



Joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation

Shiva Noori Saray¹ · Jafar Tahmoresnezhad¹

Received: 4 February 2020 / Revised: 10 June 2020 / Accepted: 13 July 2020 / Published online: 29 July 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

In many real-world knowledge transfer and transfer learning scenarios, the known common problem is distribution discrepancy (i.e., the difference in type, distribution and dimensionality of features) between source and target domains. In this paper, we introduce joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation (JDSC) method, which is an iterative two-step framework. JDSC is based on hybrid of feature-based and classifier-based approaches that uses the feature-based techniques to tackle the challenge of domain shift and classifier-based techniques to learn a reliable model. In addition, for subspace alignment, weighted joint geometrical and statistical alignment is proposed to learn two coupled projections for mapping the source and target data into respective subspaces by accounting the importance of marginal and conditional distributions, differently. The proposed method has been evaluated on various real-world image datasets. JDSC gets 86.2% average classification accuracy on four standard domain adaptation benchmarks. The experiments demonstrate that our proposed method achieves a significant improvement compared to other state of the arts in average classification accuracy. Our source code is available at <https://github.com/jtahmores/JDSC>.

Keywords Transfer learning · Domain adaptation · Subspace alignment · Classification · Distribution discrepancy

1 Introduction

Nowadays, communications across social media and content sharing applications increase the information volume (i.e., image, text and video) exponentially where the classification is an essential requirement to take the advantages of information explosion, efficiently [1]. However, the manual classification of data may be prohibitive. Therefore, the machine learning models are used to classify the information with a basic assumption of machine learning models on which the used data for training and test sets must be drawn from the same or similar distributions. But, in real world, this assumption is not guaranteed in many applications and consequently, the trained machine learning models in source domain may not work well in target domain under various conditions. Thus, domain adaptation (DA) [2] as one of the transfer learning (TL) [3] solutions is used to solve such cross-domain learning problems with different distributions.

DA is a technique for knowledge transfer from the labeled source domain to unlabeled target domain by exploiting domain invariant structures that facilitate the transfer between different domains with different distributions [4]. In this paper, we focus on unsupervised domain adaptation where the target data labels are not accessible in transfer learning phase.

Based on the type of transferred information, the TL algorithms can be classified into three different learning paradigms as follows [5]: (i) instance-based transfer learning, feature-representation transfer learning and classifier-based transfer learning. In instance-based transfer learning approaches, instead of using the entire source domain, some parts of the source data that have similar distribution with target data are reused in the learning phase. (ii) Feature-representation transfer learning approaches aim to obtain new representation of source and target domains to minimize the distribution discrepancy between domains. (iii) Classifier-based transfer learning approaches assume that the performance of target classifier can be improved using source classifier. Ensemble learners can be called as an example of classifier-based TL methods that combine multiple source classifiers to create an improved target classifier. This paper

✉ Jafar Tahmoresnezhad
j.tahmores@it.uut.ac.ir

¹ Faculty of IT and Computer Engineering, Urmia University of Technology, Urmia, Iran

follows combination of feature-based and classifier-based approaches.

In this paper, we propose a novel joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation (JDSC) method that improves the model accuracy, significantly, in an unsupervised manner.

In this method, subspace alignment and label prediction function are learned iteratively to find better representation for data and consequently better prediction function for labeling. In subspace alignment, for reducing the distributional and geometrical divergences between domains, two coupled projections are obtained that map the source and target data into respective subspaces, simultaneously. Also, a domain-invariant classifier is learned in new representations of data with structural risk minimization, while consistency between the classifier and intrinsic manifold structure of data is maximized using marginal distributions. The contributions of this work are summarized as follows.

- (1) In this paper, a novel unsupervised domain adaptation approach is introduced that is based on hybrid of feature-based and classifier-based approaches, which uses the feature-based techniques to address the challenge of domain divergence and classifier-based techniques to learn the reliable classifier.
- (2) For subspace alignment, weighted joint geometrical and statistical alignment (WJGSA) is proposed where it is the modified version of joint geometrical and statistical alignment (JGSA) [4] to learn two coupled projections to map the source and target data into respective subspaces by accounting the importance of marginal and conditional distributions, separately and quantitatively.
- (3) JDSC reduces the divergence of source and target subspaces and increases the variance of target data while maintaining the data structure.
- (4) The proposed method has been evaluated on following four real-world image datasets: object recognition (Office-10 and Caltech-10) [6], handwritten digit recognition (USPS and MNIST) [7,8], large image recognition dataset (ImageNet, VOC 2007) [9] and face dataset [10] to compare it against several novel state-of-the-art methods where the experiments demonstrate that our proposed method achieves a significant improvement in average classification accuracy.

In the rest, the paper is organized as follows. The second part of paper provides an overview on related work in this field. In the third section, the proposed method is described in detail. In the fourth section, the evaluated datasets are presented in detail. In the fifth section, the results of the proposed algorithm against other machine learning and domain adaptation methods are reported. Finally, the paper concludes with some suggestions for future works in the last section.

