Conditional Probability Distribution Divergence Reduction in Visual Domain Adaptation

Elham Hatefi
Artificial Intelligence Department
Faculty of Computer Engineering
University of Isfahan
Isfahan, Iran
el.hatefi@eng.ui.ac.ir

Hossein Karshenas
Artificial Intelligence Department
Faculty of Computer Engineering
University of Isfahan
Isfahan, Iran
h.karshenas@eng.ui.ac.ir

Peyman Adibi
Artificial Intelligence Department
Faculty of Computer Engineering
University of Isfahan
Isfahan, Iran
adibi@eng.ui.ac.ir

***Abstract*— The rapid evolution of data has challenged traditional machine learning methods and leads to the failure of many learning models. As a possible solution to the lack of sufficient labeled data, transfer learning aims to exploit the accumulated knowledge in the auxiliary domain to develop new predictive models. This article studies a specific type of transfer learning called domain adaptation, which works based on subspace learning in order to minimize distance between class conditional probability distributions of source and target domains and to preserve source discriminative information. Efficient Classifier trained on source domain data has been used to predict target domain data labels to facilitate subspace learning. In this work, subspace learning is formulated as an optimization problem and experiments have been carried out on the real-world datasets. The results of experiments indicate that the proposed method outperforms several existing methods at this field in term of accuracy on three datasets: Office-Caltech10, Office and SS5 datasets.**

Keywords—Transfer Learning, Domain Adaptation, Class Conditional Probability Distribution.

I. INTRODUCTION

Machine learning algorithms require a large number of training examples to learn and often fail to use the learned model to predict data obtained from different domains and tasks. Learning a new model requires a lot of labeled data that may not always be available, especially if new type of information is involved. In addition, training the new model from scratch is time consuming and costly. For example, in visual detection due to many factors (e.g., environment, light, background, type of sensors, angle of view, etc.), there is always a distribution change or even divergence of feature space between two domains that can degrade the performance. As a solution, the model or knowledge gained from one domain can be transferred to another. This process is called transfer learning and has been an active research subject**.** In fact, the purpose of this type of learning is to transfer knowledge from auxiliary to main domain. The auxiliary and main domains are named source and target domains, respectively.

A specific type of transfer learning is domain adaptation where the label set of the source and target are the same but the distribution or feature spaces of two domains are different [1]. In fact, domain adaptation can be either homogeneous or heterogeneous. In the first case, the feature and label spaces are the same but the difference between domains is based on the divergence between their distributions. In the second case, the label spaces are the same but the feature spaces are different between domains. In this paper, we assume homogeneous domain adaptation. Many of the introduced domain adaptation methods are presented at the sample, feature, or classification level. In the sample level, some source samples are re-weighted based on differences with the target domain samples. In the feature level, it is attempted to provide a more acceptable representation of the features in order to minimize inter-domain discrepancy. In the classification level, an optimal classifier for the target domain is created based on the source and target domain data as well as the trained model based on the source domain data.

In this paper, we learn feature space in multiple steps in order to reduce class conditional probability distribution shift as it was ignored in many previous works and to preserve source discriminative information. In most of the existing methods at this field, marginal distribution was used to minimize inter-domain discrepancies. In fact, minimizing class conditional probability distributions shift can preserve more discriminative information than marginal distribution. We use SVM and logistic regression classifiers in order to predict the best labels for target domain data and improve underlying subspace in the iterative process. These classifiers are used not only for subspace learning, but also as the base classifiers to predict target data after feature space learning is accomplished.

The rest of this paper is organized as follows: Section II reviews several related researches. Section III describes the proposed method. Experiments will be presented in section IV. Finally, Section V gives a conclusion and offers some future works.

II. RELATED WORK

Typical approaches to domain adaptation fall into six different categories [2]: statistical approaches, geometric approaches, higher-level representations, class-based approaches, self-labeling and hybrid approaches. The details of each of these approaches will be discussed below:

- **Statistical approach:** The purpose of this type of models is to minimize the difference of the statistical distributions between the source and target domains[3,4,5,6,7].
- **Geometric approach:** The relationship between the two domain datasets is based on their geometric properties. In fact, it is assumed that domain variations can be reduced based on the relationship between the geometric structures of the source and target data[8,9,10].
- **Higher-level representation:** The purpose is to use a higher level to represent the relationship between domains more concisely and unambiguously. This

approach can be used by other techniques to better transfer knowledge. In addition, it can be used independently to reduce inter-domain discrepancy [11,12,13,14,15,16,17].

- **Class-based methods:** These approaches apply label information as a guide to connect the source and target domains [18,19,20].
- **Self-labeling approach:** These methods use domain examples to train the initial model and obtain pseudo-label for the target domain and then add the target domain and pseudo-label data to retrain the model and continue the process until convergence [21,22].
- **Hybrid approach:** Combines two or more of the above methods to better transfer the knowledge. For example statistical and geometric approach, statistical and class based approach, etc.[23,24,25,26].

