Boosted Multifeature Learning for Cross-Domain Transfer

XIAOSHAN YANG, TIANZHU ZHANG, and CHANGSHENG XU, Institute of Automation,
Chinese Academy of Sciences and China-Singapore Institute of Digital Media
MING-HSUAN YANG, University of California, Merced

Conventional learning algorithm assumes that the training data and test data share a common distribution. However, this assumption will greatly hinder the practical application of the learned model for cross-domain data analysis in multimedia. To deal with this issue, transfer learning based technology should be adopted. As a typical version of transfer learning, domain adaption has been extensively studied recently due to its theoretical value and practical interest. In this article, we propose a boosted multifeature learning (BMFL) approach to iteratively learn multiple representations within a boosting procedure for unsupervised domain adaption. The proposed BMFL method has a number of properties. (1) It reuses all instances with different weights assigned by the previous boosting iteration and avoids discarding labeled instances as in conventional methods. (2) It models the instance weight distribution effectively by considering the classification error and the domain similarity, which facilitates learning new feature representation to correct the previously misclassified instances. (3) It learns multiple different feature representations to effectively bridge the source and target domains. We evaluate the BMFL by comparing its performance on three applications: image classification, sentiment classification and spam filtering. Extensive experimental results demonstrate that the proposed BMFL algorithm performs favorably against state-of-the-art domain adaption methods.

Categories and Subject Descriptors: H.4.3 [**Information Systems Applications**]: Communications Applications—*Information browsers*; I.4.8 [**Image Processing and Computer Vision**]: Feature Measurement—*Feature representation*; I.5.4 [**Pattern Recognition**]: Applications—*Text processing*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Domain adaptation, multifeature, boosting, denoising auto-encoder

ACM Reference Format:
Xiaoshan Yang, Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. 2015. Boosted multifeature learning for cross-domain transfer. ACM Trans. Multimedia Comput. Commun. Appl. 11, 3, Article 35 (January 2015), 18 pages.
DOI: http://dx.doi.org/10.1145/2700286

1. INTRODUCTION

With the recent boom of smart phones, digital cameras, and social media sites (e.g., Flickr, YouTube, and Facebook), it is convenient for people to capture and share social media data online. As a result, a large amount of user-contributed media data (e.g., images, videos, and texts) has been generated. These media data contain useful

---

This work is supported in part by the National Program on Key Basic Research Project (973 Program, Project No. 2012CB316304), and National Natural Science Foundation of China (61225009, 61303173). This work is also supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme Office, and supported by National Natural Science Foundation of China (61373122).

Authors' addresses: X. Yang, T. Zhang, and C. Xu (corresponding author), National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China, and China-Singapore Institute of Digital Media, Singapore 119613, Singapore; email: xiaoshan.yang@nlpr.ia.ac.cn, tzzhang10@gmail.com, csxu@nlpr.ia.ac.cn; M.-H. Yang, Department of Electrical Engineering and Computer Science, University of California at Merced, CA 95334; email: mhyang@ucmerced.edu.

DOI: http://dx.doi.org/10.1145/2700286

information and have been adopted for many promising applications. For example, media data on Flickr can predict the winner of the 2008 United Sates president election, monitor the product distribution in the world, and provide successful prediction of product sales [Jin et al. 2010]. Facial expressions in social photos are explored to measure public opinion during the election [Ma and Luo 2013]. In several studies [Bao et al. 2013; Zaharieva et al. 2013; Petkos et al. 2012; Brenner and Izquierdo 2012; Wang et al. 2012; Orlando et al. 2013; Reuter and Cimiano 2012; Liu and Huet 2013; Qian et al. 2014a, 2014b; Yang et al. 2014; Zhang and Xu 2014], social media data are adopted for social event detection and classification. Most of the existing applications use the metadata, such as time, location and descriptions, as features. For example, in Jin et al. [2010], the number of photos uploaded in a fixed time duration is used to predict the election winner. In Roy et al. [2012], only text descriptions of the video are used for recommendation. These metadata are easy to be extracted. However, they may be missed in some data samples and cannot be obtained as features. To deal with this issue, many methods [Snoek et al. 2006; Qi et al. 2012; Effelsberg 2013; Yang et al. 2013] have been proposed to learn effective features to represent the semantic content of the data.

Based on these features, most conventional methods assume that the training and test data are drawn from the same distribution, thus the learned model can be directly applied to the test data. However, it is a common scenario for cross-domain data analysis in multimedia that the training and test distributions differ significantly and it is extremely difficult, if possible, to generalize from limited training data [Pan and Yang 2010]. As a result, this assumption often does not hold and it greatly hinders many real-world applications. In image classification, changes in the camera, image resolution, lighting, background, viewpoint, and post-processing will cause visual domain shift, such as when shifting from typical object category datasets mined from internet search engines to images captured in real-world surroundings, for instance, by a mobile robot [Saenko et al. 2010]. In sentiment classification, a sentiment analysis model that is learned on book reviews does not perform well on kitchen appliance reviews if applied directly [Blitzer et al. 2007]. In spam filtering, we want to recognize spams by using the trained classifier. The challenge is that the distributions among various users are different. Besides, the spam emails always change their information overtime. For these scenarios, the main problem is how to transfer the learned model from training (source) to test (target) instances when they follow different distributions.

Domain adaptation, as one of the transfer learning methods, mainly focuses on the above mentioned distribution mismatch problem between training and test data [Daumé 2007; Liu et al. 2012; Pan and Yang 2010]. Due to its theoretical value and practical interest, domain adaptation has been extensively studied in recent years. Existing domain adaptation methods can be categorized into two groups. One is the semi-supervised domain adaptation where a small subset of labeled instances in the target domain can be used for learning [Dai et al. 2007; Yao and Doretto 2010; Duan et al. 2010; Al-Stouhi and Reddy 2011]. The other one is the unsupervised domain adaptation where only unlabeled instances of the target domain are available for learning [Blitzer et al. 2006, 2007; Gopalan et al. 2011; Glorot et al. 2011; Chen et al. 2012; Gong et al. 2012; Habrard et al. 2013; Gong et al. 2013; Ni et al. 2013]. In both semisupervised and unsupervised domain adaptation methods, the key challenge is still the distribution mismatch between the source domain and the target domain. To tackle this problem, a number of approaches have been proposed to reduce the domain difference which are based on instance sub-sampling or re-weighting [Huang et al. 2006, 2007; Dai et al. 2007; Sugiyama et al. 2009; Yao and Doretto 2010; Al-Stouhi and Reddy 2011; Chen et al. 2011; Habrard et al. 2013; Gong et al. 2013], and joint feature representation learning [Blitzer et al. 2006, 2011; Ben-David et al. 2006; Duan et al.

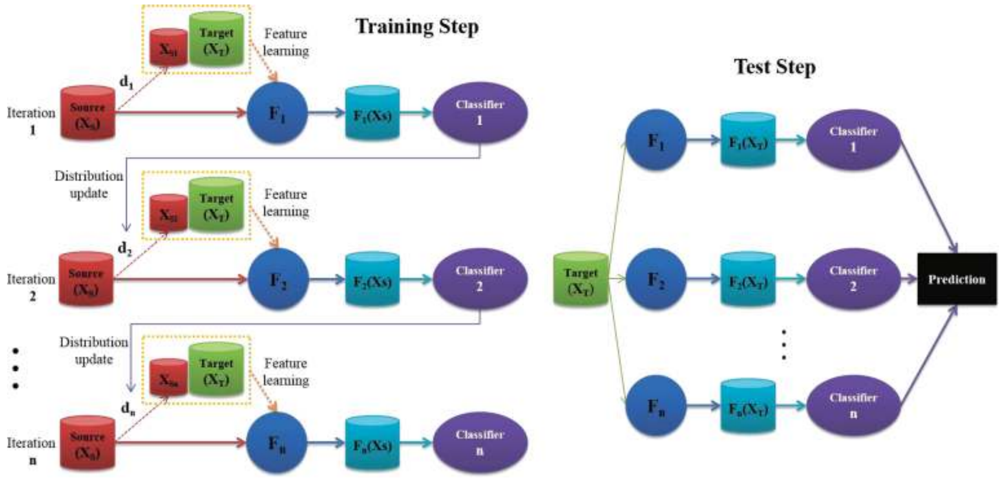

Fig. 1. The flowchart of the training and test process of our proposed boosted multifeature learning for unsupervised domain adaption. For details, please see the corresponding text.

