Subspace Distribution Adaptation Frameworks for Domain Adaptation

Sentao Chen, Le Han, Xiaolan Liu, Zongyao He, and Xiaowei Yang

***Abstract*— Domain adaptation tries to adapt a model trained from a source domain to a different but related target domain. Currently, prevailing methods for domain adaptation rely on either instance reweighting or feature transformation. Unfortunately, instance reweighting has difficulty in estimating the sample weights as the dimension increases, whereas feature transformation sometimes fails to make the transformed source and target distributions similar when the cross-domain discrepancy is large. In order to overcome the shortcomings of both methodologies, in this article, we model the unsupervised domain adaptation problem under the *generalized covariate shift* assumption and adapt the source distribution to the target distribution in a subspace by applying a distribution adaptation function. Accordingly, we propose two frameworks: Bregman-divergence-embedded structural risk minimization (BSRM) and joint structural risk minimization (JSRM). In the proposed frameworks, the subspace distribution adaptation function and the target prediction model are jointly learned. Under certain instantiations, convex optimization problems are derived from both frameworks. Experimental results on the synthetic and real-world text and image data sets show that the proposed methods outperform the state-of-the-art domain adaptation techniques with statistical significance.**

***Index Terms*— Convex optimization, covariate shift, feature transformation, risk minimization, unsupervised domain adaptation.**

I. Introduction

DOMAIN adaptation emerges as an extension to the traditional supervised learning to tackle the learning problem where the training data are sampled from a source distribution (source domain), and the test data are sampled from a different target distribution (target domain) [1]–[3]. In the field of domain adaptation, the training data and test data are usually called source data and target data, respectively. In real-world applications, the *distribution mismatch* between training and test data occurs in different fields. In computer vision, it is caused by different view angles, lighting conditions, or acquisition devices. In natural language processing, it is induced by different usage of terms and term frequencies for different subjects. Domain adaptation aims at correcting this *distribution mismatch* and further learning a target prediction model by utilizing both the source and target data. As evidenced by a stream of recent works, domain adaptation has made impressive progress in a wide range of applications, such as visual object recognition [1]–[7], face recognition [8], [9], and natural language processing [10]–[14].

Generally speaking, there are two kinds of domain adaptation settings: the unsupervised and the semisupervised. Unsupervised domain adaptation assumes that the labeled source and unlabeled target data are available for training the adaptation model, and semisupervised domain adaptation additionally assumes that a small fraction of the labeled target data is also available [9], [15], [16]. In this article, we concentrate on the former setting, which is usually regarded as a more challenging problem. In the literature of unsupervised domain adaptation, a stream of works reweight the source samples in the original feature space and train a target prediction model with these weighted samples [17]–[21]. This reweighting approach enjoys a nice theoretical property: the weighted model converges to the true target model if the covariate shift assumption holds [22]. The estimation of the weights is crucial to this instance reweighting approach. Unfortunately, the difficulty of weight estimation increases as the dimension increases [23]. To some extent, this constrains the applications of instance reweighting in text and image classification tasks whose samples originally lie in a high-dimensional feature space. Besides, in many real-world applications, the conditional distribution may change across domains due to noisy or dynamic factors underlying the observed data [24], which makes the covariate shift assumption invalid in the original feature space. Another stream of works applies feature transformation to make the source and target distributions similar [6], [12], [24], [25]. These methods are intuitive and interpretable since they explicitly minimize the empirical statistical distance between the source and target distributions to correct the *distribution mismatch*. Learning the feature transformation is the main concern of these methods. However, even after the feature transformation, the source

Manuscript received November 5, 2018; revised May 15, 2019, August 16, 2019, and December 18, 2019; accepted January 1, 2020. Date of publication January 24, 2020; date of current version December 1, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61273295, Grant 61502175, Grant 61503141, and Grant 61906069, in part by the Guangdong Natural Science Funds under Grant 2019A1515011411 and Grant 2019A1515011700, and in part by the China Postdoctoral Science Foundation under Grant 2019M662912. *(Corresponding authors: Sentao Chen; Xiaowei Yang.)*

Sentao Chen and Xiaowei Yang are with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: sentaochen@yahoo.com; xwyang@scut.edu.cn).

Le Han and Xiaolan Liu are with the School of Mathematics, South China University of Technology, Guangzhou 510640, China (e-mail: hanle@scut.edu.cn; liuxl@scut.edu.cn).

Zongyao He is with the School of Computer and Data Science, Henan University of Urban Construction, Pingdingshan 467036, China (e-mail: zyhe2@hncj.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2964790

distribution may still deviate from the target distribution when the cross-domain discrepancy is substantially large [7], [26].

In light of the above-mentioned discussion, the drawback of instance reweighting originates from the high dimensionality and the invalid covariate shift assumption in the original feature space, whereas the weakness of feature transformation stems from the potential dissimilarity between the transformed source and target distributions. This naturally guides us to model the unsupervised domain adaptation problem in a low-dimensional feature space where the similarity between the transformed source and target distributions is not assumed. We therefore introduce a *generalized covariate shift* assumption: after performing feature transformation, the subspace conditional distributions of the two domains are the same, whereas the source distribution is still different from the target distribution in the transformed feature space.

Under the *generalized covariate shift* assumption, we first search for a low-dimensional subspace where the covariate shift holds. Subsequently, we adapt the source distribution by a distribution adaptation function to make it similar to the target distribution. This consequently makes the source joint distribution similar to the target joint distribution in the subspace. Meanwhile, we minimize the generalization error of a hypothesis with respect to this adapted source joint distribution to learn a target prediction model. Ideally, the obtained subspace should be a space where both domains share the same conditional distribution. Inspired by the idea in [27], we choose the subspace, which is spanned by the principal components of the target data. To learn the subspace distribution adaptation function, we minimize the Bregman divergence between it and the density ratio function $P_t(z)/P_s(z)$, where $P_t(z)$ and $P_s(z)$ are the target distribution and the source distribution in the subspace, respectively. Using the empirical data, we correspondently propose the Bregman-divergence-embedded structural risk minimization (BSRM) framework, in which the subspace distribution adaptation function and the target prediction model are jointly learned. In particular, by choosing certain loss function, a convex optimization problem can be derived from the framework. Furthermore, by transforming the estimation of the distribution adaptation function into a binary classification problem, we propose the joint structural risk minimization (JSRM) framework. JSRM is very similar to BSRM except that the subspace distribution adaptation function in JSRM is learned via structural risk minimization. By carefully choosing the two-loss functions in JSRM, a convex optimization problem can also be derived. The major contributions of this article can be summarized as follows.

1) From the statistical perspective, we propose the BSRM framework, which jointly adapts the source distribution to the target distribution in the subspace and learns the target prediction model under the newly introduced *generalized covariate shift* assumption. The proposed framework contributes to generalizing the traditional structural risk minimization framework [28] to the unsupervised domain adaptation setting. Moreover, we instantiate the BSRM framework to a convex optimization problem by reasonably choosing the loss function.
2) We further introduce the JSRM framework to tackle the unsupervised domain adaptation problem via joint loss minimization. In this framework, distribution adaptation and prediction model training can both be achieved via minimizing the familiar loss functions. Importantly, a convex optimization problem can also be derived from this framework.
3) Comprehensive experiments are conducted on several real-world problems, including document classification, spam detection, digit recognition, face recognition, and sentiment analysis. The comparison results of linear and nonlinear models demonstrate that the proposed methods outperform the state-of-the-art unsupervised domain adaptation techniques with statistical significance.

The remainder of this article is organized as follows. Section II reviews the related works on unsupervised domain adaptation. Section III models the unsupervised domain adaptation problem under the *generalized covariate shift* assumption. Sections IV and V introduce the BSRM and JSRM frameworks, respectively. Section VI designs the learning algorithm for the distribution adaptation function and the target prediction model. Section VII discusses the relationships between BSRM and some existing unsupervised domain adaptation methods. Section VIII evaluates the performances of our proposed methods on both synthetic and real-world data sets. Section IX concludes this article and indicates possible directions for future research.

II. Related Work

In this part, we review previous unsupervised domain adaptation methods and theories. Specifically, we roughly divide the adaptation techniques into two groups: instance reweighting and feature transformation, and review them respectively.

Instance reweighting first emerged as an approach for correcting sample selection bias [29] and covariate shift [22], which can be regarded as special cases of the *distribution mismatch*. The weights, which are in fact the estimates of the target to source density ratio function $P_t(\boldsymbol{x})/P_s(\boldsymbol{x})$ evaluated at the source samples, are key to this technique. Kernel mean matching (KMM) [17] estimates the weights by minimizing the means between the weighted source data and target data in the reproducing kernel Hilbert space (RKHS). Since KMM lacks a decent model selection strategy, Miao *et al.* [30] proposed to use the normalized mean square error as a criterion for tuning the hyperparameters of KMM. As a more general approach, some works estimate the whole density ratio function by performing probabilistic classification [18], density matching [31], or density ratio fitting [32]. In addition, other works also focus on lowering the computational complexity of these existing techniques and scaling them to large scale problems [33], [34].

The essence of instance reweighting is to reweight the training loss at the source samples and learn a target prediction model by minimizing the reweighted loss. Several works [19], [21] achieve this via a two-step procedure: 1) estimate the weights for the source samples and 2) minimize the

reweighted training loss to learn the target model. However, this is suboptimal since the parameters of the target model, and the parameters which control the weights are not independent [18]. Based on this observation, Bickel *et al.* [18] applied the maximum a posterior parameter estimation method to derive an integrated optimization problem in which the weight function and the target model are jointly learned. Similarly, a selective transfer machine (STM) [20] learns the KMM weights and the target model in a single-optimization problem. This method was extended to the distribution matching machines (DMM) [7] by additionally embedding a feature transformation matrix into the optimization problem. Our proposed frameworks, BSRM and JSRM, also belong to this one-step approach. However, BSRM is more general than Bickel's integrated model, STM and DMM. Detailed analysis of the relationships among them will be given in Section VII.

Feature transformation is a much richer family than instance reweighting for domain adaptation. In the early works, feature transformation is performed by simply augmenting the original features [35], [36]. Later, a lot of works learn the feature mapping or mappings via explicitly minimizing different statistical divergences, such as maximum mean discrepancy (MMD) [6], [12], [24], Bregman divergence [37], Jensen–Shannon divergence [38], or Hellinger distance [25]. After transforming the features, a classifier can then be trained on the source data to label the target data. An outstanding representative among these works is the distribution-matching embedding [25], which first models the source and target distributions by kernel density estimation and then learns linear or nonlinear feature mapping by minimizing the Hellinger distance between the two distributions. Recently, some works exploit the neural network structure to concatenate distribution matching and classifier training in an end-to-end learning paradigm [10], [13], [39] and achieve remarkable performances in image classification tasks, such as digit recognition [40]–[42]. For example, Zellinger *et al.* [10] proposed to train a neural network based on a joint objective that simultaneously minimizes the source error and the central moment discrepancy (CMD) between the source and target activation distributions. However, most of the deep methods involve nonconvex objective functions, rely on large-size source domains, and are computationally expensive to train [43]. On the contrary, the convex instantiations derived from our frameworks can accelerate the optimization process and obtain a global optimal solution. Besides, some deep methods, such as the CMD method, implicitly assume that all the source instances are equally important for the target domain. This may not hold because when the cross-domain discrepancy is substantially large, there may still exist some source instances irrelevant to the target domain even using domain-invariant representations. And these instances may introduce large bias to the domain adaptive classifier. In our frameworks, such irrelevant source instances will be downweighted by the distribution adaptation function and be considered less important to the target domain.