Domain adaptation is accomplished by supervised, unsupervised and semi-supervised methods and domain generalization. In supervised case, there are little labeled data in the target domain. In the second case, there will be no labeled data in the target domain but there are enough unlabeled data in the target domain. In the third case, in addition to the small amount of labeled data in the target domain, there are also a large amount of unlabeled data that will generate additional structural information. In domain generalization, the target domain data are not available and as a solution we can use multiple source domains to transfer the immutable knowledge to the new domain. In this paper, we assume unsupervised domain adaptation. Unsupervised learning in domain adaptation usually uses statistical, geometric, higher-level, and hybrid approaches. The statistical approach usually uses Maximum Mean Discrepancy (MMD) criterion [3]. This criteria compute the distance between the sample means of the source and target data in the k-dimensional embedding subspace. In addition to the MMD technique, other statistical criterion is used to compare the two distributions, such as Kullback-Leibler divergence [4], Hellinger distance [5], Quadratic divergence [6] and Mutual Information [7]. The hybrid approach [26] used a combination of statistical and geometric methods to form a mapping for the source and target domain that minimizes structural form and statistical discrepancies simultaneously. In [10], a method based on a geometrical approach was proposed to combine two domains based on the geodesic flow on the Grassmann manifold.

What makes the method proposed in this paper different from previous work is that in this method, we minimize class conditional probability distribution of the two domains. As mentioned, in unsupervised domain adaptation the label of target data is not available. In order to obtain class conditional distribution of the target domain, these labels must be estimated. In this paper, we use SVM and logistic regression classifiers in order to predict best labels for target domain data.

III. Proposed Method

In this paper, we use nonlinear mapping function Φ to transform source and target data to a high dimensional space. In order to minimize divergence between source and target domain we use projection P for the source domain and Q for the target domain. These transitions should satisfy the following conditions:

- Reduce the conditional probability distribution divergence
- Preserve source discriminative information
- Minimize subspace divergence

All of the notations used in this paper are listed in Table I.

TABLE I. Notations

Notation	Description
X_s	Source domain data.
X_t	Target domain data.
Y_s	Source domain label.
$\hat{Y}_{tp}$	Target pseudo-label.
$\hat{Y}_t$	Target predict label.
C	Number of class.
X_s^c	Source domain data that belongs to class c.
X_t^c	Target domain data that belongs to class c.
P	Projection for source domain.
Q	Projection for target domain.
Φ	Nonlinear mapping function for source and target data.
N_s	Number of source domain data.
n_s^c	Number of source domain data that belongs to class c.
n_t^c	Number of source domain data that belongs to class c.
M	Number of iteration.
α	Trade-off parameter.
β	Trade-off parameter.

A. *Reducing the Class Conditional Probability Distribution Divergence*

We predict the label of target data based on the classifier trained on source data and then compute the conditional probability distribution for both source and target domains. In order to minimize class conditional probability distribution divergence, we use the following problem [26]:

$$\min_{P,Q} \sum_{c=1}^{C} \left\| \frac{1}{n_s^c} \sum_{x_i \in X_s} P^T \Phi(x_i) - \frac{1}{n_t^c} \sum_{x_j \in X_t} Q^T \Phi(x_j) \right\|_F^2 \quad (1)$$

where n_s^c, n_t^c represent the number of source and target domain data that belongs to class c. The above problem can be transformed to its matrix form as follows:

$$\min_{P,Q} Tr\left(\begin{bmatrix} P^T & Q^T \end{bmatrix} \begin{bmatrix} A_s & A \\ A & A_t \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} \right) \quad (2)$$

in which A_s is computed as follows:

$$A_s = \Phi(X_s) \sum_{c=1}^{C} D_s^c \Phi(X_s)^T \quad (3)$$

where D_s^c is defined as follows:

$$(D_s^c)_{i,j} = \begin{cases} \frac{1}{n_s^c * n_s^c} & x_i, x_j \in X_s^c \\ 0 & otherwise \end{cases} \quad (4)$$

Similarly, A_t is computed as follows:

$$A_t = \Phi(X_t) \sum_{c=1}^{C} D_t^c \Phi(X_t)^T \quad (5)$$

where D_t^c is defined as follows:

$$(D_t^c)_{i,j} = \begin{cases} \frac{1}{n_t^c * n_t^c} & x_i, x_j \in X_t^c \\ 0 & otherwise \end{cases} \quad (6)$$

