Hindawi
Complexity
Volume 2018, Article ID 8425821, 13 pages
https://doi.org/10.1155/2018/8425821

Research Article

Unsupervised Domain Adaptation Using Exemplar-SVMs with Adaptation Regularization

Yiwei He,[1] **Yingjie Tian**,[2,3,4] **Jingjing Tang**,[5] **and Yue Ma**[5]

[1]*School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China*
[2]*Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China*
[3]*Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China*
[4]*School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China*
[5]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Yingjie Tian; tyj@ucas.edu.cn

Received 28 August 2017; Accepted 18 February 2018; Published 22 April 2018

Academic Editor: Shirui Pan

Domain adaptation has recently attracted attention for visual recognition. It assumes that source and target domain data are drawn from the same feature space but different margin distributions and its motivation is to utilize the source domain instances to assist in training a robust classifier for target domain tasks. Previous studies always focus on reducing the distribution mismatch across domains. However, in many real-world applications, there also exist problems of sample selection bias among instances in a domain; this would reduce the generalization performance of learners. To address this issue, we propose a novel model named Domain Adaptation Exemplar Support Vector Machines (DAESVMs) based on exemplar support vector machines (exemplar-SVMs). Our approach aims to address the problems of sample selection bias and domain adaptation simultaneously. Comparing with usual domain adaptation problems, we go a step further in slacking the assumption of i.i.d. First, we formulate the DAESVMs training classifiers with reducing Maximum Mean Discrepancy (MMD) among domains by mapping data into a latent space and preserving properties of original data, and then, we integrate classifiers to make a prediction for target domain instances. Our experiments were conducted on Office and Caltech10 datasets and verify the effectiveness of the model we proposed.

1. Introduction

Over the past decades, machine learning technologies have achieved significant success in various areas, such as computer vision [1], natural language processing [2], and video detection [3]. However, traditional machine learning methods assume that training and testing data come from the same domain, which implies that training or testing data are drawn from the same distribution and represented in the same feature spaces. This assumption is too violated to be held in the real world as collecting suitable and enough labeled data is time consuming and an expensive manual effort. Lacking labeled data, most of traditional machine learning methods always lose their generalization performance in reality. Therefore, it is desired to utilize the data of the relational domain to help training a robust learner for target domains. Driven by this requirement, transfer learning has rapidly developed in recent years [4]. Transfer learning slacks the assumption of the traditional machine learning in which data or labels are drawn from the same distribution and represented in the same feature space. In the transfer learning settings, it is always assumed that domains are similar or related, with even no relationships, which is instead of i.i.d. assumption. Thus, transfer learning has a strong motivation when developing the classical machine learning functions or applying the functions to real-world applications. Besides, transfer learning can be regarded as a supplement of classical machine learning methods. One is the problem of covariate shift or sample selection bias. Another motivation is that we want to train a universal or general model as a predictor for all the tasks, viewed as the parameter or learner shared. It is also considered as a goal of Artificial General Intelligence. Transfer learning aims to utilize source or related domains to help target domain tasks. It has achieved significant success

in various practical applications, such as face recognition [5], natural language processing [6], cross-language text classification [7], WiFi localization [8], or medicine image [9].

Domain adaptation is a subproblem of transfer learning which assumes that source and target domain data are generated from the same feature and label space but different margin probability distributions. It aims to solve the problems that there is none or less labeled data in the target domain and usually use labeled data in the source domain to assist the training of target domain tasks. Massive works focus on the domain adaptation problems, and they also extend to some applications, such as WiFi location, text sentiment analysis, and image classification for multidomains. Since distribution mismatch generally exists in the real-world applications, there is also some other research area concern about domain adaptation. For example, extreme learning machine (ELM) is an efficient model for training single-hidden layer networks [10]. There are also some ELM works in a domain adaptation setting [11, 12]. They utilize most previous domain adaptation classifiers that have added constraint term which is based on using instance reweighting to minimize Maximum Mean Discrepancy (MMD) [13]. However, these methods need to assume that the difference between the source and target domain is not too large. Namely, this idea requires that different domains are similar.

Most pattern recognition problems can be transformed into several basic classification tasks. Generally speaking, classification tasks assume that a category can be represented by a hyperplane [14, 15], and most of the machine learning algorithms aim to learn hyperplanes to predict for unseen instances. Meanwhile, to improve the ability of representation by a hyperplane, there are some works which cluster the samples first and then solve the classification tasks on the clusters. In contrast to the category classification tasks, a cluster classifier can include more information about the positive category, but the more risks of overfitting. Motivated by the object detection, [16] proposed an extreme classification model training the classifiers for every positive instance and all the negative instances named exemplar support vector machines (E-SVMs). In fact, exemplar-SVMs can be viewed as an extreme situation of cluster-level SVM, in which every positive sample is regarded as a cluster. There are two viewpoints about the reason why the exemplar-SVM achieves a surprising generalization performance. One of the viewpoints is taking the exemplar-SVMs as a representation with complete details of positive instances. In other words, every classifier captures details of the positive instance like background, corner, color, or orientations and most of the classifiers can describe the category more intrinsically. From transfer learning viewpoint, training data cannot satisfy the underlying assumption of i.i.d., as every instance in the training set may be different from each other, namely, sample selection bias [17]. Each exemplar-SVMs classifier is trained on a high weight positive sample and other negative samples; it can represent the positive sample well in the same distribution. Recently, [18] extends exemplar-SVMs into a transfer learning form which uses loss function reweighting and adds a low-rank regularization item for classifiers.

In this work, we propose a novel model to address unsupervised domain adaptation problems that there is no label on target domain data. Furthermore, it permits distribution mismatch among instances. In our model, we train kernel exemplar classifiers for every positive instance and then integrate the classifier to make a prediction for target domain data. To align the distribution mismatch, we embed the regularization item based on TCA in our classifiers. In our opinion, the model constructs the bridge to transfer the knowledge, and we use the information in the kernel matrix which includes the instances representation in the high-dimension space to assist classifier training across domains. For the problem of sample selection bias, we integrate the classifiers to make a prediction. Basically, the step of integration is to expand the representation of hyperplanes that entirely take advantage of details learned before.

Our contributions are as follows. (1) We propose a novel unsupervised domain adaptation model based on exemplar-SVMs named Domain Adaptation Exemplar Support Vector Machines (DAESVMs), and it improves standard domain adaptation prediction accuracy by transferring knowledge across domains. (2) Every DAESVM classifier constructs a bridge that transmits knowledge from the source domain to target domain. Compared with the traditional two-step method, this strategy thoroughly searches the optimization point of the model which makes the classification hyperplane more precious about domains. (3) To solve the problem of sample selection bias, we use the ensemble methods to integrate the classifiers. The process of the ensemble is similar to slacking the classification hyperplane, which drops off some unreliable classification results and use the reliable parts to make a prediction. (4) We bring in the method of the pseudo label in DAESVMs inspired by [19] to supplement the information of target domain, and the experiments verify the effectiveness of the pseudo label. (5) We push a step further to extend to implementing DAESVMs on the multidomain adaptation. The rest of this paper is organized as follows. In Section, we introduce the notation of the problem. Meanwhile, we review the related works of domain adaptation, exemplar-SVM, and Transfer Component Analysis (TCA). In Section, we introduce the deduction process of DAESVM and formulate the model. In Section, we propose the optimization algorithm for our model. In Section, we integrate all the DAESVMs classifiers to make a prediction. In Section, we analyze the experiments on some transfer learning dataset to verify the effectiveness of DAESVMs. In Section, we conclude our work and give an expectation.