Instead of focusing on learning the feature mapping that minimizes certain statistical divergence, several techniques exploit intermediate subspaces to link the source data to the target data. Sampling geodesic flow [44] models the subspaces as points on the Grassmann manifold and obtains the intermediate subspaces by sampling points along the geodesic between the source and target subspaces. Gong *et al.* [45] extended this method by considering all the intermediate subspaces and integrating them along the geodesic. Fernando *et al.* [27] proposed the subspace alignment (SA), which aligns the source principle component analysis (PCA) subspace to the target PCA subspace. Zhang *et al.* [46] proposed the guide subspace learning (GSL) method to learn an invariant subspace based on the subspace-guided, the data-guided and the label-guided terms. Besides these subspace-based approaches, some methods rely on covariance matching for domain adaptation. For instance, correlation alignment (CORAL) [47] aligns the source and target covariance matrices in the Euclidean space. Zhang *et al.* [14] extended CORAL to the RKHS and aligned two RKHS covariance matrices. There are also some works that explore feature reweighting [3] or feature selection [8] for domain adaptation. Our proposed methods relate closely to SA, but we do not require the source distribution to be very similar to the target distribution after feature transformation. In fact, we only assume that the transformed source distribution overlaps with the transformed target distribution so that the adaptation is possible.

Besides these domain adaptation techniques, theoretical error bounds for domain adaptation have also been studied. Ben-David *et al.* [48] proposed the $\mathcal{H}$*-divergence* and a target domain error bound based on the *VC-dimension*[28]. Building on this article, Mansour *et al.* [49] introduced the *discrepancy distance*, which further takes arbitrary loss functions into consideration and presented a target domain error bound based on the *Rademacher complexity* [50]. By considering labels in the source domain, Kuroki *et al.* [51] proposed the *source-guided discrepancy* which lower bounds the *discrepancy distance* and consequently derived a tighter error bound than the one in [49]. These theoretical results also motive the development of error bounds for some domain adaptation algorithms [7], [52].

III. Unsupervised Domain Adaptation Under Generalized Covariate Shift

In this section, we define the unsupervised domain adaptation problem, formally introduce the *generalized covariate shift assumption,* and model the adaptation problem under it.

A. Problem Definition

Let $\boldsymbol{x} \in \boldsymbol{\mathcal{X}} \in \mathbb{R}^d$ be an input variable and $y \in \mathcal{Y}$ be an output variable. $\mathcal{Y}$ is a real space, $\mathbb{R}$ for regression, or a set of discrete labels $\{1, 2, \ldots, C\}$ for classification. Let the source joint distribution be $P_s(\boldsymbol{x}, y)$, and the target joint distribution be $P_t(\boldsymbol{x}, y)$. Domain *distribution mismatch* means $P_s(\boldsymbol{x}, y) \neq P_t(\boldsymbol{x}, y)$. Let the source and target distributions be $P_s(\boldsymbol{x})$ and $P_t(\boldsymbol{x})$, respectively. In unsupervised domain adaptation, we are given a set of m_s source examples $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, y_i^s)|\boldsymbol{x}_i^s \sim P_s(\boldsymbol{x}), y_i^s \sim P_s(y|\boldsymbol{x})\}_{i=1}^{m_s}$ and a set of m_t unlabeled target instances $\mathcal{D}_t = \{\boldsymbol{x}_j^t|\boldsymbol{x}_j^t \sim P_t(\boldsymbol{x})\}_{j=1}^{m_t}$. Let h be a hypothesis from a hypothesis space $\mathcal{H}$ and $\ell : \mathcal{Y} \times \mathbb{R} \to [0, +\infty)$ a loss function. Given $\mathcal{D}_s$ and $\mathcal{D}_t$, unsupervised domain adaptation

consists of selecting a hypothesis $h \in \mathcal{H}$ to achieve the minimum expected target error $E(h) = \int \ell(h(\boldsymbol{x}), y) P_t(\boldsymbol{x}, y) d\boldsymbol{x} dy$.

B. Generalized Covariate Shift

The traditional covariate shift assumes that $P_s(\boldsymbol{x}) \neq P_t(\boldsymbol{x})$ and $P_s(y|\boldsymbol{x}) = P_t(y|\boldsymbol{x})$. However, in many real-world applications, such as image and text classification tasks, the conditional distribution may change across domains due to noisy or dynamic factors underlying the observed data [24]. Moreover, comparing distributions directly in the original feature space may not be well-suited, since the features may have been distorted by the *domain shift* and that some of the features may only be relevant to one specific domain [25]. Based on these considerations, we introduce the following *generalized covariate shift* assumption.

Assumption 1 *(Generalized Covariate Shift):* Let $F : \mathbb{R}^d \to \mathbb{R}^{d_1}$ be a feature mapping. Under this mapping, $P_s(y|F(\boldsymbol{x})) = P_t(y|F(\boldsymbol{x}))$ but $P_s(F(\boldsymbol{x})) \neq P_t(F(\boldsymbol{x}))$.

In this assumption, $P_s(F(\boldsymbol{x})) = P_t(F(\boldsymbol{x}))$ is not required. Because when two domains have a large discrepancy, $P_s(F(\boldsymbol{x})) = P_t(F(\boldsymbol{x}))$ may not hold [7], [26].

Inspired by the idea in [27], we decide the feature mapping F as a dimension reduction matrix $\boldsymbol{W}(\boldsymbol{W} \in \mathbb{R}^{d \times d_1})$, whose columns are the principal components of the target data. This F denoises the noisy information specific to domains but irrelevant to pattern classification. Due to the intrinsic similarity between the two domains, we can expect $P_s(y|F(\boldsymbol{x})) = P_t(y|F(\boldsymbol{x}))$ to hold. In fact, the invariance of the subspace conditional distribution between domains has been assumed in previous domain adaptation works [7], [24]. The assumption is feasible in real-world applications. For instance, in sentiment analysis, a book might be described as tedious, whereas a DVD might be described as boring. Both words have a negative connotation and are very likely to appear together with other negative words, such as bad, horrible, or poor. By projecting the reviews into a low-dimensional latent feature space, these words will probably indicate the common sentiment shared by both domains, which implies $P_s(y|F(\boldsymbol{x})) = P_t(y|F(\boldsymbol{x}))$. In the following, we will model the domain adaptation problem in the subspace $\mathcal{Z} = \{\boldsymbol{z} = \boldsymbol{W}^T \boldsymbol{x} | \boldsymbol{W}^{d \times d_1} \in \mathbb{R}^{d \times d_1}, \boldsymbol{x} \in \boldsymbol{\mathcal{X}}\} \subseteq \mathbb{R}^{d_1}$.

C. Modeling Under Generalized Covariate Shift

Considering that $P_s(z) \neq P_t(z)$ and $P_s(y|z) = P_t(y|z)$, we present the following optimization problem based on target risk minimization:

$$\begin{aligned} h^* &= \underset{h \in \mathcal{H}}{\operatorname{argmin}} \int_{(z,y)} \ell(h(z), y) P_t(z, y) dz dy \\ &= \underset{h \in \mathcal{H}}{\operatorname{argmin}} \int_{(z,y)} \ell(h(z), y) \frac{P_t(z)}{P_s(z)} P_s(z) P_t(y|z) dz dy \\ &= \underset{h \in \mathcal{H}}{\operatorname{argmin}} \int_{(z,y)} \ell(h(z), y) \frac{P_t(z)}{P_s(z)} P_s(z, y) dz dy. \end{aligned} \tag{1}$$

For the unknown density ratio function $P_t(z)/P_s(z)$, we approximate it by a nonnegative distribution adaptation function $r(z)$ and use the Bregman divergence to measure the approximation error. $r(z)$ plays the role of changing the source distribution from $P_s(z)$ to $r(z)P_s(z)$ to mimic the target distribution $P_t(z)$. Hence, a single-step optimization problem can be given to learning the distribution adaptation function and the target prediction model simultaneously

$$\begin{aligned} (r^*, h^*) = \underset{r,h}{\operatorname{argmin}} & \int_{(z,y)} \ell(h(z), y) r(z) P_s(z, y) dz dy \\ & + \int_z B_g \left(\frac{P_t(z)}{P_s(z)} || r(z) \right) P_s(z) dz \end{aligned} \tag{2}$$

where $B_g(x_1||x_2)$ is the Bregman divergence defined as $B_g(x_1||x_2) = g(x_1) - [g(x_2) + g'(x_2)(x_1 - x_2)]$. g is the generator function, which must be differential and strictly convex. The derivative of g is denoted as g'. Obviously, the Bregman divergence measures the discrepancy between x_1 and x_2 by calculating the difference between $g(x_1)$ and the first-order Taylor expansion of g around x_2 evaluated at x_1. The following theorem justifies the validity of formulation (2) for domain adaption.

Theorem 1: Assume that the loss function ℓ is bounded, $\ell(\cdot, \cdot) \leq M$ for some $M > 0$, and $r(z)P_s(z)$ is a probability distribution. Let the generator function be $g(x) = x \log x$. Then, for any hypothesis $h \in \mathcal{H}$,

$$\begin{aligned} E_{P_t(z,y)}[\ell(h(z), y)] \leq & E_{r(z)P_s(z,y)}[\ell(h(z), y)] \\ & + \sqrt{2} M \sqrt{E_{P_s(z)} \left[B_g \left(\frac{P_t(z)}{P_s(z)} || r(z) \right) \right]}. \end{aligned} \tag{3}$$

Furthermore, if

$$E_{P_s(z)} \left[B_g \left(\frac{P_t(z)}{P_s(z)} || r(z) \right) \right] \geq 1$$

then

$$\begin{aligned} E_{P_t(z,y)}[\ell(h(z), y)] \leq & E_{r(z)P_s(z,y)}[\ell(h(z), y)] \\ & + M \sqrt{2} E_{P_s(z)} \left[B_g \left(\frac{P_t(z)}{P_s(z)} || r(z) \right) \right]. \end{aligned} \tag{4}$$

Proof: Please see Appendix A.

From Theorem 1, we can conclude that under certain circumstances, optimization problem (2) can generate a target model with low generalization error. Meanwhile, Theorem 1 also gives us a general guideline for choosing the generator function of the Bregman divergence.