2 Related work

In general, TL aims to adapt trained models in an existing domain (source) to solve the classification problem in a new (target), yet related, domain. Based on what is transferred, TL algorithms are categorized into three different paradigms as follows.

The strategy behind instance-based approaches is to use the reweighted instances in the source domain to label the target domain. Asgarian et al. [11] proposed a hybrid instance-based transfer learning method that uses a probabilistic weighting strategy to transfer knowledge from the source domain to learn a model for target domain.

Feature-representation transfer learning approaches can be categorized into two different types, data-oriented and subspace-oriented methods where data-oriented approaches are divided into symmetric and asymmetric feature-based TL [4,12]. The data-oriented category focuses on subspace learning by exploiting the underlying representative structures between both domains to find common latent space (features) to reduce the marginal distribution differences between source and target domains (i.e., symmetric) [13] or focuses on distribution alignment by transforming the features of source domain to be closer to target domain to reduce the marginal or conditional distribution divergences between domains (i.e., asymmetric) [14]. For reducing the domain shift in the subspace-oriented category, subspaces of both domains without clearly considering the distribution shift between projected data of domains are manipulated for final mapping [15]. In this approach, the assumption of existing a unified transformation to reduce the domain shifts does not exist.

In classifier-based transfer approaches, transferring of prior knowledge of parameters from source to target domain is considered. Rubin et al. [16] focused on creating an ensemble model from two boosting-based classifiers, gradient tree boosting and adaptive boosting, based on prediction average to predict the transferring of pediatric populations from the hospital general ward to the pediatric intensive care unit. Our work belongs to the feature-representation transfer learning and classifier-based transfer categories. In feature-representation transfer, the distribution shift across domains is reduced by two coupled projections for source and target data to map into respective subspaces, while the data properties are preserved. Moreover, the shift across subspace geometries is reduced alongside reducing the distribution shifts of both domains by quantitative importance evaluation of both distributions (i.e., marginal and conditional distributions) via considering their different effects. Hence, the proposed feature-representation algorithm in this paper is a hybrid of data-based symmetric and subspace-based categories.

Also, in classifier-based transfer, a domain-invariant classifier is learned on new obtained representation of data to

overcome the feature distortions. In addition, for maximizing the consistency between the classifier and the intrinsic manifold structure of data, manifold regularization is used.

3 Joint subspace and model learning

In this section, we first define the problem setting and the purpose of domain transfer learning. Then, we present our proposed approach, joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation, in detail.

3.1 Notation

We first define the notations that are frequently used in this paper. A domain D consists of the following two terms: a feature space X and a marginal probability distribution $P(\mathbf{x})$, i.e., $D = \{X, P(\mathbf{x})\}$ where X is drawn from distribution $P(x)$ and $\mathbf{x} \in X$. Subsequently, given domain D , a task T is defined by a label space Y and a prediction function $f(\mathbf{x})$, i.e., $T = \{Y, f(\mathbf{x})\}$, where $y \in Y$, and $f(\mathbf{x}) = Q(y|\mathbf{x})$ that can be interpreted as the conditional probability distribution. In unsupervised domain adaptation, the source domain $D_s = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ has sufficient labeled data, while in target domain $D_t = \{\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+m}\}$ no labeled data exist. The goal of domain transfer learning is to learn a target model $f: x_t \rightarrow y_t$ with labeled source domain to minimize the prediction error in target domain, under the following assumptions, $X_s = X_t, Y_s = Y_t, P_s(x_s) \neq P_t(x_t)$ and $Q_s(y_s|x_s) \neq Q_t(y_t|x_t)$. Moreover, $tr(\cdot)$ and I are defined as the trace of matrix and identity matrix, in turn. Also, $\|\cdot\|_F^2$ and $\|\cdot\|_K^2$ denote the squared of Frobenius norm and squared norm in reproducing kernel Hilbert space, respectively.

3.2 Proposed method

In this paper, we focus on following three main goals to achieve: (1) obtaining two coupled projections for source and target domains, to reduce the domain divergence, specifically, by accounting the different importance among the marginal and conditional distributions; (2) minimizing the classification error on new representation of source domain labeled data; (3) maximizing the manifold consistency underlying the marginal distributions of source and target domains; and (4) finding the optimal representation and classifier, iteratively.

3.2.1 Weighted joint geometrical and statistical alignment

In this section, we introduce weighted joint geometrical and statistical alignment method which is the modified version

of joint geometrical and statistical alignment. Our proposed method adapts the marginal and conditional distributions with different importance to adapt across domains. In fact, JGSA finds two coupled subspaces to obtain the new representations of source and target domains by considering equal importance for each distribution, whereas our idea considers the relative importance of each distribution, quantitatively and separately. According to domain shift scale, one of the distributions (i.e., marginal or conditional) becomes more important in domain adaptation. Therefore, we define Eq. (1) which aims to find two coupled subspaces A and B for source and target domains, respectively, by quantitative evaluation of marginal and conditional distributions significance, as follows:

$$\min_{A, B} tr \left([A^T \ B^T] \begin{bmatrix} M_s & M_{st} \\ M_{ts} & M_t \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right) \quad (1)$$