2. Notation and Related Works

This section will introduce the notation and related works about this paper.

2.1. Notation. In this paper, we use the notation of [4] definition in transfer learning, and the definition just considers the condition of one source domain and one target domain. First, it needs to define the *Domain* and *Task*. Domain $\mathcal{D}$ is composed of a feature space $\mathcal{X}$ and a margin probability distribution $P(x)$, namely, $\mathcal{D} = \{\mathcal{X}, P(x)\}$, $x \in \mathcal{X}$. Task $\mathcal{T}$

is composed of a label space $\mathcal{Y}$ and a prediction model $f(x)$, namely, $\mathcal{T} = \{\mathcal{Y}, f(x)\}$, $y \in \mathcal{Y}$. From view of probability, $f(x) = P(y \mid x)$. Notations in this paper which are frequently used are summarized in the Notations and Descriptions section. The definition of transfer learning is as follows: Give a source domain data $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), \ldots, (x_{S_{n_S}}, y_{S_{n_S}})\}$ and a source task $\mathcal{T}_S$ and a target domain data which is unlabeled $\mathcal{D}_T = \{(x_{T_1}), \ldots, (x_{T_{n_T}})\}$ and a target task $\mathcal{T}_T$. Transfer learning aims to utilize $\mathcal{D}_S$ and $\mathcal{D}_T$ to help train a robust prediction model $f_t(x)$ on the condition of $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

2.2. Domain Adaptation. As a subproblem of transfer learning, domain adaptation has achieved great success and is utilized in many applications. It assumes that source and target domain data have the same feature space, label space, and prediction function, from the view of probability, equaling conditional probabilities distribution, namely, $f_S(x) \neq f_T(x)$ or $P_S(y \mid x) \neq P_T(y \mid x)$. It is agreed that the approaches of domain adaptation can be divided into three parts, reweighting approach, feature transfer approach, and parameter shared approach.

(1) Reweighting Approaches. In the transfer learning tasks, the basic idea of utilizing the source data to help training target predictor is to reduce the discrepancy between the source and target data as far as possible. Under the assumption that source and target domains have a lot of overlapping features, a conventional method is reweighting or selecting the source domain instances to correct the marginal probability distribution mismatch. Based on the metric distance method between distributions named Maximum Mean Discrepancy (MMD), [20] proposed a technique called Kernel Mean Minimum (KMM) revising the weight of every instance to minimize MMD between the source and target domain. Being similar to KMM, [21] used the same idea but a different metric method to adjust the discrepancy of domains. Reference [22] used the strategy of AdaBoost to update the weights of source domain data, which improved the weight of instances in favor of classification task. It also introduced the generalization error bounds of model based on the PAC learning theory. In recent years, [23] used a two-step approach; first is sampling the instances which are similar with other domains as landmarks, and then use these landmarks to map the data into a high-dimension space, after which it is more overlapping. Reference [24] solved the same problem but slacked the similarity assumption; it assumes that there are no relationships between the source and target domain. The model named Selective Transfer Machine (STM) reweights the instance of personal faces to train a generic classifier. Most of instance-based transfer learning techniques use KMM to measure the difference of the distributions, and these methods are applied in many areas, such as facial action unit detection [25] and prostate cancer mapping [26].

(2) Feature Transfer Approaches. Compared with instance-based approaches, feature-based approaches slack the similarity assumption. It assumes that source and target domain share some features named shared features, and domains have their own features named spec-features [27]. For example, when we train a task that uses movie critical to help sofa critical sentiment analysis classification task. The word "comfortable" is always nonzero in the sofa domain features but always zero in the movie domain features. This word is the spec-feature of sofa domain feature. Feature transfer approaches aim to find a shared latent subspace where the distance between the source and target domain is minimized. Reference [28] proposed an unsupervised domain adaptation approach named Geodesic Flow Kernel (GFK) based on kernel method. GFK maps data into Grassmann manifolds and constructs geodesic flows to reduce the mismatch among domains. It effectively exploits intrinsic low-dimensional structures of data in domains. To solve problems of cross-domain natural language processing (NLP), [29] proposed a general method structural correspondence learning (SCL) to learn a discriminative predictor by identifying correspondences from features in domains. Primarily, SCL finds the pivot features and then links the shared features with each other. Reference [7] learned a predictor by mapping the target kernel matrix to a submatrix of the source kernel matrix. The deep neural network is used not for learning essential features but also for domain adaptation. Reference [30] proposed a neural network architecture for domain adaptation named Deep Adaptation Network (DAN) and extended it to joint adaptation networks (JAN) [31]. Reference [32] discussed the transferable domain features on the deep neural network.

(3) Parameter-Based Approaches. The core idea of parameter-based approaches aims to transfer parameters from source to target domain tasks. It assumes that different domains share some parameters and these parameters could be utilized for domains. Reference [33] proposed Adaptive Support Vector Machine (A-SVM) as a general method to adopt new domains. A-SVM trains an auxiliary classifier firstly and then learns the target predictor based on the original parameters. Reference [34] reweighted prediction of the source classifier on target domain by signing distance between domains.

2.3. Exemplar Support Vector Machines. Reference [16] is proposed for object detection and getting high performance. It trains classifiers on every positive instance from all negative instances. Every positive instance is an exemplar and the classifier corresponding to it can be viewed as a representation of the positive instance. In the process of the prediction, every classifier predicts a value for the test instance and uses a function to make a calibration for the value and then gets the high score classifiers result as a predicted class. The exemplar-SVMs solve the problem that a hyperplane is hard to represent a category instance and utilize an extreme strategy to train predictor. In [35], they gather the training procession into one model and enter the nuclear norm regularization to the scene of domain generalization which assumes target domain is unseen. They also extend the model to the problem of domain generalization and multiview [36, 37]. In [38], they reduced two hyperparameters into one and spread exemplar-SVMs to a kernel form.

2.4. Transfer Component Analysis. Reference [39] proposed a dimension reduction method called maximum mean discrepancy embedding (MMDE). By minimizing the distance of source and target domain data distribution in a shared latent space, the source domain data is utilized to assist training classifier on the target domain. MMDE is not only to minimize the distance between the domains in the latent space but also preserve the properties of data by maximum of the variance of data. Based on the MMDE, [40] extended it to have the ability of deal with the unseen instance and reduce the computation complexity of MMDE. Substantially, TCA simplifies the process of learning kernel matrix instead by transforming init kernel matrix. The optimization of this problem is equal to a solution in m leading eigenvectors of object matrix.

3. Domain Adaptation Exemplar Support Vector Machine

In this section, we present the formulation of Domain Adaptation Exemplar Support Vector Machine (DAESVM). In the remainder of this paper, we use a lowercase letter in boldface to represent a column vector and an uppercase in boldface to represent a matrix. The notation mentioned in Section is extended. We use $\mathbf{x}_i^+, i \in \{1, \ldots, n_S^+\}$, where n_S^+ is the number of positive instances, to represent a positive instance, and $\mathbf{x}_j^-, j \in \{1, \ldots, n_S^-\}$, where n_S^- is the number of negative instances, to represent a negative instance. The set of negative samples are written as N^-. This section introduces the formulation procession of an exemplar classifier. In fact, we need to train exemplar classifiers in the number of source domain instances and the method which integrates these classifiers is proposed in Section.

