IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 25, 2019, accepted September 3, 2019, date of publication September 27, 2019, date of current version October 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944226

Informative Feature Selection for Domain Adaptation

FENG SUN[1], HANRUI WU[1], ZHIHANG LUO[2], WENWEN GU[3], YUGUANG YAN[1], AND QING DU[1]

[1]School of Software Engineering, South China University of Technology, Guangzhou 510006, China
[2]Business School, The Hongkong University of Science and Technology, Hong Kong
[3]School of Business, La Trobe University, Bundoora, VIC 3086, Australia

Corresponding authors: Yuguang Yan (yan.yuguang@mail.scut.edu.cn) and Qing Du (duqing@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61876208, in part by the Guangdong Provincial Scientific and Technological funds under Grant 2017B090901008 and Grant 2018B010108002, in part by the Pearl River S&T Nova Program of Guangzhou under Grant 201806010081, in part by the CCF-Tencent Open Research Fund under Grant RAGR20190103, and in part by Natural Science Foundation of Guangdong Province under Grant 2016A030310423.

ABSTRACT Domain adaptation aims at extracting knowledge from an auxiliary source domain to assist the learning task in a target domain. When the data distribution of the target domain is different from that of the source domain, the direct use of source data for building a classifier for the target learning task cannot achieve promising performance. In this work, we propose a novel unsupervised domain adaptation method called *Feature Selection for Domain Adaptation (FSDA)*, in which we aim to select a set of informative features. The benefits are two-fold. The first is to reduce the mismatch between the data distributions in the source and target domains by selecting a set of informative features in which they share similar properties. The second is to remove noisy features in the source domain such that the learning performance can be enhanced. We formulate a new sparse learning model for structured multiple outputs, including a vector to select informative features that can be used to jointly minimize the domain discrepancy and eliminate noisy features, and a classifier to perform prediction on the selected features. We develop a cutting-plane algorithm to solve the resulting optimization problem. Extensive experiments on real-world data sets are tested to demonstrate the effectiveness of the proposed method compared with the other existing methods.

INDEX TERMS Domain adaptation, feature selection, structured multi-output learning, transfer learning.

I. INTRODUCTION

In standard machine learning, in order to obtain an effective classifier, one usually has to collect and label a large amount of training data, which is often labor intensive and expensive. In order to reduce the effort of collecting labeled training data in a target domain of interest, domain adaptation is employed to leverage abundant labeled data from an auxiliary source domain [1]–[3]. Domain adaptation has drawn much attention and been applied in many real-world applications, including sentiment analysis [4], [5], text classification [6], [7], WiFi location [8], [9], visual object recognition [10], [11], *etc*.

According to the availability of labeled target data, we can divide domain adaptation into two categories: supervised and unsupervised settings. Supervised domain adaptation requires some labeled target data for training [12]; while unsupervised domain adaptation uses unlabeled target data for training [13], [14]. Here, we consider unsupervised domain adaptation, which is more challenging compared to the supervised setting.

One simple and straightforward method to leverage labeled source data is to classify target data using a classifier trained on labeled source data directly. However, due to the distribution difference between source and target data, this method may not achieve promising performance on the target task. Therefore, the principal problem in domain adaptation is how to reduce the difference between the source and target distributions. A majority of works have been proposed to learn a feature mapping [9], [15]–[17], such that the domain distribution mismatch is reduced in the new feature space. Then, one can learn a classifier in the new feature space without considering the domain difference. However, while

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

being able to reduce the domain discrepancy, such two-stage approaches cannot guarantee the learnt representation in the first stage is also informative to the classifier learning task in the second stage, making the discriminative ability of the representation limited.

In this paper, we firstly explore a question that if there exists a feature subset such that the distribution difference measured on it can be reduced, even if the data distributions of source and target domains are different. Motivated by this study, we propose a novel sparse learning model to learn structured multiple outputs, which aim to find both a shared feature representation and an effective classifier. As a result, the domain difference and the classification loss are jointly reduced. Different from the existing two-step methods, our joint learning model is able to select informative features to align source and target distributions, and also avoid sub-optimal solutions resulted from two-step learning models.

Specifically, we introduce a parameter vector with binary $0/1$ values to select a feature subset with a strong discriminative ability, and measure the distribution difference on the new representation based on *Maximum Mean Discrepancy* (MMD) [18], [19]. Furthermore, in order to explicitly leverage label information to train the classifier on the selected features, we apply the support vector machine (SVM) model with the squared hinge loss [20] to minimize the training loss on the new representation of source data. The resulting problem is difficult to solve because of the discrete parameters. To address it, we develop a cutting-plane algorithm to iteratively pick up informative features and train the classifier on them.

We highlight our principal contributions as follows.

- We propose a novel sparse learning model to learn structured multiple outputs, which jointly reduce the domain difference and training loss.
- A cutting-plane algorithm is developed to solve the resulting optimization problem.
- We perform extensive experiments on three real-world data sets to evaluate the performance of our proposed algorithm on the domain adaptation problem.

The paper is organized as follows. In Section II, we review important works regarding domain adaptation. We introduce our proposed method for domain adaptation in Section III. The experimental results are presented in Section IV. Section V concludes the whole paper.

II. RELATED WORKS

Existing domain adaptation problems can be divided into two categories: supervised domain adaptation and unsupervised domain adaptation [1]–[3]. The supervised setting, where some labeled target data are collected beforehand to train the classifier, has been extensively studied for decades [21], [22]. Some researchers further leverage unlabeled target data with semi-supervised techniques to boost the performance [23]–[26]. On the contrary, this paper focuses on the unsupervised setting, where no labeled target data are available for training, see for instance [15]–[17], [27].

To address domain adaptation problems, most approaches are based on two strategies: instance re-weighting and feature representation matching. Instance re-weighting, which aims to minimize the marginal distribution difference between the source and target domains by re-weighting the source domain samples. [28] learns a weighting vector for source data to reduce the marginal distribution difference, and then trains on the re-weighting source domain samples to obtain the target domain classifier. Chu *et al.* [29], [30] propose to jointly adapt source and target marginal distributions and learn an SVM classifier. These methods make an assumption that the conditional distributions of source and target domains are the same.

To relax the assumption of the same conditional distribution, feature representation matching methods seek to transform the source and target data to a shared subspace while preserving the commonalities between the source and target domains. Correlation alignment (CORAL) [31] minimizes domain shift by aligning the second-order statistics of source and target distributions. Geodesic flow kernel (GFK) [27] integrates an infinite number of subspaces to learn new feature representations that are robust to change in domains, and uses the symmetrized KL divergences between the source and target domains to measure the domain difference. After that, the classifier is trained on the learned representations. Manifold Criterion guided Transfer Learning (MCTL) [32] exploits the domain locality to learn feature projections for domain matching. LDA-inspired Domain Adaptation (LDADA) [33] proposes a linear-discriminant-analysis-like framework, so that the samples from the sample class in different domains are sufficiently close, and the samples from different classes are separated by large margins. Reference [34] proposes to exploit low-rank constraints for cross-domain metric learning.

Several methods use MMD instead of the KL divergence to measure domain differences. Transfer component analysis (TCA) [9] discovers a low-dimensional linear transformation such that the marginal distribution difference between the source and target domains is minimized, and adopts MMD to measure the distribution difference in a reproducing kernel Hilbert space (RKHS) [35]. Joint distribution analysis (JDA) [15] improves TCA through simultaneously adopting the marginal and conditional distributions in a principled dimensionality reduction process, where pseudo target labels generated from a classifier are used to measure conditional distribution difference. Transfer joint matching (TJM) [16] jointly performs feature representation matching and instance re-weighting techniques in a unified formulation. Joint geometrical statistical alignment (JGSA) [17] uses the feature representation matching strategy to reduce the distributional and geometrical divergence between the source and target domains. In [36], MMD is introduced into graph to learn domain-invariant representations based on geometric information.

The above-mentioned feature matching methods are devoted to learning feature transformation functions, and the

TABLE 1. Notations and descriptions used in this paper.