Also matrix A in problem (2) is computed as below:

$$A = \Phi(X_s) \sum_{c=1}^{C} D_{st}^c \Phi(X_t)^T \quad (7)$$

where D_{st}^c is defined as follows:

$$(D_{st}^c)_{i,j} = \begin{cases} -\frac{1}{n_s^c n_t^c} & x_i \in X_s^c, x_j \in X_t^c \\ 0 & otherwise \end{cases} \quad (8)$$

B. *Source Discriminative Information Preserving*

In unsupervised domain adaptation, the label of source data is available and can be used to learn data subspace to be more discriminative than before. For example, we can use this information to construct between class scatter matrix S_b of source domain data[27], which is defined as follows:

$$\max_P \quad Tr(P^T S_b P) \quad (9)$$

$$S_b = \sum_{c=1}^{C} n_s^c (\mu_s^c - \mu_s)(\mu_s^c - \mu_s)^T \quad (10)$$

where μ_s^c and μ_s given x_i, x_i^c belong to X_s are defined as follows:

$$\mu_s^c = \sum_{i=1}^{n_s^c} \Phi(x_i^c) \quad (11)$$

$$\mu_s = \sum_{i=1}^{N_s} \Phi(\mathrm{x}_i) \quad (12)$$

where μ_s^c is the mean of source domain data that belongs to class c and μ_s is the mean of total source domain data. By considering between class scatter matrix and trying to maximize it, the distance between the samples in different classes of source domain is maximized.

C. *Minimizing Subspace Divergenc*

Finally, subspaces P and Q must be close together which can be defined by the following relation:

$$\min_{P,Q} \|P - Q\|_F^2 \quad (13)$$

D. *Objective Function*

The goal of the proposed method is to minimize class conditional divergence distribution and subspace divergence, and maximize between class scatter matrix, simultaneously. By considering (2), (9) and (13) the objective function is specified as follows:

$$\max_{P,Q} \frac{Tr([P^T \ Q^T] \begin{bmatrix} \alpha S_b & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix})}{Tr\left([P^T \ Q^T] \begin{bmatrix} A_s + \beta I & A - \beta I \\ A - \beta I & A_t + \beta I \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix}\right)} \quad (14)$$

where α and β are trade-off parameters and I is the identity matrix. Embedding data to nonlinear space and calculating the inner product is costly. So, we reformulate the objective function (14) and then use kernel trick [28] to calculate inner product in nonlinear space. By considering $P = \Phi(\mathrm{X})\mathrm{H}$ and $Q = \Phi(\mathrm{X})R$, where $X = [X_s, X_t]$, (14) is reformulated as below:

$$\max_{H,R} \frac{Tr(\mathrm{H}^T \ \mathrm{R}^T] \begin{bmatrix} \alpha S_b & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{H} \\ \mathrm{R} \end{bmatrix})}{Tr\left(\mathrm{H}^T \ \mathrm{R}^T] \begin{bmatrix} A_s + \beta G & A - \beta G \\ A - \beta G & A_t + \beta G \end{bmatrix} \begin{bmatrix} \mathrm{H} \\ \mathrm{R} \end{bmatrix}\right)} \quad (15)$$

where $G = \Phi(X)^T \Phi(X)$ and $\mathrm{A_t}$, $\mathrm{A_s}$, A are estimated as follows:

$$A_s = G_s \sum_{c=1}^{C} D_s^c G_s^T \quad (16)$$

$$A_t = G_t \sum_{c=1}^{C} D_t^c G_t^T \quad (17)$$

$$A = \mathrm{G_s} \sum_{c=1}^{C} D_{st}^c G_t^T \quad (18)$$

where $G_s = \Phi(\mathrm{X})^T \Phi(X_s)$ and $G_t = \Phi(\mathrm{X})^T \Phi(X_t)$. To calculate S_b , tow vectors μ_s^c and μ_s must be computed. According to (11) and (12) and using G_s instead of $\Phi(\mathrm{x}_i)$ and $\Phi(x_i^c)$, μ_s^c and μ_s are computed. For example, to calculate μ_s the mean of G_s is considered. In order to calculate inner product in nonlinear space we use kernel trick. In this paper we use linear and Gaussian kernel which can be computed as below:

$$\Phi(\mathrm{X}_i)^T\Phi(\mathrm{X}_j) = K(\mathrm{X}_i, \mathrm{X}_j) \tag{19}$$

where linear kernel is defined as follow:

$$\mathrm{K}(\mathrm{X}_i, \mathrm{X}_j) = \mathrm{Linear}(\mathrm{X}_i, \mathrm{X}_j) = X_i * X_j^T, \tag{20}$$

and Gaussian kernel is defined as follow:

$$K(\mathrm{X}_i, \mathrm{X}_j) = \mathrm{Gussian}(\mathrm{X}_i, \mathrm{X}_j) = \exp(-\frac{||\mathrm{X}_i - \mathrm{X}_j||^2}{2\sigma^2}) \tag{21}$$

where σ is hyper-parameter. To optimize (15), we rewrite $[H^T \ R^T]$ as V^T. Then the objective function is reformulated as follows:

$$\max_V \frac{Tr(V^T \begin{bmatrix} \alpha S_b & 0 \\ 0 & 0 \end{bmatrix} V)}{Tr\left((V^T \begin{bmatrix} A_s + \beta G & A - \beta G \\ A - \beta G & A_t + \beta G \end{bmatrix} V\right)} \tag{22}$$