3.1. Exemplar-SVM. The exemplar-SVM is constructed by an extreme idea of training a classifier by a positive instance from all the negative instances and then calibrating the outputs of classifiers into a probability distribution to separate the samples. The model trains the number of positive instance classifiers. Learning a classifier which aims to separate a positive instance from all the negative instance can be modeled as

$$f(\mathbf{w}, b) = \|\mathbf{w}\|^2 + C_1 h\left(\mathbf{w}^T \mathbf{x}^+ + b\right) + C_2 \sum_{\mathbf{x}_i^- \in n_S^-} h\left(-\mathbf{w}^T \mathbf{x}_i^- - b\right), \tag{1}$$

where $\|\cdot\|$ is 2-norm of a vector and C_1 and C_2 are the tradeoff parameters corresponding to C in SVM for balancing the positive and negative error cost. $h(x) = \max(0, 1 - x)$ is a hinge loss function.

The formulation (1) is the primal problem of exemplar-SVM, and we can find the dual problem for utilizing kernel method. The dual formulation can be written as follows [38]:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^T \widetilde{\mathbf{K}} \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha}, \\ &\text{s.t.} \quad \alpha_0 - \sum_{i=1}^{n_S^-} \alpha_i = 0, \\ &\qquad 0 \le \alpha_0 \le C_1, \\ &\qquad 0 \le \alpha_i \le C_2, \\ &\qquad \forall i \ge 1. \end{aligned} \tag{2}$$

$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{n_S^-}) \in \mathcal{R}^{n_S^- + 1}$ are Lagrangian multipliers. $\mathbf{e}$ is an identity vector. We take this model as an exemplar learner. The matrix $\widetilde{\mathbf{K}} \in \mathcal{R}^{(n_S^- + 1) \times (n_S^- + 1)}$ is composed of

$$\begin{aligned} &\widetilde{\mathbf{K}} = \begin{bmatrix} k(\mathbf{x}^+, \mathbf{x}^+) & -\mathbf{k}^T \\ -\mathbf{k} & \mathbf{K} \end{bmatrix} \in \mathcal{R}^{(n_S^- + 1) \times (n_S^- + 1)} \\ &k \in \mathcal{R}^{N_S^-}, \\ &k_i = k(\mathbf{x}, \mathbf{x}_i^-), \\ &\mathbf{K}_{ij} = k(\mathbf{x}_i^-, \mathbf{x}_j^-). \end{aligned} \tag{3}$$

3.2. Pseudo Label for Kernel Matrix. To make the best use of samples in source or target, we construct the kernel matrix on both domain data. However, in the dual problem of SVM, kernel matrix $\mathbf{K}$ needs to be supplied labeled data. Our model is based on the unsupervised domain adaptation problem in which only source domain data are labeled. Motivated by [19], we use the pseudo label to help model training. Pseudo labels are predicted by classical classifiers, SVM in our model, which train on the source labeled data. Due to the distribution mismatch between source and target domain, there may be many labels incorrect. Followed by [19], we assume that the pseudo class centroids calculated by them may reside not far apart from the true class centroids. Thus, we use both domain data to supplement the kernel matrix $\mathbf{K}$ with label information. In our experiments, we testify this method is effective.

3.3. Exemplar Learner in Domain Adaptation Form. In fact, each exemplar learner is an SVM in kernel form which is trained by a positive instance and all the negative instances. In the opinion of [16], a discriminative exemplar classifier can be taken as a representation of a positive instance. However, in the task of object detection or image classification, this parametric form representation is feasible because of some characteristics in samples, such as angle, color, orientations, and background, which are hard to represent. The instance-based parametric discriminative classifier can include more information about positive samples. Similarly, with the motivation of transfer learning, we can view a positive instance as a domain, and there is some mismatch among domains. Our model aims to correct this mismatch and reduce the distance from the target domain. We construct an exemplar learner distance metric of domains from MMD and it can be written as

$$\begin{aligned} &\text{dist}(\mathbf{x}_S, \mathbf{x}_T) \\ &= \left\| \phi(\mathbf{x}_S^+) + \frac{1}{n_S^-} \sum_{i=1}^{n_S} \phi(\mathbf{x}_S^-) - \frac{2}{n_T} \sum_{i=1}^{n_T} \phi(\mathbf{x}_T) \right\|_{\mathcal{H}}^2. \end{aligned} \tag{4}$$

However, it is just a metric of distance which is satisfied with our requirement of minimizing this distance by some transformation. Motivated by Transfer Component Analysis (TCA), we want to map the instance into a latent space that the instances from source and target domain are more similar and assume this mapping is $P(x)$. Namely, we aim to minimize MMD distance between domains by mapping instances into another space. We extend the distance function as follows:

$$\begin{aligned}\operatorname{dist}\left(\mathbf{x}_S, \mathbf{x}_T\right) = \Bigg\| \phi\left(P\left(\mathbf{x}_S^+\right)\right) + \frac{1}{n_S}\sum_{i=1}^{n_S}\phi\left(P\left(\mathbf{x}_S^-\right)\right) \\ - \frac{2}{n_T}\sum_{i=1}^{n_T}\phi\left(P\left(\mathbf{x}_T\right)\right)\Bigg\|_{\mathscr{H}}^2.\end{aligned} \tag{5}$$

Corresponding to a general approach, it always reformulates (4) to construct a kernel matrix form. We define the Gram matrices on the source positive domain, source negative domain, and target domain. The kernel matrix $\mathbf{K}$ is composed of nine submatrices, $\mathbf{K}_{T,+}$, $\mathbf{K}_{T,-}$, $\mathbf{K}_{T,T}$, $\mathbf{K}_{+,+}$, $\mathbf{K}_{-,-}$, $\mathbf{K}_{+,-}$, $\mathbf{K}_{+,S}$, $\mathbf{K}_{-,T}$, $\mathbf{K}_{-,+}$ where $\mathbf{K} = [\phi(x_i)^T \ \phi(x_j)]$.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{+,+} & \mathbf{K}_{+,-} & \mathbf{K}_{+,T} \\ \mathbf{K}_{-,+} & \mathbf{K}_{-,-} & \mathbf{K}_{-,T} \\ \mathbf{K}_{T,+} & \mathbf{K}_{T,-} & \mathbf{K}_{T,T} \end{bmatrix} \in \mathscr{R}^{(1+n_S^- + n_T)\times(1+n_S^- + n_T)}, \tag{6}$$

and it constructs the coefficient matrix $\mathbf{L}$,

$$\mathbf{L}_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_S^+, \\ \dfrac{1}{n_S^-}, & \text{if } \mathbf{x}_i \in \mathbf{X}_S^+,\ \mathbf{x}_j \in \mathbf{X}_S^-, \\ -\dfrac{2}{n_T}, & \text{if } \mathbf{x}_i \in \mathbf{X}_S^+,\ \mathbf{x}_j \in \mathbf{X}_T, \\ -\dfrac{2}{n_S^- n_T}, & \text{if } \mathbf{x}_i \in \mathbf{X}_T,\ \mathbf{x}_j \in \mathbf{X}_S^-, \\ \dfrac{1}{{n_S^-}^2}, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_S^-, \\ \dfrac{4}{{n_T}^2}, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_T. \end{cases} \tag{7}$$