Notation	Description	Notation	Description
$\mathcal{D}_s$	source domain	$\mathcal{D}_t$	target domain
$\mathcal{X}_s$	source feature space	$\mathcal{X}_t$	target feature space
$\mathcal{Y}_s$	source label space	$\mathcal{Y}_t$	target label space
$\mathbf{x}_{s,i}$	a source instance	$\mathbf{x}_{t,i}$	a target instance
$y_{s,i}$	a source label	$y_{t,i}$	a target label
n_s	#source examples	n_t	#target examples
$P_s(\mathbf{x}_s)$	source marginal probability distribution	$P_t(\mathbf{x}_t)$	target marginal probability distribution
$P_s(y_s\|\mathbf{x}_s)$	source conditional probability distribution	$P_t(y_t\|\mathbf{x}_t)$	target conditional probability distribution
K	number of classes	$\mathbf{1}$	a vector with all elements one
$\mathbf{a}^\top$	transpose of vector $\mathbf{a}$	$\mathbf{a}\odot\mathbf{b}$	element-wise product of $\mathbf{a}$ and $\mathbf{b}$

feature representation and the classifier are learned separately. Different from these works, we propose to select informative features across domains from original features, and formulate a unified learning model to jointly learn a domain-invariant representation and a classifier. Feature selection with MMD (f-MMD) [37] picks up domain-invariant features of source and target domains, but ignores the label information of training data, thus cannot assure the discriminative ability of the selected features. In [38], optimal transport between domains is used to select domain-invariant features, and then a traditional classifier is conducted on the selected features. Instead, we propose to select features that can jointly reduce the domain difference and minimize the training loss on the labeled source data, thus can find discriminative features shared by source and target domains.

Sparse learning has been widely studied in existing researches [39]–[41]. In [42], the ℓ_1 norm is used to induce a sparse learning model for robust face representations. In [43], the ℓ_1 norm is introduced into PCA to obtain informative features. Different from them, we apply a binary feature selection vector into the learning model, so that the selected features can be used to reduce domain discrepancy and training loss simultaneously.

III. FEATURE SELECTION FOR DOMAIN ADAPTATION

A. PROBLEM STATEMENT

In this part, we firstly provide the definitions of terminologies, and then describe the notations. TABLE 1 lists the notations and their descriptions used in this paper.

Definition 1 (Domain): A domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X} = \mathbb{R}^d$ where d is the number of features, and a marginal probability distribution $P(\mathbf{x})$ where $\mathbf{x} \in \mathcal{X}$.

Definition 2 (Task): Given a specific domain $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$, a task consists of two components: a label space $\mathcal{Y} = \{1, \ldots, K\}$ where K is the number of labels, and a classifier $h(\mathbf{x})$ that can be modeled as a conditional probability distribution $P(y|\mathbf{x})$, where $y \in \mathcal{Y}$.

In our considered unsupervised domain adaptation problem, we have a source domain $\mathcal{D}_s = \{\mathcal{X}_s, P_s(\mathbf{x}_s)\}$ and a target domain $\mathcal{D}_t = \{\mathcal{X}_t, P_t(\mathbf{x}_t)\}$, where both feature and label spaces are the same, *i.e.*, $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$; while both marginal and conditional probability distributions of two domains are different, *i.e.*, $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$ and $P_s(y_s|\mathbf{x}_s) \neq P_t(y_t|\mathbf{x}_t)$. Our goal is to leverage labeled source data $\{(\mathbf{x}_{s,i}, y_{s,i})\}_{i=1}^{n_s}$ to classify unlabeled target data $\{\mathbf{x}_{t,i}\}_{i=1}^{n_t}$, where n_s and n_t are the numbers of source and target data, respectively. Specifically, we consider the binary classification problem, *i.e.*, $\mathcal{Y}_s = \mathcal{Y}_t = \{+1, -1\}$. The multi-class classification problem can be handled by the one-vs-rest approach [44].

B. LEARNING MODEL

In the domain adaptation scenario, to pick up informative features, we introduce a binary vector $\boldsymbol{\beta} \in \{0,1\}^d$ to scale an instance $\mathbf{x}$ by $(\mathbf{x} \odot \boldsymbol{\beta})$, where the value 1 indicates that the corresponding feature is selected. Let the set $\mathcal{B} = \{\boldsymbol{\beta} | \boldsymbol{\beta} \in \{0,1\}^d, ||\boldsymbol{\beta}||_0 \leq B\}$ be the domain of the parameters $\boldsymbol{\beta}$, where B is the maximum number of the selected features.

We seek to learn structured multiple outputs, which aim to find a feature subset and a classifier such that the domain difference and empirical loss are reduced simultaneously. Motivated by this, we propose a sparse learning model as follows:

$$\min_{\boldsymbol{\beta}\in\mathcal{B}} \min_{\mathbf{w},b,\boldsymbol{\xi}_s} \frac{1}{2}||\mathbf{w}||_2^2 + \frac{1}{2}C\sum_{i=1}^{n_s}\xi_{s,i}^2 + \Omega(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t),$$
$$\text{s.t. } y_{s,i}(\mathbf{w}^\top(\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) + b) \geq 1 - \xi_{s,i}, \quad i = 1, \ldots, n_s, \quad (1)$$

where the squared hinge loss is applied to train the classifier, C is a trade-off parameter, and the term $\Omega(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t)$ is used to reduce the difference between the source and target domains.

Existing works have demonstrated that empirical *Maximum Mean Discrepancy* (MMD) is an effective metric to measure the difference between two distributions [9], [15]–[17], [24]. We calculate the difference between source and target domains based on MMD. Specifically, we define

$$\Omega(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t) = \Omega_m(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t) + \Omega_c(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t), \quad (2)$$

where $\Omega_m(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t)$ measures the difference between source and target marginal probability distributions, and $\Omega_c(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t)$ measures the difference between source and target label-conditional probability distributions.

To measure the difference between the source and target marginal distributions $P_s(\mathbf{x}_s)$ and $P_t(\mathbf{x}_t)$ on the selected features, we compute the empirical MMD based on source and target data samples as follows:

$$\Omega_m(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t) = \lambda \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) - \frac{1}{n_t} \sum_{i=1}^{n_t} \phi(\mathbf{x}_{t,i} \odot \boldsymbol{\beta}) \right\|^2, \quad (3)$$

where λ is a trade-off parameter and $\phi(\cdot)$ is a feature transformation function.

Note that reducing the difference in the marginal distribution does not guarantee that the conditional distributions between domains can also be drawn close. Indeed, minimizing the difference between the conditional distributions is crucial for robust distribution adaptation [45]. Therefore, besides the difference between marginal distributions, we further measure the difference between source and target conditional distributions on the selected features. It is difficult to directly measure the difference between $P_s(y_s|\mathbf{x}_s)$ and $P_t(y_t|\mathbf{x}_t)$; thus, based on Bayes' Theorem, we instead measure the difference between $P_s(\mathbf{x}_s|y_s)$ and $P_t(\mathbf{x}_t|y_t)$. Considering that the target data are unlabeled, we use the classifier trained on the source data to generate pseudo labels for the target data. As a result, by using labeled source data and target data with pseudo labels, we can measure the difference between source and target conditional distributions. During the training procedure, we iteratively refine the classifier and pseudo target labels.

The difference between source and target conditional distributions is given as follows:

$$\Omega_c(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t) = \gamma \sum_{k=1}^{K} \left\| \frac{1}{n_s^k} \sum_{i=1}^{n_s^k} \phi(\mathbf{x}_{s,i}^k \odot \boldsymbol{\beta}) - \frac{1}{n_t^k} \sum_{i=1}^{n_t^k} \phi(\mathbf{x}_{t,i}^k \odot \boldsymbol{\beta}) \right\|^2, \quad (4)$$

where γ is a trade-off parameter, k is a label index, $\mathbf{x}_{s,i}^k$ is a source instance of label k, and n_s^k is the number of source instances with label k. Similarly, $\mathbf{x}_{t,i}^k$ is a target instance with pseudo label k, and n_t^k is the number of target instances with pseudo label k.