IV. Bregman-Divergence-Embedded Structural Risk Minimization

By the definition of Bregman divergence, optimization problem (2) can be rewritten as the following form:

$$\begin{aligned} (r^*, h^*) = \underset{r,h}{\operatorname{argmin}} & \int_{(z,y)} \ell(h(z), y) r(z) P_s(z, y) dz dy \\ & + \int_z [g'(r(z)) r(z) - g(r(z))] P_s(z) dz \\ & - \int_z g'(r(z)) P_t(z) dz. \end{aligned} \tag{5}$$

The three terms in the right-hand side of (5) are expectations. Following the law of large numbers, given two sets of samples

$\mathcal{D}_s$ and $\mathcal{D}_t$, we propose the BSRM framework as the empirical estimate of the right-hand side of (5)

$$\min_{r,h} \frac{1}{m_s}\sum_{i=1}^{m_s} r(z_i^s)\ell\big(h(z_i^s), y_i^s\big) + \frac{1}{m_s}\sum_{i=1}^{m_s}\big[g'\big(r(z_i^s)\big)r(z_i^s) - g\big(r(z_i^s)\big)\big] - \frac{1}{m_t}\sum_{i=1}^{m_t} g'\big(r(z_i^t)\big) + \lambda_1\Omega(r) + \lambda_2\Omega(h). \tag{6}$$

Note that we add two regularization terms $\lambda_1\Omega(r)$ and $\lambda_2\Omega(h)$ in the empirical estimate to prevent overfitting. $\lambda_1, \lambda_2 > 0$ are regularization parameters and Ω is a regularizer. We term optimization problem (6) as a framework since the loss function ℓ and the generator function g are flexible, and there can be multiple choices for them. In this framework, the subspace distribution adaptation function r and the target model h are jointly learned.

We now instantiate the framework (6) to make it a convex optimization problem. First, we set the nonnegative distribution adaptation function to $e^{u(z)}$, where $u \in \mathcal{H}$ is chosen from the same hypothesis space as the target prediction model h. In this article, we choose two hypothesis spaces for learning linear and nonlinear models. One is the linear model space $\mathcal{H} = \{h(z;\boldsymbol{\theta}) = \boldsymbol{\theta}^T z | \boldsymbol{\theta} \in \mathbb{R}^{d_1+1}\}$, where $z = (1, z_1, \ldots, z_{d_1})^T$. The other one is the radial basis function network space [53] $\mathcal{H} = \{h(z;\boldsymbol{\theta}) = \boldsymbol{\theta}^T\phi(z) | \phi(z) = (k(z,\boldsymbol{c}_1), \ldots, k(z,\boldsymbol{c}_b))^T, \boldsymbol{\theta} \in \mathbb{R}^b\}$, where $k(z,\boldsymbol{c}_j) = \exp(-(||z-\boldsymbol{c}_j||^2)/(2\sigma^2))$ is the jth radial basis function centered at a vector $\boldsymbol{c}_j = z_j^t$ $(1 \le j \le m_t)$, and $\sigma > 0$ is the bandwidth. Obviously, in these two hypothesis spaces, the functions are all linear with respect to the parameters. Therefore, we focus on analyzing the linear model space since it can be easily generalized to the nonlinear model by simply changing the features. Let $r(z) = e^{\boldsymbol{\alpha}^T z}$, $h(z) = \boldsymbol{\theta}^T z$, and $g(x) = x\log x$ as suggested by Theorem 1, and Ω be the L_2 regularizer. Then, the optimization problem (6) becomes

$$\min_{\boldsymbol{\alpha},\boldsymbol{\theta}} \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{\alpha}^T z_i^s}\ell(\boldsymbol{\theta}^T z_i^s, y_i^s) + \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{\alpha}^T z_i^s} - \frac{1}{m_t}\sum_{i=1}^{m_t}\boldsymbol{\alpha}^T z_i^t + \lambda_1\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \lambda_2\boldsymbol{\theta}^T\boldsymbol{\theta}. \tag{7}$$

Generally speaking, for classification problems, the loss functions, such as the hinge loss, the logistic loss, the exponential loss, and the square loss, can be uniformly expressed as $\ell(\boldsymbol{\theta}^T z_i^s, y_i^s) = \varphi(y_i^s\boldsymbol{\theta}^T z_i^s)$. Using this notation, optimization problem (7) can be rewritten as the following compact form:

$$\min_{\boldsymbol{w}} \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T\boldsymbol{v}_i^s}\varphi\big(y_i^s\boldsymbol{w}^T\boldsymbol{v}_i^s\big) + \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T\boldsymbol{v}_i^s} - \frac{1}{m_t}\sum_{i=1}^{m_t}\boldsymbol{w}^T\boldsymbol{v}_i^t + \boldsymbol{w}^T\boldsymbol{\Lambda}\boldsymbol{w} \tag{8}$$

where

$$\boldsymbol{w} = \begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\theta}\end{pmatrix}, \quad \boldsymbol{v}^s = \begin{pmatrix}z^s\\ \boldsymbol{0}\end{pmatrix}, \quad \boldsymbol{v}^t = \begin{pmatrix}\boldsymbol{0}\\ z^t\end{pmatrix}, \quad \text{and} \quad \boldsymbol{\Lambda} = \begin{pmatrix}\lambda_1\boldsymbol{I}\\ \lambda_2\boldsymbol{I}\end{pmatrix}.$$

$\boldsymbol{0}$ is a d_1+1 dimensional column vector and $\boldsymbol{I}$ is an identity matrix of size $(d_1+1)\times(d_1+1)$. The following Theorem 2 specifies the relationship between the convexity of (8) and the choice of the loss function φ.

Theorem 2: Optimization problem (8) is convex if $\log\varphi$ is convex. In particular, when φ is the exponential loss, (8) is convex.

Proof: Please see Appendix B.

According to Theorem 2, we can easily check that (8) may not be convex with the other loss functions. Hence, we use the exponential loss and derive the following convex instantiation of the BSRM framework:

$$\min_{\boldsymbol{w}} \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T(\boldsymbol{v}_i^s - y_i^s\boldsymbol{v}_i^s)} + \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T\boldsymbol{v}_i^s} - \frac{1}{m_t}\boldsymbol{w}^T\sum_{i=1}^{m_t}\boldsymbol{v}_i^t + \boldsymbol{w}^T\boldsymbol{\Lambda}\boldsymbol{w}. \tag{9}$$

We call it as Convex BSRM (CBSRM).

V. Joint Structural Risk Minimization

In optimization problem (2), we use a distribution adaptation function to adapt the source distribution to the target distribution in the subspace and learn it via minimizing the Bregman divergence. In this section, based on the result of [54], we transform the estimation of the subspace distribution adaptation function into a binary classification problem.

Let $P(z,l)$ be a probability distribution over $\mathcal{Z}\times\{-1,+1\}$ and $\eta(z) = P(l=+1 \mid z)$. From [54], we know that the loss function ℓ is strictly proper composite with (invertible) link function $\psi : [0,1] \to \mathbb{R}$ if $\psi\eta = f^* = \operatorname{argmin}_{f\in\mathcal{H}} E_{P(z,l)}[\ell(f(z),l)]$.

By the following Theorem 3, we can transform the estimation of the distribution adaptation function r into the estimation of a binary classification model f.

Theorem 3: Let $P(z,l)$ be a probability distribution over $\mathcal{Z}\times\{-1,+1\}$, $P(z|l=-1) = P_s(z)$, $P(z|l=+1) = P_t(z)$, and $\pi = P(l=+1)$. Let ℓ_1 be a strictly proper composite loss function with invertible link function ψ, whose inverse is denoted as ψ^{-1}. Then optimization problem (2) can be transformed into the following problem:

$$(f^*, h^*) = \operatorname*{argmin}_{f,h\in\mathcal{H}} \int_{(z,y)} P_s(z,y)\ell(h(z),y)\frac{1-\pi}{\pi}\frac{(\psi^{-1}f)(z)}{1-(\psi^{-1}f)(z)}dzdy + 2\int_{(z,l)} P(z,l)\ell_1(f(z),l)dzdl. \tag{10}$$

Proof: Please see Appendix C.

Similar to the derivation of the BSRM framework, we use empirical means to approximate the expectations in (10) and propose the JSRM framework as follows:

$$\min_{f,h\in\mathcal{H}} \frac{1}{m_t}\sum_{i=1}^{m_s}\frac{(\psi^{-1}f)(z_i^s)}{1-(\psi^{-1}f)(z_i^s)}\ell\big(h(z_i^s), y_i^s\big) + \frac{2}{m_s+m_t}\sum_{i=1}^{m_s+m_t}\ell_1(f(z_i),l_i) + \lambda_1\Omega(f) + \lambda_2\Omega(h) \tag{11}$$

where $\{z_i\}_{i=1}^{m_s+m_t} = \{z_i^s\}_{i=1}^{m_s} \cup \{z_i^t\}_{i=1}^{m_t}$. $l_i = -1$ when $z_i \in \{z_i^s\}_{i=1}^{m_s}$, and $l_i = +1$ when $z_i \in \{z_i^t\}_{i=1}^{m_t}$. Note that we replace π with its maximum likelihood estimate $m_t/(m_s+m_t)$. In this framework, we also have multiple choices for the two-loss functions ℓ_1 and ℓ.

Let $f(z) = \boldsymbol{\alpha}^T z$, $h(z) = \boldsymbol{\theta}^T z$, ℓ_1, and ℓ be the exponential loss, and Ω be the L_2 regularizer, then the JSRM framework can be instantiated to the following convex optimization problem:

$$\min_{\boldsymbol{w}} \frac{1}{m_t}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T(2\boldsymbol{v}_i^s - y_i^s \boldsymbol{v}_i^s)} + \frac{2}{m_s+m_t}\left(\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T \boldsymbol{v}_i^s} + \sum_{i=1}^{m_t} e^{-\boldsymbol{w}^T \boldsymbol{v}_i^t}\right) + \boldsymbol{w}^T \boldsymbol{\Lambda} \boldsymbol{w}. \quad (12)$$

We call it as Convex JSRM (CJSRM).

VI. Learning Algorithm

In this section, we design algorithms for solving the CBSRM and CJSRM models. For convenience, we denote $J_1(\boldsymbol{w})$ and $J_2(\boldsymbol{w})$, respectively, as the objective functions in (9) and (12). Their gradients are

$$\nabla_{\boldsymbol{w}} J_1(\boldsymbol{w}) = \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T(\boldsymbol{v}_i^s - y_i^s \boldsymbol{v}_i^s)}(\boldsymbol{v}_i^s - y_i^s \boldsymbol{v}_i^s) + \frac{1}{m_s}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T \boldsymbol{v}_i^s}\boldsymbol{v}_i^s - \frac{1}{m_t}\sum_{i=1}^{m_t} \boldsymbol{v}_i^t + 2\Lambda \boldsymbol{w} \quad (13)$$

$$\nabla_{\boldsymbol{w}} J_2(\boldsymbol{w}) = \frac{1}{m_t}\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T(2\boldsymbol{v}_i^s - y_i^s \boldsymbol{v}_i^s)}(2\boldsymbol{v}_i^s - y_i^s \boldsymbol{v}_i^s) + \frac{2}{m_s+m_t}\left(\sum_{i=1}^{m_s} e^{\boldsymbol{w}^T \boldsymbol{v}_i^s}\boldsymbol{v}_i^s - \sum_{i=1}^{m_t} e^{-\boldsymbol{w}^T \boldsymbol{v}_i^t}\boldsymbol{v}_i^t\right) + 2\Lambda \boldsymbol{w}. \quad (14)$$

Based on these gradients, we can make use of the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [55] to solve optimization problems (9) and (12). Due to their convexity, the global optimal solution can be easily found. Algorithm 1 summarizes the learning algorithm for CBSRM. Replacing the third step in Algorithm 1 by solving the optimization problem (12) gives the algorithm for CJSRM.