Thus, the primal distance function is represented by $\mathbf{KL}$. Motivated by TCA [40], the mapping for primal data is equal to the transformation of kernel matrix generated by the source and target domain data. Utilizing the low-dimension transform matrix $\widetilde{\mathbf{M}} \in \mathscr{R}^{(1+n_S^- + n_T)\times m}$ reduces the dimension of the primal kernel matrix. It maps the empirical kernel map $\mathbf{K} = (\mathbf{K}\mathbf{K}^{-1/2})(\mathbf{K}^{-1/2}\mathbf{K})$ into an m-dimensional shared space. Mostly, we replaced the distance function $\mathbf{KL}$ by $(\mathbf{K}\mathbf{M}\mathbf{M}^T\mathbf{K}\mathbf{L})$. In our case, we follow [40] and minimize the trace of the distance,

$$\operatorname{dist}\left(\mathbf{x}_S^+, \mathbf{x}_S^-, \mathbf{x}_T\right) = \operatorname{tr}\left(\mathbf{M}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{M}\right). \tag{8}$$

For controlling the complexity of $\mathbf{M}$ and preserving the data characteristic, we add the regularization and constraint item. The domain adaptation item is formulated followed from TCA and written as

$$\begin{aligned}\Omega\left(\mathbf{x}_S^+, \mathbf{x}_S^-, \mathbf{x}_T\right) &= \operatorname{tr}\left(\mathbf{M}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{M}\right) + \mu \operatorname{tr}\left(\mathbf{M}^T\mathbf{M}\right) \\ \text{s.t.} \quad & \mathbf{M}^T\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M} = \mathbf{I}_m,\end{aligned} \tag{9}$$

where $\mu > 0$ is a tradeoff parameter and $\mathbf{I}_m \in \mathscr{R}^{(m\times m)}$ is an identity matrix. $\mathbf{H} = \mathbf{I}_{n_S^- + n_T + 1} - (1/(n_S^- + n_T + 1))\mathbf{e}\mathbf{e}^T$ is a centering matrix.

Furthermore, the objective function of dual SVM needs to be added to the training label information which is similar to our model. Thus, we construct the training label matrix $\mathbf{U}$

$$\mathbf{U} = \operatorname{diag}\left(\mathbf{y}_S^+, \mathbf{y}_S^-, \mathbf{y}_T\right). \tag{10}$$

$\mathbf{y}_S^+$ is the label of a positive instance, $\mathbf{y}_S^-$ is the label vector of negative source instances, and $\mathbf{y}_T$ is the pseudo labels of target instances which are predicted by SVM before. It can be rewritten in another form:

$$\mathbf{U} = \operatorname{diag}\left(1, \underbrace{-1, \ldots, -1}_{n_S^-}, \underbrace{y_T^1, \ldots, y_T^{n_T}}_{n_T}\right). \tag{11}$$

Label matrix $\mathbf{U}$ provides the information of source domain data labels and target domain pseudo labels. The matrix $\widetilde{\mathbf{K}}$ in a dual problem of exemplar-SVM (2) is primal data kernel matrix. We want to replace it by mapping the kernel matrix into a latent subspace. Namely, replace $\mathbf{K}$ by $\widehat{\mathbf{K}}$ and the final objective function of each DAESVM model is formulated as follows:

$$\begin{aligned}\min_{\alpha,\mathbf{M}} \quad & \boldsymbol{\alpha}^T\widehat{\mathbf{K}}\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} + \lambda \operatorname{tr}\left(\mathbf{M}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{M}\right) \\ & + \mu \operatorname{tr}\left(\mathbf{M}^T\mathbf{M}\right), \\ \text{s.t.} \quad & \alpha_0 - \sum_{i=1}^{n_S^- + n_T} \alpha_i = 0, \\ & 0 \le \alpha_0 \le C_1, \\ & 0 \le \alpha_i \le C_2, \\ & \forall i \ge 1, \\ & \mathbf{M}^T\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M} = \mathbf{I}_m, \\ & \widehat{\mathbf{K}} = \mathbf{U}\mathbf{K}\mathbf{M}\mathbf{M}^T\mathbf{K}\mathbf{U}.\end{aligned} \tag{12}$$

4. Optimization Algorithm

To minimize problem (12), we adopt the alternated optimization method which alternates between solving two subproblems over parameter $\boldsymbol{\alpha}$ and mapping matrix $\mathbf{M}$. Under these methods, the alternated optimization approach is guaranteed to decrease the objective function. Algorithm 1 summarizes the optimization procedure of problem (12) which we formulated.

Input: $\mathbf{X}^{tr}$, $\mathbf{X}^{te}$; parameter λ, μ, m, C_1 and C_2;
Output: optimal $\boldsymbol{\alpha}$ and $\mathbf{M}$
(1) initial $\boldsymbol{\alpha} = \mathbf{0}$;
(2) Construct kernel matrix $\mathbf{K}$ from $\mathbf{X}^{tr}$ and $\mathbf{X}^{te}$ based on (6); coefficient matrix $\mathbf{L}$ based on (7); centering matrix $\mathbf{H}$; label matrix $\mathbf{U}$ based on (11).
(3) **repeat**
(4) Update transformation matrix $\mathbf{M}$ when fix $\boldsymbol{\alpha}$
(5) Eigendecompose the optimization matrix of $(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m)^{-1}\mathbf{KHK}$ and select m leading eigenvectors to construct the transformation matrix $\mathbf{M}$
(6) Solve the convex optimization problem for fixed $\mathbf{M}$ to optimize $\boldsymbol{\alpha}$
(7) **until** Convergence

Algorithm 1: Domain Adaptation Exemplar Support Vector Machine.

Minimizing over $\boldsymbol{\alpha}$. The optimization over $\boldsymbol{\alpha}$ can be rewritten into the following form:

$$\begin{aligned} \min_{\mathbf{M}} \quad & \boldsymbol{\alpha}^T\mathbf{UKMM}^T\mathbf{KU}\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} + \lambda \operatorname{tr}\left(\mathbf{M}^T\mathbf{KLKM}\right) \\ & + \mu \operatorname{tr}\left(\mathbf{M}^T\mathbf{M}\right), \\ \text{s.t.} \quad & \mathbf{M}^T\mathbf{KHKM} = \mathbf{I}_m. \end{aligned} \tag{13}$$

Being similar to TCA, the formulation is containing a non-convex norm constraint, and we transform this optimization problem by reformulating as

$$\max_{\mathbf{M}} \quad \operatorname{tr}\left(\left(\mathbf{M}^T\left(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m\right)\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{KHKM}\right). \tag{14}$$

Proof. The Lagrangian of (12) is

$$\begin{aligned} \mathscr{L}(\mathbf{M}, \mathbf{Z}) = {} & \boldsymbol{\alpha}^T\mathbf{UKMM}^T\mathbf{KU}\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \\ & + \lambda \operatorname{tr}\left(\mathbf{M}^T\mathbf{KLKM}\right) - \mu \operatorname{tr}\left(\mathbf{M}^T\mathbf{M}\right) \\ & - \operatorname{tr}\left(\left(\mathbf{M}^T\mathbf{KHKM} - \mathbf{I}_m\right)\mathbf{Z}\right). \end{aligned} \tag{15}$$

Because the initial kernel matrix $\mathbf{K}$ is a symmetric matrix and we can rewrite the first term of (15),

$$\begin{aligned} \boldsymbol{\alpha}^T\mathbf{UKMM}^T\mathbf{KU}\boldsymbol{\alpha} &= \operatorname{tr}\left(\boldsymbol{\alpha}^T\mathbf{UKMM}^T\mathbf{KU}\boldsymbol{\alpha}\right) \\ &= \operatorname{tr}\left[\left(\mathbf{M}^T\mathbf{K}^T\mathbf{U}\boldsymbol{\alpha}\right)^T \left(\mathbf{M}^T\mathbf{K}^T\mathbf{U}\boldsymbol{\alpha}\right)\right] \\ &= \operatorname{tr}\left[\left(\mathbf{M}^T\mathbf{K}^T\mathbf{U}\boldsymbol{\alpha}\right) \left(\mathbf{M}^T\mathbf{K}^T\mathbf{U}\boldsymbol{\alpha}\right)^T\right] \\ &= \operatorname{tr}\left(\mathbf{M}^T\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UKM}\right). \end{aligned} \tag{16}$$