where

$$\begin{aligned} M_s &= X_s((1-\gamma)L_s + \gamma \sum_{c=1}^C L_s^{(c)})X_s^T, \\ L_s &= \frac{1}{n_s^2} \mathbf{1}_s \mathbf{1}_s^T, \\ (L_s^{(c)})_{ij} &= \begin{cases} \frac{1}{(n_s^{(c)})^2} x_i, x_j \in X_s^{(c)} \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (2)$$

$$\begin{aligned} M_t &= X_t((1-\gamma)L_t + \gamma \sum_{c=1}^C L_t^{(c)})X_t^T, \\ L_t &= \frac{1}{n_t^2} \mathbf{1}_t \mathbf{1}_t^T, \\ (L_t^{(c)})_{ij} &= \begin{cases} \frac{1}{(n_t^{(c)})^2} x_i, x_j \in X_t^{(c)} \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (3)$$

$$\begin{aligned} M_{st} &= X_s((1-\gamma)L_{st} + \gamma \sum_{c=1}^C L_{st}^{(c)})X_t^T, \\ L_{st} &= -\frac{1}{n_s n_t} \mathbf{1}_s \mathbf{1}_t^T, \\ (L_{st}^{(c)})_{ij} &= \begin{cases} -\frac{1}{n_s^{(c)} n_t^{(c)}} x_i \in X_s^{(c)}, x_j \in X_t^{(c)} \\ 0 & \text{otherwise,} \end{cases} \\ M_{ts} &= X_t((1-\gamma)L_{ts} + \gamma \sum_{c=1}^C L_{ts}^{(c)})X_s^T, \\ L_{ts} &= -\frac{1}{n_s n_t} \mathbf{1}_t \mathbf{1}_s^T, \end{aligned} \quad (4)$$

and

$$(L_{ts}^{(c)})_{ij} = \begin{cases} -\frac{1}{n_s^{(c)} n_t^{(c)}} x_j \in X_s^{(c)}, x_i \in X_t^{(c)} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\mathbf{1}_s \in \mathbb{R}^{n_s}$ and $\mathbf{1}_t \in \mathbb{R}^{n_t}$ denote the column vector with all ones related to the source and target domains, in turn. In addition, γ is an adaptive parameter that induces the importance of marginal and conditional distributions, quantitatively, which is computed through Eq. (6),

$$\gamma \approx 1 - \frac{d_M}{d_M + \sum_{c=1}^C d_c} \tag{6}$$

where d_M and d_c are the marginal and conditional \mathcal{A} distances [17], respectively. \mathcal{A} -distance is defined as Eq. (7) in which $\epsilon(h)$ is a linear classifier error in source D_s and target D_t domains' classification.

$$d_M(D_s, D_t) = 2(1 - 2\epsilon(h)) \tag{7}$$

In addition, for reducing shift across source and target subspaces (i.e., A and B), Eq. (8) is utilized:

$$\min_{A, B} \|A - B\|_F^2 \tag{8}$$

Moreover, Eq. (9) maximizes the variance of target domain with the goal of preserving target data properties by projecting the features into the relevant dimensions,

$$\max_B \text{tr} \left(B^T S_t B \right), \text{ s.t., } S_t = X_t H_t X_t^T \tag{9}$$

where S_t is the scatter matrix of target domain and H_t is the centering matrix. For a good domain adaptation, it is better to maintain the discriminative information of source data within finding a new subspace for source domain. Therefore, Eqs. (10) and (11) are used to preserve the information of source domain using labeled samples in source domain. The purpose of this work is to find a subspace (A) for source domain that converges the samples with same classes and diverges the samples in different classes as follows:

$$\begin{aligned} &\max_A \text{tr} \left(A^T S_b A \right), \text{ s.t.,} \\ &S_b = \sum_{c=1}^C n_s^{(c)} (m_s^{(c)} - \bar{m}_s) (m_s^{(c)} - \bar{m}_s)^T \end{aligned} \tag{10}$$

$$\begin{aligned} &\min_A \text{tr} (A^T S_w A), \text{ s.t.,} \\ &S_w = \sum_{c=1}^C X_s^{(c)} (H_s^{(c)}) (X_s^{(c)})^T \end{aligned} \tag{11}$$

where S_b is the between-class scatter matrix and S_w is the within-class scatter matrix of source domain. Also, $m_s^{(c)}$ and \bar{m}_s are the average of source samples that belong to class c and the average of source samples, respectively. Considering Eqs. (1), (8), (9), (10) and (11), the objective function is achieved as follows:

$$\max_{A, B} \frac{\text{tr} \left([A^T B^T] \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)}{\text{tr} \left([A^T B^T] \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu) I \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)} \tag{12}$$

By optimizing Eq. (12), the following equation is achieved:

$$\begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} W = \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu) I \end{bmatrix} W \phi \tag{13}$$

where W consists of corresponding eigenvectors of k leading eigenvalues of ϕ . Due to the lack of label in target domain, the computation of the conditional distribution $Q_t(y_t|x_t)$ is not possible. Therefore, we use the idea in [18] to compute the class conditional distribution $Q_t(x_t|y_t)$ instead of conditional distribution $Q_t(y_t|x_t)$. For evaluation of $Q_t(x_t|y_t)$, soft target labels \hat{y}_t is used instead of true target labels y_t . Soft labels of target domain is predicted using a base classifier trained on source domain in first iteration that is refined, iteratively.