C. OPTIMIZATION

Problem (1) is difficult to solve because of the discrete parameters $\boldsymbol{\beta}$. However, with the fixed $\boldsymbol{\beta}$, the inner optimization problem involving $\mathbf{w}$, b and $\boldsymbol{\xi}_s$ is a standard SVM problem and can be written as follows:

$$\min_{\mathbf{w}, b, \boldsymbol{\xi}_s} \frac{1}{2}||\mathbf{w}||_2^2 + \frac{1}{2} C \sum_{i=1}^{n_s} \xi_{s,i}^2, \quad \text{s.t. } y_{s,i}(\mathbf{w}^\top(\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) + b) \geq 1 - \xi_{s,i}, \quad i = 1, \ldots, n_s. \quad (5)$$

By introducing non-negative Lagrangian multipliers $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{n_s}]^\top$, we achieve the Lagrangian as

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}_s, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||_2^2 + \frac{1}{2} C \sum_{i=1}^{n_s} \xi_{s,i}^2 + \sum_{i=1}^{n_s} \alpha_i \big(1 - \xi_{s,i} - y_{s,i}(\mathbf{w}^\top(\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) + b)\big). \quad (6)$$

After setting the partial derivatives *w.r.t.* $\mathbf{w}$, b and $\boldsymbol{\xi}_s$ to zero, respectively, we obtain

$$\begin{aligned} &\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}_s, \boldsymbol{\alpha}) = \mathbf{0} \\ &\qquad \Rightarrow \mathbf{w} = \sum_{i=1}^{n_s} \alpha_i y_{s,i} (\mathbf{x}_{s,i} \odot \boldsymbol{\beta}), \\ &\nabla_b \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}_s, \boldsymbol{\alpha}) = \mathbf{0} \Rightarrow \sum_{i=1}^{n_s} \alpha_i y_{s,i} = 0, \\ &\nabla_{\boldsymbol{\xi}_s} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}_s, \boldsymbol{\alpha}) = \mathbf{0} \Rightarrow \boldsymbol{\xi}_s = \frac{1}{C} \boldsymbol{\alpha}. \end{aligned} \quad (7)$$

By substituting the above results into Eq. (6), we obtain the dual form of Problem (5) as

$$\max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2} \left|\left| \sum_{i=1}^{n_s} \alpha_i y_{s,i} (\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) \right|\right|_2^2 - \frac{1}{2C} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbf{1}, \quad (8)$$

where the definition domain $\mathcal{A} = \{\boldsymbol{\alpha} | \sum_{i=1}^{n_s} \alpha_i y_{s,i} = 0, \alpha_i \geq 0, \forall i\}$. By involving $\boldsymbol{\beta}$, we obtain an equivalent problem to Problem (1) as

$$\min_{\boldsymbol{\beta} \in \mathcal{B}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2} \left|\left| \sum_{i=1}^{n_s} \alpha_i y_{s,i} (\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) \right|\right|_2^2 - \frac{1}{2C} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbf{1} + \Omega(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t). \quad (9)$$

For convenience, we define a function $f(\boldsymbol{\beta}, \boldsymbol{\alpha})$ as

$$f(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \frac{1}{2} \left|\left| \sum_{i=1}^{n_s} \alpha_i y_{s,i} (\mathbf{x}_{s,i} \odot \boldsymbol{\beta}) \right|\right|_2^2 + \frac{1}{2C} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{1} - \Omega(\boldsymbol{\beta}, \mathcal{D}_s, \mathcal{D}_t). \quad (10)$$

Thus, Problem (9) can be simplified as

$$\min_{\boldsymbol{\beta} \in \mathcal{B}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -f(\boldsymbol{\beta}, \boldsymbol{\alpha}). \quad (11)$$

According to the minimax inequality [46], we interchange the order of $\min_{\boldsymbol{\beta} \in \mathcal{B}}$ and $\max_{\boldsymbol{\alpha} \in \mathcal{A}}$ to get the lower-bound of the above problem as follows:

$$\max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\beta} \in \mathcal{B}} -f(\boldsymbol{\beta}, \boldsymbol{\alpha}). \quad (12)$$

Finally, we achieve the following minimax problem

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}} \max_{\boldsymbol{\beta} \in \mathcal{B}} f(\boldsymbol{\beta}, \boldsymbol{\alpha}). \quad (13)$$

1) CUTTING PLANE ALGORITHM

By introducing a variable $\theta \in \mathbb{R}$, we reformulate Problem (13) as a quadratically constrained linear programming problem as follows:

$$\min_{\boldsymbol{\alpha}\in\mathcal{A},\theta\in\mathbb{R}} \theta, \quad \text{s.t. } \theta \geq f(\boldsymbol{\beta},\boldsymbol{\alpha}),\ \forall \boldsymbol{\beta}\in\mathcal{B}, \tag{14}$$

which involves $\sum_{i=0}^{B}\binom{d}{i}$ quadratic constraints, thus is difficult to solve.

However, not all the constraints in Problem (14) are active at the optimum, if only a subset of features makes contribution to classification and domain adaptation. Motivated by this, we address Problem (14) using a cutting plane algorithm [47], which is summarized in Algorithm 1. Specifically, at the τ-th iteration, we firstly find the most violated constraint $\boldsymbol{\beta}^{\tau}$ based on $\boldsymbol{\alpha}^{\tau-1}$ by solving the subproblem as follows:

$$\max_{\boldsymbol{\beta}\in\mathcal{B}} f(\boldsymbol{\beta},\boldsymbol{\alpha}^{\tau-1}). \tag{15}$$

After that, we add $\boldsymbol{\beta}^{\tau}$ into the active constraint set $\mathcal{B}^{\tau}$, and then obtain $\boldsymbol{\alpha}^{\tau}$ by solving the subproblem with the constraints in $\mathcal{B}^{\tau}$ as

$$\min_{\boldsymbol{\alpha}\in\mathcal{A},\theta\in\mathbb{R}} \theta, \quad \text{s.t. } \theta \geq f(\boldsymbol{\beta},\boldsymbol{\alpha}),\ \forall \boldsymbol{\beta}\in\mathcal{B}^{\tau}. \tag{16}$$

Next, we discuss the method to solve the subproblems (15) and (16).

Algorithm 1 Feature Selection for Domain Adaptation (FSDA).

Require: $\boldsymbol{\alpha}_0 = C\mathbf{1}$, the active constraint set $\mathcal{B}_0 = \emptyset$

1: Set $\tau = 1$.
2: **repeat**
3: Find the most violated constraint $\boldsymbol{\beta}^{\tau}$ based on $\boldsymbol{\alpha}^{\tau-1}$ by solving the subproblem (15).
4: Update the active constraint set

$$\mathcal{B}^{\tau} = \mathcal{B}^{\tau-1} \cup \{\boldsymbol{\beta}^{\tau}\}.$$

5: Solve $\boldsymbol{\alpha}^{\tau}$ by solving the subproblem (16).
6: Update $\tau = \tau + 1$.
7: **until** Convergence.