The original Lagrangian formulation is written as follows:

$$\begin{aligned} & \operatorname{tr}\left(\mathbf{M}^T\left(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m\right)\mathbf{M}\right) \\ & - \operatorname{tr}\left(\left(\mathbf{M}^T\mathbf{KHKM} - \mathbf{I}_m\right)\mathbf{Z}\right). \end{aligned} \tag{17}$$

The derivative of (17) is

$$\left(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m\right)\mathbf{M} - \mathbf{KHKMZ}. \tag{18}$$

We set the derivative above to zero, and we get $\mathbf{Z}$ as

$$\begin{aligned} \mathbf{Z} = {} & \left(\mathbf{M}^T\mathbf{KHKM}\right)^{\dagger}\mathbf{M}^T\left(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m\right) \\ & \cdot \mathbf{M}. \end{aligned} \tag{19}$$

Substituting $\mathbf{Z}$ into (17), we obtain

$$\min_{\mathbf{M}} \quad \operatorname{tr}\left(\left(\mathbf{M}^T\mathbf{KHKM}\right)^{\dagger}\mathbf{M}^T\left(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m\right)\mathbf{M}\right). \tag{20}$$

Final, we obtain an equivalent maximization problem (14). □

Being similar to TCA, the solution is finding the m leading eigenvectors of $(\mathbf{KU}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{UK} + \lambda\mathbf{KLK} - \mu\mathbf{I}_m)^{-1}\mathbf{KHK}$.

Minimizing over $\mathbf{M}$. The optimization over $\mathbf{M}$ can be rewritten into the following QP form:

Input: $\mathbf{y}_S$ $\boldsymbol{\alpha}$, $\mathbf{X}^{te}$; parameter $\mathscr{P}$
Output: prediction labels $\mathbf{y}$
(1) Compute the weights $\mathbf{w}$ of the classifiers.
(2) Construct weight matrix $\mathbf{W}$ and bias $\mathbf{b}$ of predictors based on $\boldsymbol{\alpha}$.
(3) **repeat**
(4) Compute scores of each classifier in this category.
(5) Find top $\mathscr{P}$ scores.
(6) Compute the sum of these top scores.
(7) **until** The number of categories
(8) Choose the max score owned category as the prediction label $\mathbf{y}$.

Algorithm 2: Ensemble Domain Adaptation Exemplar Classifiers.

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^T \widehat{\mathbf{K}} \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha}, \\ \text{s.t.} \quad & \alpha_0 - \sum_{i=1}^{n_S^- + n_T} \alpha_i = 0, \\ & 0 \le \alpha_0 \le C_1, \\ & 0 \le \alpha_i \le C_2, \\ & \forall i \ge 1. \end{aligned} \tag{21}$$

$\widehat{\mathbf{K}} = \mathbf{UKMM}^T\mathbf{KU}$ which represents the kernel matrix has been transformed by transformation matrix $\mathbf{M}$. It is obvious that this problem is a QP problem and it could be solved efficiently using interior point methods or other successive optimization procedures such as Alternating Direction Method of Multipliers (ADMM).

5. Ensemble Domain Adaptation Exemplar Classifiers

In this section, we introduce the method of integration exemplar classifiers. As mentioned before, we get the number of source domain instances classifiers and this section aims to predict labels for target domain instances. In our opinions, the classification hyperplane of an exemplar classifier is representation for a source domain positive instance. However, most of the hyperplanes contain information which comes from various samples, such as images of different background or source. In fact, we aim to search the exemplar classifiers which are from instances similar to the testing sample. Thus, we utilize integrating method to filter out classifiers which include details different with the testing sample. Another view for the integration method is that it slacks the part of hyperplanes. Namely, it removes some exemplar classifiers which are trained by large instances distribution mismatch.

In our method, we first construct the classifiers from Lagrange multipliers $\boldsymbol{\alpha}$. The classifier construction equation is

$$\mathbf{w} = \alpha_0 \mathbf{x}^+ - \sum_{i=1}^{n_S^- + n_T} \alpha_i \mathbf{x}_i^-, \tag{22}$$

where $\mathbf{w}$ is the weight of classifier.

$$b = y_j - \alpha_0 \widehat{\mathbf{K}}_{0,j} - \sum_{i=1}^{n_S^- + n_T} y_i \alpha_i \widehat{\mathbf{K}}_{ij}, \tag{23}$$

where b is the bias of classifier. The classifier is given by

$$s = \boldsymbol{\alpha}^\top \mathbf{x} + b. \tag{24}$$

And then we compute the scores by every classifier and the testing instance. Second, we find the top $\mathscr{P}$ numbers of scores for each class classifier and compute the sum of those scores. At last, we get a score for each class, and the highest score is the category that we predict. The prediction method is described in Algorithm 2.

6. Experiments

In this section, we conduct experiments onto the four domains, Amazon, DSLR, Caltech, and Webcam, to evaluate the performance of proposed Domain Adaptation Exemplar Support Vector Machines. We first compare our method to baselines and other domain adaptation methods. Next, we analyze the effectiveness of our approach. At last, we introduce the problem of parameter sensitivity.

6.1. Data Preparation. We run the experiments on Office and Office Caltech datasets. Office dataset contains three domains Amazon, Webcam, and DSLR. Each of them includes images from amazon.com or Office environment images taken with varying lighting and pose changes using a Webcam or a DSLR camera. Office Caltech dataset contains the ten overlapping categories between the Office dataset and Caltech-256 dataset. By the standard transfer learning experiment method, we merge two datasets; it entirely includes four domains Amazon, DSLR, Caltech, and Webcam which are studied in [41]. The dataset of Amazon is the images downloaded from Amazon merchants. The images in the Webcam also come from the online web page, but they are of low quality as they are taken by web camera. The domain of DSLR is photographed by the digital SLR camera by which the images are of high quality. Caltech is always added to domain adaptation experiments, and it is collected by object detection tasks. Each domain has its characteristic. Compared to the other domains, the quality of images in

the DSLR is higher than others and the influence factors such as object detection and background are less than images downloaded from the web. Amazon and Webcam come from the web, and images in the domains are of low quality and more complexity. However, there are some different details on each of them. Instances in the Webcam are object alone, but the composition of samples in Amazon is more complex including background and other goods. Figure 1 shows the example of the backpack from four domain samples. In the view of transfer learning, the datasets come from different domains and the different margin probabilities for the images. In our model, we aim to solve this problem and get a robust classifier for the cross-domain.

We chose ten common categories among all four datasets: backpack, bike, bike helmet, bookcase, bottle, calculator, desk chair, desk lamp, desktop computer, and file cabinet. There are 8 to 151 samples per category in a domain: 958 images in Amazon, 295 images in Webcam, 157 images in DSLR, 1123 images in Caltech, and 2533 images total in the dataset. Figure 1 shows examples for datasets.