In the following, we analyze the computational complexity of Algorithm 1. The computational cost for the first step is $O(m_t d^2 + d^3)$, the second step $O((m_s + m_t)dd_1)$, and the third step $O(md_1)$, where m is the number of correction pairs declared in the L-BFGS algorithm [55]. Hence, the total computational complexity of Algorithm 1 is $O(m_t d^2 + d^3 + (m_s + m_t)dd_1 + md_1)$.

VII. Relationship to Existing Methods

Here, we discuss the relationships between BSRM and the other methods, including CJSRM, Bickel's integrated model [18], STM [20], and DMM [7]. For relating BSRM to CJSRM and Bickel's integrated model, we assume that the source and target samples are of equal size. Let the generator function be $g(x) = -2\sqrt{x}$, the loss function be the exponential loss, and the distribution adaptation function be $e^{2\boldsymbol{\alpha}^T z}$, then BSRM becomes CJSRM. Let the subspace be the original feature space, and the generator function be $g(x) = x\log x - (x+1)\log(x+1)$, then BSRM becomes Bickel's integrated model. When the generator function is $g(x) = x\log x$, BSRM adapts the source distribution to the target distribution in the subspace to minimize their KL divergence. STM and DMM reweight the source samples to minimize the MMD distance. Hence, BSRM, STM, and DMM share a common characteristic: they jointly minimize certain distribution divergence and learn the target prediction model in a single-optimization problem. For the convexity of the corresponding optimization problems, BSRM includes convex instantiation, such as CBSRM, STM is biconvex [56], whereas DMM is nonconvex.

Algorithm 1 Learning Algorithm for CBSRM

Input: Source data $\mathcal{D}_s$ and target data $\mathcal{D}_t$; subspace dimensionality d_1, regularization parameters λ_1, λ_2.
Output: Subspace distribution adaptation function $r(z)$, target prediction model $h(z)$.
1: $\boldsymbol{W} = PCA(\mathcal{D}_t, d_1)$;
2: Map the original data to the subspace to get the embeddings: $z_i^s = \boldsymbol{W}^T \boldsymbol{x}_i^s$, $z_j^t = \boldsymbol{W}^T \boldsymbol{x}_j^t$,$(i = 1, \ldots, m_s, j = 1, \ldots, m_t)$;
3: Solve optimization problem (9) via L-BFGS.

VIII. Experiments

In this part, we first visualize and investigate the behavior of CBSRM on a synthetic classification problem. Subsequently, we comprehensively evaluate CBSRM and CJSRM on a series of real-world text and image classification data sets and compare them with the state-of-the-art methods. Specifically, we start by presenting the data sets and the experimental protocol, then provide the experimental results and discussions, and finish by performing a statistical test on the obtained results. Eventually, we experimentally analyze the characteristic of CBSRM in detail. Note that the evaluations are performed in a transductive learning setting, i.e., we measure the performance of the classifier on the already given, but unlabeled target data.

A. Synthetic Data

The experiment on synthetic data is designed to serve for two purposes: 1) the feature transformation approach could fail to make the source and target distributions similar and 2) CBSRM can learn a robust target model even when the source and target distributions are not aligned after feature transformation. For a clear demonstration, we consider adapting a binary classification model from one domain to the other. The joint probability distribution of Domain 1 (D1) is $P_1(\boldsymbol{x}, y) = P_1(\boldsymbol{x})P_1(y|\boldsymbol{x})$, where

$$P_1(\boldsymbol{x}) = \frac{1}{2}N\left(\boldsymbol{x}; \begin{pmatrix}3\\-2\\1\end{pmatrix}, \begin{pmatrix}2 & & \\ & 1 & \\ & & 0.1\end{pmatrix}\right) + \frac{1}{2}N\left(\boldsymbol{x}; \begin{pmatrix}3\\2\\1\end{pmatrix}, \begin{pmatrix}2 & & \\ & 1 & \\ & & 0.1\end{pmatrix}\right) \quad (15)$$

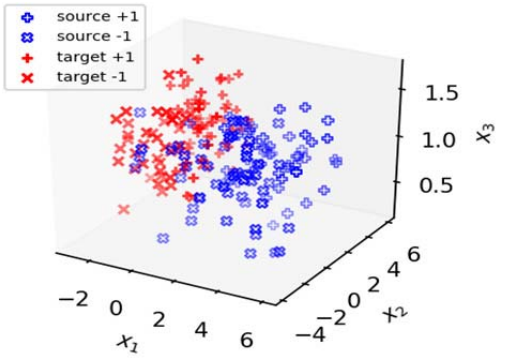


Fig. 1. Sample data from the source domain D1 and the target domain D2.

$$P_1(y=+1|\boldsymbol{x}) = \frac{1+\tanh(\min(0,x_1)+x_2+0.01x_3)}{2} \quad (16)$$

$\boldsymbol{x}=(x_1,x_2,x_3)^T$ and $N(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ is the multivariate Gaussian density with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $P_1(y=-1|\boldsymbol{x})=1-P_1(y=+1|\boldsymbol{x})$. The joint probability distribution of Domain 2 (D2) is $P_2(\boldsymbol{x},y)=P_2(\boldsymbol{x})P_2(y|\boldsymbol{x})$, where

$$P_2(\boldsymbol{x}) = \frac{1}{2}N\left(\boldsymbol{x};\begin{pmatrix}-1\\0\\1\end{pmatrix},\begin{pmatrix}1&&\\&1&\\&&0.1\end{pmatrix}\right) + \frac{1}{2}N\left(\boldsymbol{x};\begin{pmatrix}-1\\4\\1\end{pmatrix},\begin{pmatrix}1&&\\&1&\\&&0.1\end{pmatrix}\right) \quad (17)$$

$$P_2(y=+1|\boldsymbol{x}) = \frac{1+\tanh(\min(0,x_1)+x_2-0.01x_3)}{2}. \quad (18)$$

$P_2(y=-1|\boldsymbol{x})=1-P_2(y=+1|\boldsymbol{x})$. In this setup, the marginal distributions are different, whereas the conditional distributions are similar. We treat D1 and D2 as the source domain and the target domain, respectively, and show some samples drawn from D1 and D2 in Fig. 1.

Let the number of the source and target samples be $m_s=m_t=200$. We compare CBSRM with three typical feature transformation methods, which explicitly or implicitly assume that the source and target distributions are similar after transformation. These methods are listed as follows.

1) Transfer component analysis (TCA) [24] is a classic feature-based method that minimizes the MMD distance and maximizes the data variance to learn the new feature representation.
2) SA [27] is a simple approach that aligns the source PCA subspace to the target PCA subspace.
3) Joint Geometrical and Statistical Alignment (JGSA) [6] learns two projections to reduce the geometrical shift and *distribution shift* simultaneously.

We fix the dimensionality of the subspace at 2 for all the methods and use the hyperparameters recommended by the original articles for the comparison methods. To learn the classifiers for TCA, SA, and JGSA, we use the exponential loss and minimize the structural risk (SRM) on the transformed source data. Linear model space is chosen for all the methods.

TABLE I

AVERAGE ACCURACY (%) WITH STANDARD DEVIATION (IN PARENTHESES) OF 30 TRIALS ON THE SYNTHETIC DATA

Task	TCA	SA	JGSA	CBSRM
D1 → D2	81.4(2.7)	82.5(1.9)	85.2(5.2)	**89.4**(1.1)
D2 → D1	67.7(3.4)	74.7(2.8)	88.4(4.3)	**92.1**(1.0)

Maximum value in each task is highlighted in bold.

Fig. 2 shows the domain adaptation results of different methods on the data set. Unfortunately, TCA, SA, and JGSA fail to make the source and target distributions similar by their feature transformations. As a consequence, the models learned from the transformed source data do not separate the transformed target data well. Although the source distribution is also different from the target distribution in the CBSRM subspace, by further applying the subspace distribution adaptation function, CBSRM learns a robust target model that achieves the highest classification accuracy 90%.

Next, we compare the performances of the four methods in different settings of the domains. Let the number of the source and target samples be $m_s=m_t=1000$. We denote a domain adaptation task by $S\rightarrow T$, where S is the source domain and T is the target domain. This notation will be used for the rest of this article. We construct two adaptation tasks and report the average classification accuracy and standard deviation of the four methods over 30 trials in Table I. As can be observed from Table I, CBSRM performs the best in both tasks, and JGSA achieves the second highest mean classification accuracy. However, the large standard deviation shows that JGSA is not very stable.

B. Real-World Data Sets

Five types of classification problems are considered here: document classification, spam detection, digit recognition, face recognition, and sentiment classification. Correspondently, seven benchmark text and image data sets *20-Newsgroups*, *Reuters-21578*, *SPAM*, *IMDb reviews*, *Digit*, *CMU face*, and *Amazon reviews* are adopted in the experiments.

20-Newsgroups[1] is a collection of 18 774 newsgroup documents organized in a hierarchical structure of 6 top categories and 20 subcategories. This is a popular text data set in domain adaptation [12], [14], [57], [58]. We use the top four categories: *Com*, *Rec*, *Sci*, and *Talk* to generate cross-domain adaptation tasks. Specifically, for every pair of the top categories, we treat one category as a positive class and the other category as a negative class. Then, we construct the source domain by selecting the largest subcategories, respectively, from the two top categories. Similarly, the target domain is constructed by selecting the second-largest subcategories respectively from the two top categories. In this manner, we can construct six domain adaptation tasks: *Com* versus *Rec*, *Com* versus *Sci*, *Com* versus *Talk*, *Rec* versus *Sci*, *Rec* versus *Talk*, and *Sci* versus *Talk*. By switching the domains, we obtain another six adaptation tasks denoted as *Rec* versus *Com*, *Sci* versus *Com*, *Talk* versus *Com*, *Sci* versus *Rec*, *Talk* versus *Rec*, and *Talk* versus *Sci*. The task is to predict the top category to which

[1] http://qwone.com/~jason/20Newsgroups/

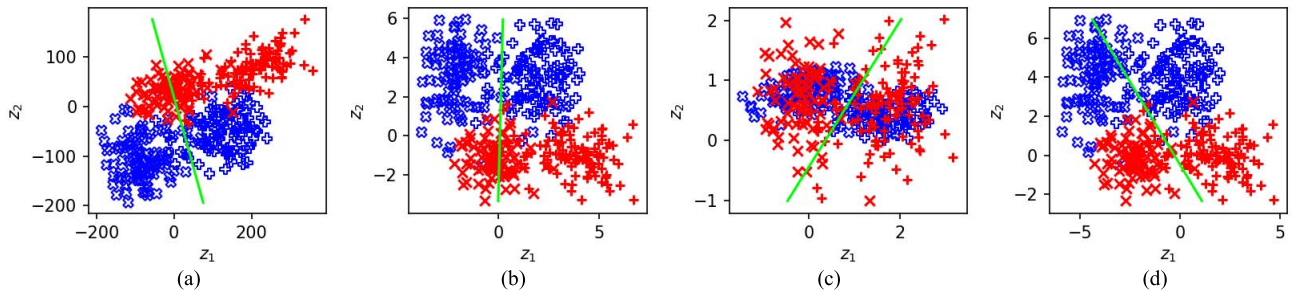


Fig. 2. Classification accuracy of TCA, SA, and JGSA, and the proposed CBSRM on synthetic data. The straight line is the learned target prediction model, and numbers in the brackets indicate the target domain classification accuracy. (a) TCA (80%). (b) SA (84%). (c) JGSA (87%). (d) CBSRM (90%).

the sample belongs. We extract the features in a similar way to [58], which is implemented by the scikit-learn[2] package.