2) SOLVING SUBPROBLEM (15)

To avoid too many subscripts, we use $\mathbf{a}[i]$ to represent the i-th element of the vector $\mathbf{a}$. Problem (15) can be rewritten as

$$\begin{aligned}
&\max_{\boldsymbol{\beta}\in\mathcal{B}} f(\boldsymbol{\beta},\boldsymbol{\alpha}^{\tau-1}) \\
&= \max_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{2}\Big|\Big|\sum_{i=1}^{n_s}\alpha_i^{\tau-1}y_{s,i}(\mathbf{x}_{s,i}\odot\boldsymbol{\beta})\Big|\Big|_2^2 - \Omega(\boldsymbol{\beta},\mathcal{D}_s,\mathcal{D}_t) \\
&= \max_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{2}\Big|\Big|\sum_{i=1}^{n_s}\alpha_i^{\tau-1}y_{s,i}(\mathbf{x}_{s,i}\odot\boldsymbol{\beta})\Big|\Big|_2^2 \\
&\quad - \lambda\Big|\Big|\frac{1}{n_s}\sum_{i=1}^{n_s}\phi(\mathbf{x}_{s,i}\odot\boldsymbol{\beta}) - \frac{1}{n_t}\sum_{i=1}^{n_t}\phi(\mathbf{x}_{t,i}\odot\boldsymbol{\beta})\Big|\Big|^2 \\
&\quad - \gamma\sum_{k=1}^{K}\Big|\Big|\frac{1}{n_s^k}\sum_{i=1}^{n_s^k}\phi(\mathbf{x}_{s,i}^k\odot\boldsymbol{\beta}) - \frac{1}{n_t^k}\sum_{i=1}^{n_t^k}\phi(\mathbf{x}_{t,i}^k\odot\boldsymbol{\beta})\Big|\Big|^2.
\end{aligned} \tag{17}$$

In general, this problem is difficult to solve for an arbitrary mapping function $\phi(\cdot)$. Nevertheless, with the identity mapping $\phi(\mathbf{x}) = \mathbf{x}$, this integer programming problem can be handled by a closed-form solution. Specifically, we can define a score vector $\mathbf{z} = [z_1,\ldots,z_d]^{\top}$, where each element corresponds to a feature. The score for the ι-th feature is

$$\begin{aligned}
z_\iota = &\frac{1}{2}\Big(\sum_{i=1}^{n_s}\alpha_i^{\tau-1}y_{s,i}\mathbf{x}_{s,i}[\iota]\Big)^2 \\
&- \lambda\Big(\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{x}_{s,i}[\iota] - \frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{x}_{t,i}[\iota]\Big)^2 \\
&- \gamma\sum_{k=1}^{K}\Big(\frac{1}{n_s^k}\sum_{i=1}^{n_s^k}\mathbf{x}_{s,i}^k[\iota] - \frac{1}{n_t^k}\sum_{i=1}^{n_t^k}\mathbf{x}_{t,i}^k[\iota]\Big)^2.
\end{aligned} \tag{18}$$

As a result, Problem (15) can be rewritten as

$$\max_{\boldsymbol{\beta}\in\mathcal{B}} f(\boldsymbol{\beta},\boldsymbol{\alpha}^{\tau-1}) = \max_{\boldsymbol{\beta}\in\mathcal{B}}\sum_{\iota=1}^{d} z_\iota\beta_\iota, \tag{19}$$

where β_ι is the ι-th element of $\boldsymbol{\beta}$. The domain $\mathcal{B} = \{\boldsymbol{\beta}|\boldsymbol{\beta}\in\{0,1\}^d, ||\boldsymbol{\beta}||_0 \leq B\}$ constrains that there are at most B elements with value 1 in $\boldsymbol{\beta}$, while the remaining elements are all 0. Therefore, to maximize the total score weighted by $\boldsymbol{\beta}$, we can just find B features with the B largest scores, and then assign 1 to their corresponding β's and 0 to the others.

3) SOLVING SUBPROBLEM (16)

For the subproblem (16), we solve its primal form *w.r.t.* the primal variables $\mathbf{w}$. Specifically, at the τ-th iteration, for the feature selection vector $\boldsymbol{\beta}^{\tau}$, let $\tilde{\mathbf{x}}_{s,i}^{\tau} \in \mathbb{R}^B$ be an instance with features selected by $\boldsymbol{\beta}^{\tau}$, *i.e.*, $\tilde{\mathbf{x}}_{s,i}^{\tau} = \mathbf{x}_{s,i}\odot\boldsymbol{\beta}^{\tau}$, and the corresponding parameter vector is represented by $\tilde{\mathbf{w}}_{\tau}\in\mathbb{R}^B$. As a result, the primal form of Subproblem (16) is given by

$$\min_{\tilde{\mathbf{w}},b} \frac{1}{2}(\sum_{\kappa=1}^{\tau}||\tilde{\mathbf{w}}_\kappa||)^2 + \frac{1}{2}C\sum_{i=1}^{n_s}\xi_{s,i}^2, \tag{20}$$

where $\tilde{\mathbf{w}} = [\tilde{\mathbf{w}}_1^{\top},\ldots,\tilde{\mathbf{w}}_\tau^{\top}]^{\top}$, and $\xi_{s,i} = \max\big(1 - y_{s,i}(\sum_{\kappa=1}^{\tau}\tilde{\mathbf{w}}_\kappa^{\top}\tilde{\mathbf{x}}_{s,i}^{\kappa} - b), 0\big)$. Problem (20) can be solved by a dual proximal gradient method [48].

4) COMPUTATIONAL COMPLEXITY ANALYSIS

Here we provide the computational complexity of Algorithm 1. At each iteration, the complexity for solving Subproblem (15) is $O(n_s d + n_t d + d\log B)$, and the complexity for solving Subproblem (16) is $O(n_s d)$. Therefore, the complexity of each iteration is $O(n_s d + n_t d + d\log B)$.

TABLE 2. Statistical information of the data sets.

Domain	#Examples	#Features	#Classes
Amazon (A)	2,817	4,096	31
DSLR (D)	498	4,096	31
Webcam (W)	795	4,096	31
Amazon (A)	958	1,000	10
Caltech (C)	1,299	1,000	10
DSLR (D)	157	1,000	10
Webcam (W)	295	1,000	10
Caltech256 (C)	3,847	4,096	40
ImageNet (I)	4,000	4,096	40
Sun(S)	2,626	4,096	40
MSRC (M)	1,269	240	6
VOC2007 (V)	1,530	240	6

D. PREDICTION

At the τ-th iteration, when Algorithm 1 stops, we have τ feature selection vectors $\{\boldsymbol{\beta}^{\kappa}\}_{\kappa=1}^{\tau}$ and the parameters $\{\tilde{\mathbf{w}}_{\kappa}\}_{\kappa=1}^{\tau}$ and b. At last, we can make predictions for the unlabeled target data by

$$\hat{y}_{t,i} = \sum_{\kappa=1}^{\tau} \tilde{\mathbf{w}}_{\kappa}^{\top} \tilde{\mathbf{x}}_{t,i}^{\tau} + b. \tag{21}$$

E. CONVERGENCE

Note that Problem (14) is a convex problem. According to [49], our proposed algorithm can achieve a global solution to Problem (14) within a finite number of steps.

Theorem 1: Assume that in each iteration, the subproblems (15) and (16) can be solved, FSDA stops after a finite number of steps with a global solution of Problem (14).

Please refer to [49] for the proof.

IV. EXPERIMENTS

A. DATA SETS

We perform experiments on several visual recognition data sets: Office, Cross-Dataset Testbed, MSRC + VOC2007. TABLE 2 presents statistical information of the data sets.

- Office
  The Office data set contains images of 31 categories over three object domains [50]: Amazon (A) contains images downloaded from the Amazon website, DSLR (D) contains high-resolution images obtained from a digital SLR camera, and Webcam (W) contains low-resolution images taken from a web camera. The data set contains a total of 4, 110 images with a minimum of 7 and a maximum of 100 samples per category. We use the publicly available DeCAF$_6$ [51] features with 4, 096 dimensions, which are obtained from a convolutional neural network trained on ImageNet.
- Office-Caltech
  The Caltech-256 (C) data set [52] contains 256 categories, including 10 categories overlapped with the Office data set. We use the images with these 10 categories in both Office and Caltech-256 data sets to construct domain adaptation problems with four domains, *i.e.*, A, C, D, and W. The DeCAF$_8$ features with 1, 000 dimensions is used to conduct experiments.
- Cross-Dataset Testbed
  The Cross-Dataset Testbed data set [53] contains 10,473 images with 40 categories. This data set is collected from three domains. Caltech256 (C) contains 256 categories with a minimum of 80 and a maximum of 827 images per category, ImageNet (I) contains around 21,000 object classes organized according to the Wordnet hierarchy, Sun (S) contains a total of 142,165 pictures and it was created as a comprehensive collection of annotated images covering a large variety of environmental scenes, places and objects. These three domains share 40 classes and we use the publicly available DeCAF$_7$ features with 4,096 dimensions [31].
- MSRC + VOC2007
  The MSRC (M) data set [54], which is provided by Microsoft Research Cambridge, contains a set of digital photographs grouped into 22 categories spanning over objects and scenes. The VOC2007 (V) data set [55] is collected from Flickr and contains 20 object categories. These two data sets follow significantly different distributions with 240 feature dimensions, and they share 6 semantic classes: "aeroplane", "bicycle", "bird", "car", "cow", and "sheep".

