the four best performances and none of the worst performances. From Table 5, we note that the results of JDA[24] and JGSA[26] are close to ours. Although the average

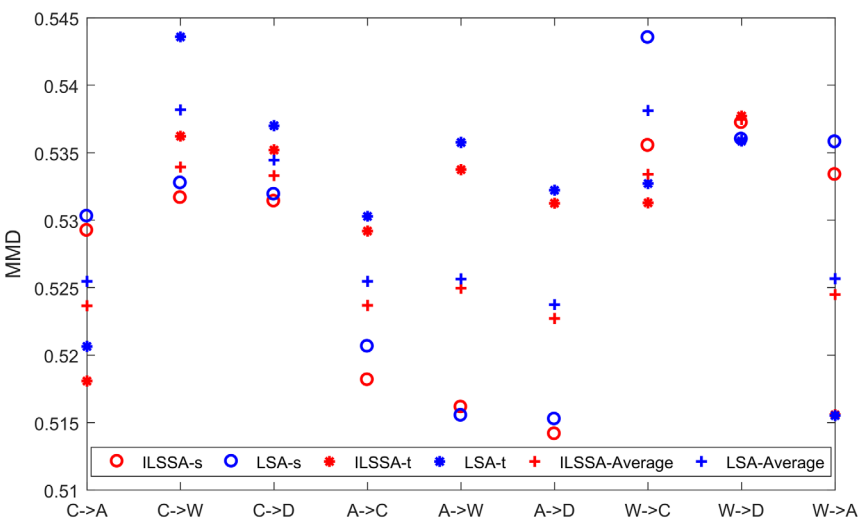

Fig. 5 MMD between the selected landmarks and the source (and target) domain for LSA and ILSSA (in color online).

Table 5 Accuracy (%) compared with other state-of-the-art methods.

Method	A→D	A→W	A→C	W→A	W→D	W→C	C→A	C→D	C→W	Average
TCA[23]	39.1	40.1	40	40.2	77.5	33.7	46.7	41.4	36.2	43.88
GFK[29]	40.1	37.0	40.7	27.6	85.4	24.8	46.0	40.8	37.0	42.13
SA[10]	38.8	39.6	39.9	39.4	77.9	31.8	46.1	39.4	38.9	43.53
JDA[24]	40.1	46.8	44.0	39.0	85.4	33.6	54.6	47.1	51.9	49.17
TJM[25]	40.8	46.8	40.3	31.6	84.7	32.0	43.6	43.3	41.0	44.90
SDA[30]	33.8	30.9	39.5	39.3	75.8	34.7	49.70	40.1	39.0	42.53
CORAL[32]	38.3	38.7	40.3	37.8	84.9	34.6	47.2	40.7	39.2	44.63
JGSA[26]	47.1	54.6	41.1	40.6	71.3	30.6	55.7	48.4	51.5	48.99
ILSSA	43.3	43.4	44.1	37.3	87.9	32.6	56.1	49.5	53.5	49.74

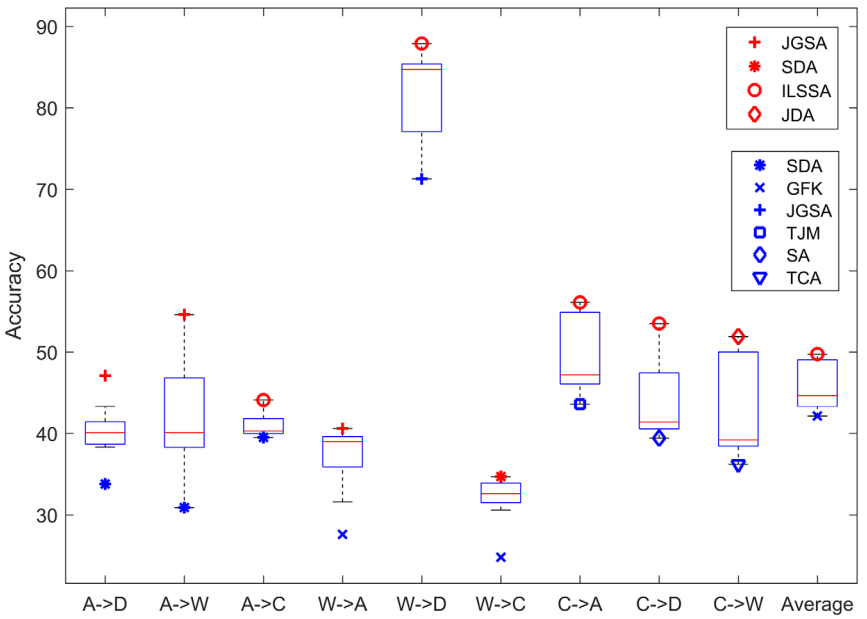

Fig. 6 Accuracy shown in a box-plot comparison of the proposed method and other state-of-the-art approaches. Red (blue) symbols indicate the methods for achieving the best (worst) performance in each subproblem (in color online).

result of ILSSA is only slightly better than JDA,[24] it achieves four of the best performances on nine subproblems, whereas JDA[24] achieves only one. JGSA[26] achieves three of the best performances, whereas our average accuracy is higher than those of all cited methods.