2009a, 2009b, 2011; Glorot et al. 2011; Gong et al. 2012; Chen et al. 2012; Ni et al. 2013].

For the instance subsampling or reweighting based methods, the key idea is that only part of the instances in the source domain can be used to help the learning task in the target domain. Thus, distinctive instances are sampled to bridge the source domain and target domain. For the joint feature representation based methods, the main idea is to learn domain invariant features. Despite the demonstrated success of these approaches [Blitzer et al. 2006; Lai and Fox 2010; Bergamo and Torresani 2010; Saenko et al. 2010; Glorot et al. 2011], they can be further improved in several aspects. First, most instance reweighting or sub-sampling based methods only sample part of the instances in the source domain for domain adaption and discard all other labeled instances. Second, joint feature representation based methods learn a general feature representation for all instances without considering their distribution. In practice, a single feature representation is unlikely to capture the intrinsic data structure and discrepancy between the source and target domains. Therefore, it is necessary to learn the feature representations differently to fully describe the different structure of the data [Duan et al. 2010; Gong et al. 2012, 2013; Ni et al. 2013].

To address the above issues, we propose a novel boosted multifeature learning (BMFL) method. The basic idea is to iteratively learn multiple feature representations for unsupervised domain adaption. Each iteration of boosting begins by learning a feature representation according to the weights assigned by the previous step. The resulting feature representation is then applied to train a new classifier and obtain the domain similarity between the source and target domains. Then, the learned classifier is applied to the instances in the source domain to obtain the classification scores. Finally, the new weights of instances are updated by use of the classification scores and the domain similarity. Based on the above procedure, it is clear that the instances are iteratively reused with different weights to learn multiple feature representations to bridge the source and target domains. Figure 1 shows the main steps of the proposed algorithm. In the training step, we iteratively learn multiple feature representations for instances from both source and target domains, and obtain the corresponding weak classifiers. In the test step, the instances are described by the learned multiple features and classified with the combination of the corresponding weak classifiers to determine their final class labels.

Compared with the existing methods, the contributions of the proposed BMFL algorithm are threefold as follows.

(1) It iteratively reuses all instances with different weights and better exploits instances than existing methods.
(2) It effectively models the instance weight distribution with classification error and domain similarity, which helps learning new feature representation to correct the previously misclassified instances.
(3) It iteratively learns multiple different deep feature representations to effectively bridge the source and target domains due to the boosting procedure. We demonstrate the effectiveness and the applicability of our approach on three applications: image classification, sentiment classification and spam classification.

2. RELATED WORK

In the literature, there are extensive methods about domain adaptation. Generally, domain adaptation can be categorized as either semisupervised methods or unsupervised methods. In this section, we briefly review the domain adaption methods which are the most related to ours. Then, we introduce several domain transfer methods for multimedia data analysis.

Semisupervised adaption methods adopt labeled instances in a target domain to help bridge the gap between two domains. Daumé [2007] model the data distribution corresponding to the source and target domains to consist of a shared component and a component that is specific to the individual domains. In Dai et al. [2007], the TrAdaBoost method is adopted to update instance weight according to its predicted label. In Yao and Doretto [2010] and Al-Stouhi and Reddy [2011], the TrAdaBoost is improved by introducing multiple source domains and multiplying a dynamic correction factor, respectively. Duan et al. [2009a, 2009b, 2010] reduce domain mismatch by using the kernel trick of support vector machines (SVM). In Duan [2010], a kernel function and a robust classifier are learned by minimizing both the structural risk functional and the distribution mismatch between the labeled and unlabeled samples from the source and target domains. Bergamo and Torresani [2010] perform an empirical analysis of several variants of SVM for the domain shift problem. In Lai and Fox [2010], object recognition from 3D point clouds is carried out by generalizing the small amount of labeled training data onto the pool of weakly labeled data obtained from the Internet. Metric learning approaches [Saenko et al. 2010; Kulis et al. 2011] are also proposed to learn a cross domain transformation to link two domains. Recently, Jhuo et al. [2012] utilize low-rank reconstruction to learn a transformation such that the transformed source samples can be linearly reconstructed by the target samples. Theoretical study on the nature of classification error across new domains is given in Ben-David et al. [2010]. Though these semisupervised methods perform well on several public datasets, the condition that a subset of labeled instances on target domain must be provided will inevitably hinder their applications.

Different from the preceding methods, unsupervised domain adaption is more challenging because there are no labeled instances in the target domain. Therefore certain priors are used to relate two domains. In Blitzer et al. [2006], a structural correspondence learning method is proposed to induce correspondence among features from two domains by modeling their relations with pivot features that appear frequently in both domains. The techniques in Pan et al. [2009] reduce the distance across two domains by learning a latent feature space where domain similarity is measured through maximum mean discrepancy. In Wang and Mahadevan [2009], the proposed manifold-alignment domain adaption computes similarity between data points in different domains through the local geometry of data points within each domain. In other works [Blitzer et al. 2011; Gopalan et al. 2011; Gong et al. 2012], the source and target domains are linked by sampling finite or infinite number of intermediate subspaces on the Grassmannian manifold. In Habrard et al. [2013], the boosting scheme is applied in unsupervised domain adaptation by optimizing the source classification error and margin constraints over the unlabeled target instances. In Ni et al. [2013], a dictionary learning approach is proposed for unsupervised domain adaptation. The recently proposed method [Gong et al. 2013] is more related to our work, where the source and target domains are linked by the learned domain invariant features. During the learning phase, a subset of instances in the source domain is selected as landmark. Multiple feature representations are computed by several fixed kernel functions in Gong et al. [2013], while our BMFL method automatically and iteratively learns them inside a boosting procedure. Conventional boosting based methods are all for semisupervised domain, where labeled instances in target domain are adopted to tune weights of weak learners and the distribution of instances. Due to the lack of labeled instances on target domain, there are seldom boosting based methods, except Habrard et al. [2013], for unsupervised domain adaptation. Based on the boosting scheme, the method [Habrard et al. 2013] mainly focuses on finding the classifiers which are able to move closer source and target distributions. In our BMFL, besides tuning the weights of weak learners and distribution of instances, we attempt to learn more suitable feature representations with the iteration process of boosting.

In the multimedia community, there are also several algorithms proposed to improve the learning task in the target domain by leveraging the source domain. A knowledge adaptation method for ad-hoc multimedia event detection is proposed in Ma et al. [2012]. In Lu et al. [2013], cross-domain correlation knowledge is used for web multimedia object classification. In Qi et al. [2011], a feature transformation method is proposed to indirectly transfer semantic knowledge between text and images. In Roy et al. [2012], the authors use a graph based framework to model the distribution discrepancy problem between the social and the video domains. Real time social streams (Twitter) are utilized in two multimedia applications, socialized query suggestion for video search and socialized video recommendation [Roy et al. 2012]. In Tan et al. [2011], social relationship information is adopted to improve user-level sentiment analysis. In addition, real-time social media data have been utilized in semantic video indexing, social event prediction, image/video context annotation [Naaman 2012]. Most of these methods focus on specific domain transfer tasks. Different from these methods, we propose a general algorithm to learn new feature representations where the discrepancy between the source domain and the target domain can be reduced.