B. BASELINE METHODS

- **SVM**. We perform SVM [56] on labeled source data to train a classifier, and then use it to make predictions for unlabeled target data. SVM is a straightforward method without considering domain discrepancy.
- **TCA**. Transfer component analysis (TCA) [9] finds common latent features that have the similar marginal distribution across the source and target domains.
- **GFK**. Geodesic flow kernel (GFK) [27] learns a new feature representation based on the proposed geodesic flow kernel.
- **f-MMD**. Feature selection with MMD (f-MMD) [37] distinguishes variant and invariant features across source and target data based on MMD.
- **ITL**. Information-theoretical learning (ITL) [57] jointly learns a domain-invariant representation and optimizes information-theoretical metrics for classification.
- **JDA**. Joint distribution adaptation (JDA) [15] adopts MMD to measure the difference between both marginal and conditional distributions, and a new feature representation is constructed to train the classifier.
- **TJM**. Transfer joint matching (TJM) [16] minimizes the domain difference by jointly re-weighting source data and learning a new feature representation.
- **CORAL**. Correlation alignment (CORAL) [31] aligns the input feature distributions of the source and target domains by exploring their second-order statistics.

TABLE 3. Accuracies (%) on the Office data set.

Task	SVM	TCA	GFK	f-MMD	ITL	JDA	TJM	CORAL	JGSA	MCTL	LDADA	FGM	FSDA
A→D	46.79	45.98	45.47	50.00	54.02	53.41	52.01	54.15	52.01	58.84	**61.21**	56.83	60.84
A→W	39.12	45.91	34.50	51.07	51.95	52.08	50.82	48.25	53.08	53.08	**55.14**	50.57	53.08
D→A	34.61	35.89	36.69	35.96	36.85	40.89	40.82	43.58	41.21	42.07	44.29	39.94	**44.59**
D→W	81.26	89.94	78.08	91.45	93.58	89.81	90.69	**95.75**	88.81	94.59	86.88	94.21	95.22
W→A	29.22	32.77	28.54	37.17	38.05	40.11	36.00	42.17	40.33	40.18	41.28	40.89	**43.70**
W→D	90.56	94.18	89.09	92.77	99.00	93.98	93.78	99.04	92.17	99.40	90.78	99.20	**99.60**
Average	53.59	57.44	52.06	59.74	62.24	61.71	60.69	63.82	61.27	64.70	63.36	63.61	**66.17**

TABLE 4. Accuracies (%) on the Office-Caltech data set.

Task	SVM	TCA	GFK	f-MMD	ITL	JDA	TJM	CORAL	JGSA	MCTL	LDADA	FGM	FSDA
A→C	63.43	78.98	65.27	64.82	81.68	81.06	76.67	75.97	86.91	85.45	83.07	84.83	**88.30**
A→D	66.88	73.25	71.05	74.52	78.34	80.25	80.25	79.71	**88.54**	84.08	85.07	85.35	87.90
A→W	59.32	72.88	73.65	58.64	69.15	83.39	81.69	77.53	**93.22**	80.00	90.44	78.98	82.71
C→A	64.93	86.74	70.51	80.06	90.08	90.19	87.37	87.45	92.28	91.96	90.98	92.48	**92.80**
C→D	62.42	74.52	61.69	75.80	71.34	80.89	80.89	75.89	89.17	85.35	84.40	89.81	**91.08**
C→W	62.03	76.61	57.60	74.24	75.93	85.08	82.71	75.51	**93.90**	83.39	85.21	86.10	88.81
D→A	79.65	82.88	68.10	78.71	85.49	79.96	78.50	79.90	**90.29**	86.64	80.85	87.37	89.25
D→C	55.04	69.52	55.34	73.29	77.06	70.44	67.44	66.75	**81.99**	77.44	68.90	78.14	79.98
D→W	86.78	95.59	87.34	93.22	97.63	97.29	95.93	**98.41**	89.15	97.97	83.94	97.63	97.97
W→A	75.05	84.34	70.20	71.82	79.65	82.46	83.30	84.57	**91.44**	82.99	90.63	85.49	87.58
W→C	53.27	69.13	51.26	68.44	73.13	66.36	65.36	67.63	**84.06**	75.83	70.66	77.98	80.06
W→D	91.08	92.99	81.00	98.09	97.45	**99.36**	96.82	96.50	96.82	98.09	92.34	**99.36**	**99.36**
Average	68.32	79.79	67.75	75.97	81.41	83.06	81.41	80.48	**89.81**	88.65	83.87	86.96	88.82

- **JGSA**. Joint geometrical and statistical alignment (JGSA) [17] learns two coupled projections to map the source and target data into a shared subspace, in which the distributional and geometrical divergences between domains are reduced.
- **FGM**. Feature generating machine (FGM) [48], [49] is a SVM-based feature selection method on single domain. We perform FGM on labeled source data to select features and train a classifier, and then use it to make predictions for the unlabeled target data. We use FGM to evaluate the effectiveness of the selected features on source data for the target classification task.
- **MCTL**. Manifold Criterion guided Transfer Learning [32] exploits the locality structure to learn an intermediate domain for reducing both global and local discrepancies.
- **LDADA**. LDA-inspired Domain Adaptation [33] extends linear discriminant analysis to learn classs-pecific linear projections.

For the representation learning methods, *i.e.*, TCA, f-MMD, ITL, JDA, TJM, CORAL, JGSA, MCTL and LDADA, SVM classifiers are trained on the new representations of labeled source samples. For the metric learning method GFK, we follow the approach in the original paper to adopt the one-nearest-neighbor classifier.

C. EXPERIMENTAL SETTING

For the Office data set, we follow the experimental setting in [31] in which all 31 classes are used. As a result, we consider all the combinations of source and target domain pairs and construct $3 \times 2 = 6$ tasks, *i.e.*, A → D, A → W, D → A, D → W, W → A and W → D. Similar to the setting of the Office data set, we generate $4 \times 3 = 12$ tasks on the Office-Caltech data sets, *i.e.*, A → C, A → D, A → W, C → A, C → D, C → W, D → A, D → C, D → W, W → A, W → C, W → D. For the Cross-Dataset Testbed data set, we generate $3 \times 2 = 6$ tasks, *i.e.*, C → I, C → S, I → C, I → S, S → C, S → I. For the MSRC + VOC2007 data sets, we use data with the common classes between MSRC and VOC2007 to construct learning tasks, including M → V and V → M.

For simplicity and fair comparison, we adopt linear SVM with the trade-off parameter $C = 1$. For FSDA, we empirically set $B = 100$ for the Office and Cross-Data Testbed data sets, and $B = 8$ for the MSRC + VOC2007 data sets.

D. RESULTS AND DISCUSSIONS

In this experiment, we evaluate the performance of all the compared algorithms in terms of the classification accuracy. TABLES 3, 4, 5 and 6 present the results on all the used data sets. We discuss several interesting observations as follows.

- The base classifier SVM obtains the lowest accuracies on most tasks. Actually, the predictions of SVM are the pseudo labels used in Eq. (4). This observation indicates the pseudo labels that are obtained from the classifier trained on source data are not accurate, which demonstrates the necessity of domain adaptation.

TABLE 5. Accuracies (%) on the Cross-Dataset Testbed data set.

Task	SVM	TCA	GFK	f-MMD	ITL	JDA	TJM	CORAL	JGSA	MCTL	LDADA	FGM	FSDA
C→I	58.38	59.20	51.00	58.17	63.55	64.68	58.45	61.20	65.55	66.22	**70.72**	65.50	66.47
C→S	18.66	20.98	16.92	23.31	23.00	21.10	19.23	20.38	23.23	22.05	22.81	23.12	**23.69**
I→C	63.58	62.98	58.50	66.55	71.15	75.23	65.53	68.95	74.34	75.44	**77.04**	75.23	75.75
I→S	20.41	22.96	19.30	23.57	23.50	22.66	21.17	23.26	24.68	24.30	22.74	24.60	**24.87**
S→C	19.29	25.16	17.49	25.99	26.88	27.71	27.68	23.49	22.67	26.75	23.31	28.85	**30.13**
S→I	22.95	23.33	15.14	25.05	26.75	27.55	25.12	23.20	23.28	27.70	28.75	29.88	**31.20**
Average	33.88	35.77	29.73	37.11	39.14	39.82	36.20	36.75	38.96	40.41	40.90	41.20	**42.02**

TABLE 6. Accuracies (%) on the MSRC + VOC2007 data sets.