JDA[24] applies statistical properties (marginal and conditional distribution) to seek a unified common subspace in a principled dimensionality reduction procedure. JGSA[26] applies statistical and geometrical properties to learn two coupled projections that project two domain data items into low-dimensional subspaces, where SA is performed to reduce the domain shift. However, as mentioned earlier, data-centric methods (e.g., JDA[24] and JGSA[26]) will fail when the two domains have a large discrepancy. This is because such a low-dimensional common subspace may not exist, where the statistical distributions of two domains are the same and the data properties are also maximally preserved. ILSSA selects landmarks similar to both domains to construct the intermediate common subspace related to both the source and target domains. Thus, the common subspace serves as a bridge between the two domains, wherein performing SA can further reduce the domain shift. Hence, it is direct and effective.

The runtime complexities on the top five DA methods (ILSSA, JDA,[24] JGSA,[26] TJM,[25] CORAL[32]) were evaluated on all DA pairs with their SURF features. All experiments were run by MATLAB-2016b on a system with Windows 10, and the CPU version was an Intel(R) Core(TM) i7-7700 CPU @3.6 GHz.

The results are reported in Table 6. From the average runtime, CORAL[32] achieves the best results, followed by TJM,[25] JDA,[24] JGSA,[26] and ILSSA. CORAL[32] aligns the input feature distributions of the source and target domains by exploring their second-order statistics. Thus, it only requires computation of the covariance statistics in each domain and applying the whitening and recoloring linear transformation to the source features. ILSSA solves two constrained quadratic programming functions. Its performance depends on the number of iterations. Thus, it requires the most time. Both TJM and JDA involve computing the kernel matrix and solving the generalized eigen-decomposition problem. However, TJM[25] applies a low-rank approximation

Table 6 Time complexity of ILSSA and the top five methods.

Runtime (s)	A→D	A→W	A→C	W→A	W→D	W→C	C→A	C→D	C→W	Average runtime	Average accuracy
TJM[25]	8.16	9.58	25.78	8.36	1.39	10.32	27.83	12.25	14.12	13.09	44.9
JDA[24]	22.59	27.55	69.18	25.43	3.66	32.02	71.89	29.96	35.4	35.3	49.17
CORAL[32]	0.56	0.59	0.76	0.72	0.55	0.76	0.71	0.54	0.58	0.64	44.63
JGSA[26]	37.28	44.87	112.40	40.21	5.96	52.37	114.19	44.83	57.66	56.64	48.99
ILSSA	61.14	78.85	185.12	135.58	32.77	156.75	170.91	68.19	85.91	108.34	49.74

to solve the optimization function. Therefore, its complexity can be greatly reduced. JGSA is ∼1.6 times slower than JDA.[24] This is because JGSA[26] simultaneously learns two mappings, and the matrix size for eigen-decomposition is doubled compared to JDA.[24] Although ILSSA has a longer running time than other methods, its average accuracy is the highest.

5 Conclusion

This paper proposed the ILSSA method for the unsupervised DA problem. ILSSA automatically selects source landmarks from the source domain and iteratively selects target landmarks from the target domain. These well-selected landmarks are used to construct the common subspace, where the landmarks' distribution similarity with both the domains has a significant impact on the prediction performance of the target samples. The proposed method estimates the initial target pseudolabels using the original source classifier and then updates them using a classifier trained with the source samples that are projected to the common subspace and aligned with the target domain. The target sample prediction errors decrease during iteration, such that some target landmarks can be selected and have smaller distribution shift with the source domain. The balance of the number of selected samples in each class in both domains is used as the constraint condition for the MMD function, making the MMD function a quadratic optimization function. Comprehensive experimental results demonstrate that ILSSA is effective and it outperforms state-of-the-art adaptation methods.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (Grant Nos. 61671175, 61370162, and l61672190).