3. PROPOSED ALGORITHM

In this section, we introduce how the proposed boosted multifeature learning algorithm iteratively learns multiple feature representations for unsupervised domain adaption. We first show our problem description. Then, the formulation of BMFL is discussed. Finally, we give the discussion to show the difference with the existing methods.

3.1. Overview

Let $\mathbf{X}_{\mathcal{S}} = \{\mathbf{x}_i^s | i = 1, \ldots, l\}$ denote the instances drawn from the source distribution $\mathcal{S}$. The corresponding labels for $\mathbf{X}_{\mathcal{S}}$ are denoted as $\mathbf{Y}_{\mathcal{S}} = \{y_i | i = 1, \ldots, l, y_i \in \{1, 2, \ldots, K\}\}$, where K is the number of classes and l is the number of instances in the source domain. Let $\mathbf{X}_{\mathcal{T}} = \{\mathbf{x}_i^t | i = 1, \ldots, m\}$ be the unlabeled instances drawn from the target domain

$\mathcal{T}$, where m is the number of instances in the target domain. In unsupervised domain adaption, we only have the labeled instances $\mathbf{X}_{\mathcal{S}}$ in the source domain and the unlabeled instances $\mathbf{X}_{\mathcal{T}}$ in the target domain. Our aim is to learn a domain transfer classifier $\mathbf{H}(\mathbf{x})$ for $\forall \mathbf{x} \in \{\mathbf{x}_i^t | \mathbf{x}_i^t \in \mathbf{X}_{\mathcal{T}}\}$ with the assistance of the labeled set $\mathbf{X}_{\mathcal{S}}$ and the unlabeled set $\mathbf{X}_{\mathcal{T}}$.

To achieve this goal, we use the BMFL method to iteratively obtain multiple feature representations to bridge the domain gap. As shown in Figure 1, the proposed BMFL algorithm has two steps. In the training step, multiple feature representations are learned inside a boosting procedure [Zhang et al. 2009, 2011]. In each iteration n, we sample a subset $\mathbf{X}_{\mathcal{S}_n}$ of the instances set $\mathbf{X}_{\mathcal{S}}$ in the source domain according to their weights $\mathbf{d}_n$ assigned by the previous boosting iteration. This subset $\mathbf{X}_{\mathcal{S}_n}$ is then used as a guide to learn a feature representation function $\mathbf{F}_n(\mathbf{x})$ using a recent method [Chen et al. 2012]. The resulting feature representation is applied to compute the domain similarity $\mathbf{c}_n$ and learn a new weak learner $\mathbf{h}_n(\mathbf{x})$ by considering the instance weights $\mathbf{d}_n$. The learned classifier is adopted to classify the instances in the source domain to obtain their classification errors ϵ_n. Finally, the new weights of instances are updated by using the classification error ϵ_n and the domain similarity $\mathbf{c}_n$. In this way, instances in the source domain with large domain similarities to instances in the target domain are more likely to be selected for training a new feature representation function $\mathbf{F}_{n+1}(\mathbf{x})$ in the next iteration. Once this procedure converges, we obtain a set of feature representation functions $\{\mathbf{F}_n(\mathbf{x})\}_{n=1}^N$, and a set of weak learners $\{\mathbf{h}_n(\mathbf{x})\}_{n=1}^N$ and their corresponding combination coefficients $\{\alpha_n\}_{n=1}^N$ to get the final strong classifier $\mathbf{H}(\mathbf{x})$ as shown in (11). Here, N is the number of iterations. In the test step, each instance $\mathbf{x} \in \mathbf{X}_{\mathcal{T}}$ is mapped with functions $\{\mathbf{F}_n(\mathbf{x})\}_{n=1}^N$ to obtain N feature spaces. Then, each mapped feature $\mathbf{F}_n(\mathbf{x})$ is classified by its corresponding weak classifier $\mathbf{h}_n(\mathbf{x})$. The predicted results of all N weak classifiers are combined to decide the final class labels $\mathbf{H}(\mathbf{x})$.

For each iteration n, our approach has three major components, feature learning, weak learner, and instance weight update, described as follows.

3.2. Distribution Sensitive Feature Learning

The simple yet effective marginalized stacked denoising autoencoder (mSDA) method has been successfully applied to domain adaptation. The basic idea is to combine the instances in the source and target domains together to learn a common feature representation [Glorot et al. 2011; Chen et al. 2012]. The basic building block of mSDA [Chen et al. 2012] is a one-layer denoising autoencoder. Let $\{\mathbf{x}_i | i = 1, \ldots, l+m\} = \mathbf{X}_{\mathcal{S}} \cup \mathbf{X}_{\mathcal{T}}$ be all instances from the source and target domains. The mSDA method reconstructs the original feature with a single mapping function by minimizing the following squared reconstruction loss.

$$\frac{1}{2(l+m)} \sum_{i=1}^{l+m} \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_i\|^2. \tag{1}$$

Here, $\tilde{\mathbf{x}}_i$ is the corrupt version of $\mathbf{x}_i$. Specifically, $\tilde{\mathbf{x}}_i$ is obtained by stochastically setting some elements of the input $\mathbf{x}_i$ to zero [Vincent et al. 2008]. Hence denoising autoencoder as shown in Equation (1) is trying to predict the missing values from the nonmissing values. The corruptions are useful for capturing the statistical dependencies between the inputs [Bengio 2009]. W denotes the mapping matrix which projects the corrupted feature $\tilde{\mathbf{x}}_i$ to $\mathbf{x}_i$. Though it is just a single linear mapping, more representative domain invariant features can be learned when combined with the nonlinear activation function and the layer-wise stacking scheme [Chen et al. 2012]. With r

different corruptions, Equation (1) can be written as

$$\mathcal{L}_{sq}(\mathbf{W}) = \frac{1}{2(l+m)r} \sum_{j=1}^{r} \sum_{i=1}^{l+m} \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_{ij}\|^2. \tag{2}$$

This equation can be solved using the closed-form solution for ordinary least squares. A more simplified solution is given in Chen et al. [2012] by marginalizing all the noises when $r \to \infty$.

$$\mathbf{W} = E[\mathbf{P}]E[\mathbf{Q}]^{-1}, \mathbf{Q} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top, \mathbf{P} = \bar{\mathbf{X}}\tilde{\mathbf{X}}^\top, \tag{3}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{l+m}]$, $\bar{\mathbf{X}} = [\mathbf{X}, \ldots, \mathbf{X}]$. In addition, the corrupted version of $\bar{\mathbf{X}}$ is denoted as $\tilde{\mathbf{X}}$. Here $\mathbf{W}$ can be considered practically as a linear mapping function. After the linear feature mapping, as in traditional deep learning methods, a nonlinear activation function (e.g., $\tanh(\cdot)$) is applied. To construct a deep learning structure, such one layer auto-encoders are stacked together. In practice, the mSDA structure for feature representation are fixed by weight matrices where each layer has one weight matrix and a nonlinear $\tanh(\cdot)$ function. We denote the mSDA feature representation as a single function $\mathbf{F}(\mathbf{x})$. Take a 2-layer mSDA as an example, we use the function $\mathbf{F}(\mathbf{x}) = \tanh(\mathbf{W}_2 \tanh(\mathbf{W}_1\mathbf{x}))$ to represent the mSDA feature representation method.

Existing deep learning methods ignore the instance weight distribution in learning multiple different feature representations. It is assumed that instances in both source and target domains can be mapped into a common feature space where they share a common distribution. They do not consider the differences of instances due to the domain discrepancy. In this work, we propose a distribution sensitive deep feature learning within the proposed unsupervised domain adaption framework. We use $\mathbf{X}_{\mathcal{T}}$ and a sampled subset $\mathbf{X}_{\mathcal{S}_n}$ of instances in the source domain to learn deep feature representation function $\mathbf{F}_n(\mathbf{x})$ for domain adaptation. The subset $\mathbf{X}_{\mathcal{S}_n}$ is sampled according to the iteratively updated instance distribution $\mathbf{d}_n$ as discussed in Section 3.4.