Reuters-21578[3] also has a hierarchical structure of five top categories and many subcategories. We select the three top categories, *Orgs*, *People*, and *Places*, for our experiments and follow the strategy in [58] to preprocess the data. We create six domain adaptation tasks: *Orgs* versus *Places*, *Orgs* versus *People*, *Places* versus *Orgs*,*Places* versus *People*, *People* versus *Orgs*, and *People* versus *Places*.

SPAM [3] is a spam detection data set concatenated by two data sets from the University of California at Irvine (UCI) machine learning repository: one containing 4205 emails from the Enron spam database and the other one containing 5338 text messages from the short message service (SMS)-spam data set. The domain difference is due to the fact that text messages are often written in shortened words, whereas email messages tend to be more formal. We consider two adaptation tasks here: Email→Text and Text→Email. Following the processing steps in [3], a 4272-D feature vector is extracted by using bag-of-words representation.

Internet Movie Database (IMDb)*reviews* [59] contain movie reviews labeled with 1–10 stars from the IMDb. Following the protocol in [3], we label the reviews as positive if the values of their ratings are higher than five and negative for the remaining ones. From the original bag-of-words representation, we select only the features with more than 100 nonzero values in the entire data set, resulting in a 4180-D feature vector. We then split the data set by genre and obtain three domains: Action, Family, and War, from which six adaptation tasks are constructed to predict the viewer sentiment.

Digit is composed of three handwritten digit data sets: MNIST,[4] USPS,[5] and Optdigits.[6] These data sets contain images for digits from 0 to 9. MNIST consists of 70 000 images of size 28 × 28, USPS contains 9298 images of size 16 × 16, and Optdigits includes 5620 images of size 8 × 8. Fig. 3 shows some sample images from these three data sets. Apparently, data distributions of these data sets are quite different from each other, which makes domain adaptation necessary. To speed up experiments, we respectively choose 1000 images from MNIST, USPS, and Optdigits. The classes are balanced in the subsets. Besides, following [3], all the images are resized to 16 × 16 to create a common feature space. Finally, three-domain adaptation tasks are constructed here: MNIST→USPS, USPS→Optdigits, and Optdigits→ MNIST.

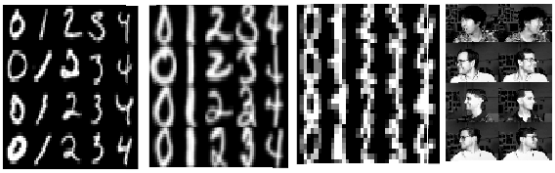


Fig. 3. Image samples from MNIST, USPS, Optdigits, and CMU face.

CMU face[7] consists of 640 black and white face images of people taken with the varying pose (straight, left, right, and up), expression (neutral, happy, sad, and angry), and eye status (open and sunglasses). There are 20 persons, and each person has 32 images capturing every combination of features. The following protocol is used to construct the domain adaptation tasks. We choose the left pose and right pose face images from all the 20 subjects and regard the two pose directions as two different domains. Thus, each domain has 20 classes and 160 images. The sample images are shown in Fig. 3. We use the half-resolution images (64 × 60), and apply the histogram of oriented gradients (HOGs) [60] technique implemented by scikit to extract the features. This results in a 1512-D feature vector. Two domain adaptation tasks are constructed here: Left→Right and Right→Left. In each task, the goal is to predict the identity of the person from one pose direction by using labeled samples from another pose direction.

Amazon reviews[35] contain online reviews of different products collected on the Amazon website. This data set includes four domains; each one consists of reviews from a special kind of product (books, DVD, electronics, and kitchen). Following the protocol in [13], the reviews are encoded as a 5000-D feature vector. If the product is ranked higher than three stars, the corresponding labels are positive; otherwise, the labels are negative. We construct 12 domain adaptation tasks. In each task, the goal is to predict the sentiment of

[2]http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[3]http://www.daviddlewis.com/resources/testcollections/reuters21578/

[4]http://yann.lecun.com/exdb/mnist/

[5]https://cs.nyu.edu/roweis/data.html

[6]http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

[7]https://archive.ics.uci.edu/ml/datasets/CMU+Face+Image

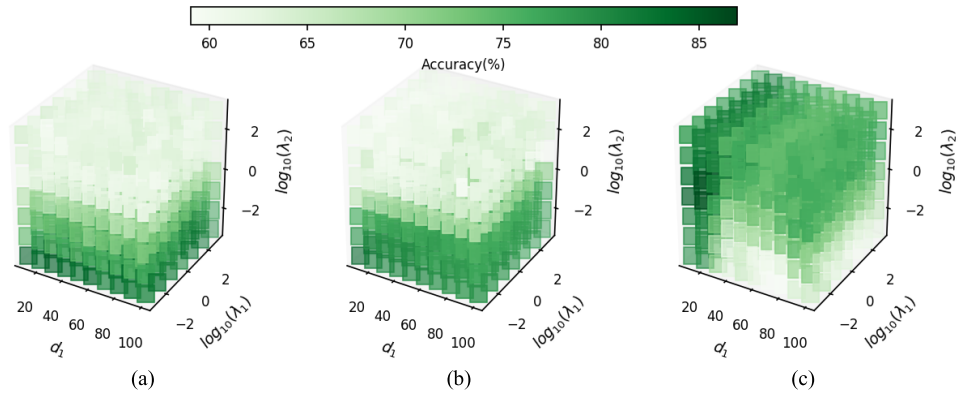


Fig. 4. Hyperparameter sensitivity of CBSRM on three domain adaptation tasks (best viewed in color). (a)–(c) Heat maps of the effect of hyperparameters on the tasks *Rec* versus *Talk*, *Com* versus *Sci*, and *Orgs* versus *Places*, respectively.

TABLE II
CLASSIFICATION ACCURACY (%) OF LINEAR MODELS OVER 43 ADAPTATION TASKS

Task	SRM	IEM	TCA	SA	TJM	CORAL	STM	OT-GL	JGSA	MCTL	GSL	CBSRM	CJSRM
Com vs Rec	85.15	85.56	86.32	72.28	91.01	85.20	85.36	88.69	88.54	90.27	89.60	88.44	**91.11**
Com vs Sci	79.49	79.54	74.98	63.49	87.24	79.54	78.88	86.88	81.92	88.76	87.09	**89.01**	88.70
Com vs Talk	89.70	89.97	82.11	81.84	92.82	89.76	89.34	90.71	91.76	89.23	92.40	**93.71**	88.97
Rec vs Com	89.53	89.88	94.26	75.69	90.03	89.68	91.84	96.47	84.39	93.56	93.26	**96.57**	96.02
Rec vs Sci	67.33	67.48	67.78	56.73	79.04	67.43	67.38	90.55	88.84	89.75	92.36	**94.52**	94.32
Rec vs Talk	75.65	75.76	81.58	54.51	85.36	75.65	72.56	86.46	81.58	88.72	**90.24**	88.40	86.14
Sci vs Com	80.69	80.04	69.78	49.01	85.14	80.69	77.21	**92.21**	84.94	92.17	87.37	89.54	88.58
Sci vs Rec	61.02	60.92	45.61	88.82	72.65	61.07	60.57	74.92	78.90	83.48	81.67	87.26	**89.87**
Sci vs Talk	47.84	47.94	49.52	52.52	50.57	47.78	49.68	54.57	22.78	59.42	28.16	**60.36**	43.68
Talk vs Com	92.99	93.36	92.94	85.63	93.25	92.99	93.20	96.31	89.31	95.59	91.13	**96.59**	93.25
Talk vs Rec	79.06	78.54	75.39	47.18	**93.24**	79.11	79.62	90.40	87.46	93.04	91.90	87.46	92.88
Talk vs Sci	56.24	56.65	44.74	42.67	55.20	56.24	56.49	59.50	63.95	54.12	**78.51**	57.84	55.04
Orgs vs Places	71.33	70.94	74.40	60.69	70.08	70.18	67.30	74.97	61.07	75.36	56.38	**76.12**	75.55
Orgs vs People	68.87	68.46	69.61	67.21	**77.64**	68.12	70.69	70.61	59.68	71.11	54.64	70.61	70.44
Places vs Orgs	64.07	64.86	68.99	64.07	55.01	64.86	58.07	66.63	52.06	69.19	57.38	**69.30**	67.22
Places vs People	50.88	51.16	57.47	64.43	51.06	56.17	52.55	58.86	58.12	67.13	60.26	67.22	**68.24**
People vs Orgs	76.55	76.55	82.37	73.16	79.30	76.63	77.36	81.56	59.90	82.46	81.65	**83.29**	82.21
People vs Places	52.73	51.99	65.08	59.88	58.31	54.13	53.76	60.53	44.19	63.70	57.66	**69.26**	68.98
Text → Email	52.08	51.81	53.10	52.05	54.95	50.22	51.96	52.79	44.39	**55.77**	53.94	54.26	55.24
Email → Text	57.99	57.24	58.20	58.58	59.72	53.50	57.96	49.32	64.05	64.48	60.68	**68.97**	65.04
Action → Family	85.50	87.26	83.42	70.53	79.26	82.94	85.26	66.85	66.61	81.18	**87.43**	86.86	86.06
Action → War	83.98	85.04	85.42	73.51	83.68	82.44	83.90	70.05	62.59	81.31	75.20	85.88	**86.09**
Family → Action	81.71	81.56	81.18	75.30	80.36	81.89	82.01	70.16	65.02	80.98	49.21	**83.77**	83.30
Family → War	80.68	81.20	82.14	73.11	80.20	80.33	81.28	71.76	63.67	79.85	51.66	**82.87**	81.82
War → Action	83.24	84.15	84.18	74.72	82.71	81.86	83.01	69.90	64.66	80.75	**86.10**	84.97	84.62
War → Family	83.18	84.38	84.38	72.21	81.34	80.54	81.82	66.45	63.41	78.78	67.33	**86.06**	86.00
MNIST → USPS	48.40	45.80	38.70	47.70	53.80	48.80	39.10	62.60	77.10	75.56	68.80	**79.90**	79.30
USPS → Optdigits	52.70	56.30	51.20	52.20	54.30	52.60	54.20	64.30	65.70	70.38	65.00	**73.40**	71.90
Optdigits → MNIST	16.40	28.40	16.10	20.40	36.80	16.40	29.50	40.90	**51.90**	42.94	37.40	43.60	44.70
Left → Right	85.16	84.55	89.03	92.90	83.23	93.29	84.52	89.68	87.10	92.21	92.84	**94.19**	93.55
Right → Left	81.53	80.25	74.52	79.62	73.25	84.08	81.53	**89.81**	86.62	80.25	80.25	83.44	82.80
Books → DVD	77.40	76.55	73.60	68.05	76.96	75.60	77.60	68.20	61.70	75.80	79.10	**79.70**	78.70
Books → Kitchen	74.30	73.80	71.25	66.15	74.70	72.35	74.75	63.85	61.40	73.55	**79.85**	78.10	78.00
Books → Electronics	72.50	71.35	53.30	63.95	66.30	69.50	71.80	65.40	59.75	69.95	71.75	**73.80**	72.10
DVD → Books	77.45	76.95	74.60	70.10	77.00	75.55	77.90	66.05	60.35	75.30	54.20	**78.50**	78.50
DVD → Kitchen	76.00	75.40	70.00	68.20	74.81	74.65	76.10	59.25	63.90	73.40	50.65	**78.35**	78.10
DVD → Electronics	75.20	74.50	70.10	66.80	74.03	72.85	74.35	61.40	62.60	74.40	61.80	77.65	**78.40**
Kitchen → Books	72.45	72.05	71.70	69.85	72.70	71.25	72.80	60.00	57.90	71.05	50.60	73.60	**75.45**
Kitchen → DVD	74.15	73.90	49.95	65.55	66.31	73.50	73.50	63.90	61.40	72.80	56.15	**75.50**	74.70
Kitchen → Electronics	82.85	82.60	78.00	75.65	81.78	79.65	82.85	70.90	70.00	80.40	82.60	**84.50**	84.15
Electronics → Books	69.60	69.20	67.80	66.40	69.51	68.45	69.60	61.50	55.20	67.70	55.35	**72.15**	71.85
Electronics → DVD	71.60	71.50	67.45	65.60	70.35	68.40	70.50	59.95	53.40	70.05	56.05	**73.10**	71.65
Electronics → Kitchen	83.65	83.25	80.95	76.70	83.35	82.55	83.50	70.15	65.00	81.25	82.45	**85.60**	83.55