3.2.2 Prediction function

The original data are mapped via A and B to find the new representations of source and target domains (i.e., $Z_s = A^T X_s$ and $Z_t = B^T X_t$). Our main goal is to learn an adaptive classifier f on labeled source domain D_s for target domain classification. To learn f , the structural risk functional is minimized as follows:

$$f = \arg \min_{f \in \mathcal{H}_K} \sum_{i=1}^n \ell(f(Z_{s_i}), y_i) + \eta \|f\|_K^2 \tag{14}$$

where \mathcal{H}_K consists of classifiers in the reproducing kernel Hilbert space, η is the regularization parameter and ℓ is the squared loss function $\ell = (y_i - f(Z_{s_i}))^2$ that measures the performance of classifier f on prediction of training labels. Therefore, the Representer theorem [19] is used to define the classifier f as follows:

$$f(z) = \sum_{i=1}^{n+m} a_i K(z_i, z) \tag{15}$$

where $K(., .)$ is the kernel function and a_i is the coefficient. Considering the squared loss function, and Eq. (15), the Eq. (14) is reformulated as follows:

$$f = \arg \min_{f \in \mathcal{H}_K} \left\| (Y - A^T K) E \right\|_F^2 + \eta \text{tr} \left(A^T K \Lambda \right) \tag{16}$$

where E is the diagonal domain indicator matrix with each element $E_{ii} = 1$ if $z_i \in D_s$, and $E_{ii} = 0$ otherwise. Also,

$\Lambda=(a_1, \dots, a_{n+m})^T$ consists of the vector of coefficients and $\mathbf{Y} = [y_1, \dots, y_{n+m}]$ is the label matrix of source and target data.

3.2.3 Manifold regularization

In addition, the manifold regularization term (i.e., Eq. (17)) is added into Eq. (16) to maximize the consistency between the intrinsic manifold structure of data and predictive structure of f using the marginal distributions of source and target domains (i.e., $P_s(Z_s)$ and $P_t(Z_t)$) as follows:

$$M_f(P_s, P_t) = \sum_{i,j=1}^{n+m} V_{ij} (f(z_i) - f(z_j))^2. \tag{17}$$

By incorporating Eq. (15) into Eq. (17) and adding the obtained equation into Eq. (16), we achieve

$$f = \arg \min_{f \in \mathcal{H}_K} \left\| (Y - \Lambda^T K) E \right\|_F^2 + \eta \text{tr}(\Lambda^T K \Lambda) + \rho \text{tr}(\Lambda^T K L K \Lambda) \tag{18}$$

where $L = D - V$ is the Laplacian matrix, which is normalized with diagonal matrix $D_{ii} = \sum_{j=1}^{n+m} V_{ij}$. Also, V is the affinity matrix which is computed by Eq. (19) as follows:

$$V_{ij} = \begin{cases} \cos(z_i, z_j), & \text{if } z_i \in N_P(z_j) \vee z_j \in N_P(z_i) \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where $N_P(z_j)$ is the set of P-nearest neighbors of point z_j . Setting derivative of objective function in Eq. (18) to $\mathbf{0}$ leads to

$$\Lambda^* = ((E + \rho L) K + \eta I)^{-1} E Y^T. \tag{20}$$

The cross-domain function f is learned through Eq. (15) using Eq. (20), directly, without the need of explicit classifier training.

4 Experimental setup

In this section, we consider data description to evaluate the performance of our JDSC. Also, we compare the performance of several state-of-the-art domain adaptation methods with the performance of our proposed method. Finally, the implementation details are described in the last subsection.

4.1 Data description

In this paper, the following four datasets: Office-Caltech-10 [6], Digits (USPS, MNIST) [7,8], ImageNet and VOC 2007 [9] and Pie (Face) [10], are used to evaluate the performance of JDSC.

The Office-31 dataset consists of the following three domains: Amazon (collected images from online merchants), Webcam (images taken by web camera) and DLSR (images taken by digital SLR camera), each of which contains a set of images of different objects with different qualities in each domain. The Office-31 dataset has 4652 images with 4096 features per image and 31 classes. The Caltech-256 (collected from Google images) dataset is another object recognition dataset that has 30,607 images with 4096 features per image and 256 classes. Ten common classes of four domains are used in experiments (i.e., keyboard, bike, calculator, headphones, mouse, mug, laptop, monitor, backpack and projector). The Office-Caltech-10 dataset consists of 12 tasks; in each task, one domain (e.g., Amazon) is considered as source domain and another domain (e.g., Caltech) as target domain. Differences in distribution of Office and Caltech datasets have a beneficial effect on performance evaluation of domain adaptation methods.