Task	SVM	TCA	GFK	f-MMD	ITL	JDA	TJM	CORAL	JGSA	FGM	FSDA
M→V	31.50	33.27	31.32	30.26	33.79	34.18	34.38	33.08	33.59	33.53	**34.44**
V→M	41.92	42.24	38.84	49.96	49.17	51.69	49.49	42.45	50.20	53.66	**54.77**
Average	36.71	37.75	35.08	40.11	41.48	42.94	41.93	37.77	41.90	43.60	**44.61**

- f-MMD and FGM achieve better performance than SVM, which indicates that there exists a feature subset that is beneficial for classification.
- FSDA, ITL, JDA, TJM, CORAL, JGSA, MCTL and LDADA outperform SVM on most tasks, which indicates that reducing domain discrepancy is an effective method to achieve promising performance in domain adaptation.
- FSDA outperforms f-MMD, which is also a feature selection method for domain adaptation. This indicates that leveraging label information, which is adopted in FSDA but not in f-MMD, can help find more discriminative features for classification.
- FSDA achieves the best or highly comparative performance on all the tasks, which clearly demonstrates that FSDA is able to find informative features that enjoy a strong discriminative ability and can also reduce the domain difference.

E. EFFECTIVENESS ANALYSIS

In this part, we empirically study if the features selected by FSDA can reduce the domain difference between domains.

1) Firstly, we take the task W → D as an example to evaluate the distribution matching effect of FSDA. For each one-vs-rest binary classification problem in W → D, we employ MMD to evaluate the domain differences on the features selected by FGM and FSDA, respectively. The domain differences on the selected features are measured by Eq. (2), in which the feature selection vector $\boldsymbol{\beta}$ is obtained by FGM and FSDA, respectively. FIGURE 1 shows the domain differences on the selected features of all the one-vs-rest binary classification problems on W → D, where the results are scaled so that the minimal value is scaled to be 1. Furthermore, we average the domain differences on the selected features over all the one-vs-rest binary classification problems in one domain adaptation task, and plot the results in FIGURE 2. Compared to FGM, FSDA achieves the lower domain differences between the source and target domains, which means that FSDA is able to find more similar properties between two domains, and the features selected by FSDA is able to reduce the domain discrepancy.

TABLE 7. Results of a representative task on the A→D task on the Office data set.

	SVM	f-MMD	FSDA
# features	4096	554	700
Domain difference	31.82	1.00	5.92
Accuracy (%)	72.73	77.27	**79.55**

TABLE 8. Results of a representative task on the C→S task.

	SVM	f-MMD	FSDA
# features	4096	1585	300
Domain difference	13.99	2.32	1.00
Accuracy (%)	79.65	82.30	**84.96**

2) In addition, we take one-vs-one classification problems as examples to analyze the effectiveness of our proposed method by comparing FSDA with SVM and f-MMD. TABLES 7, 8 and 9 show the results of three representative tasks, which are from the tasks A→D, C→S and M→V, respectively. In the tables, "# features" is the number of the selected features for classification, the domain difference is calculated based on Eq. (2), and the accuracy is evaluated on the one-vs-one classification problem. We draw some observations as follows.
   - SVM uses all the features and has the largest MMD values and the lowest performance, since it ignores the difference between the source and target domains.

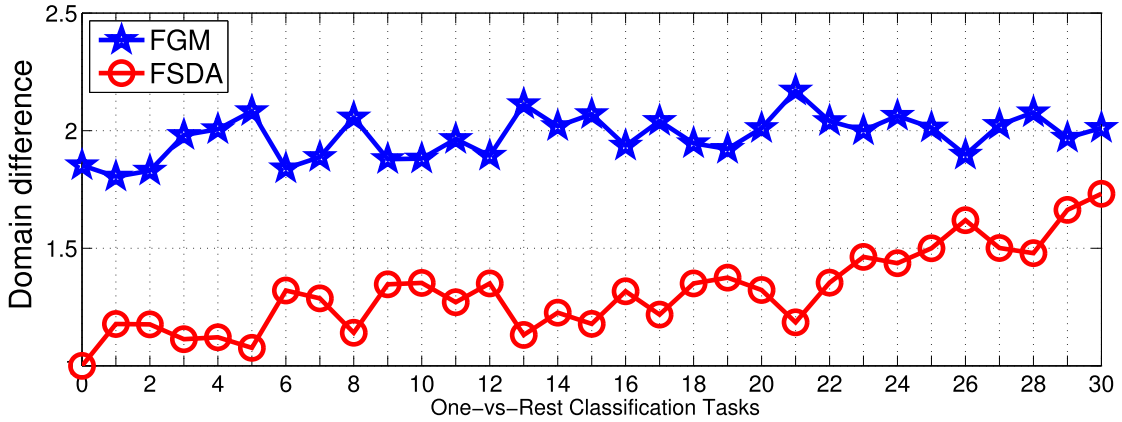


FIGURE 1. Domain difference of each binary task learned by FGM and FSDA on the task W → D on the office data set.

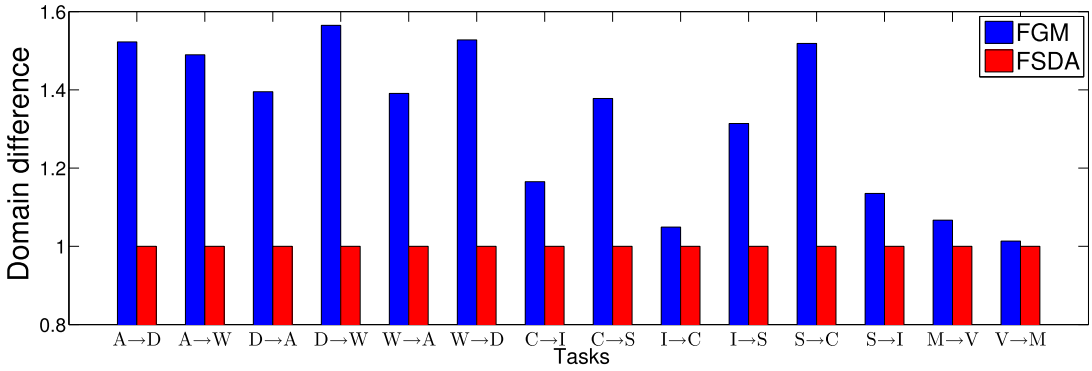


FIGURE 2. Domain difference learned by FGM and FSDA on several the learning tasks.

TABLE 9. Results of a representative task on the M→V task.

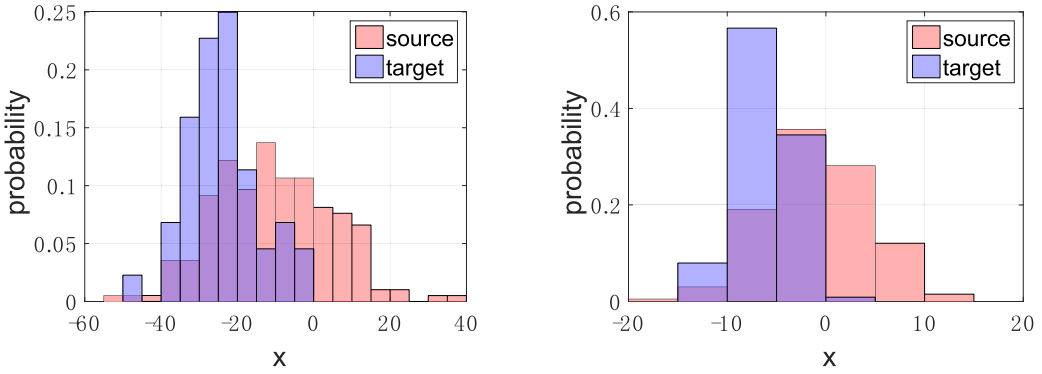
	SVM	f-MMD	FSDA
# features	256	68	56
Domain difference	18.88	1.45	1.00
Accuracy (%)	68.06	74.57	**76.30**

- f-MMD has the comparable domain difference compared to FSDA. However, FSDA obtains better performance than f-MMD, which demonstrates that FSDA can find more discriminative features for classification.
- FSDA achieves the highest accuracies, which verifies that FSDA is able to find discriminative features that can jointly reduce the domain difference and the classification loss.