3.3. Weak Learner

Once we obtain the learned features $\{\mathbf{F}_n(\mathbf{x}_i^s)|\mathbf{x}_i^s \in \mathbf{X}_{\mathcal{S}}\}$, we need to design an effective and efficient weak learner $\mathbf{h}_n(\mathbf{x})$ by considering the current instance weights $\mathbf{d}_n$. For simplicity, we adopt the linear weighted support vector machine (WSVM) [Yang et al. 2007] due to its efficiency. Note that any other classifier can also be applied, such as decision trees. In our experiments, we show that the WSVM achieves comparable results as other alternatives. We rewrite $\mathbf{d_n} = [d_n^1, d_n^2, \ldots, d_n^l]^\top$ and d_n^i is the weight of the i-th instance $\mathbf{x}_i^s$ in the source domain. Then, a 2-class linear support vector machine can be written as

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{i=1}^{l} d_n^i \xi_i \\ &s.t.\ \ y_i(\mathbf{w}^\top\mathbf{x}_i^s + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l \\ &\xi_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{4}$$

Note that (4) can be viewed as assigning a penalty parameter $d_n^i C$ to $\mathbf{x}_i^s$. Thus different instances will be constrained with different penalties in the learning process. For the multiclass case, one-vs-the-rest strategy can be adopted.

Similar to the conventional multiclass AdaBoost scheme [Zhu et al. 2009], after constructing the weak learner, we compute its classification error ϵ_n and assign a weight α_n for the weak learner $\mathbf{h}_n(\mathbf{x})$ as shown in (5) and (6), respectively. Here, $\mathbb{I}(.)$ is the indicator function.

$$\epsilon_n = \frac{1}{\sum_{i=1}^{l} d_n^i} \sum_{i=1}^{l} d_n^i \cdot \mathbb{I}\big(y_i \neq \mathbf{h}_n(\mathbf{x}_i^s)\big), \tag{5}$$

$$\alpha_n = \ln((1-\epsilon_n)/\epsilon_n) + \ln(K-1). \tag{6}$$

3.4. Instance Weight Update

In unsupervised domain adaptation, different from the semisupervised case, there are only unlabeled instances in the target domain to learn domain invariant features. Thus we can not train weak learners and update the weights with the same way as in conventional boosting based semisupervised domain adaptation methods, such as TrAdaBoost [Dai et al. 2007]. Instead, our instance weight distribution update scheme is shown in (7)), where $\mathbf{p}_n$ and $\mathbf{q}_n$ are vectors and their elements are defined as follows: p_n^i is computed according to the classification error of the current weak learner $\mathbf{h}_n(\mathbf{x})$ for instance $\mathbf{x}_i^s$ in the source domain, and q_n^i is calculated for instance $\mathbf{x}_i^s$ via the domain similarity criterion. Details of the $\mathbf{p}_n$ and $\mathbf{q}_n$ are introduced in Section 3.4.1 and 3.4.2, respectively.

$$d_{n+1}^i = d_n^i \cdot \exp\big(p_n^i \cdot q_n^i\big), \quad i = 1, \ldots, l. \tag{7}$$

3.4.1. Classification Error Criterion $\mathbf{p}_n$. In conventional AdaBoost, the weak learners are trained with instances from a single domain, weights of the misclassified instances are increased while weights of the correctly classified instances are decreased. In unsupervised domain adaptation, there are two key problems which hinder the direct use of the conventional Adaboost. (1) There is a large mismatch between the source and target domains. (2) There are no labeled instances in the target domain which can be used for supervised training of the weak learners and updating of instance weights. For the first problem, our solution is to map the original instances from both domains into a common feature space where a common distribution exists in both domains. The common feature learning for the source domain and the target domain has been illustrated in Section 3.2. For the second problem, in the mapped common feature space, we use the labeled instances in the source domain to mimic the labeled instances in the target domain. Specifically, in the mapped common feature space, weights of the misclassified instances in the source domain are increased while weights of the correctly classified instances are decreased. Thus p_n^i in (7) is calculated as

$$p_n^i = \alpha_n \cdot \mathbb{I}\big(y_i \neq \mathbf{h}_n(\mathbf{x}_i^s)\big). \tag{8}$$

By this updating scheme, the misclassified instances are more likely to be sampled in the guide set.

3.4.2. Domain Similarity Criterion $\mathbf{q}_n$. In practice, it is not possible to guarantee that instances in two domains have the same distribution in the learned single feature space. As a remedy of the weight update criterion $\mathbf{p}_n$ as shown in Section 3.4.1, we propose a new criterion $\mathbf{q}_n$. This is implemented by considering domain similarity distribution to distinguish contributions of different instances in domain adaptation.

According to the application of $\mathcal{MMD}$ [Borgwardt et al. 2006] in discrete data, we adopt the following formulation to describe the domain discrepancy:

$$\begin{gathered}\mathcal{MMD}[\mathbf{F}, \mathbf{X}_{\mathcal{S}}, \mathbf{X}_{\mathcal{T}}] = \Big\| E\big(\mathbf{F}_n(\mathbf{x}^s)\big) - E\big(\mathbf{F}_n(\mathbf{x}^t)\big) \Big\|_2 \\ \mathbf{x}^s \in \mathbf{X}_{\mathcal{S}}, \quad \mathbf{x}^t \in \mathbf{X}_{\mathcal{T}}.\end{gathered} \tag{9}$$

Here, $E()$ is the expectation, the learned deep feature representation function $\mathbf{F}_n(\mathbf{x})$ is viewed as a mapping function. In the domain adaptation problem, the discrepancy between original instances from both domains is decided by the data distribution. Thus, the value denoted by (9) reflects the domain adaptation power of the feature mapping function $\mathbf{F}_n(\mathbf{x})$.

To decide which instances in the source domain have more similar distribution to the instances in the target domain, a similarity vector $\mathbf{q}_n$ can be computed by (10), y_{ik} is a class indicator of the i^{th} instance in the source domain for class k. Three constraints are used to balance the classes. Once we obtain the domain similarity measurement $\mathbf{q}_n$ for all instances in the source domain, we use it to update the weight distribution $\mathbf{d}_{n+1}$ using (7).

$$\begin{aligned}
&\arg\min_{\mathbf{q}_n} \left\| \sum_{i=1}^{l} (q_n^i \mathbf{F}_n(\mathbf{x}_i^s)) - \sum_{j=1}^{m} \mathbf{F}_n(\mathbf{x}_j^t) \right\|_2 \\
&s.t. \quad \sum_{i=1}^{l} q_n^i y_{ik} = \frac{1}{l} \sum_{i=1}^{l} y_{ik}, \quad k = 1, \ldots, K \\
&\qquad \sum_{i=1}^{l} q_n^i = 1, \\
&\qquad 0 \le q_n^i \le 1, \ \forall i = 1, \ldots, l. \qquad (10)
\end{aligned}$$

This $\mathcal{MMD}$ scheme to measure domain discrepancy has also been used in Gong et al. [2013] and Duan et al. [2010]. Different from them using a previously defined kernel matrix, our method is based on the iteratively learned boosted deep features.

3.5. Domain Transfer Classifier

Once the boosting procedure converges, we obtain a set of feature representation functions $\{\mathbf{F}_n(\mathbf{x})\}_{n=1}^N$, and a set of weak learners $\{\mathbf{h}_n(\mathbf{x})\}_{n=1}^N$ and their corresponding combination coefficients $\{\alpha_n\}_{n=1}^N$. Then, the learned domain transfer classifier $\mathbf{H}(\mathbf{x})$ is

$$\mathbf{H}(\mathbf{x}) = \arg\max_k \sum_{n=1}^{N} \alpha_n \cdot \mathbb{I}(\mathbf{h}_n(\mathbf{x}) = k), k \in \{1, \ldots, K\}. \qquad (11)$$

3.6. Discussion

The details of the proposed BMFL algorithm is summarized in Algorithm 1. Compared with the existing domain adaption methods, the differences are the following.