*E. **Optimization***

The objective function (22) can be transformed into a constrained problem form:

$$\max_V \ Tr(V^T \begin{bmatrix} \alpha S_b & 0 \\ 0 & 0 \end{bmatrix} V) \tag{23}$$

$$s.t: \ Tr\left((V^T \begin{bmatrix} A_s + \beta G & A - \beta G \\ A - \beta G & A_t + \beta G \end{bmatrix} V\right) = 1$$

By using Lagrange multiplier θ, the Lagrangian function of (23) can be reformulated as follows:

$$L = Tr\left(V^T \begin{bmatrix} \alpha S_b & 0 \\ 0 & 0 \end{bmatrix} V\right) + Tr((V^T \begin{bmatrix} A_s + \beta G & A - \beta G \\ A - \beta G & A_t + \beta G \end{bmatrix} V - I)\theta) \tag{24}$$

In order to solve the above equation we set the derivative $\frac{\partial L}{\partial V} = 0$ and get the following equation:

$$\begin{bmatrix} \alpha S_b & 0 \\ 0 & 0 \end{bmatrix} V = \begin{bmatrix} A_s + \beta G & A - \beta G \\ A - \beta G & A_t + \beta G \end{bmatrix} V\theta \tag{25}$$

This equation can be solved through eigenvalue decomposition. We select d eigenvectors $[V_1, \dots, V_d]$ corresponding to d largest eigenvalues $\theta = diag(\lambda_1, \dots, \lambda_d)$ as the transformation V, and obtain transformations H and R. Then the embedding subspace of source data ($P^T\Phi(\mathrm{X}_s)$) and target data ($Q^T\Phi(\mathrm{X}_t)$) can be estimated as follows:

$$P^T\Phi(\mathrm{X}_s) = \mathrm{H}^T\Phi(\mathrm{X})^T\Phi(\mathrm{X}_s) = \mathrm{H}^T G_s \tag{26}$$

$$Q^T\Phi(\mathrm{X}_t) = \mathrm{R}^T\Phi(\mathrm{X})^T\Phi(\mathrm{X}_t) = \mathrm{R}^T G_t \tag{27}$$

As mentioned, the proposed subspace learning method is done iteratively. In iteration m, after mapping source and target data to their subspaces according to (26) and (27), the classifier trained on projected source domain data is used to predict target domain data labels. Then the target pseudo-labels are created and used to estimate the class conditional probability of target data in the next iteration (m+1). In fact, by considering the iterative process, target data labels are estimated more accurately.

The proposed method is outlined in the Algorithms I and II. According to Algorithm I, $\hat{\mathrm{Y}}_{\mathrm{tp}}$ is first initialized according to the classifier trained on source data domain. Then, within the iterative process of the algorithm, after learning the subspace (Algorithm II) and projecting the source and target domain data to this space, the labels of target domain are predicted according to classifier. According to Algorithm I, "Classifier_Train*"* function is used to train classifier with two input parameters: projected source data domain ($H_m^T G_s$) and their corresponding labels (Y_s) and one output parameter: "Model". "Predict_Labels" function is used to estimate target domain pseudo-label with two input parameters: projected target data domain ($R_m^T G_t$) and "Model" and one output parameter: $\hat{Y}_{tp}$. Finally, target domain labels ($\hat{Y}_t$) are inferred.

Algorithm I: Proposed Method

Input: *Source data and labelsX_s, Y_s , Target data X_t*
Intermediate: *Target pseudo-label $\hat{Y}_{tp}$*
Output*: Inferred target labels: $\hat{Y}_t$*
Initialize: *Maximum Iteration M, $\hat{Y}_{tp}$*
Begin
For m=1 to M
  H_m, R_m, G_s, G_t=SubspaceLearning($X_s, Y_s, X_t, \hat{Y}_{tp}$)
  $Model$= Classifier _Train($H_m^T G_s, Y_s$)
  $\hat{Y}_{tp}$=Predict_Labels(Model,$R_m^T G_t$)
End
  $Model$= Classifier_Train ($H_M^T G_s, Y_s$)
  $\hat{Y}_t$= Predict_Labels (Model,$R_M^T G_t$)
End