We follow both SURF and DeCAF features extraction in the experiments. First, we use SURF features encoding the images into 800-bin histograms. Next, we use DeCAF feature which is extracted by 7 layers of Alex-net [42] into 4096-bin histograms. At last, we normalized the histograms and then z-scored to have zero mean and unit standard deviation in each dimension.

We run our experiments on a standard way for visual domain adaptation. It always uses one of four datasets as source domain and another one as target domain. Each dataset provides same ten categories and uses the same representation of images which is considered as the problem of homogeneous domain adaptation. For example, we choose images taken by the set of DSLR (denoted by D) as source domain data and use images in Amazon (denoted by A) as target domain data. This problem is denoted as $\mathscr{D} \rightarrow \mathscr{A}$. Using this method, we can compose 12 domain adaptation subproblems from four domains.

6.2. Experiment Setup

(1) Baseline Method. We compare our DAESVM method with three kinds of classical approaches: one is classified without regularization of transfer learning, the second is conventional transfer learning methods, and the last one is the foundation model, which is low-rank exemplar support vector machine. The methods are listed as follows:

(1) Transfer Component Analysis (TCA) [40]

(2) Support Vector Machine (SVM) [43]

(3) Geodesic Flow Kernel (GFK) [28]

(4) Landmarks Selection-based Subspace Alignment (LSSA) [23]

(5) Kernel Mean Maximum (KMM) [20]

(6) Subspace Alignment (SA) [44]

(7) Joint Matching Transfer (TJM) [45]

(8) Low-Rank Exemplar-SVMs (LRESVMs) [18]

TCA, GFK, and KMM are the classical transfer learning methods. We compare our model with these methods. Besides, we prove our method is more robust than models without domain adaptation items in the transfer learning scenery. TCA is the foundation of our model, and it is similar to GFK and SFA which are based on the idea of feature transfer. KMM transfer knowledge by instance reweighting. TJM is a popular model utilizing the problem of unsupervised domain adaptation. SA and LSSA are the models using landmarks to transfer knowledge.

(2) Implementation Details. For baseline method, SVM is trained on the source data and tested on the target data [46]. TCA, SA, LSSA, TJM, and GFK are first viewed as dimension reduction process and then train a classifier on the source data and make a prediction for the target domain [19]. Being similar to dimension reduction, KMM is first to compute the weight of each instance and then train predictor on the reweighting source data.

Under the assumption of unsupervised domain adaptation, it is impossible to tune the optimal parameters for the target domain task by cross validation, since there exists distribution mismatch between domains. Therefore, in the experiments, we adopt the strategy of Grid Search to obtain the best parameters and report the best results. Our method involves five tunable parameters: tradeoff in ESVM C_1 and C_2, tradeoff in regularization items λ and μ, and parameter of dimension reduction m. The parameters of tradeoff in ESVM C_1 and C_2 are selected over $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{-0}, 10^{1}, 10^{2}, 10^{3}\}$. We fix $\lambda = 1$, $\mu = 1$, $m = 40$ empirically and select radial basic function (RBF) as the kernel function. In fact, our model is relatively stable under a wide range of parameter values. We train a classifier for every positive instance in the source domain data and then we put them into a probability distribution. We deal with the multiclass classifier in a one versus the others way. To measure the performance of our method, we use the average accuracy and the standard deviation over ten repetitions. The average testing accuracies and standard errors for all 12 tasks of our methods are reported in Table 1. For the rest of baseline experiments, most of them are cited by the papers which are published before.

6.3. Experiments Results. In this section, we compare our DAESVM with baseline methods regarding classification accuracy.

Table 1 summarizes the classification accuracy obtained by all the 10 categories and generates 12 tasks in 4 domains. The highest accuracy is in a bold font which indicates that the performance of this task is better than others. First, we implement the traditional classifiers without domain adaptation items that we train the predictors on the source domain data and make a prediction for target domain dataset. Second, we compared our DAESVM with unsupervised domain adaptation methods, such as TCA or GFK, implemented to use the same dimension reduction with the parameter m in our model. At last, we also compared DAESVM with newly transfer learning models, like low-rank ESVMs [18].

FIGURE 1: Example images from the backpack category in Amazon, DLSR ((a) from left to right), Webcam, and Caltech-256 ((b) from left to right). The different domain images are various. The images have different style, background, or sources.

Overall, in a usual transfer learning way, we run datasets across different pairs of source and target domain. The accuracy of DAESVM for the adaptation from DSLR to Webcam can achieve 92.1% which make the improvement over LRESVM by 1.2%. Compared with TCA, DAESVMs make a consideration about the distribution mismatch among instances or different domains. For the adaptation from Webcam to DSLR, this task can get the accuracy of 91.8%. For the domain datasets Amazon and Caltech which are more significant than DSLR and Webcam, DAESVM gets the accuracy of 77.5% which improves about 36.2% compared to the method of TJM. For the ability which transfers knowledge

Table 1: Classification accuracies of different methods for different tasks of domain adaptation. We conduct the experiments on conventional transfer learning methods. Comparing with traditional methods, DAESVMs gain a big improvement in the prediction accuracy. And they also improve confronted with the approach of LRESVM which is proposed recently [average ± standard error of accuracy (%)].

Task	SVM	KMM	TCA	TJM	SA	GFK	LSSA	LRESVM	DAESVMs
$\mathscr{A} \to \mathscr{C}$	45.4	42.2	45.3	56.9	51.8	49.6	54.8	**79.8**	77.5 ± 0.79
$\mathscr{A} \to \mathscr{D}$	50.7	42.7	60.3	56.4	56.4	55.7	57.3	74.9	**76.8 ± 0.76**
$\mathscr{A} \to \mathscr{W}$	47.4	42.4	61.3	51.0	54.7	56.9	56.7	**75.4**	73.2 ± 1.08
$\mathscr{C} \to \mathscr{A}$	50.7	48.3	54.7	58.6	57.1	51.2	58.4	77.2	**80.2 ± 0.39**
$\mathscr{C} \to \mathscr{D}$	53.2	53.5	56.4	57.4	59.0	57.1	59.1	87.1	**89.0 ± 0.23**
$\mathscr{C} \to \mathscr{W}$	44.2	45.8	50.4	58.8	62.7	57.1	58.1	74.1	**74.7 ± 0.38**
$\mathscr{D} \to \mathscr{A}$	40.8	42.2	53.8	46.1	58.9	59.2	58.4	80.4	**83.4 ± 1.41**
$\mathscr{D} \to \mathscr{C}$	48.3	41.6	43.9	49.6	54.3	59.4	57.7	**79.0**	73.0 ± 1.04
$\mathscr{D} \to \mathscr{W}$	67.8	72.9	82.4	82.0	83.4	80.2	87.1	91.0	**92.1 ± 0.25**
$\mathscr{W} \to \mathscr{A}$	42.4	41.9	53.0	50.8	57.0	66.2	59.7	74.3	**77.8 ± 0.33**
$\mathscr{W} \to \mathscr{C}$	41.2	39.0	53.7	54.8	34.7	52.4	54.2	**70.6**	66.5 ± 0.54
$\mathscr{W} \to \mathscr{D}$	80.2	82.0	87.9	83.4	78.9	81.2	87.2	89.2	**91.8 ± 0.59**
Average	51.0	49.5	58.6	58.8	59.1	60.5	62.4	79.4	**80.0 ± 0.67**

Table 2: We also conduct our experiments for the tasks of multidomain and gain an improvement comparing with methods proposed before. The experiments adopt the same strategy as the single domain adaptation. We treat multidomain as one source or target to find the shared features in a latent space. However, the complexity of the multidomain shared features limits the accuracy of tasks [average ± standard error of accuracy (%)].