(1) Compared with AdaBoost-based domain adaption methods our BMFL method (a) iteratively learns multiple features for domain adaption, instead of only using the original feature as in the conventional AdaBoost-based methods; (b) it adopts domain similarity to update instance weight, which is ignored in previous methods; (c) our BMFL algorithm is unsupervised as Habrard et al. [2013] whereas most existing methods are semisupervised [Dai et al. 2007; Sugiyama et al. 2009; Yao and Doretto 2010; Al-Stouhi and Reddy 2011; Chen et al. 2011].

(2) Compared with deep learning-based methods our BMFL method iteratively learns multiple deep features by considering the instance weight distribution inside the boosting procedure, which is more robust than traditional methods that only learn a general representation for all instances without considering their weight distribution.

ALGORITHM 1: Boosted Multifeature Learning (BMFL) for Unsupervised Domain Adaptation

```
input  : Source Domain: $\mathbf{X}_{\mathcal{S}} = \{\mathbf{x}_i^s\}_{i=1}^l$, $\mathbf{Y}_{\mathcal{S}} = \{y_i\}_{i=1}^l$. Target Domain: $\mathbf{X}_{\mathcal{T}} = \{\mathbf{x}_i^t\}_{i=1}^m$. $K$ and $N$.
         $d_1^i = 1/l$, $q_1^i = 1/l$, $\forall i = 1, \ldots, l$
output: weak classifiers $\{\mathbf{h}_n(\mathbf{x})\}_{n=1}^N$, coefficients $\{\alpha_n\}_{n=1}^N$, feature functions $\{\mathbf{F}_n(\mathbf{x})\}_{n=1}^N$.
for $n = 1$ to $N$ do
    Sample $\mathbf{X}_{\mathcal{S}_n}$ from $\mathbf{X}_{\mathcal{S}}$ according to $\mathbf{d}_n$.
    Learn deep feature representation $\mathbf{F}_n(\mathbf{x})$ on $\mathbf{X}_{\mathbf{S}_n} \cup \mathbf{X}_{\mathcal{T}}$ as in Section 3.2.
    Learn a weak classifier $\mathbf{h}_n(\mathbf{x})$ according to $\mathbf{d}_n$ as in Section 3.3.
    Compute the error $\epsilon_n$ and $\alpha_n$ according to (5) and (6), respectively.
    Compute $\mathbf{q}_n$ by optimizing (10).
    Update instance weight distribution $\mathbf{d}_{n+1}$ as in (7) in Section 3.4.
end
```

(3) Compared with other recent adaption methods, such as Gong et al. [2013], both our BMFL method and Gong et al. [2013] adopt a subset sampled in the source domain for domain adaption. However, there are three differences. (a) In Gong et al. [2013], multiple feature representations are computed by several fixed kernel functions while the BMFL method automatically and iteratively learns them inside a boosting procedure. (b) Our BMFL method adopts the instance weight distribution updated by classification error and domain similarity to iteratively learn multiple features. (c) Different from Gong et al. [2013], which combines the subset sampled in the source domain and the instances in the target domain to simulate a semisupervised domain adaptation, our BMFL method uses the sampled subset to iteratively learn deep features.

4. EXPERIMENTAL RESULTS

To test the effectiveness of the proposed BMFL algorithm for cross-domain data analysis, we evaluate it against state-of-the-art methods on three popular applications of unsupervised domain adaptation, including image classification, sentiment classification and spam filtering. Since our work focuses on the unsupervised domain transfer, all instances in the source domain are labeled while all instances in the target domain are unlabeled. The domain transfer model is trained based on both the labeled instances in the source domain and the unlabeled instances in the target domain. Then the learned model is tested on the target domain. The experimental results are reported and discussed as follows.

4.1. Image Classification

The image classification experiment illustrated in Section 4.1 of the manuscript is carried out on a total of 2,533 images in 10 categories which are common to all four public datasets as in Gong et al. [2012]: "backpack," "touring-bike," "calculator," "headphones," "computer-keyboard," "laptop-101," "computer-monitor," "computer-mouse," "coffee-mug," and "video-projector."

The four public datasets, Caltech, Amazon, Webcam and DSLR, are popularly used for evaluating domain adaptation methods [Griffin et al. 2007; Saenko et al. 2010]. (1) The Caltech dataset has been extensively used for image classification which contains a total of 30607 images belong to 256 categories. The Caltech-256 is collected by choosing a set of object categories, downloading examples from Google Images and then manually screening out all images that did not fit the category. (2) Amazon dataset

Table I.
Accuracy on each task and their average on the image classification dataset of several domain adaption methods. GFK1 and GFK2 are methods proposed in [Gong et al. 2012], GFS is the method proposed in [Gopalan et al. 2011], Metric is the method proposed in [Saenko et al. 2010], Landmark is proposed in [Gong et al. 2013], SIDL is the method proposed in [Ni et al. 2013] and mSDA is proposed in [Chen et al. 2012]. Based on the results, it is clear that our BMFL achieves the best on average.

Method	A-C	A-D	A-W	C-A	C-D	C-W	D-A	D-C	D-W	W-A	W-C	W-D	Avg
GFK1	37.9	-	35.7	40.4	41.1	-	36.1	-	79.1	35.5	29.3	-	41.9
GFS	39.2	36.3	33.6	43.6	40.8	36.3	-	-	-	33.5	30.9	75.7	41.1
GFK2	42.2	42.7	40.7	44.5	43.3	44.7	-	-	-	31.8	30.8	75.6	44.0
Metric	42.4	42.9	49.8	46.6	47.6	42.8	-	-	-	38.6	33.0	87.1	47.9
Landmark	45.5	47.1	46.1	56.7	**57.3**	49.5	-	-	-	40.2	35.4	75.2	50.3
SIDL	40.4	-	37.9	45.4	42.3	-	39.1	-	**86.2**	38.3	36.3	-	45.7
mSDA	46.6	44.0	41.4	58.4	49.0	51.5	38.5	34.5	80.7	34.0	31.9	**87.9**	49.9
BMFL	**47.5**	**49.7**	**51.9**	**58.7**	54.8	**53.6**	**43.4**	**37.0**	86.1	**42.3**	**37.3**	86.6	**54.1**

consists of images from the web downloaded from online merchants.[1] These images are of products shot at medium resolution typically taken in an environment with studio lighting conditions. (3) DSLR dataset consists of images that are captured with a digital SLR camera in realistic environments with natural lighting conditions. The images have high resolution (4288 × 2848) and low noise. (4) Webcam dataset consists of images of the 31 categories recorded with a simple webcam. The images are of low resolution (640 × 480) and show significant noise and color as well as white balance artifacts. Many current imagers on robotic platforms share a similarly sized sensor, and therefore also possess these sensing characteristics. The resulting webcam dataset contains the same 5 objects per category as in DSLR, for a total of 795 images.

We evaluate the proposed BMFL algorithm against several state-of-the-art unsupervised adaptation methods including the GFS [Gopalan et al. 2011], GFK [Gong et al. 2012], METRIC [Saenko et al. 2010], Landmark [Gong et al. 2013], subspace Interpolation via dictionary learning (SIDL) [Ni et al. 2013], and mSDA [Chen et al. 2012] methods. For fair comparison, we use the same 800-dimensional SURF features as in Gong et al. [2013]. Table I shows the reported results of the first five methods where fewer than eight different pairs of the source and target combinations are evaluated. Different from these methods, we report experimental results on 12 tasks. These tasks are created from the 4 image domains Caltech (C), Amazon (A), Webcam (W) and DSLR (D). For example, "A-C" means that the cross domain task where Amazon is used as the source domain and Caltech as the target domain. The layer number of stacked auto-encoders used in our BMFL and mSDA is set to 2. Due to the dataset size, the performance of the proposed method is not improved when more layers are used as shown in Chen et al. [2012]. For the BMFL method, the maximum number of the iterations N is set to 30 and the size of the sampled subset for learning feature representation is set to about 300. For all 6 methods, the regularization parameter is set to 10 when a linear SVM is used for classification.