Digit dataset consists of two domains, USPS and MNIST, which contains handwritten numbers from 0 to 9. The USPS dataset has 7291 training images and 2007 test images of size 16×16 pixels with 256 features for each image, while the MNIST dataset has 60,000 training images and 10,000 test images of size 28×28 pixels with 256 features for each image. Ten common classes (i.e., digits 0–9) of both domains are used in experiments. Two experiments are performed using these two domains; in each experiment, one of them is considered as source and another one as target domain. It is worth noting that the distribution of each number is different in USPS and MNIST domains.

The Pie dataset is used for face recognition. It consists of the following five domains with 41,368 images of 68 different persons in different imaging modes for each domain: Pie1 (face image from left), Pie2 (face image from top), Pie3 (face image from bottom), Pie4 (face image from front) and Pie5 (face image from right). Therefore, 20 tasks are achieved from the above five domains to evaluate the performance of the proposed method, in which two domains are selected from five domains as source and target domains.

ImageNet and VOC 2007 are two large datasets of natural images with different distributions. ImageNet has over 14 million images with more than 20,000 categories, while VOC 2007 dataset consists of 9963 images containing 24,640 annotated objects. Five common classes of both datasets are exploited in our experiments (i.e., dog, chair, cat, bird and person). Therefore, two tasks I–V and V–I are considered in experiments.

Table 1 Optimal parameters for different datasets

Dataset	P	η	λ	ρ	γ	β	K	μ
ImageNet-Voc2007	2	0.1	10	1.0	1.5	0.001	100	0.1
Pie	2	0.01	0.8	0.9	0.1	0.01	200	0.1
Digit	2	0.01	0.4	0.9	0.3	0.01	100	1
Office-Caltech10	5	0.2	1	0.9	0.3	0.1	40	0.9

4.2 Implementation details

The number of images and type of features in different datasets are described as follow. In office dataset, 1410 images are selected, randomly; each image is defined by DeCaf6 features (which are the activations of sixth fully connected layer of a convolutional network trained on ImageNet). Moreover, in Caltech dataset, 1123 images with DeCaf6 features are selected, randomly. In Digit dataset, 1800 images of USPS domain and 2000 images of MNIST domain with 256 features are selected, randomly. In Pie dataset, 3332, 1629, 1632, 3329 and 1632 images with 1024 features are selected for Pie1, Pie2, Pie3, Pie4 and Pie5 domains, respectively. In ImageNet and VOC 2007 datasets, 7341 and 3376 images with 4096 DeCaf6 features are sampled, respectively. The optimal parameters for mentioned datasets are summarized in Table 1. The iteration number, T , and the used kernel are 10 and RBF (radial basis functions), respectively. The accuracy of classifier is computed through Eq. (21) where $\hat{y}(x)$ and $y(x)$ are the predicted and true labels for target domain, respectively,

$$\text{Accuracy} = \frac{|x : x \in D_t \wedge \hat{y}(x) = y(x)|}{|x : x \in D_t|}. \quad (21)$$

5 Experimental results and discussions

In this section, the classification accuracy results on Office-Caltech-10, Digit, ImageNet-VOC 2007 and Pie datasets are shown in Tables 2 and 3. We describe our observations and analyze the parameter sensitivity of JDSC on different types of datasets in the rest.

5.1 Result evaluation

JDSC outperforms other state-of-the-art domain adaptation and transfer learning methods (LRSR [20], ARTL [19], DICD [21], JGSA [4], VDA [22], D-CORAL [23], UTML [24]) on most of experiments (24 out of 36 tasks). The average classification accuracy of JDSC on 36 tasks is 86.2%, and the improvement in average performance is significant against the best compared method. In the rest, we compare our proposed method with other methods in detail.

Low-rank and sparse representation (LRSR) is a subspace learning method that obtains a common subspace which represents target domain by sparse and low-rank minimization problem to reduce the domain shift between source and target domains. However, LRSR does not address cross-domain distribution discrepancy completely. While JDSC is able to adapt domains both geometrically and statistically, JDSC performs (8.6%), (18.9%), (1.0%) and (21.2%) better than LRSR in prediction accuracy in Office-Caltech-10, Digit, ImageNet-VOC 2007 and Pie datasets, respectively.

Adaptation regularization-based transfer learning (ARTL) is a transfer learning method to learn domain-invariant classifier in original space, whereas JDSC learns adaptive classifier in a new space with better features, which prevents the feature distortion in model building. Our results show that JDSC gets (1.7%), (4.8%), (6.0%) and (12.7%) significant classification accuracy improvement compared to ARTL in Office-Caltech-10, Digit, ImageNet-VOC 2007 and Pie datasets, respectively.

Domain-invariant and class discriminative feature learning (DICD) creates a common subspace by reducing the difference in conditional and marginal distributions while important data properties are preserved. In addition, DICD reduces the distance of samples from same classes, while it increases the distance of samples from different classes. JDSC maximizes target variance to prevent feature distortions. Also, our method preserves source label information to get discriminative representation. JDSC in Office-Caltech-10 datasets obtains (2.5%) improvement and in Digit and Pie datasets obtains (11.6%) and (10.3%) performance improvement against DICD, respectively.