Algorithm II: SubspaceLearning

Input*:$X_s, Y_s, X_t, \hat{Y}_{tp}$*
Output*: H_m, R_m, G_s, G_t*
Begin
1. *Compute G_s, G_t,G , A_t , A_s , A, S_b*
2. *Solve the Eigen-decomposition problem in (25)*
3. *Select d corresponding eigenvectors of d largest eigenvalues as V*
4. *Obtain H_m and R_m*

End

As was mentioned, two classifiers were used to predict target domain data labels: SVM and logistic regression classifier. Logistic regression is a statistical learning technique which is used for the classification problems and based on the concept of probability. This classifier gives a set of outputs or classes based on probability when the inputs are passed through a prediction function. In other words, in order to map predicted values to probabilities, the sigmoid function is used which maps any real value into another value between 0 and 1. SVM classifier is a supervised machine learning

model and its objective is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. For example, to separate the two classes of data points, there are many possible hyperplanes that could be chosen. The objective of SVM is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. These classifiers can be used for binary and multi-class classification problems.

IV. Result

Three datasets were used to evaluate the performance of the proposed algorithm: Office dataset, Office + Caltech10 dataset and SS5 dataset. Details of these datasets are described below:

- **Office Dataset** (O31): This dataset has images in three domains: Amazon (A), Webcam (W), and DSLR (D). Each contains images from 31 categories including office stuff like backpack, laptop, keyboard, etc. The three domains Amazon, Webcam, and DSLR contain images from Amazon's website(amazon.com), a webcam (low-resolution images by a web camera), and a DSLR (high-resolution images by a digital single-lens reflex camera), respectively, with different lighting, pose changes and backgrounds. Fig.1 shows the sample images from the office dataset.
- **Office + Caltech10 Dataset** (OC10): This dataset has 10 common object categories from an Office (e.g., keyboard, laptop, etc.) and Caltech10 datasets. In particular, OC10 contains a subset of O31 (3 domains of Amazon, DSLR, Webcam) and another Caltech (C) domain. Fig.2 shows the sample images from Office+Caltech10 dataset.
- **SS5 Dataset :** The evaluations on a cross-place satellite scene dataset is also considered. Three publicly available datasets are chosen. Specifically, they are Banja Luka (B) [31], UC Merced Land Use (U) [32], and 18-class Satellite Scene (S) [33] datasets. Five shared scene categories are chosen from three datasets, respectively. These categories are farmland/field, trees/forest, industry, residential, and river. Some images are shown in Fig.3.

Fig. 1. Sample images from Office dataset.

Fig. 2. Sample images from Office +Caltech10 dataset.

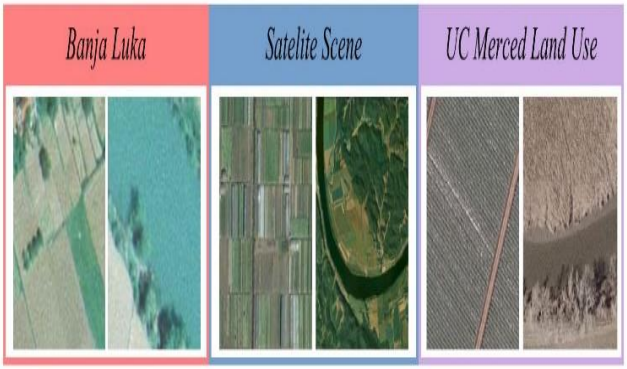

Fig. 3. Sample image from SS5 Dataset

A domain adaptation tasks is denoted by $S \rightarrow T$, where S and T denote the source and target domains, respectively. For example $A \rightarrow D$ means that the source domain is Amazon and the target domain is DSLR. The used datasets O31 and OC10 lead to 6 and 12 domain adaptation tasks, respectively. Similar to O31, there are 6 domain adaptation tasks on the SS5 dataset. Our method was compared with NA, GFK, ISL, JGSA and LDADA methods. The details of these methods are discussed below:

- **NA**: (No Adaptation) A basic bassline that learns logistic regression on the source domain data and applies it to the target domain data.
- **GFK:** Geodesic Flow Kernel (GFK) is an improvement over the SGF[29] technique where instead of sampling a few points on the geodesic, the whole curve is used for domain adaptation[10].
- **ILS**: This method is used to learn an Invariant Latent Space (ILS) to reduce the discrepancy between the source and target domains and uses Riemannian optimization techniques to match statistical properties between samples projected into the latent space from different domains [9].
- **JGSA**: The inter-domain differences are minimized geometrically and statistically. This can be done by mapping to a smaller dimension to minimize statistical and geometric differences in the data [26].
- **LDADA**: This is a recent method that learns class specific linear projections. Learning these projections is naturally cast into a linear-discriminant-analysis-

like framework, which gives an efficient, closed form solution [30].