Table I shows the results where the BMFL performs well in almost all 12 tasks except three tasks, D-W, C-D, W-D, which are comparable to the results of other methods. We also show the average results of the 12 tasks denoted as Avg. Since some reported results of the evaluated methods are not available, we only average the reported results.

Compared with the recent mSDA method, the proposed BMFL algorithm has about 4% improvement on average, which is because instances in the small dataset are not

---

[1] www.amazon.com.

Table II.
Accuracy on each task and their average on the sentiment dataset. GFK-d20 and GFK-d5 are methods proposed in [Gong et al. 2012], Landmark is the method proposed in [Gong et al. 2013], mSDA1 and mSDA2 are methods proposed in [Chen et al. 2012]. Based on the results, it is clear that our BMFL achieves the best on average.

Method	B-D	B-E	B-K	D-B	D-E	D-K	E-B	E-D	E-K	K-B	K-D	K-E	Avg
GFK-d20	69.1	65.0	67.5	67.9	67.2	67.5	64.7	65.5	76.4	65.9	66.1	76.1	68.2
GFK-d5	70.9	67.7	70.3	70.1	66.4	70.1	66.8	66.5	76.6	65.5	67.7	73.8	69.4
Landmark	-	78.5	-	79.0	-	-	-	-	83.4	-	75.1	-	79.0
mSDA1	76.7	77.0	72.6	79.1	76.8	75.9	70.0	71.4	84.5	74.0	73.2	82.5	76.1
mSDA2	78.8	79.3	75.1	80.9	80	77.7	71.4	72.8	86.3	76.2	76.7	84.8	78.3
BMFL-tree	79.7	78.1	**84.5**	80.2	80.6	83.3	72.3	76.4	85.9	76.8	77.5	84.5	80.0
BMFL-SVM1	78.9	**79.6**	81.9	79.2	**81.6**	81.9	72.8	73.3	86.4	77.8	77.8	84.4	79.6
BMFL-SVM2	**81.2**	73.9	81.6	**83.7**	78.6	**86.2**	**77.0**	**77.8**	**87.5**	**79.5**	**80.7**	**85.9**	**81.1**

sufficient to train a deeper mSDA structure. Therefore, it is important to learn multiple feature representations.

4.2. Sentiment Classification

We evaluate the proposed BMFL for sentiment classification on the Amazon reviews benchmark dataset. This dataset contains more than 340, 000 reviews from 25 different types of products from Amazon. As in Chen et al. [2012], we consider only the binary classification problem whether a review is positive or negative. We use the same features as Chen et al. [2012], where the raw bag-of-words features are extracted. As in Blitzer et al. [2006], a smaller dataset is created to evaluate the existing domain adaption methods. We evaluate our BMFL on the same small dataset which contains four types of products: books (B), DVDs (D), electronics (E), and kitchen (K) appliances. Each domain contains about 6,000 instances with 5000 dimensional feature.

With these four domain instances, there are 12 tasks in total when we take every pair as a task. Similarly to the image classification experiment in Section 4.1, we denote the cross domain task from the books to the DVDs as “B-D”, electronics to kitchen as “E-K”, etc. We compare the BMFL algorithm with the GFK [Gong et al. 2012], Landmark [Gong et al. 2012] and mSDA [Chen et al. 2012] on these transfer tasks. For the GFK, we test it using different dimensions of the learned subspace. In the first two rows of Table II, we show the best two sets of results, where GFK-d20 denotes the GFK method using a 20-dimensional subspace and GFK-d5 denotes the GFK method using a 5-dimensional subspace. In the third row of Table II, we show the experimental results reported in Gong et al. [2013].

For the BMFL method, we also evaluate the effectiveness of different weak learners. In our experiments, two different weak learners are used, decision trees and linear SVM, and the results are shown in Table II denoted as BMFL-tree and BMFL-SVM1, respectively. Although the results with a decision tree in Table II are slightly better than those using a linear SVM, we use a linear SVM as our weak learner in other experiments due to its efficiency. In the experimental results shown in Table II, the mSDA1, BMFL-tree and BMFL-SVM1 methods only use the learned features without combing the original features. In contrast, the mSDA2 and BMFL-SVM2 methods combine the original features and the learned features together. Furthermore, for the mSDA and BMFL algorithms, we adopt 5 layer auto-encoders for feature learning.

The results show that the methods using combined features achieve better results, which are also demonstrated in Chen et al. [2012]. The results in Table II show that the BMFL based methods perform well in all tasks. To better evaluate these 12 tasks, we show the average accuracy of all tasks for each method. The results are shown in the last column of Table II. Table II shows that the proposed BMFL algorithms performs the best against all the evaluated methods.

Table III. Average Results of 9 Methods on the Spam Dataset, and Our BMFL Achieves the Best

Method	Avg
SVM1	62.0
Adaboost [Freund and Schapire 1996]	40.6
DASVM [Bruzzone and Marconcini 2010]	62.5
SVM-W [Huang et al. 2007]	62.1
SLDAB-H [Habrard et al. 2013]	62.9
SLDAB-gn [Habrard et al. 2013]	64.2
SVM2	63.5
mSDA [Gong et al. 2013]	68.4
BMFL	**72.1**

Table IV. Accuracy of Each Task for Three Methods on the Spam Dataset

Method	1	2	3	4	5	Avg
SVM2	63.3	65.2	62.6	62.6	64	63.5
mSDA [Gong et al. 2013]	68.5	**70.7**	67.0	69.3	66.4	68.4
BMFL	**70.3**	70.5	**74.3**	**74.4**	**71.0**	**72.1**

4.3. Spam Filtering

The UCI Spam dataset[2] contains 4,601 emails with 2,788 nonspam instances and 1,813 spam instances, which are represented by 57-dimensional features. For fair comparison with state-of-the-art methods, we use the same scheme as Habrard et al. [2013] for the domain adaption task. The original UCI Spam dataset is randomly split into three different sets of equivalent size. The first sample set is used to represent the source domain. The other two sets are used as unlabeled training samples from the target domain and test samples in the target domain. To simulate different distribution, the last two sample sets are created by adding Gaussian noise. Specifically, Gaussian noise is generated for the n-th element of the original features according to $\mathcal{G}(\mu_n, \delta_n)$. The mean μ_n and the standard deviation δ_n are sampled from a uniform distribution among $[-0.15, 0.15]$ and $[0, 0.5]$, respectively. This process is repeated for 5 times for 5 different domain adaptation tasks in the experiments.

We compare BMFL with the state-of-the-art boosting-based unsupervised domain adaptation method [Habrard et al. 2013] using the reported results. The results in Table III show that the proposed BMFL method performs the best against all the other methods. The first 6 results in Table III are reported in Habrard et al. [2013]. To fairly compare with the SLDAB methods, we denote the SVM in Habrard et al. [2013] as SVM1, and our method as SVM2. For the mSDA method [Chen et al. 2012] and the proposed BMFL algorithm, we use two layers in the auto-encoders. Furthermore, the regularization parameters used for linear SVM in SVM2, mSDA [Chen et al. 2012] and BMFL methods are all set to be 10 for the best results. Although the randomly generated tasks are different, the average performance show the domain adaptation strength of the BMFL method. In addition, the accuracy on each task is shown in Table IV. We can see that our BMFL performs better than mSDA [Chen et al. 2012] on all tasks except the task 2. For the task 2, it may be because our BDFL cannot learn much more appropriate representations than mSDA [Chen et al. 2012] for domain transfer.