References

1. S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010).
2. K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data* **3**(1), 9 (2016).
3. L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: a survey," *IEEE Trans. Neural Networks Learn. Syst.* **26**(5), 1019–1034 (2015).
4. V. M. Patel et al., "Visual domain adaptation: a survey of recent advances," *IEEE Signal Process. Mag.* **32**(3), 53–69 (2015).
5. J. Zhang, W. Li, and P. Ogunbona, "Transfer learning for cross-dataset recognition: a survey,"
6. A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," in *Advances in Neural Information Processing Systems*, pp. 478–486 (2010).
7. A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach," in *Advances in Neural Information Processing Systems*, pp. 181–189 (2010).
8. R. Mehrotra, R. Agrawal, and S. A. Haider, "Dictionary based sparse representation for domain adaptation," in *Proc. of the 21st ACM Int. Conf. on Information and Knowledge Management*, pp. 2395–2398, ACM (2012).
9. H. Daumé, III, "Frustratingly easy domain adaptation," in *ACL*, Vol. 256 (2007).
10. B. Fernando et al., "Unsupervised visual domain adaptation using subspace alignment," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 2960–2967 (2013).
11. M. Baktashmotlagh et al., "Unsupervised domain adaptation by domain invariant projection," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 769–776 (2013).
12. R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(11), 2288–2302 (2014).
13. Y. Lin et al., "Cross-domain recognition by identifying joint subspaces of source domain and target domain," *IEEE Trans. Cybern.* **47**(4), 1090–1101 (2017).
14. W. Dai et al., "Boosting for transfer learning," in *Proc. of the 24th Int. Conf. on Machine Learning*, pp. 193–200, ACM (2007).
15. Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1855–1862, IEEE (2010).
16. Q. Sun et al., "A two-stage weighting framework for multi-source domain adaptation," in *Advances in Neural Information Processing Systems*, pp. 505–513 (2011).
17. B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Int. Conf. on Machine Learning*, pp. 222–230 (2013).
18. J. Huang et al., "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems*, pp. 601–608 (2007).
19. J. Geng and Z. Miao, "Domain adaptive boosting method and its applications," *J. Electron. Imaging* **24**(2), 023038 (2015).
20. M. Ghifary et al., "Scatter component analysis: a unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(7), 1414–1430 (2017).
21. M. Baktashmotlagh, M. Harandi, and M. Salzmann, "Distribution-matching embedding for visual domain adaptation," *J. Mach. Learn. Res.* **17**(1), 3760–3789 (2016).
22. S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. of the Twenty-Third AAAI Conf. on Artificial Intelligence*, pp. 677–682 (2008).
23. S. J. Pan et al., "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks* **22**(2), 199–210 (2011).
24. M. Long et al., "Transfer feature learning with joint distribution adaptation," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 2200–2207 (2013).
25. M. Long et al., "Transfer joint matching for unsupervised domain adaptation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1410–1417 (2014).
26. J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1859–1867 (2017).
27. S. Tang et al., "Domain adaptation of image classification based on collective target nearest-neighbor representation," *J. Electron. Imaging* **25**(3), 033006 (2016).
28. R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 999–1006, IEEE (2011).
29. B. Gong et al., "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2066–2073, IEEE (2012).
30. B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation," in *British Machine Vision Conf.*, pp. 24.1–24.10 (2015).
31. R. Aljundi et al., "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 56–63 (2015).
32. B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. of the 30th AAAI Conf. on Artificial Intelligence*, pp. 2058–2065 (2016).
33. E. Zhong et al., "Cross domain distribution adaptation via kernel mapping," in *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 1027–1036, ACM (2009).
34. J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. of the 2006 Conf. on Empirical Methods in Natural Language Processing*, pp. 120–128, Association for Computational Linguistics (2006).
35. A. Gretton et al., "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*, pp. 513–520 (2007).
36. M. Baktashmotlagh et al., "Domain adaptation on the statistical manifold," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2481–2488 (2014).
37. Y. Yang and T. Hospedales, "Zero-shot domain adaptation via kernel regression on the grassmannian," arXiv:1507.07830 (2015).
38. A. Margolis, "A literature review of domain adaptation with unlabeled data," Technical Report, pp. 1–42 (2011).
39. A. Gretton et al., "A kernel two-sample test," *J. Mach. Learn. Res.* **13**(March), 723–773 (2012).
40. S. Ben-David et al., "A theory of learning from different domains," *Mach. Learn.* **79**(1), 151–175 (2010).
41. K. M. Borgwardt et al., "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics* **22**(14), e49–e57 (2006).
42. B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1785–1792, IEEE (2011).

43. G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," (2007).
44. J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proc. of the 25th Int. Conf. on Machine Learning*, pp. 376–383, ACM (2008).

Ting Xiao is a PhD degree candidate at the School of Computer Science and Technology, Harbin Institute of Technology (HIT). She received her master's degree in computer application technology from Harbin Institute of Technology in 2016. Her research interests cover image processing, computer vision, and machine learning.

Peng Liu is an associate professor at the School of Computer Science and Technology, HIT. He received his doctoral degree in microelectronics and solid-state electronics from HIT in 2007. His research interests cover image processing, computer vision, transfer learning, reinforcement learning, pattern recognition, and design of very large-scale integration circuit.

Wei Zhao is an associate professor at the School of Computer Science and Technology. She received her doctoral degree in computer application technology from HIT in 2006. Her research interests cover pattern recognition, image processing, and deep-space target visual analysis.

Xianglong Tang is a professor at the School of Computer Science and Technology, HIT. He received his doctoral degree in computer application technology from HIT in 1995. His research interests cover pattern recognition, aerospace image processing, medical image processing, and machine learning.