Task	SVM	KMM	TCA	TJM	SA	GFK	LSSA	LRESVM	DAESVM
$\mathscr{D}, \mathscr{W} \to \mathscr{A}$	45.7	37.4	40.5	57.1	59.4	47.3	61.7	**80.1**	77.2 ± 1.27
$\mathscr{A}, \mathscr{D} \to \mathscr{C}, \mathscr{W}$	37.1	31.6	43.0	60.2	48.7	47.6	74.2	**86.9**	84.7 ± 0.65
$\mathscr{D} \to \mathscr{A}, \mathscr{C}, \mathscr{W}$	41.4	43.8	57.2	63.9	51.9	51.4	77.0	82.9	**88.4 ± 0.21**
$\mathscr{A}, \mathscr{D}, \mathscr{W} \to \mathscr{C}$	43.9	50.6	54.9	69.0	60.2	60.4	63.7	87.7	**90.1 ± 0.34**
$\mathscr{A}, \mathscr{D} \to \mathscr{W}$	71.0	61.0	54.0	61.3	54.0	47.0	71.9	80.8	**83.8 ± 0.78**
$\mathscr{A}, \mathscr{C} \to \mathscr{D}, \mathscr{W}$	81.4	53.9	77.4	71.8	57.4	64.1	80.7	89.3	**92.4 ± 0.25**
Average	53.4	46.4	54.5	63.9	55.2	53.0	71.5	84.6	**86.1 ± 0.58**

from large dataset to small domain dataset, from Amazon to DSLR, we get the accuracy of 76.8%. Contrarily, from DSLR to Amazon, the prediction accuracy is 83.4%. Totally speaking, our DAESVM trained on one domain has good performance and will also have robust performance on multidomain.

We also complement tasks of multidomains adaptation, which utilized one or more domains as source domain data and made an adaptation to other domains. The results are shown in Table 2. The accuracy of DAEVM for the adaptation from Amazon, DSLR, and Webcam to Caltech achieves 90.1% which get the improvement over LERSVM. For the task of adaptation from Amazon and Caltech to Webcam, DSLR can get the accuracy of 92.4%. The experiments prove that our models are effective not only for single domain adaptation but also for multidomain adaptation.

Two key factors may contribute to the superiority of our method: The feature transfer regularization item is utilized to slack the similarity assumption. It just assumes that there are some shared features in different domains instead of the assumption that different domains are similar to each other. This factor makes the model more robust than models with reweighting item. The second factor is the exemplar-SVMs which are proposed from a motivation of transfer learning which makes a consideration that instances are distribution mismatch from each other. Our model combines these two factors to resist the problem of distribution mismatch among domains and sample selection bias among instances.

6.4. Pseudo Label Effectiveness. Following [19], we use pseudo labels to supplement training model. In our experiments, we test the prediction results which are influenced by the accuracy rate of pseudo labels. As a result, described by Figure 2, the prediction accuracy is improved following the increasing accuracy of pseudo labels. It is proved that the method of the pseudo label is effective and we can do the iteration by using the labels predicted by the DAESVM as the pseudo labels. The iteration step can efficiently enhance the performance of the classifiers.

6.5. Parameter Sensitivity. There are five parameters in our model, and we conduct the parameter sensitivity analysis which can achieve optimal performance under a wide range of parameter values and discuss the results.

(1) Tradeoff λ. λ is a tradeoff to control the weight of MMD item which aims to minimize the distribution mismatch between source and target domain. Theoretically, we want this term to be equal to zero. However, if we set this

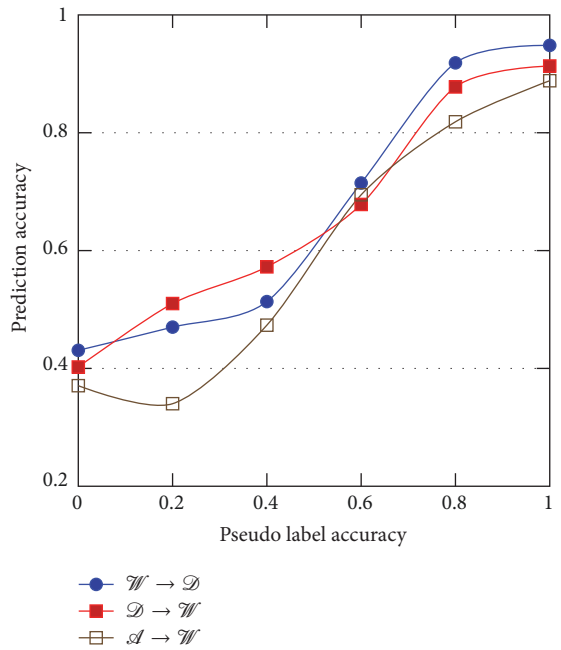

Figure 2: The accuracy of DAESVMs is improved with the improvement of the pseudo label accuracy. The results verify the effectiveness of the pseudo label method.

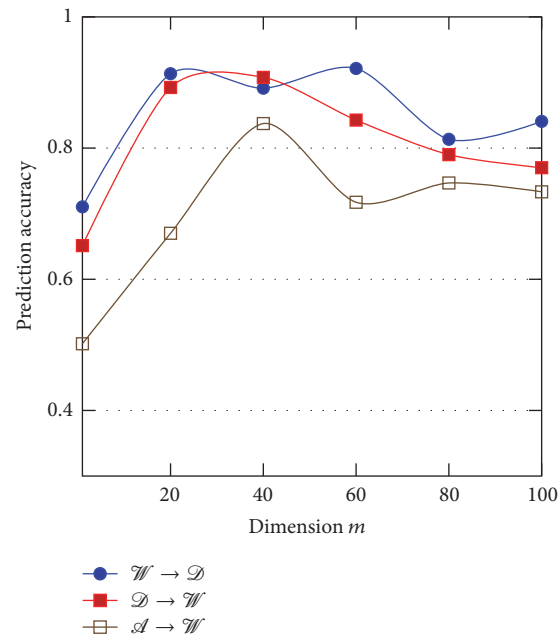

Figure 3: When the dimension is 20 or 40, the prediction accuracy is higher than others.

parameter to infinite, $\lambda \to \infty$, it may lose the data properties when we transform source and target domain data into high-dimension space. Contrarily, if we set λ to zero, the model would lose the function of correcting the distribution mismatch.

(2) Tradeoff μ. μ is a tradeoff to control the weight of data variance item which aims to preserve data properties. Theoretically, we want this item to be equal to zero. However, if we set this parameter to infinite, $\mu \to \infty$, it may augment the data distribution mismatch among different domains; namely, transformation matrix $\mathbf{M}$ cannot utilize source data to assist the target task. Contrarily, if we set μ to zero, the model cannot preserve the properties of original data.

(3) Dimension Reduction m. m is the dimension of the transformation matrix, namely, the dimension of the subspace which we want to map samples into. Similarly, minimizing m too less may lead to losing the properties of data which may lead to the classifier failure. If m is too large, the effectiveness of correct distribution mismatch may be lost. We conduct the classification results influenced by the dimension of m, and the results are displayed in Figure 3.

(4) Tradeoff in ESVM C_1 and C_2. Parameters C_1 and C_2 are the upper bound of the Lagrangian variables. In the standard SVM, positive and negative instances share the same standard of these two parameters. In our models, we expect the weights of the positive samples to be higher than negative samples. In our experiments, the value of C_1 is one hundred times C_2 which could gain a high-performance predictor. The visual analysis of these two parameters is in Figure 4.