the reviews from a product by using the labeled reviews from another product. During training time, all algorithms are given 2000 labeled source examples and 2000 unlabeled target examples. At test time, we evaluate their performances on the 2000 target examples with labels.

C. Experimental Setup

In addition to TCA, SA, and JGSA, we compare CBSRM and CJSRM with more state-of-art domain adaptation methods on the real-world data sets. The comparison methods are described as follows.

1) SRM is a baseline for all domain adaptation methods. The classifier is trained via minimizing the structural risk on the source data.
2) IEM [18] is an integrated exponential model that reweights the source instances by logistic regression and minimizes the exponential loss to learn the target classifier.
3) Transfer joint matching (TJM) [26] is a feature learning method that extends TCA by heuristically reweighting the instances.
4) CORAL [47] aligns the source and target covariance matrices in the original feature space to correct the *distribution mismatch*.
5) STM [20] jointly minimizes the KMM [17] objective and the weighted structural risk to learn the target classifier.
6) Optimal Transport-Group Lasso (OT-GL) [5] applies optimal transport to align the source and target distributions and regularizes the transport with the group-lasso regularization term.
7) Kernel whitening coloring (KWC) [14] uses a linear operator to align the source and target covariance matrices in the RKHS. The linear operator in the RKHS is the whitening-coloring map.
8) Kernel optimal transport (KOT) [14] uses a linear operator to align the source and target covariance matrices in the RKHS. The linear operator in the RKHS is the optimal transport map.
9) Manifold Criterion Guided Transfer Learning (MCTL) [61] considers the data locality structure and generates a new intermediate domain sharing similar distribution with the true target domain.
10) GSL [46] learns an invariant, discriminative, and domain-agnostic subspace by the guide learning mechanism.

For a fair comparison, the exponential loss for classification is used in all the methods. We employ a linear model space and an RBF network hypothesis space to learn both linear and nonlinear classification models. For the model hyperparameters, if there are default values recommended in the original articles, we directly use their default values. Otherwise, we estimate the hyperparameters via cross validation on the source data. Because of the *distribution shift*, it should be noted that hyperparameter values searched via cross validation on the source data are not guaranteed to be optimal for generalizing to the target data [62]. In this article, the subspace dimension, the regularization parameters, and the Gaussian kernel bandwidth are, respectively, searched in $\{5, 10, 20, 50, 100, 500, 800\}$, $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, and $\{0.5, 1, 5, 10, 15, 20, 50, 100\}$.

In the experiments, we use centering or z-score standardization for data preprocessing. Regarding the implementation of the comparison methods, we use the released codes of the authors if they are available. Otherwise, we implement the algorithms according to the original articles.

D. Experimental Results and Discussion

We report the linear and nonlinear classification results in Table II and Table III, respectively, and highlight the best result in bold for each task. From Table II, we observe that CBSRM and CJSRM, respectively, achieve the best results on 60.47% and 13.95% of the tasks. In addition, they also obtain the second-best results on 13.95% and 37.21% of the tasks. In Table III, CJSRM and CBSRM respectively achieve the best results on 41.38% and 20.69% of the tasks. Besides, they also obtain the second-best results on 17.24% and 31.03% of the tasks. These results indicate that the proposed methods are feasible for domain adaptation. Comparing Table III with Table II, we find an interesting fact: unlike traditional machine learning, nonlinear models do not always outperform linear models in unsupervised domain adaptation. This is because nonlinear models are complex models, and they can easily overfit the source data.

E. Statistical Test

We conduct the Wilcoxon signed-ranks test [63] to check if the proposed methods are significantly better than the other algorithms using linear classification model. The test compares the performances of two algorithms on multiple tasks. Therefore, we fix CBSRM as a control algorithm and conduct 12 pairs of tests: SRM versus CBSRM,…, CJSRM versus CBSRM. To run the test, we rank the differences in the performances of two algorithms for each task. The differences are ranked according to their absolute values. The smallest absolute value gets the rank of 1, the second smallest gets the rank of 2, and so on. In case of equality, the average ranks are assigned. The statistics of the Wilcoxon signed-ranks test is

$$z(a,b) = \frac{T(a,b) - N(N+1)/4}{\sqrt{N(N+1)(2N+1)/24}} \tag{19}$$

where $T(a,b) = \min\{R^+(a,b), R^-(a,b)\}$. $R^+(a,b)$ is the sum of ranks for the tasks on which algorithm b outperforms algorithm a and $R^-(a,b)$ is the sum of ranks for the opposite. They are defined as follows:

$$R^+(a,b) = \sum_{\text{diff}_i>0} \text{rank}(\text{diff}_i) + \frac{1}{2}\sum_{\text{diff}_i=0} \text{rank}(\text{diff}_i) \tag{20}$$

$$R^-(a,b) = \sum_{\text{diff}_i<0} \text{rank}(\text{diff}_i) + \frac{1}{2}\sum_{\text{diff}_i=0} \text{rank}(\text{diff}_i) \tag{21}$$

where diff_i is the difference between the accuracy of two algorithms on the ith task out of N tasks, and $\text{rank}(\text{diff}_i)$ is the rank of $|\text{diff}_i|$. We fix b as CBSRM and let a vary from SRM to CJSRM in turn. Based on formulas (19)–(21), we compute $z(a,b)$ for the 12 pairs of tests and report the results in Table IV. The second row of Table IV shows that none of the z values exceeds the critical value -1.96. This indicates that with the significance level $\alpha = 0.05$, CBSRM is statistically better than the other algorithms using a linear model.

Following the same test procedure as the linear case, we find that with significance level $\alpha = 0.05$, CBSRM and CJSRM both statistically outperform the other comparison methods in the nonlinear model experiments.

F. Characteristic Analysis of CBSRM

1) *Hyperparameter Sensitivity:* Recall that CBSRM involves three hyperparameters—subspace dimensionality d_1 and two regularization parameters

TABLE III
CLASSIFICATION ACCURACY (%) OF NONLINEAR MODELS OVER 29 ADAPTATION TASKS

Task	SRM	IEM	TCA	SA	TJM	CORAL	STM	OT-GL	JGSA	KWC	KOT	MCTL	GSL	CBSRM	CJSRM
Com vs Rec	85.00	91.21	86.32	73.75	90.96	84.35	90.50	88.69	88.28	89.85	**92.88**	91.32	89.95	90.76	90.35
Com vs Sci	78.22	82.63	74.98	64.50	87.18	78.37	85.97	86.88	88.20	87.59	86.58	87.09	87.54	88.70	**89.01**
Com vs Talk	89.02	92.55	82.11	81.10	92.87	89.07	92.29	90.71	91.34	91.97	**94.35**	88.29	92.56	90.87	90.07
Rec vs Com	89.93	92.04	94.26	57.22	90.03	90.08	95.82	96.47	82.78	95.36	92.60	93.11	94.72	95.97	**97.12**
Rec vs Sci	67.88	75.32	67.73	53.31	79.84	68.84	86.33	90.55	88.24	93.61	86.78	92.31	91.26	**95.17**	94.87
Rec vs Talk	76.39	82.26	81.58	83.63	85.36	76.28	87.61	86.46	79.53	88.30	87.82	87.93	88.61	86.30	**88.90**
Sci vs Com	81.85	84.23	69.78	54.52	85.14	81.96	90.04	92.21	81.80	89.99	87.92	91.71	85.45	**92.82**	92.67
Sci vs Rec	60.52	67.01	45.61	89.12	73.16	60.87	73.41	74.92	78.90	81.52	81.72	78.70	81.92	89.22	**91.44**
Sci vs Talk	51.84	47.47	49.52	54.57	50.57	56.15	52.73	54.57	25.52	54.63	55.42	**60.84**	32.53	53.21	56.57
Talk vs Com	92.42	92.47	92.94	84.64	93.25	93.41	94.24	**96.31**	91.64	94.34	92.58	92.38	92.53	93.10	93.41
Talk vs Rec	79.16	88.13	75.39	44.55	93.39	80.60	92.72	90.40	88.70	91.49	91.18	92.83	88.45	92.77	**93.77**
Talk vs Sci	56.60	60.07	44.74	39.77	54.94	56.03	62.29	59.50	65.35	58.26	55.51	51.48	**77.21**	61.88	66.23
Orgs vs Places	64.23	70.37	65.10	60.59	68.64	69.22	69.03	68.45	67.30	73.63	**76.70**	65.48	61.36	71.62	72.29
Orgs vs People	71.19	76.07	72.43	59.27	76.82	74.25	**78.64**	71.02	50.00	73.75	70.28	71.19	67.14	73.92	73.92
Places vs Orgs	66.73	70.27	67.32	42.42	58.75	64.76	71.06	65.55	46.45	64.86	65.05	69.88	62.40	71.16	**71.45**
Places vs People	54.78	54.03	54.13	66.01	58.58	56.73	57.93	62.11	47.81	56.54	58.86	57.47	58.50	68.33	**69.08**
People vs Orgs	77.84	79.46	74.13	73.96	75.99	76.55	82.53	79.30	62.32	78.90	80.11	74.78	80.27	**83.10**	83.10
People vs Places	54.41	56.91	50.97	60.53	55.33	55.52	60.72	61.46	40.76	63.88	62.85	61.65	52.37	65.45	**66.10**
Text → Email	51.10	47.15	53.76	50.84	55.33	50.29	51.81	52.45	56.71	**57.19**	48.63	46.78	50.89	53.76	52.88
Email → Text	66.63	54.66	67.96	75.45	65.72	56.01	57.99	68.32	63.01	74.37	49.43	73.83	68.30	78.17	**83.10**
Action → Family	87.26	87.18	68.85	77.66	80.28	86.46	**87.75**	67.09	66.53	75.98	76.46	86.31	87.03	87.26	87.18
Action → War	85.96	86.34	67.78	77.27	81.68	85.20	87.09	71.62	64.37	82.28	82.17	86.66	76.60	**87.51**	86.33
Family → Action	82.21	84.36	67.96	76.48	79.37	84.00	84.47	69.98	49.20	82.27	82.30	82.77	49.24	**85.59**	83.89
Family → War	82.12	**85.23**	65.26	73.60	81.25	82.87	85.15	67.48	51.66	81.33	81.11	81.85	51.66	84.96	82.49
War → Action	85.68	**86.39**	69.69	78.80	82.71	85.21	85.74	70.48	66.37	82.24	81.95	85.39	86.13	85.44	83.68
War → Family	85.42	86.30	67.09	77.58	84.34	83.58	86.06	66.53	65.09	76.62	76.94	86.07	71.26	86.30	**86.46**
MNIST → USPS	52.80	55.80	40.30	52.30	60.80	52.10	40.10	62.40	78.70	37.40	51.20	76.43	68.70	80.30	**80.50**
USPS → Optdigits	58.70	60.30	53.70	59.10	64.50	59.20	59.20	64.70	67.40	61.90	65.50	72.31	68.00	**73.70**	69.90
Optdigits → MNIST	18.50	27.40	21.50	21.60	33.30	18.50	29.10	40.90	**53.30**	27.40	35.00	42.15	38.00	43.10	44.70