Joint geometrical and statistical alignment (JGSA) is an unsupervised domain adaptation framework, which obtains two subspaces for source and target domains to mitigate both geometrical and distribution shifts, jointly. However, JDSC reduces distribution discrepancies across domains by accounting the different importance of marginal and conditional distributions. Compared to JGSA, the average performance improvement of JDSC is (2.4%), (8.8%), (11.8%) and (6.4%) in Office-Caltech-10, Digit, ImageNet-VOC 2007 and Pie datasets, respectively.

Visual domain adaptation (VDA) is a transfer learning and domain adaptation approach, which reduces joint marginal and conditional distribution shifts, iteratively, by domain-invariant clustering in an embedding representation to discriminate different classes alongside with domain transfer. Despite VDA, JDSC preserves manifold consistency and performs dynamic distribution alignment. JDSC obtains (6%), (14.1%), (9.9%) and (12.4%) improvement against VDA in average accuracy in Office-Caltech-10, Digit, ImageNet-VOC 2007 and Pie datasets, respectively.

Table 2 Accuracy (%) of JDSC against compared methods in Office-Caltech-10 dataset using DeCaf6 features

Dataset	ARTL	LRSR	DICD	JGSA	VDA	D-CORAL	JDSC
C-A	92.4	91.5	91	91.4	91	92.3	94.1
C-W	87.8	76.6	92.2	86.8	82.7	90.1	88.1
C-D	86.6	87.9	93.6	93.6	86	88.6	94.3
A-C	87.4	85.8	86	84.9	82	83.7	89.1
A-W	88.5	73.6	81.4	81	78	89.8	89.2
A-D	85.4	82.2	83.4	88.5	79.6	90.5	91.1
W-C	88.2	72.8	84	85	80.2	81.5	88.4
W-A	92.3	75.5	89.7	90.7	87.6	91.9	92.6
W-D	100	100	100	100	100	100	100
D-C	87.3	75.2	86.1	86.2	79.5	80.1	88
D-A	92.7	85	92.2	92	89.9	91	94
D-W	100	99.3	99	99.7	99.7	97.3	99.7
Avg.	90.7	83.8	89.9	90	86.4	89.7	92.4

The bold values indicate the best results

Table 3 Accuracy (%) of JDSC against compared methods in Digit, ImageNet-VOC 2007 and Pie datasets

Dataset	ARTL	LRSR	DICD	JGSA	VDA	UTML	JDSC
U-M	67.7	54.5	65.2	68.2	63	59.8	76.7
M-U	88.8	73.8	77.8	80.4	75	76.1	89.4
I-V	62.4	67	–	52.3	62	–	68.3
V-I	72.2	77.5	–	70.6	64.7	–	78.3
P1-P2	64.1	65.9	73	74.3	73	80.4	86.4
P1-P3	56.1	64.1	72	74.9	61.6	83.5	81.7
P1-P4	85.4	82	92.2	92.9	90.1	96.6	96.8
P1-P5	51.6	54.9	66.9	61	42.4	66.9	82.4
P2-P1	71.7	45.5	69.9	71.1	72.9	79.4	80.4
P2-P3	64.6	53.5	65.9	73.6	75.6	77.8	81.5
P2-P4	87.9	71.4	85.3	89	83.6	92.6	89.1
P2-P5	55.6	48	48.7	62.3	57.7	67.4	75.3
P3-P1	61.4	52.5	69.4	68.2	58.8	75.9	71.3
P3-P2	68.3	55.6	65.4	80.1	74.7	79.7	80.7
P3-P4	88.2	77.5	83.4	87.5	87.5	93	89.7
P3-P5	61.4	54.1	61.4	67.2	52.6	65	80.9
P4-P1	91.9	81.5	93.1	93.4	92.4	94.4	94.7
P4-P2	92.1	58.4	90	94	92.3	94.5	93.4
P4-P3	87.1	82.2	89	89.6	90.4	93.4	91.2
P4-P5	70.7	72.6	75.6	80.2	70	82.8	86
P5-P1	56.3	52.2	62.9	57.4	49.9	74	64.5
P5-P2	56.4	49.4	57	68.9	62.3	74	77.4
P5-P3	63.6	58.5	65.9	70	61.3	72.7	79.5
P5-P4	80.2	64.3	74.8	84.4	71.2	85.4	84.9
Avg.(Digit)	78.3	64.2	71.5	74.3	69	68	83.1
Avg.(ImVO)	67.3	72.3	–	61.5	63.4	–	73.3
Avg.(Pie)	70.7	62.2	73.1	77	71	81.5	83.4
Avg.(all)	72.1	66.2	72.3	70.9	67.8	74.8	79.9

The bold values indicate the best results

Unsupervised transfer metric learning (UTML) tackles domain shift problem by minimizing the intraclass and max-

imizing the interclass distribution discrepancies between source and target domains via maximum mean discrepancy.

