Cross-Domain Learning from Multiple Sources: A Consensus Regularization Perspective

Fuzhen Zhuang, Ping Luo, *Member*, *IEEE Computer Society*, Hui Xiong, *Senior Member*, *IEEE*, Yuhong Xiong, Qing He, and Zhongzhi Shi, *Senior Member*, *IEEE*

Abstract—Classification across different domains studies how to adapt a learning model from one domain to another domain which shares similar data characteristics. While there are a number of existing works along this line, many of them are only focused on learning from a single source domain to a target domain. In particular, a remaining challenge is how to apply the knowledge learned from multiple source domains to a target domain. Indeed, data from multiple source domains can be semantically related, but have different data distributions. It is not clear how to exploit the distribution differences among multiple source domains to boost the learning performance in a target domain. To that end, in this paper, we propose a consensus regularization framework for learning from multiple source domains to a target domain. In this framework, a local classifier is trained by considering both local data available in one source domain and the prediction consensus with the classifiers learned from other source domains. Moreover, we provide a theoretical analysis as well as an empirical study of the proposed consensus regularization framework. The experimental results on text categorization and image classification problems show the effectiveness of this consensus regularization learning method. Finally, to deal with the situation that the multiple source domains are geographically distributed, we also develop the distributed version of the proposed algorithm, which avoids the need to upload all the data to a centralized location and helps to mitigate privacy concerns.

Index Terms—Classification, multiple source domains, cross-domain learning, consensus regularization.

1 Introduction

Traditional learning techniques have the assumption that training and test data are drawn from the same data distribution, and thus they are not suitable for dealing with the situation where new unlabeled data are obtained from fast evolving, related but different information sources. This leads to the cross-domain learning problem which targets on adapting the knowledge learned from one or more source domains to target domains. In this paper, we focus on the classification task when source and target domains have different distributions but have the same feature sets in cross-domain learning.

Previous work is mainly focused on learning from a single *source domain* $\mathcal{D}_s$ to a *target domain* $\mathcal{D}_t$ [2], [3], [4]. However, in many real-world learning scenarios, there are training data from multiple source domains. These training data may follow different data distributions but are semantically related and share some commonality. Therefore, in this paper, we investigate the problem of cross-domain learning from multiple source domains to a target domain.

More specifically, we have m source domains as $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m$ and a target domain $\mathcal{D}_t$ (note that $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m$, and $\mathcal{D}_t$ also represent their corresponding data sets throughout this paper), and the labeled source domains and the unlabeled target domain may be geographically distributed. We assume that the class labels in $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m$ and the labels to be predicted in $\mathcal{D}_t$ are drawn from the same class-label set $\mathcal{C}$, and the labeled and unlabeled data share the same feature space $\Theta_{\mathcal{F}}$. We also assume that these source domains and the target domain are semantically related to each other in the sense that similar features would describe similar categories, but they have different distributions. Under this assumption, we study the problem of adapting the knowledge learned from these m source domains for classifying the unlabeled data in the target domain.

1.1 Motivating Examples

In practice, learning from multiple source domains can be a promising direction for cross-domain learning. Here, we provide the following two application scenarios to motivate this cross-domain learning problem. The first example is the application of **Webpage categorization**. Let us consider the task of using classification techniques to find course main pages from all the webpages in a university Website. To do this, we may create training data by manually labeling a collection of main course pages, while this needs a lot of human efforts. An alternative way is to use already-labeled main course pages from other universities as the training data. However, different universities usually use different templates for course pages, in which the terms used can also be different. For example,

- *F. Zhuang is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Kexueyuan Nanlu #6, Zhongguan Cun, Haidian District, Beijing, China 100190, and the Graduate University of Chinese Academy of Sciences, China. E-mail: zhuangfz@ics.ict.ac.cn.*
- *P. Luo and Y. Xiong are with the Hewlett-Packard Labs, China. E-mail: {ping.luo, yuhong.xiong}@hp.com.*
- *H. Xiong is with the Management Science and Information Systems Department, Rutgers Business School—Newark and New Brunswick, Rutgers, The State University of New Jersey, Washington Park, Washington Street, Newark, NJ 07102. E-mail: hxiong@rutgers.edu.*
- *Q. He and Z. Shi are with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Kexueyuan Nanlu #6, Zhongguan Cun, Haidian District, Beijing, China 100190. E-mail: {heq,shizz}@ics.ict.ac.cn.*

Manuscript received 18 Dec. 2008; revised 29 May 2009; accepted 7 Sept. 2009; published online 20 Nov. 2009.
Recommended for acceptance by S. Greco.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-12-0665.
Digital Object Identifier no. 10.1109/TKDE.2009.205.

Fig. 1. Image samples from the TRECVID collection including CCTV, CBC, CNN, and NBC.

the terms indicating the reading materials may include "Required Reading List," "Textbooks," "Reference," etc. Therefore, the data distributions of the main course webpages from different universities are likely to be different. Traditional classification learning will be difficult to use in this situation. Indeed, if we consider the labeled data from one university as one domain, and then the labeled data from multiple source domains could be obtained. Now, the goal is to find new course webpages in a target university via learning from multiple source domains. This is the cross-domain learning problem addressed in this paper.

To further motivate our work, let us consider the problem of **video concept detection**, which aims to generalize models built for detecting semantic concepts from multiple source video data to other domains. Here, a domain is a TV channel, such as CCTV, CBS, CNN, and NBC. For instance, the TRECVID collection [5] is such a multidomain video corpus which has news video from different TV channels. As shown in Fig. 1, video shots of easily recognizable anchors from four different famous TV channels exhibit dissimilar visual features. In other words, there exists "semantic gap" between visual features and semantic content. The problem of mismatch among the distributions of different domains is particularly severe in multimedia area [6]. As a result, concept classifiers trained from only one source domain might perform poorly on the target domain.