TABLE IV
SUMMARY OF STATISTICS OF THE WILCOXON SIGNED-RANKS TEST

Statistics	SRM	IEM	TCA	SA	TJM	CORAL	STM	OT-GL	JGSA	MCTL	GSL	CJSRM
z	-5.71	-5.70	-5.71	-5.69	-4.94	-5.70	-5.70	-5.06	-5.34	-4.65	-4.67	-2.72

λ_1, λ_2. Here, we examine the sensitivity of its performance with respect to different choices of these parameters. Specifically, we run the experiments on the tasks *Rec* versus *Talk*, *Com* versus *Sci*, and *Orgs* versus *Places* with the parameter ranges $d_1 \in \{10, 20, \ldots, 100\}$, and $\lambda_1, \lambda_2 \in \{10^3, 10^2, \ldots, 10^3\}$. Following [9], we use a heat map to present the sensitivity result. Fig. 4 shows the target domain classification accuracy of CBSRM versus different parameter pairs in terms of $(d_1, \log_{10}(\lambda_1), \log_{10}(\lambda_2))$ on the three-domain adaptation tasks. As can be observed from Fig. 4(a) and (b), CBSRM is sensitive to the regularization parameter λ_2 on the two adaptation tasks *Rec* versus *Talk* and *Com* versus *Sci*. Fig. 4(c) shows that the sensitivity of CBSRM on the task *Orgs* versus *Places* depends on the subspace dimensionality d_1 and the regularization parameter λ_1. Since CBSRM attains its superior performance with different choices of the hyperparameters on different tasks, as a general guideline for choosing hyperparameters, we would suggest picking d_1, λ_1, and λ_2 via cross validation to obtain their suitable values for different tasks.

2) *Learning Curves:* CBSRM is a classifier-embedded domain adaptation model. Therefore, we can investigate its behavior in the target domain as the number of training samples increases. Note that the training samples of CBSRM consist of the labeled source samples and the unlabeled target samples, both of which are set to be equal in size here, i.e., $m_s = m_t$. We calculate the classification accuracy of CBSRM as a function of the number of samples ($= m_s = m_t$) on the task *Rec* versus *Talk*. The classification accuracy is measured on a target test data set consisting of 800 samples. For comparison purposes, we also train an SRM using the labeled target data, which is denoted as SRM-T. We gradually increase the number of samples from 100 to 900 and measure the classification accuracy of CBSRM and SRM-T on the test set. For every sample size, the above procedure is repeated 30 times to calculate the mean and standard deviation of the classification accuracy. The learning curves of SRM-T and CBSRM are shown in Fig. 5. We observe from Fig. 5 that SRM-T consistently outperforms CBSRM for every sample size. This is unsurprising since SRM-T is trained on the labeled target data and it just serves as an

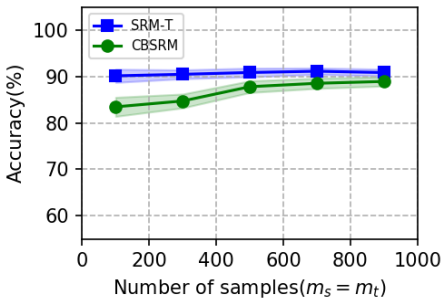


Fig. 5. Learning curves of SRM-T and CBSRM in the target domain.

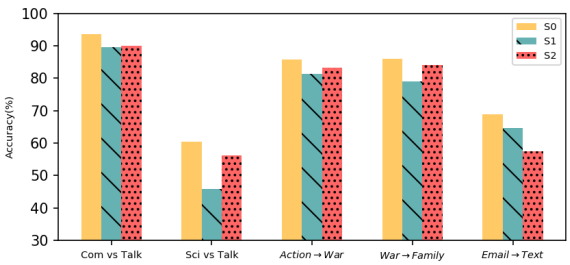


Fig. 6. Adaptation results from three different settings of CBSRM on five domain adaptation tasks.

upper bound here. The interesting thing is that as the sample size increases, CBSRM gradually corrects the *distribution mismatch* and approximates its upper bound SRM-T, which is a very desirable property and demonstrates the success of domain adaptation.

3) *Ablative Analysis:* The proposed CBSRM contains two components—dimension reduction and subspace distribution adaptation. To test whether these two components are essential to CBSRM, we consider removing one of them each time and observe how the adaptation result will change. Specifically, we study the following settings:
   a) S0: the standard CBSRM model;
   b) S1: remove the subspace distribution adaptation from CBSRM;
   c) S2: remove the dimension reduction from CBSRM.

Fig. 6 shows the adaptation results from these settings on five domain adaptation tasks. Obviously, the standard setting S0 outperforms the other two settings S1 and S2, implying that removing either the subspace distribution adaptation or the dimension reduction will decrease the performance of the model. Thus, dimension reduction and subspace distribution adaptation are both indispensable components in the CBSRM model.

IX. Conclusion

In this article, two learning frameworks: BSRM and JSRM are proposed for the unsupervised domain adaptation problem. The main goal of these two frameworks is to adapt the source distribution to the target distribution in the subspace to learn the unbiased target prediction model. We derive convex instantiations from both frameworks, which accelerates the optimization process and makes the solution more accurate. The extensive experimental evaluations on both synthetic and real-world data sets demonstrate the effectiveness of the proposed methods.

There are a few points worth discussing and further exploring. The first one is to study the nonconvex instantiations of BSRM and JSRM. The second one is to match the joint distributions of both domains, instead of only matching the marginal distributions. However, since no target labels are available for unsupervised domain adaptation, this will be a more challenging task. The last one is to extend the current frameworks to end-to-end learning paradigms such that they can be directly applied to big natural image classification problems. This may be achieved by reweighting the classification loss and incorporating the Bregman divergence term into the existing end-to-end deep learning architectures.

Appendix

A. Proof of Theorem 1

On the one hand, when $g(x) = x \log x$, we have

$$\begin{aligned} &E_{P_s(z)}\left[B_g\left(\frac{P_t(z)}{P_s(z)}||r(z)\right)\right] \\ &\quad= \int_z P_s(z)r(z)dz - \int_z P_t(z)\log r(z)dz - 1 \\ &\qquad+ \int_z P_t(z)\log\frac{P_t(z)}{P_s(z)}dz \\ &\quad= D_{KL}(P_t(z)||r(z)P_s(z)) \end{aligned} \tag{22}$$

where $D_{KL}(\cdot||\cdot)$ is the KL divergence. Equation (22) makes use of the assumption that $r(z)P_s(z)$ is a probability distribution and, therefore, $\int_z r(z)P_s(z)dz = 1$. On the other hand, for any hypothesis $h \in \mathcal{H}$, we have

$$\begin{aligned} &|E_{P_t(z,y)}[\ell(h(z), y)] - E_{r(z)P_s(z,y)}[\ell(h(z), y)]| \\ &\quad= \left|\int_{(z,y)} \ell(h(z), y)P_s(y|z)(P_t(z) - r(z)P_s(z))dzdy\right| \\ &\quad\le \int_{(z,y)} |\ell(h(z), y)P_s(y|z)(P_t(z) - r(z)P_s(z))|dzdy \\ &\quad\le M\int_z |P_t(z) - r(z)P_s(z)|dz \\ &\quad\le \sqrt{2}M\sqrt{D_{KL}(P_t(z)||r(z)P_s(z))}. \end{aligned} \tag{23}$$

The first equality exploits the chain rule and the *generalized covariate shift* assumption. The first inequality uses the property of integral. The second inequality utilizes the assumption $\ell(\cdot,\cdot) \le M$ and the fact $P_s(y|z) \le 1$. The last inequality is due to Pinsker's inequality [64]. Hence, it holds that

$$\begin{aligned} E_{P_t(z,y)}[\ell(h(z), y)] &\le E_{r(z)P_s(z,y)}[\ell(h(z), y)] \\ &+\sqrt{2}M\sqrt{E_{P_s(z)}\left[B_g\left(\frac{P_t(z)}{P_s(z)}||r(z)\right)\right]}. \end{aligned} \tag{24}$$

Based on the inequality $\sqrt{x} \le x$ for $x \ge 1$, when

$$E_{P_s(z)}\left[B_g\left(\frac{P_t(z)}{P_s(z)}||r(z)\right)\right] \ge 1$$

we have

$$E_{P_t(z,y)}[\ell(h(z),y)] \le E_{r(z)P_s(z,y)}[\ell(h(z),y)] + M\sqrt{2}E_{P_s(z)}\left[B_g\left(\frac{P_t(z)}{P_s(z)}||r(z)\right)\right]. \quad (25)$$