Table 4 Run time (s) of LRSR, ARTL, JGSA, VDA and JDSC

Task	#Sample×#Feature	LRSR	ARTL	JGSA	VDA	JDSC
C-A	2081 × 4096	203.1	20.1	224.6	35.3	207.2
I-V	10717 × 4096	4345.2	656.7	1364.9	2276.7	2720.9
M-U	3800 × 256	269.9	29.4	17.7	82.9	43.2

Moreover, in UTML, the property of domains is preserved by maintaining variance of samples. Unlike UTML, which adapts only conditional distribution, JDSC adapts both marginal and conditional distributions with different significance. JDSC has (15.1%) and (1.9%) improvement compared to the best baseline method UTML in the classification accuracy on Digit and Pie datasets, respectively.

Also, deep learning methods were widely considered in recent years [25]. JDSC can be compared with deep learning methods under the following two circumstances: (1) use of data with pretrained features by deep learning networks as input data and (2) using deep learning networks instead of label prediction function in classifier-based step. For this purpose, we use circumstance 1 to compare JDSC with deep methods, and the experiment results on Office-Caltech-10 with pretrained DeCaf6 features learned on convolutional networks are given in Table 2. As can be seen from Table 2, JDSC outperforms D-CORAL method [23] (which adapts the second-order subspaces using deep neural networks) and has 2.7% improvement.

5.2 Time complexity

Table 4 presents the run time of JDSC and other baseline and state-of-the-art methods on different tasks. By considering high time complexity of deep methods for backpropagations, they are not compared in this challenge. As is clear from Table 4, JDSC has modest run time (i.e., 207.2 s) in task C-A compared to the total run time (i.e., $224.6 + 20.1 = 244.7$) of the two baseline methods JGSA and ARTL. Therefore, JDSC has an acceptable and comparable time complexity against other compared methods, due to its performance in classification accuracy where the test environment is an Intel® Core™ i7-8550 CPU with 8 GB memory. Also, the MATLAB is selected as the coding language.

5.3 Parameters impact

We evaluate the parameter sensitivity of JDSC on selected tasks of four benchmark datasets (i.e., C-A from Office-Caltech-10 dataset, P1-P2 from Pie dataset, V-I from Image Net-VOC 2007 dataset and U-M from Digit dataset) to validate its performance on a wide range of parameter values. Figure 1 illustrates the relationship between various parameters and accuracy. Each of $\gamma, \lambda, \mu, \beta, \eta, \rho, K$ and P

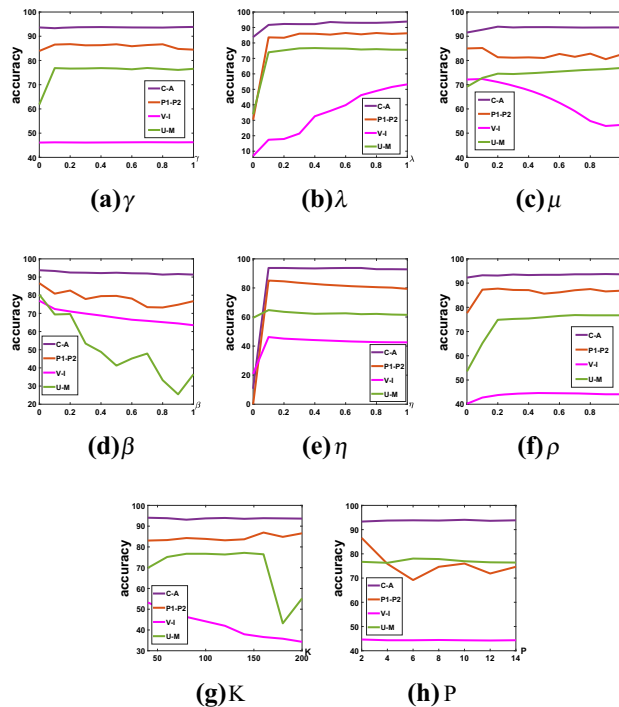


Fig. 1 Parameter evaluation with respect to classification accuracy (%) for $\gamma, \lambda, \mu, \beta, \eta, \rho, K$ and P parameters on C-A, P1-P2, V-I and U-M tasks

parameters has been validated on different types of datasets by fixing other parameters. Figure 1a illustrates γ which is a trade-off parameter for marginal and conditional distribution alignments. Also, in Fig. 1b–d, λ, μ and β are the trade-off parameters to balance the importance of each component in Eq. (13). In addition, Fig. 1e and 1f show parameter sensitivity of η and ρ parameters in Eq. (20). Figure 1g and 1h illustrate the impact of K (the dimension of embedded subspaces) and P (the number of neighbors in Laplacian graph) parameters in prediction accuracy. All $\gamma, \lambda, \mu, \beta, \eta,$ and ρ parameters are evaluated in range between 0.0 to 1.0. Also, parameters K and P are evaluated in ranges of [40, 200] and [2, 14], respectively. As is clear from Fig. 1a–c and 1e–f the results of Pie and Digit datasets for $\gamma, \lambda, \mu, \eta$ and ρ parameters illustrate stability of accuracy values after several iterations. Classification accuracy on C-A, P1-P2 and V-I tasks for β and K parameters is almost steady, while for high values of β and K , the classification accuracy on U-M task is low. Also, for C-A, U-M and V-I tasks, the accuracy has no obvious change for parameter P , while in P1-P2 task, the predicted accuracy is sensitive to values of P .