Table II and III show the accuracy of different methods on OC10 dataset for the first and second six domain adaptation tasks where features are extracted from a VGG-M neural network [34] pre-trained on ImageNet. Features of three datasets O31, OC10 and SS5 can be extracted from handcrafted features (SURF) [35] or VGG features. In all of the previous works, it has been shown that domain adaptation based on VGG features has better result than SURF features [9,30]. Thus, in this paper we use VGG features for domain adaptation. "Proposed Method-LR" indicates the accuracy of proposed methods for logistic regression classifier and "Proposed Method-SVM" shows the accuracy for SVM classifier reported in our conference paper [36]. For conference paper the Gaussian kernel has better result than linear and polynomial kernels. For "Proposed Method-LR" the linear kernel has better result than other kernels. The results obtained for "Office" datasets show that the proposed algorithm for two classifiers leads to better results in most cases of domain adaptation tasks.

TABLE II. ACCURACY (%) ON OC10 DATASET WITH THE FIRST SIX DOMAIN ADAPTATION TASKS

Method	$A \rightarrow C$	$C \rightarrow A$	$A \rightarrow D$	$D \rightarrow A$	$A \rightarrow W$	$W \rightarrow A$
NA	88.55	93.8	87.04	83.37	85.44	88.86
GFK	85.1	93.2	84.7	91.8	84.8	91.2
ILS	86.5	93.1	83.6	92.2	89.9	92.6
JGSA	86.02	93.2	93.6	93.53	87.12	92.80
LDADA	88.5	95.1	90.0	94.2	92.7	94.2
Proposed Method-LR	**89.19**	94.51	**93.84**	93.72	**93.49**	**94.61**
Proposed Method-SVM	**89.76**	**95.71**	92.34	**94.31**	**93.16**	**95.3**

TABLE III. ACCURACY (%) ON OC10 DATASET WITH THE SECOND SIX DOMAIN ADAPTATION TASKS

Method	$C \rightarrow D$	$D \rightarrow C$	$C \rightarrow W$	$W \rightarrow C$	$D \rightarrow W$	$W \rightarrow D$
NA	92.34	78.07	88.22	84.46	92.13	97.58
GFK	91	82.3	86.8	82.3	97.6	98.3
ILS	88.6	85.7	88.8	87.3	96.3	96.5
JGSA	92.36	82.64	89.83	84.51	**98.31**	98.73
LDADA	93.8	84.3	94.4	88.3	95.0	**99.2**
Proposed Method-LR	**95.89**	**87.44**	**96.37**	**88.34**	97.29	97.53
Proposed Method-SVM	**96.71**	85.3	**96.02**	**88.52**	95.703	97.52

Accuracy with SURF and VGG feature on two domain tasks $A \rightarrow C$ and $C \rightarrow A$ of OC10 dataset for logistic regression classifier is shown in Fig.4. This result implies that good feature representation leads to better classification performance. For two tasks $A \rightarrow C$ and $C \rightarrow A$, the improvement in accuracy is about 50% and 40%, respectively.

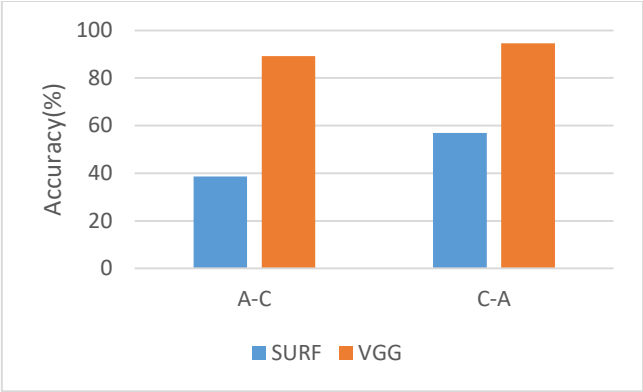

Fig. 4. Accuracy with SURF and VGG feature on two domain tasks $A \rightarrow C$ and $C \rightarrow A$ of OC10 dataset

Table IV shows the accuracy of the proposed and other competitive methods on OC31 dataset. For regression classifier the proposed algorithm outperforms other approaches in all cases of domain adaptation tasks.

TABLE IV. ACCURACY (%) ON OC31 DATASET

Method	$A \rightarrow D$	$D \rightarrow A$	$A \rightarrow W$	$W \rightarrow A$	$D \rightarrow W$	$W \rightarrow D$
NA	69.17	48.45	63.93	52.81	84.25	89.71
GFK	59.9	48.3	53.6	45.7	82.7	86.0
ILS	57.3	53.9	52.3	48.3	83.3	82.5
JGSA	69.58	57.76	66.3	55.3	85.4	89.5
LDADA	69.6	60.3	64.6	58.7	83.5	88.2
Proposed Method-LR	**77.23**	**64.03**	**75.62**	**61.59**	**90.88**	**91.35**
Proposed Method-SVM	**78.81**	57.63	**71.9**	58.51	**86.14**	**96.68**

In order to validate that the proposed method is general and can be applied to other images with different characteristics, the results on the SS5 dataset are reported. These results shown in Table V implies that the proposed method can also be applicable to other general visual recognition problems.