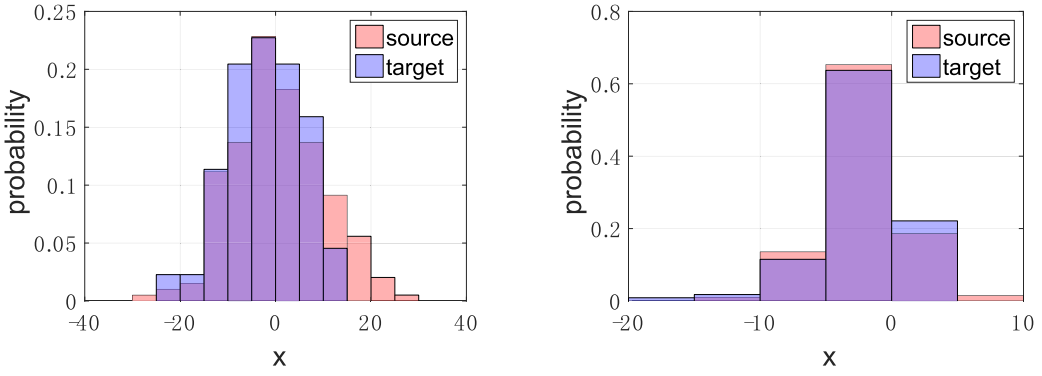
3) Moreover, for tasks A→D and C→S, we select one of the most discriminative features in SVM and FSDA on the source and target data to show the data distributions. FIGURE 3 detailedly presents the data distrib utions on the two learning tasks. FIGURE 3(a) and 3(b) show the distributions on one of the most discriminative features on task A→D in SVM and FSDA, respectively. The discriminative feature of SVM distributes more different than the discriminative feature of FSDA in the source and target domains, which indicates that FSDA is able to find discriminative features that are shared by the source and target domains. Similar conclusions can be drawn from the other figures of FIGURE 3.

F. SENSITIVITY STUDY OF PARAMETERS

We use I → C as an example task to evaluate the sensitivity of parameters B, λ and γ. Specifically, we vary $B \in \{1, 5, 10, 50, 100, 150, 200\}$ and search λ and γ in the range $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$. FIGURE 4 presents the results. To make a clearer view, we also plot the second best results, which are achieved by FGM on this task. From FIGURE 4(a), we observe that the accuracy increases as B increases until B reaches the value 100. When $B > 100$, some noisy features may be selected, resulting in the decrease of the performance. FIGURE 4(b) shows that FSDA outperforms FGM with different values of λ and reaches the best result when $\lambda = 2^6$, which indicates that the performance of FSDA is not sensitive to λ. FIGURE 4(c) shows that FSDA performs better than FGM when $\gamma > 2^2$ and obtains the best result when $\gamma = 2^8$. From the comparison to FGM, we observe that

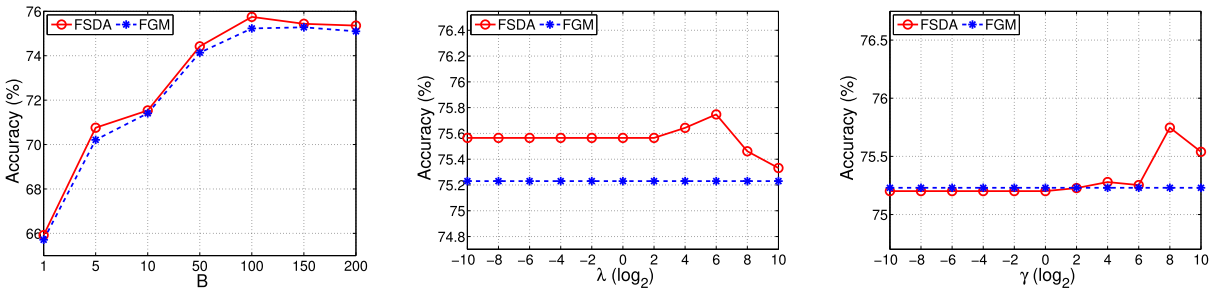


(a) A → D : SVM (c) C → S : SVM



(b) A → D : FSDA (d) C → S : FSDA

FIGURE 3. Data distributions of the three representative tasks. (a) and (b) show the distributions on one of the most discriminative features on task A→D on the Office data set in SVM and FSDA, respectively. (c) and (d) show the distributions on one of the most discriminative features on task C→S in SVM and FSDA, respectively. (a) A → D: SVM. (b) A → D: FSDA. (c) C → S: SVM. (d) C → S: FSDA.



(a) Results *w.r.t.* different values of B (b) Results *w.r.t.* different values of λ (c) Results *w.r.t.* different values of γ

FIGURE 4. Sensitivity of the parameters B, λ and γ on task I→C. (a) Results *w.r.t.* different values of B. (b) Results *w.r.t.* different values of λ. (c) Results *w.r.t.* different values of γ.

considering the discrepancies of the marginal and conditional distributions is rather important.

G. CONVERGENCE RESULTS

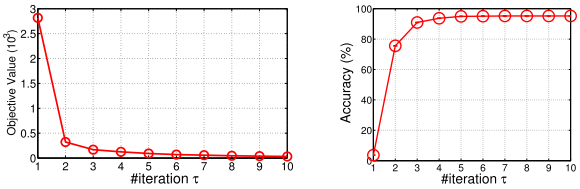
In FIGURE 5, we take two tasks as examples to investigate the convergence property of the proposed method in terms of the objective value and accuracy. From FIGURE 5(a) we observe that as the iteration increases, the objective value converges to a stable value. Similarly, FIGURE 5(b) shows that the classification accuracy gets better and converges in a few iterations.

H. RUNNING TIME RESULTS

In TABLE 10, we take the task D → W on the Office data set as an example to evaluate the efficiency of the proposed method. The experiments are conducted on a workstation with Xeon 3.40 GHz CPU and 16 GB of RAM. SVM simply trains a classifier on labeled source data without consider-

TABLE 10. Running time results (second) on the task D → W on the Office data set.

Task	SVM	TCA	GFK	f-MMD	ITL	JDA	TJM	CORAL	JGSA	MCTL	LDADA	FGM	FSDA
Time	4.96	11.39	260.77	12.64	2288.11	70.34	29.37	180.76	105.14	59.45	12.15	29.78	50.53



(a) (b)

FIGURE 5. (a) Convergence results of the objective value on A → D on the Office data set. (b) Convergence results of the accuracy on D → W on the Office data set.

ing distribution matching, thus achieves the shortest running time. Compared with the other domain adaptation methods, FSDA has the modest efficiency and achieves the best classification performance.

V. CONCLUSION

In this paper, we propose to find informative features for the unsupervised domain adaptation problem. Our proposed methods aim to find structured multiple outputs, in which a vector for selecting a subset of features that can jointly reduce the domain discrepancy and eliminating noisy features, and a classifier for predicting on the selected features. To solve the resultant optimization problems, we develop a cutting plane algorithm to iteratively select features and refine the classifier. We conduct extensive experiments on real-world data sets. The results demonstrate that the proposed methods can find features to reduce the domain discrepancy and outperforms the compared state-of-the-art algorithms.