□

B. Proof of Theorem 2

The diagonal matrix $\mathbf{\Lambda}$ is positive definite, and therefore, the last term in (8) is convex. The affine function $-\boldsymbol{w}^T\boldsymbol{v}_i^t$ is convex and $e^{\boldsymbol{w}^T\boldsymbol{v}_i^s}$ is also a convex function based on the affine composition rule [65]. Therefore, the second and third terms in (8) are both convex. The convexity of (8) finally depends on the convexity of $e^{\boldsymbol{w}^T\boldsymbol{v}_i^s}\varphi(y_i^s\boldsymbol{w}^T\boldsymbol{v}_i^s)$. Since a log-convex function is convex in itself, we only have to check whether $\log(e^{\boldsymbol{w}^T\boldsymbol{v}_i^s}\varphi(y_i^s\boldsymbol{w}^T\boldsymbol{v}_i^s)) = \boldsymbol{w}^T\boldsymbol{v}_i^s + \log(\varphi(y_i^s\boldsymbol{w}^T\boldsymbol{v}_i^s))$ is convex. Obviously, it is convex if $\log\varphi$ is convex. In particular, when φ is the exponential loss $\varphi(x) = e^{-x}$, $\log\varphi(x) = \log e^{-x} = -x$ is convex, which consequently makes (8) a convex optimization problem. □

C. Proof of Theorem 3

Based on the Bayes' rule, we have

$$\frac{P_t(z)}{P_s(z)} = \frac{P(z\mid l=+1)}{P(z\mid l=-1)} = \frac{P(l=+1\mid z)P(z)/P(l=+1)}{P(l=-1\mid z)P(z)/P(l=-1)} = \frac{P(l=+1\mid z)}{1-P(l=+1\mid z)}\frac{1-P(l=+1)}{P(l=+1)} = \frac{1-\pi}{\pi}\frac{\eta(z)}{1-\eta(z)}. \quad (26)$$

Since ℓ_1 is a strictly proper composite loss function with invertible link function ψ, given a binary classification model f with low risk, η can be estimated as $\hat{\eta} = \psi^{-1}f$. Therefore, as an estimate of $\frac{P_t(z)}{P_s(z)}$, the distribution adaptation function $r(z)$ can be expressed as

$$r(z) = \frac{1-\pi}{\pi}\frac{\hat{\eta}(z)}{1-\hat{\eta}(z)} = \frac{1-\pi}{\pi}\frac{(\psi^{-1}f)(z)}{1-(\psi^{-1}f)(z)}. \quad (27)$$

By applying [54, Proposition 3], we have

$$E_{P_s(z)}\left[B_{g_\pi}\left(\frac{P_t(z)}{P_s(z)}||r(z)\right)\right] = 2(E_{P(z,l)}[\ell_1(f(z),l)] - E_{P(z,l)}[\ell_1((\psi\eta)(z),l)]) \quad (28)$$

where g_π is a differential and strictly convex function dependent on ℓ_1, ψ, and π.

On the right-hand side of the optimization problem (2), by plugging (27) into the first term and expressing the second term (Bregman divergence) as (28), we obtain

$$(f^*,h^*) = \operatorname*{argmin}_{f,h\in\mathcal{H}} \int_{(z,y)} P_s(z,y)\ell(h(z),y)\frac{1-\pi}{\pi}\frac{(\psi^{-1}f)(z)}{1-(\psi^{-1}f)(z)}dzdy + 2\int_{(z,l)} P(z,l)\ell_1(f(z),l)dzdl. \quad (29)$$

Note that we drop the term $E_{P(z,l)}[\ell_1((\psi\eta)(z),l)]$ since it is a constant independent of f and h. □

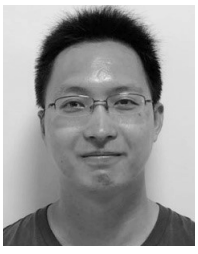
ACKNOWLEDGMENT

The authors would like to thank the associate editor and the reviewers for their valuable comments and suggestions.

REFERENCES

[1] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015.
[2] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross-domain learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1768–1779, Jun. 2019.
[3] W. M. Kouw, L. J. van der Maaten, J. H. Krijthe, and M. Loog, "Feature-level domain adaptation," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 5943–5974, 2016.
[4] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.
[5] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, Sep. 2017.
[6] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1859–1867.
[7] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptation with distribution matching machines," in *Proc. AAAI*, 2018, pp. 2795–2802.
[8] W.-Y. Deng, A. Lendasse, Y.-S. Ong, I. W.-H. Tsang, L. Chen, and Q.-H. Zheng, "Domain adaption via feature selection on explicit feature map," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1180–1190, Apr. 2019.
[9] L. A. Pereira and R. D. S. Torres, "Semi-supervised transfer subspace for domain adaptation," *Pattern Recognit.*, vol. 75, pp. 235–249, Mar. 2018.
[10] W. Zellinger, B. A. Moser, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Robust unsupervised domain adaptation for neural networks via moment alignment," *Inf. Sci.*, vol. 483, pp. 174–191, May 2019.
[11] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1682–1695, Jul. 2017.
[12] C. Pölitz, W. Duivesteijn, and K. Morik, "Interpretable domain adaptation via optimization over the Stiefel manifold," *Mach. Learn.*, vol. 104, nos. 2–3, pp. 315–336, Sep. 2016.
[13] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.
[14] Z. Zhang, M. Wang, Y. Huang, and A. Nehorai, "Aligning infinite-dimensional covariance matrices in reproducing kernel Hilbert spaces for domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3437–3445.
[15] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2240–2249, Dec. 2014.
[16] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2935608.
[17] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proc. NIPS*, vol. 2007, pp. 601–608.
[18] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, pp. 2137–2155, Sep. 2009.
[19] H. Hachiya, M. Sugiyama, and N. Ueda, "Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition," *Neurocomputing*, vol. 80, pp. 93–101, Mar. 2012.
[20] W.-S. Chu, F. D. L. Torre, and J. F. Cohn, "Selective transfer machine for personalized facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 529–545, Mar. 2017.
[21] S. Khalighi, B. Ribeiro, and U. J. Nunes, "Importance weighted import vector machine for unsupervised domain adaptation," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3280–3292, Oct. 2017.
[22] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Planning Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000.
[23] M. Sugiyama, M. Kawanabe, and P. L. Chui, "Dimensionality reduction for density ratio estimation in high-dimensional spaces," *Neural Netw.*, vol. 23, no. 1, pp. 44–59, Jan. 2010.

[24] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
[25] M. Baktashmotlagh, M. Harandi, and M. Salzmann, "Distribution-matching embedding for visual domain adaptation," *J. Mach. Learn. Res.*, vol. 17, no. 108, pp. 1–30, Jul. 2016.
[26] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1410–1417.
[27] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.
[28] V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.
[29] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 114–121.
[30] Y.-Q. Miao, A. K. Farahat, and M. S. Kamel, "Auto-tuning kernel mean matching," in *Proc. IEEE 13th Int. Conf. Data Mining Workshops*, Dec. 2013, pp. 560–567.
[31] 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 *Proc. NIPS*, 2008, pp. 1433–1440.
[32] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, Jul. 2009.
[33] J. Wen, R. Greiner, and D. Schuurmans, "Correcting covariate shift with the Frank-Wolfe algorithm," in *Proc. IJCAI*, 2015, pp. 1010–1016.
[34] S. Chandra, A. Haque, L. Khan, and C. Aggarwal, "Efficient sampling-based kernel mean matching," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 811–816.
[35] J. Blitzer, R. Mcdonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2006, pp. 120–128.
[36] H. Daumé, III, A. Kumar, and A. Saha, "Frustratingly easy domain adaptation," in *Proc. Workshop Domain Adapt. Nat. Lang. Process.*, 2007, pp. 53–59.
[37] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.
[38] B. Quanz, J. Huan, and M. Mishra, "Knowledge transfer with low-quality data: A feature extraction issue," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 10, pp. 1789–1802, Oct. 2012.
[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: https://arxiv.org/abs/1412.3474
[40] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.
[41] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4500–4509.
[42] G. Cai, Y. Wang, L. He, and M. Zhou, "Unsupervised domain adaptation with adversarial residual transform networks," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2935384.
[43] J. Liang, R. He, Z. Sun, and T. Tan, "Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation," in *Proc. CVPR*, Jun. 2019, pp. 2975–2984.
[44] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.
[45] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
[46] L. Zhang, J. Fu, S. Wang, D. Zhang, Z. Dong, and C. L. P. Chen, "Guide subspace learning for unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2944455.
[47] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016, pp. 2058–2065.
[48] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. NIPS*, 2007, pp. 137–144.
[49] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proc. Conf. Learn. Theory*, 2009.
[50] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, Nov. 2002.
[51] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, "Unsupervised domain adaptation based on source-guided discrepancy," in *Proc. AAAI*, 2019.
[52] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Jul. 2017.
[53] Q. Que and M. Belkin, "Back to the future: Radial basis function networks revisited," in *Proc. AISTATS*, 2016, pp. 1375–1383.
[54] A. Menon and C. S. Ong, "Linking losses for density ratio and class-probability estimation," in *Proc. ICML*, 2016, pp. 304–313.
[55] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
[56] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Math. Methods Oper. Res.*, vol. 66, no. 3, pp. 373–407, Nov. 2007.
[57] P. Wei, Y. Ke, and C. K. Goh, "Feature analysis of marginalized stacked denoising autoenconder for unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1321–1334, May 2019.
[58] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2007, pp. 210–219.
[59] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2004, p. 271.
[60] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2005, pp. 886–893.
[61] L. Zhang, S. Wang, G.-B. Huang, W. Zuo, J. Yang, and D. Zhang, "Manifold criterion guided transfer learning via intermediate domain generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3759–3773, Dec. 2019.
[62] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, May 2007.
[63] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[64] M. D. Reid and R. C. Williamson, "Information, divergence and risk for binary experiments," *J. Mach. Learn. Res.*, vol. 12, no. 3, pp. 731–817, Mar. 2011.
[65] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.



Sentao Chen received the B.S. degree in statistics from the Guangdong University of Technology, Guangzhou, China.

He is currently pursuing the Ph.D. degree in software engineering with the School of Software Engineering, South China University of Technology, Guangzhou. His current research interests include statistical machine learning and domain adaptation.

Le Han received the B.S. degree in pure mathematics and the M.Sc. degree in computational mathematics from Wuhan University, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in computational mathematics from Sun Yat-sen University, Guangzhou, China, in 2008.

She is currently an Associate Professor with the School of Mathematics, South China University of Technology, Guangzhou. Her current research interests include matrix optimization, tensor learning, and computer graphics.

Xiaolan Liu received the Ph.D. degree in computer application technology from the South China University of Technology, Guangzhou, China, in 2011.

She is currently an Associate Professor with the School of Mathematics, South China University of Technology. Her current research interests include machine learning, and pattern recognition, and their applications in computer vision.

Zongyao He received the M.S. degree in power electronics and power drives from Wuhan University, Wuhan, China, in 2005.

He is currently a Professor with the School of Computer and Data Science, Henan University of Urban Construction, Pingdingshan, China. His current research interests include intelligent information processing, smart city.

Xiaowei Yang received the B.S. degree in theoretical and applied mechanics, the M.Sc. degree in computational mechanics, and the Ph.D. degree in solid mechanics from Jilin University, Changchun, China, in 1991, 1996, and 2000, respectively.

He is currently a full-time Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. He has published over 100 journals and refereed international conference papers, including the areas of structural reanalysis, interval analysis, soft computing, support vector machines, and tensor learning. His current research interests include designs and analyses of algorithms for large-scale pattern recognitions, imbalanced learning, semisupervised learning, support vector machines, tensor learning, and evolutionary computation.