TABLE V. ACCURACY (%) ON SS5 DATASET WITH THE FIRST SIX DOMAIN ADAPTATION TASKS

Method	$B \rightarrow R$	$R \rightarrow B$	$B \rightarrow U$	$U \rightarrow B$	$R \rightarrow U$	$U \rightarrow R$
NA	48.38	37.49	67.6	62.56	80.6	82.88
GFK	56.9	48.5	66.2	64.4	88.6	80.7
ILS	**75.3**	56.1	78.9	66.2	95.4	76.4
JGSA	51.4	25.1	59.4	**77.3**	94.6	97.4
LDADA	51.8	58.9	81.6	70.5	**97.2**	98.1
Proposed Method-LR	52.6	**68.34**	**87.6**	70.175	94.6	**98.39**
Proposed Method-SVM	53.85	**64.72**	**90.80**	72.02	93.6	97.39

Table VI shows the average accuracy for OC10, OC31 and SS5 dataset of different methods. The result implies that the proposed algorithm leads to better results for three datasets.

TABLE VI. AVERAGE ACCURACY FOR OC10,OC31 AND SS5 DATASETS

Method	OC10	OC31	SS5
NA	88.32	68.05	63.25
GFK	89.09	62.7	67.55
ILS	90.09	62.93	74.71
JGSA	91.05	70.64	67.53
LDADA	92.47	70.81	76.35
Proposed Method-LR	**93.51**	**76.78**	**78.61**
Proposed Method-SVM	**93.36**	**74.94**	**78.73**

Domain adaptation can lead to inaccurate results if the differences in data distribution of the domains is not considered [10,30]. In addition, minimizing the difference of the marginal distribution of the domains [9,26] does not yield good results because it tries to minimize data distribution discrepancy without considering data label information. The results confirm that if minimization of distributions divergence is based on labeled information and a good classifier is selected, it can lead to better results. Considering class conditional distributions divergence between domains and using efficient classifier, the inter-domain divergences are minimized, which resulted in higher accuracies for domain adaptation tasks.

V. CONCLUSION

In this paper, a new approach was presented that is based on subspace learning in order to reduce class conditional probability distribution shift and to preserve the source discriminative information. Subspace learning was formulated as an optimization problem and was solved through eigenvalue decomposition. Prediction based on logistic regression or SVM classifier boosted this learning in an iterative process. The experimental results showed that the accuracy of the proposed algorithm for three datasets "Office", "Office + Caltech10" and "SS5" had better results in most cases of domain adaptation task. For future work, we will examine whether a better approximation of the data distribution would be possible, for example by considering probabilistic distribution of data. Also, the domain adaptation in heterogeneous case and more challenging datasets will be examined.

REFERENCES

[1] S.J.Pan, Q.Yang, "A survey on transfer learning", IEEE Trans. Knowl. Data Eng. 22, 10 (2010),1345–1359, 2010.

[2] J. Zhang, et al., "Recent Advances in Transfer Learning for Cross-Dataset Visual Recognition: A Problem-Oriented Perspective", ACM Computing Surveys (CSUR), 2019. 52(1): p. 7.

[3] S.Pan, I.W.Tsang, J.T.Y.Kwok, and Q.Yang, "Domain adaptation via transfer component analysis", In Proceedings of the International Joint Conference on Artificial Intelligence. 1187, 2009.

[4] M.Sugiyama, S.Nakajima, H.Kashima, P.V. Buenau, and M.Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation", In Proceedings of the Advances in Neural Information Processing Systems. 1433–1440, 2008.

[5] M.Baktashmotlagh, M.T.Harandi, B.C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2481–2488, 2014.

[6] S.Si, D.Tao, and B.Geng, "Bregman divergence-based regularization for transfer subspace learning". IEEE Trans. Knowl. Data Eng. 22, 7 (2010), 929–942, 2010.

[7] Y.Shi and F.Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation", In Proceedings of the International Conference on Machine Learning. 1079–1086, 2012.

[8] R.K.Sanodiya, and J. Mathew, "A framework for semi-supervised metric transfer learning on manifolds", Knowledge-Based Systems, 176: p. 1-14, 2019.

[9] S.Herath, M. Harandi, and F. Porikli. "Learning an invariant hilbert space for domain adaptation". in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[10] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation" in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2012, pp. 2066–2073,2012.

[11] S.Motiian, M.Piccirilli, D.A. Adjeroh, and G.Doretto, "Unified deep supervised domain adaptation and generalization", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.5715–5725, 2017.