---

[2] http://archive.ics.uci.edu/ml/datasets/Spambase.

Table V.
Accuracy of the object recognition on the Caltech dataset using images from Bing.

Method	Accuracy
SVM	7.83
GFK [Gong et al. 2012]	6.83
Landmark [Gong et al. 2013]	11.2
mSDA [Chen et al. 2012]	10.8
BMFL	**14.3**

Table VI.
The experimental results about the domain similarity criterion and multiple feature representations. For more details, please see the analysis in the text.

Method	A-C	A-D	A-W	C-A	C-D	C-W	D-A	D-C	D-W	W-A	W-C	W-D	Avg
BMFL$_p$	**48.1**	42.7	51.2	56.9	44.6	49.8	41.0	36.8	83.4	40.9	36.9	86.0	51.5
BMFL$_s$	47.7	37.6	41.4	55.6	43.9	51.2	40.2	36.8	84.1	41.2	36.2	86.0	50.2
BMFL$_m$	47.5	**49.7**	**51.9**	**58.7**	**54.8**	**53.6**	**43.4**	**37.0**	**86.1**	**42.3**	**37.3**	**86.6**	**54.1**

4.4. Object Recognition Using the Loosely Labeled Web Scale Images

To evaluate our algorithm in a more practical environment, we introduce a much more difficult experiment of object recognition [Zhang et al. 2013] using loosely labeled web images. We use Caltech256 [Griffin et al. 2007], which is one of the popularly used image datasets in computer vision, as the target domain of the domain transfer task. The Caltech256 dataset contains 256 object categories and about 30K images totally. For the source domain, we use the Bing dataset proposed in Bergamo and Torresani [2010] which consists of 120K weakly labeled web photos retrieved using keyword-based image search. Specifically, category names of the Caltech256 dataset are used as text queries for searching images on the Bing site without human verification. Compared with other kinds of task, such as image classification, sentiment classification and spam filtering introduced in Sections 4.1, 4.2, and 4.3, we can see that this domain transfer task is much more difficult.

In Table V, we show the average accuracies of 5 different algorithms for transferring knowledge from the web images to object recognition. We can see that our method perform better than all the baselines.

4.5. Discussion and Analysis

In this section, we show that (1) the domain similarity is effective to update instance weight distribution for domain adaption and (2) the learned multiple feature representations are better than the single one. To evaluate these observations, the experiments are done on the image classification dataset introduced in Section 4.1.

Without considering the domain similarity criterion $\mathbf{q}_n$, we only adopt the classifier error criterion $\mathbf{p}_n$ to update the instance weight distribution $\mathbf{d}_n$. The corresponding results are shown in Table VI denoted as BMFL$_p$. As shown in Table VI, compared with the proposed BMFL$_m$, the accuracy without domain similarity criteria $\mathbf{q}_n$ drops about 3% on average. Specifically, the accuracy of each task decreases except the task A-C, which is also comparable to our BMFL$_m$.

In addition, the method with only one single feature representation is denoted as BMFL$_s$. For our multiple feature representations based method, it is denoted as BMFL$_m$. The results are shown in Table VI. Compared with our BMFL$_m$, the performance of BMFL$_s$ also decreases about 4% on average accuracy. The results show that it is important to iteratively learn multiple feature representations by considering the

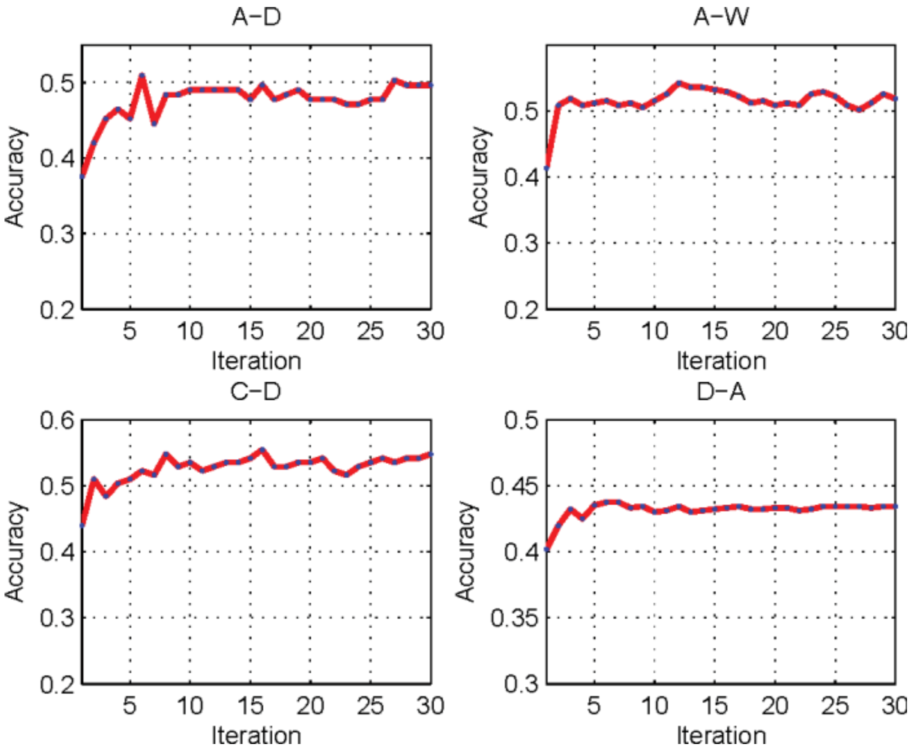

Fig. 2. Accuracy vs. iteration of the BMFL method on 4 tasks. Accuracy of the BMFL method increases with the number of iteration and the method converges quickly.

instance weight distribution. In Figure 2, we also show that the accuracy of each task will increase with the iteration of boosting procedure. Due to the space limitation, we only show the results on 4 tasks: A-D, A-W, C-D and D-A. Moreover, Figure 2 shows that the BMFL method converges well (e.g., 15 iterations in our experiments).

5. CONCLUSIONS

In this article, we propose a novel boosted multifeature learning approach to iteratively learn multiple deep feature representations within a boosting framework for unsupervised domain adaption. We evaluate our BMFL algorithm against state-of-the-art methods on three classification applications: image classification, sentiment classification, and spam filtering. Extensive experimental results demonstrate that our BMFL algorithm consistently performs favorably against existing domain adaption methods and can effectively deal with the cross-domain transfer problem. In the future, we will extend our BMFL algorithm for other applications in multimedia data analysis.

REFERENCES

Samir Al-Stouhi and Chandan K. Reddy. 2011. Adaptive boosting for transfer learning using dynamic updates. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. 60–75.

Bing-Kun Bao, Weiqing Min, Ke Lu, and Changsheng Xu. 2013. Social event detection with robust highorder co-clustering. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval (ICMR'13)*. ACM, New York, 135–142.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems*. 137–144.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. 2010. Impossibility theorems for domain adaptation. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 129–136.

Yoshua Bengio. 2009. Learning deep architectures for AI. *Found. Trends Machine Learn.* 2, 1, 1–127.

Alessandro Bergamo and Lorenzo Torresani. 2010. Exploiting weakly-labeled Web images to improve object classification: a domain adaptation approach. In *Proceedings of the Annual Conference on Neural Information Processing Systems*. 181–189.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.

John Blitzer, Sham Kakade, and Dean P. Foster. 2011. Domain adaptation with coupled subspaces. *J. Machine Learn. Rese.* 15, 173–181.

John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 120–128.

Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alexander J. Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*. 49–57.

Markus Brenner and Ebroul Izquierdo. 2012. Social event detection and retrieval in collaborative photo collections. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR'12)*. ACM, New York, 21:1–21:8.