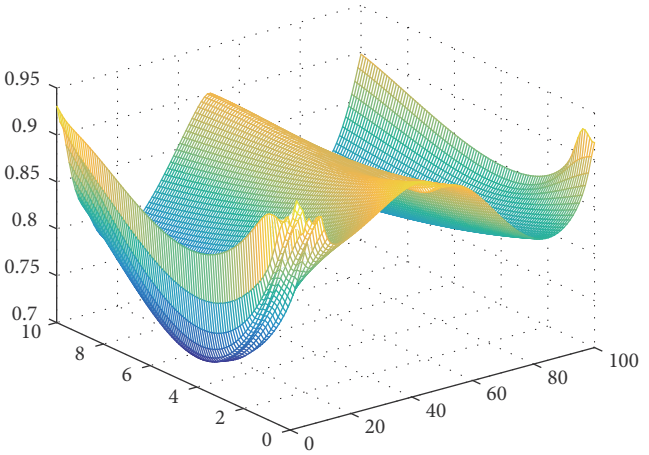

Figure 4: We fix $\lambda = 1$, $m = 20$, and $\mu = 1$ in these experiments, and C_1 is searched in $\{0.1, 0.5, 1, 5, 10, 50, 100\}$ and C_2 is searched in $\{0.001, 0.005, 0.01, 0.1, 0.5, 1, 10\}$.

7. Conclusion

In this paper, we have proposed an effective method for domain adaptation problems with regularization item which reduces the data distribution mismatch between domains and preserves properties of the original data. Furthermore, utilizing the method of integrating classifiers can predict target domain data with high accuracy. The proposed method mainly aims to solve the problem, in which domains or instances distributions mismatch occurs. Meanwhile, we extend DAESVMs to the multiple source or target domains. Experiments conducted on the transfer learning datasets transfer knowledge from image to image.

Our future works are as follows. First, we will integrate the training procession of all the classifiers in an ensemble way. It is better to accelerate training process by rewriting all the weight into a matrix form. This strategy can omit the process of matrix inversion optimization. Second, we want to make a constraint for $\boldsymbol{\alpha}$ that can hold the sparsity. At last, we will extend DAESVMs on the problem transfer knowledge among domains which have few relationships, such as transfer knowledge from image to video or text.

Notations and Descriptions

$\mathscr{D}_S, \mathscr{D}_T$: Source/target domain
$\mathscr{T}_S, \mathscr{T}_T$: Source/target task
d: Dimension of feature
$\mathbf{X}_S, \mathbf{X}_T$: Source/target sample matrix
$\mathbf{y}_S, \mathbf{y}_T$: Source/target sample label matrix
$\mathbf{K}$: Kernel matrix without label information
α: Lagrange multipliers vector
n_S, n_T: The number of source/target domain instances
$\mathbf{e}$: Identity vector
$\mathbf{I}$: Identity matrix.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been partially supported by grants from National Natural Science Foundation of China (nos. 61472390, 71731009, 91546201, and 11771038) and the Beijing Natural Science Foundation (no. 1162005).

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[2] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, ACM, July 2008.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.

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

[5] W.-S. Chu, F. D. L. Torre, and J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 3515–3522, USA, June 2013.

[6] A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," *In Advances in Neural Information Processing Systems 23*, pp. 478–486, 2010.

[7] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 54–66, 2015.

[8] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, "Adaptive localization in a dynamic WiFi environment through multi-view learning," in *Proceedings of the AAAI-07/IAAI-07 Proceedings: 22nd AAAI Conference on Artificial Intelligence and the 19th Innovative Applications of Artificial Intelligence Conference*, pp. 1108–1113, can, July 2007.

[9] A. Van Engelen, A. C. Van Dijk, M. T. B. Truijman et al., "Multi-Center MRI Carotid Plaque Component Segmentation Using Feature Normalization and Transfer Learning," *IEEE Transactions on Medical Imaging*, vol. 34, no. 6, pp. 1294–1305, 2015.

[10] Y. Zhang, J. Wu, Z. Cai, P. Zhang, and L. Chen, "Memetic Extreme Learning Machine," *Pattern Recognition*, vol. 58, pp. 135–148, 2016.

[11] M. Uzair and A. Mian, "Blind domain adaptation with augmented extreme learning machine features," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 651–660, 2017.

[12] L. Zhang and D. Zhang, "Domain Adaptation Extreme Learning Machines for Drift Compensation in E-Nose Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1790–1801, 2015.

[13] B. Scholkopf, J. Platt, and T. Hofmann, in *A kernel method for the two-sample-problem*, pp. 513–520, 2008.

[14] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance Learning with Discriminative Bag Mapping," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-1.

[15] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–16, 2017.

[16] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proceedings of the 2011 IEEE International Conference on Computer Vision, ICCV 2011*, pp. 89–96, Spain, November 2011.

[17] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the Twenty-first international conference*, p. 114, Banff, Alberta, Canada, July 2004.

[18] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain Generalization and Adaptation using Low Rank Exemplar SVMs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1.

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

[20] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems, NIPS 2006*, pp. 601–608, can, December 2006.

[21] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments*, The MIT Press, 2012.

[22] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on*

Machine Learning (ICML '07), pp. 193–200, New York, NY, USA, June 2007.

[23] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban, "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 56–63, USA, June 2015.

[24] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, "Distant domain transfer learning," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 2604–2610, usa, February 2017.

[25] W.-S. Chu, F. De La Torre, and J. F. Cohn, "Selective transfer machine for personalized facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 529–545, 2017.

[26] R. Aljundi, J. Lehaire, F. Prost-Boucle, O. Rouvière, and C. Lartizien, "Transfer learning for prostate cancer mapping based on multicentric MR imaging databases," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9487, pp. 74–82, 2015.

[27] M. Long, *Transfer learning: problems and methods [Ph.D. thesis]*, Tsinghua University, problems and methods. PhD thesis, 2014.

[28] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2066–2073, June 2012.

[29] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, pp. 120–128, Association for Computational Linguistics, July 2006.

[30] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," *and M. I. Jordan. Learning transferable features with deep adaptation networks. pages 97–105*, pp. 97–105, 2015.

[31] M. Long, J. Wang, and M. I. Jordan, *Deep transfer learning with joint adaptation networks*, 2016.

[32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 3320–3328, can, December 2014.

[33] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 188–197, September 2007.

[34] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaption," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1682–1695, 2017.

[35] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8691, no. 3, pp. 628–643, 2014.

[36] L. Niu, W. Li, D. Xu, and J. Cai, "An Exemplar-Based Multi-View Domain Generalization Framework for Visual Recognition," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[37] L. Niu, W. Li, and D. Xu, "Multi-view domain generalization for visual recognition," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4193–4201, Chile, December 2015.

[38] T. Kobayashi, "Three viewpoints toward exemplar SVM," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2765–2773, USA, June 2015.

[39] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *In Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 677–682, 2008.

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

[41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Computer Vision—ECCV 2010*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 213–226, Springer, Berlin, Germany, 2010.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[43] V. N. Vapnik, *Statistical Learning Theory*, Adaptive and Learning Systems for Signal Processing, Communications, and Control, Wiley- Interscience, New York, NY, USA, 1998.

[44] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 2960–2967, Australia, December 2013.

[45] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 1410–1417, USA, June 2014.

[46] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.