[12] M.Long and J.Wang, "Learning transferable features with deep adaptation networks", In Proceedings of the International Conference on Machine Learning. 97–105, 2015.

[13] M.Long, H.Zhu, J.Wang, and M.I. Jordan, "Unsupervised domain adaptation with residual transfer networks", In Advances in Neural Information Processing Systems. MIT Press, 136–144, 2016.

[14] H.Venkateswara, J.Eusebio, S.Chakraborty, and S.Panchanathan, "Deep hashing network for unsupervised domain adaptation", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5018–5027, 2017.

[15] P.Wei, Y.Ke, and C.K.Goh, "Deep nonlinear feature coding for unsupervised domain adaptation", In Proceedings of the International Joint Conferences on Artificial Intelligence,2016.

[16] I.Goodfellow, J.Pouget-Abadie, M.Mirza, B.Xu, D.Warde-Farley, S.Ozair, A.Courville, and Y.Bengio, "Generative adversarial nets", In Advances in Neural Information Processing Systems. MIT Press, 2672–2680, 2014.

[17] E.Tzeng, J.Hoffman, K.Saenko, and T.Darrell. 2017, "Adversarial discriminative domain adaptation", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017).

[18] K. Saenko, B. Kulis, M. Fritz, and T.Darrell, "Adapting visual category models to new domains", In Proceedings of the European Conference on Computer Vision. Springer, 213–226, 2010.

[19] J.Xu, S.Ramos, D.Vazquez, and A. M. Lopez, "Domain adaptation of deformable partbased models", IEEE Trans. Pattern Anal. Mach. Intell. 36, 12 (2014), 2367–2380, 2014.

[20] L.Duan, D.Xu, and I.W.Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach", IEEE Trans. Neural Netw. Learn. Syst. 23, 3 (2012), 504–518, 2012.

[21] W.Dai, Q.Yang, G.Rong Xue, and Y.Yu, "Boosting for transfer learning". In Proceedings of the International Conference on Machine Learning. ACM, 193–200, 2007.

[22] Q.Wang, P. Bu, and T.P. Breckon, "Unifying Unsupervised Domain Adaptation and Zero-Shot Visual Recognition", arXiv preprint arXiv:1903.10601, 2019.

[23] P.Koniusz, Y.Tas, and F.Porikli, "Domain adaptation by mixture of alignments of second-or higherorder scatter tensors", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[24] L.Duan, I. W. Tsang, and Dong Xu. 2012, "Domain transfer multiple kernel learning", IEEE Trans. Pattern Anal. Mach. Intell. 34, 3 (2012), 465–479.

[25] L.Duan, D. Xu, I. W.Tsang, and Jiebo Luo, "Visual event recognition in videos by learning from web data", IEEE Trans. Pattern Anal. Machine Intell. 34, 9 (2012), 1667–1680, 2012.

[26] J.Zhang, W.Li, and P.Ogunbona, “Joint geometrical and statistical alignment for visual domain adaptation”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[27] J.Ye, R. Janardan, and Q. Li. “Two-dimensional linear discriminant analysis”. in Advances in neural information processing systems. 2005.

[28] T.Hofmann, B. Schölkopf, and A.J. Smola, “Kernel methods in machine learning”. The annals of statistics, 2008: p. 1171-1220.

[29] R.Gopalan, R.Li, and R.Chellappa. “Domain adaptation for object recognition:An unsupervised approach”. In proc. Int. Conference on Computer Vision(ICCV), 2011, pages 999-1006.

[30] H.Lu, et al., “An embarrassingly simple approach to visual domain adaptation”. IEEE Transactions on Image Processing, 2018. 27(7): p. 3403-3417.

[31] V. Risojevi´c and Z. Babi´c, “Aerial image classification using structural texture similarity,” in Proc. IEEE Int. Symp. Signal Process. Inf. Technol.(ISSPIT), Dec. 2011, pp. 190–195.

[32] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in Proc. SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst., 2010, pp. 270–279.

[33] D. Dai and W. Yang, “Satellite image classification via two-layer sparse coding with biased image representation,” IEEE Geosci. Remote Sens. Lett., vol. 8, no. 1, pp. 173–176, Jan. 2011.

[34] K.Chatfiel, K.Simonyan, A.Vedaldi, and A.Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in Proc. Brit. Mach. Vis. Conf. (BMVC), 2014,pp.1-12.

[35] H.Bay, T. Tuytelaars, and L. Van Gool. “Surf: Speeded up robust features”. in European conference on computer vision. 2006. Springer.

[36] E.Hatefi, H.Karshenas, P.Adibi, “Subspace Learning Augmented with Class Conditional Probability based on SVM Classifier in Domain Adaptation”, 25th International Computer Conference, Computer Society of iran(CSICC), 2020.