Lorenzo Bruzzone and Mattia Marconcini. 2010. Domain Adaptation Problems: A DASVM classification technique and a circular validation strategy. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 5, 770–787.

Minmin Chen, Kilian Q. Weinberger, and Yixin Chen. 2011. Automatic feature decomposition for single view co-training. In *Proceedings of the International Conference on Machine Learning*. 953–960.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the International Conference on Machine Learning*.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In *Proceedings of the 30th International Conference on Machine Learning*. 193–200.

Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.

Lixin Duan, Ivor W. Tsang, Dong Xu, and Tat-Seng Chua. 2009a. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the International Conference on Machine Learning*. 37.

Lixin Duan, Ivor Wai-Hung Tsang, Dong Xu, and Stephen J. Maybank. 2009b. Domain Transfer SVM for video concept detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1375–1381.

Lixin Duan, Dong Xu, Ivor Wai-Hung Tsang, and Jiebo Luo. 2010. Visual event recognition in videos by learning from web data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1959–1966.

Wolfgang Effelsberg. 2013. A personal look back at twenty years of research in multimedia content analysis. *ACM Trans. Multimedia Comput. Commun. Appl.* 9, 43:1–43:4.

Yoav Freund and Robert E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning*. 148–156.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning*. 513–520.

Boqing Gong, Kristen Grauman, and Fei Sha. 2013. Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the International Conference on Machine Learning*. 222–230.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2066–2073.

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2011. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*. 999–1006.

Gregory Griffin, Alex Holub, and Pietro Perona. 2007. Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology.

Amaury Habrard, Jean-Philippe Peyrache, and Marc Sebban. 2013. Boosting for Unsupervised Domain Adaptation. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. 433–448.

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. 2006. Correcting sample selection bias by unlabeled data. In *Proceedings of the Annual Conference on Neural Information Processing Systems*. 601–608.

I.-H. Jhuo, D. Liu, D. T. Lee, and S.-F. Chang. 2012. Robust visual domain adaptation with low-rank reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2168–2175.

Xin Jin, Andrew C. Gallagher, Liangliang Cao, Jiebo Luo, and Jiawei Han. 2010. The wisdom of social multimedia: using flickr for prediction and forecast. In *Proceedings of the ACM International Conference on Multimedia (MM'10)*. 1235–1244.

B. Kulis, K. Saenko, and T. Darrell. 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1785–1792.

Kevin Lai and Dieter Fox. 2010. Object recognition in 3D point clouds using web data and domain adaptation. *Int. J. Rob. Res.* 29, 8, 1019–1037.

S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. 2012. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xueliang Liu and Benoit Huet. 2013. Heterogeneous features and model selection for event-based media classification. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval (ICMR'13)*. ACM, New York, 151–158.

Wenting Lu, Jingxuan Li, Tao Li, Weidong Guo, Honggang Zhang, and Jun Guo. 2013. Web multimedia object classification using cross-domain correlation knowledge. *IEEE Trans. Multimedia* 15, 8.

Ge Ma and Jiebo Luo. 2013. Is a picture worth 1000 votes? Analyzing the sentiment of election related social photos. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 1–6.

Zhigang Ma, Yi Yang, Yang Cai, Nicu Sebe, and Alexander G. Hauptmann. 2012. Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In *Proceedings of the ACM International Conference on Multimedia (MM'12)*. 469–478.

Mor Naaman. 2012. Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications. *Multimedia Tools Appl.* 56, 1, 9–34.

Jie Ni, Qiang Qiu, and Rama Chellappa. 2013. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 692–699.

Salvatore Orlando, Francesco Pizzolon, and Gabriele Tolomei. 2013. SEED: A framework for extracting social events from press news. In *Proceedings of the 22nd International Conference on World Wide Web Companion (WWW'13 Companion)*. International World Wide Web Conferences Steering Committee, 1285–1294.

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2009. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. 1187–1192.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 10, 1345–1359.

Georgios Petkos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2012. Social event detection using multimodal clustering and integrating supervisory signals. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR'12)*. ACM, New York, 23:1–23:8.

Guojun Qi, Charu C. Aggarwal, and Thomas S. Huang. 2011. Towards semantic knowledge propagation from text corpus to web images. In *Proceedings of the International World Wide Web Conference (WWW'11)*. 297–306.

Guo-Jun Qi, Charu C. Aggarwal, Qi Tian, Heng Ji, and Thomas S. Huang. 2012. Exploring context and content links in social media: A latent space method. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 5, 850–862.

Shengsheng Qian, Tianzhu Zhang, and Changsheng Xu. 2014a. Boosted multi-modal supervised latent Dirichlet allocation for social event classification. In *Proceedings of the 22nd International Conference on Pattern Recognition*.

Shengsheng Qian, Tianzhu Zhang, and Changsheng Xu. 2014b. Social event classification via boosted multimodal supervised latent Dirichlet allocation. *ACM Trans. Multimedia Comput. Commun. Appl.*

M. Sugiyama, A. Schwaighofer, N. D. Lawrence, and J. Candela Quiñonero, 2009. Covariate shift by kernel mean matching. In *Dataset Shift in Machine Learning*, MIT Press, 131–160.

Timo Reuter and Philipp Cimiano. 2012. Event-based classification of social media streams. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR'12)*. ACM, New York, 22:1–22:8.

Suman Deb Roy, Tao Mei, Wenjun Zeng, and Shipeng Li. 2012. SocialTransfer: Cross-domain transfer learning from social streams for media applications. In *Proceedings of the ACM International Conference on Multimedia (MM'12)*. 649–658.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision*. 213–226.

Cees Snoek, Marcel Worring, Jan van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. 2006. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the ACM International Conference on Multimedia*. 421–430.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining*. 1397–1405.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning*. 1096–1103.

Chang Wang and Sridhar Mahadevan. 2009. Manifold alignment without correspondence. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1273–1278.

Yanxiang Wang, Hari Sundaram, and Lexing Xie. 2012. Social event detection with interaction graph modeling. In *Proceedings of the 20th ACM International Conference on Multimedia (MM'12)*. ACM, New York, 865–868.

XuLei Yang, Qing Song, and Yue Wang. 2007. A weighted support vector machine for data classification. *Int. J. Pattern Recognit Artif Intell.* 21, 5, 961–976.

Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu. 2014. Cross domain feature learning in social multimedia. *IEEE Trans. Multimedia*.

Yi Yang, Zhigang Ma, Alexander G. Hauptmann, and Nicu Sebe. 2013. Feature selection for multimedia analysis by sharing information among multiple tasks. *IEEE Trans. Multimedia* 15, 3, 661–669.

Yi Yao and Gianfranco Doretto. 2010. Boosting for transfer learning with multiple sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1855–1862.

Maia Zaharieva, Matthias Zeppelzauer, and Christian Breiteneder. 2013. Automated social event detection in large photo collections. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval (ICMR'13)*. ACM, New York, 167–174.

Tianzhu Zhang, Bernard Ghanem, Si Liu, Changsheng Xu, and Narendra Ahuja. 2013. Low-rank sparse coding for image classification. In *Proceedings of the International Conference on Computer Vision*.

Tianzhu Zhang, Jing Liu, Si Liu, Yi Ouyang, and Hanqing Lu. 2009. Boosted exemplar learning for human action recognition. In *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops*.

Tianzhu Zhang, Jing Liu, Si Liu, Changsheng Xu, and Hanqing Lu. 2011. Boosted exemplar learning for action recognition and annotation. *IEEE Trans. Circuits Syst. Video Technol.* 21, 7, 853–866.

Tianzhu Zhang and Changsheng Xu. 2014. Cross-domain multi-event tracking via CO-PMHT. *ACM Trans. Multimedia Comput. Commun. Appl.* 10, 4, 31.

Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. 2009. Multi-class AdaBoost. *Statistics and Its Interface* 2, 3.

Received March 2014; revised August 2014; accepted September 2014