6 Conclusion

In this paper, we proposed a new transfer learning method referred as joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation (JDSC) to address discrepancy problem between source and target domains. JDSC finds two coupled projections for source and target domains, respectively, to minimize the domain shift, specifically, by accounting the different importance for marginal and conditional distributions. In addition, JDSC increases the manifold consistency underlying the marginal distributions of source and target domains. As a result, the optimal new representations and classifier are achieved to adapt domains. We assess the improvement of JDSC in transferring the knowledge by performing experiments on standard visual datasets where the results show the prominence of JDSC in comparison with other state-of-the-art visual domain adaptation methods. JDSC can find its applications in a wide range of classification problems, e.g., land cover classification through remote sensing [26] and recognition of anomalies in thermal images [27]. As a future work, we aim to extend JDSC using extracted features through deep neural networks. Also, the proposed method can be applied to reinforcement learning approaches to improve challenges in robotics.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

- Csurka, G.: Domain adaptation for visual applications: a comprehensive survey (2017). arXiv preprint [arXiv:1702.05374](https://arxiv.org/abs/1702.05374)
- Tahmoresnezhad, J., Hashemi, S.: Common feature extraction in multi-source domains for transfer learning. In: 2015 7th Conference on Information and Knowledge Technology (IKT), pp. 1–5 (2015)
- Tahmoresnezhad, J., Hashemi, S.: Exploiting kernel-based feature weighting and instance clustering to transfer knowledge across domains. *Turk. J. Electr. Eng. Comput. Sci.* **25**, 292–307 (2017)
- Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1859–1867 (2017)
- Azab, A.M., Toth, J., Mihaylova, L.S., Arvaneh, M.: A review on transfer learning approaches in brain–computer interface. In: *Signal Processing and Machine Learning for Brain–Machine Interfaces*, pp. 81–101. Institution of Engineering and Technology (2018)
- Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2066–2073 (2012)
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998)
- Hull, J.J.: A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 550–554 (1994)
- Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., Yu, P.S.: Visual domain adaptation with manifold embedded distribution alignment. In: *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 402–410 (2018)
- Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) database. In: *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pp. 53–58 (2002)
- Asgarian, A., et al.: A hybrid instance-based transfer learning method (2018). arXiv preprint [arXiv:1812.01063](https://arxiv.org/abs/1812.01063)
- Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. *J. Big Data* **3**, 9 (2016)
- Ghifary, M., Balduzzi, D., Kleijn, W.B., Zhang, M.: Scatter component analysis: a unified framework for domain adaptation and domain generalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1414–1430 (2016)
- Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2200–2207 (2013)
- Mahadevan, S., Mishra, B., Ghosh, S.: A unified framework for domain adaptation using metric learning on manifolds. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 843–860. Springer (2018)
- Rubin, J., et al.: An ensemble boosting model for predicting transfer to the pediatric intensive care unit. *Int. J. Med. Inform.* **112**, 15–20 (2018)
- Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: *Advances in Neural Information Processing Systems*, pp. 137–144 (2007)
- Wang, J., Chen, Y., Hao, S., Feng, W., Shen, Z.: Balanced distribution adaptation for transfer learning. In: *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 1129–1134. IEEE (2017)
- Long, M., Wang, J., Ding, G., Pan, S.J., Philip, S.Y.: Adaptation regularization: a general framework for transfer learning. *IEEE Trans. Knowl. Data Eng.* **26**, 1076–1089 (2013)
- Xu, Y., Fang, X., Wu, J., Li, X., Zhang, D.: Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE Trans. Image Process.* **25**, 850–863 (2015)
- Li, S., Song, S., Huang, G., Ding, Z., Wu, C.: Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Trans. Image Process.* **27**, 4260–4273 (2018)
- Tahmoresnezhad, J., Hashemi, S.: Visual domain adaptation via transfer feature learning. *Knowl. Inf. Syst.* **50**, 585–605 (2017)
- Sun, B., Saenko, K.: Deep coral: correlation alignment for deep domain adaptation. In: *European Conference on Computer Vision Workshops*, pp. 443–450 (2016)
- Huang, J., Zhou, Z.: Transfer metric learning for unsupervised domain adaptation. *IET Image Proc.* **13**, 804–810 (2019)
- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: *International Conference on Artificial Neural Networks*, pp. 270–279 (2018)
- Addabbo, P., Focareta, M., Marcuccio, S., Votto, C., Ullo, S.L.: Land cover classification and monitoring through multisensor image and data combination. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 902–905 (2016)
- Addabbo, P., Angrisano, A., Bernardi, M.L., Gagliarde, G., Mennella, A., Nisi, M., Ullo, S.L.: UAV system for photovoltaic plant inspection. *IEEE Aerosp. Electron. Syst. Mag.* **33**, 58–67 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.