In the future, we plan to extend our work towards three directions. First, compared with the cutting-plane algorithm used in this paper, the stochastic mirror descent method can solve the minimax problem with less computation cost per iteration [58]–[60]. Therefore, we plan to apply the stochastic mirror descent method to solve our minimax problem in the future. Second, the recent researches regarding the optimal margin distribution machine have shown its great generalization ability [61]–[63]. We will investigate how to apply the optimal margin distribution machine to address the problem of domain adaptation in the forthcoming researches. Third, compared to the existing method that performing a traditional classifier on extracted features, training a deep neural network using an end-to-end paradigm usually achieves better performance. We will explore how to learn informative features in a modern deep network in an end-to-end paradigm to further improve the performance.

ACKNOWLEDGMENT

(Feng Sun, Hanrui Wu, Zhihang Luo, and Wenwen Gu contributed equally to this work.)

REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.

[3] 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 2014.

[4] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2456–2464.

[5] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Proc. Assoc. Adv. Artif. Intell.*, 2014, pp. 2213–2220.

[6] Z. Chen and W. Zhang, "Domain adaptation with topical correspondence learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–7.

[7] J. T. Zhou, S. J. Pan, I. W. Tsang, and S.-S. Ho, "Transfer learning for cross-language text categorization through active correspondences construction," in *Proc. Assoc. Adv. Artif. Intell.*, 2016, pp. 2400–2406.

[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 *Proc. Assoc. Adv. Artif. Intell.*, 2007, pp. 1108–1113.

[9] 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.

[10] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual event recognition in videos by learning from Web data," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1959–1966.

[11] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from Web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.

[12] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, "Distant domain transfer learning," in *Proc. Assoc. Adv. Artif. Intell.*, 2017, pp. 2604–2610.

[13] T. M. H. Hsu, W. Y. Chen, C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. Frank Wang, "Unsupervised domain adaptation with imbalanced cross-domain data," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4121–4129.

[14] Y.-H. H. Tsai, C.-A. Hou, W.-Y. Chen, Y.-R. Yeh, and Y.-C. F. Wang, "Domain-constraint transfer coding for imbalanced unsupervised domain adaptation," in *Proc. Assoc. Adv. Artif. Intell.*, 2016, pp. 3597–3603.

[15] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[16] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1410–1417.

[17] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1859–1867.

[18] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

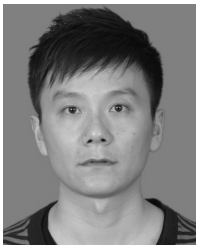
[19] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 513–520.

[20] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 408–415.

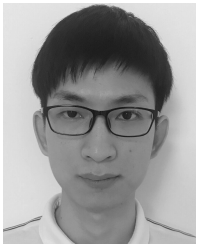
[21] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[22] E. Eaton and M. desJardins, "Selective transfer between learning tasks using task-based boosting," in *Proc. Assoc. Adv. Artif. Intell.*, 2011, pp. 1–6.

[23] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Jun. 2013.
[24] Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5081–5090.
[25] Y. Yan, W. Li, M. Ng, M. Tan, H. Wu, H. Min, and Q. Wu, "Learning discriminative correlation subspace for heterogeneous domain adaptation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 3252–3258.
[26] Y. Yan, W. Li, H. Wu, H. Min, M. Tan, and Q. Wu, "Semi-supervised optimal transport for heterogeneous domain adaptation," in *Proc. IJCAI*, 2018, pp. 2969–2975.
[27] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
[28] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 601–608.
[29] W.-S. Chu, F. De la Torre, and J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3515–3522.
[30] W.-S. Chu, F. De la 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.
[31] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. Assoc. Adv. Artif. Intell.*, 2016, pp. 1–8.
[32] 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.*, to be published.
[33] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. van den Hengel, "An embarrassingly simple approach to visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3403–3417, Jun. 2018.
[34] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.
[35] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 67–93, Mar. 2002.
[36] Z. Ding, S. Li, M. Shao, and Y. Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 37–52.
[37] S. Uguroglu and J. Carbonell, "Feature selection for transfer learning," in *Proc. Mach. Learn. Knowl. Discovery Databases*, 2011, pp. 430–442.
[38] L. Gautheron, I. Redko, and C. Lartizien, "Feature selection for unsupervised domain adaptation using optimal transport," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML-PKDD)*, 2018, pp. 759–776.
[39] C. Zhang, C. Liang, L. Li, J. Liu, Q. Huang, and Q. Tian, "Fine-grained image classification via low-rank sparse coding with general and class-specific codebooks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1550–1559, Jul. 2017.
[40] Z. Yuan, T. Lu, and C. L. Tan, "Learning discriminated and correlated patches for multi-view object detection using sparse coding," *Pattern Recognit.*, vol. 69, pp. 26–38, Sep. 2017.
[41] K. Nai, Z. Li, G. Li, and S. Wang, "Robust object tracking via local sparse appearance model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4958–4970, Jun. 2018.
[42] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[43] M. Tan, Z. Hu, Y. Yan, J. Cao, D. Gong, and W. Qingyao, "Learning sparse PCA with stabilized ADMM method on stiefel manifold," *IEEE Trans. Knowl. Data Eng.*, to be published.
[44] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2006.
[45] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, "A two-stage weighting framework for multi-source domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 505–513.
[46] S.-J. Kim and S. Boyd, "A minimax theorem with applications to machine learning, signal processing, and finance," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1344–1367, 2008.
[47] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
[48] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learn. Res.*, vol. 15, pp. 1371–1429, Apr. 2014.
[49] M. Tan, L. Wang, and I. W. Tsang, "Learning sparse svm for feature selection on very high dimensional datasets," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1047–1054.
[50] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.
[51] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 32. Jun. 2014, pp. 647–655.
[52] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep., 2007.
[53] T. Tommasi and T. Tuytelaars, "A testbed for cross-dataset analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 18–31.
[54] M. Everingham, A. Zisserman, C. K. I. Williams, and L. van Gool. *The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results*. Accessed: 2006. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf
[55] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. Accessed: 2007. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[56] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.
[57] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1275–1282.
[58] G. Lan, A. Nemirovski, and A. Shapiro, "Validation analysis of mirror descent stochastic approximation method," *Math. Program.*, vol. 134, no. 2, pp. 425–458, 2012.
[59] A. Nedic and S. Lee, "On stochastic subgradient mirror-descent algorithm with weighted averaging," *SIAM J. Optim.*, vol. 24, no. 1, pp. 84–107, 2014.
[60] C. D. Dang and G. Lan, "Stochastic block mirror descent methods for nonsmooth and stochastic optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 856–881, 2015.
[61] T. Zhang and Z.-H. Zhou, "Large margin distribution machine," in *Proc. SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 313–322.
[62] T. Zhang and Z.-H. Zhou, "Multi-class optimal margin distribution machine," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 4063–4071.
[63] T. Zhang and Z.-H. Zhou, "Optimal margin distribution machine," *IEEE Trans. Knowl. Data Eng.*, to be published.



FENG SUN is currently pursuing the Ph.D. degree with the School of Marxism, South China University of Technology, China. His current research interests include artificial intelligence and information systems.



HANRUI WU received the bachelor's degree from the School of Software Engineering, South China University of Technology, Guangzhou, China, in 2013, where he is currently pursuing the Ph.D. degree with the School of Software Engineering. His research interests include transfer learning and zero-shot learning.

ZHIHANG LUO received the B.S. degree in communication engineering from Nanchang University, in 2004, and the M.S. degree from The Hongkong University of Science and Technology, in 2016. Her research interests include machine learning and data mining.

WENWEN GU received the B.S. degree in accounting from La Trobe University, Australia, in 2014. Her research interests include machine learning and data mining.

YUGUANG YAN received the B.S. and Ph.D. degrees from the School of Software Engineering, South China University of Technology, China, in 2013 and 2019, respectively. His current research interests include transfer learning, optimal transport, and medical computing.

QING DU received the B.S. degree in computer science and technology, the master's degree in computer application, and the Ph.D. degree in computer application from the South China University of Technology, in 2002, 2005, and 2014, respectively, where she is currently an Associate Professor with the School of Software Engineering. Her research interests include information retrieval, recommendation systems, natural language processing, and deep learning.