The common ground of the above two applications is that training data are from multiple semantically related source domains. One can argue that, if we merge these multiple source domains into one source domain, this problem can be solved by existing cross-domain learning algorithms. However, the important information, such as the distribution differences among these source domains, will be lost during the merging process. This information is the key to understand the common nature of these source domains.

1.2 Contributions

In our preliminary work [1], we proposed a *consensus regularization* framework that enables cross-domain learning from multiple source domains and incorporates the unlabeled data from the target domain into the learning process. For each source domain, this consensus optimization framework will output one classifier, which is trained by considering both the local data and the prediction consensus with the other classifiers on the unlabeled target domain data. In this mutually-affected manner, the resultant classifiers not only maintain the individuality of the corresponding source domains, but also reveal the common nature of all the source domains and the target domain. Additionally, the training data from different source domains might be geographically distributed, and it is difficult to put them into a centralized location due to the communication overhead, business concern, and/or privacy concern. Under this circumstance, we also implemented the learning algorithm in a distributed *master-slave* architecture, which only share some summary statistics rather than full contents of labeled data between the slave nodes and the master node. Therefore, this distributed algorithm can alleviate privacy concerns. Note that the consensus regularization framework can be exploited with many classification models, such as exponential family models. In this work, we implement it by using the *Logistic Regression* model [7].

In this paper, we further theoretically prove that maximizing the prediction consensus of local classifiers on target domain can improve the overall learning performance. In addition, we experimentally exploit the sources of performance gain of the proposed method. It indicates that leveraging distribution differences among source domains is very promising for multisource cross-domain learning. Finally, we provide systematic experiments on text categorization and image classification to validate the effectiveness of the proposed consensus regularization framework. These new experiments provide more insights into the consensus regularization framework, such as fast convergence and the generalization ability.

1.3 Outline

The rest of this paper is organized as follows: Section 2 describes some basic concepts about logistic regression and consensus measuring. In Section 3, we present the problem formulation, introduce the consensus regularization framework, and analyze why this consensus regularization framework works. Section 4 proposes the distributed learning algorithm for consensus regularization. In Section 5, we show the experimental results and analyze the sources of performance gain for the proposed algorithm. Finally, Section 6 gives the related work, followed by our conclusions in Section 7.

2 Preliminaries

In this section, we first introduce the notations used throughout this paper, and then present some preliminary concepts about logistic regression and consensus measuring.

2.1 Notations

In this paper, we use bold letters, such as $\mathbf{p}$ and $\mathbf{a}$, to represent vectors. Also, $\mathbf{p}_{(i)}$ indicates the ith element of $\mathbf{p}$. Random variables are written in upper case, such as X and Y. Therefore, bold upper case letters, such as $\mathbf{X}$ and $\mathbf{Y}$, are used to represent vectors of random variables. Calligraphic letters, such as $\mathcal{A}$ and $\mathcal{D}$, are used to represent sets. Finally, we use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{R}_+$ to denote the set of nonnegative real numbers.

TABLE 1
Entropy and Consensus of Probability Distribution Vectors

instance	$\mathbf{p}^1$ by h^1	$\mathbf{p}^2$ by h^2	$\mathbf{p}^3$ by h^3	$\overline{\mathbf{p}}$	Entropy	Consensus
$\mathbf{x}_1$	$(1,0,0)$	$(1,0,0)$	$(1,0,0)$	$(1,0,0)$	0	0
$\mathbf{x}_2$	$(0.7,0.25,0.05)$	$(0.8,0.1,0.1)$	$(0.6,0.25,0.15)$	$(0.7,0.2,0.1)$	$(\frac{7}{10}\log\frac{10}{7}+\frac{1}{5}\log 5+\frac{1}{10}\log 10)$	$-(\frac{7}{10}\log\frac{10}{7}+\frac{1}{5}\log 5+\frac{1}{10}\log 10)$
$\mathbf{x}_3$	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(\frac{1}{3},\frac{1}{3},\frac{1}{3})$	$\log 3$	-log 3

2.2 Logistic Regression

Logistic regression [7] is an approach to learn functions of $P(Y \mid \mathbf{X})$ in the case where Y is discrete-valued, and $\mathbf{X}$ is any vector containing discrete or continuous random variables. Logistic regression assumes a parametric form for the distribution $P(Y \mid \mathbf{X})$, then directly estimates its parameters from the training data. The parametric model assumed by logistic regression in the case where Y is Boolean is

$$P(y = \pm 1 \mid \mathbf{x}; \mathbf{w}) = \sigma(y\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-y\mathbf{w}^T\mathbf{x})}, \quad (1)$$

where $\mathbf{w}$ is the parameter of the model. Under the principle of *Maximum A-Posteriori* (MAP), $\mathbf{w}$ is estimated under the Laplacian prior. Given a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, we want to find the parameter $\mathbf{w}$ which maximizes:

$$\sum_{i=1}^{N} \log \frac{1}{1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i)} - \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}. \quad (2)$$

This criterion is a concave function of $\mathbf{w}$, so that the global solution can be obtained by the nonlinear numerical optimization methods. After $\mathbf{w}$ is estimated, (1) can be used to compute the probabilities of an instance belonging to the positive and negative classes.

2.3 Consensus Measuring

In this section, we first give the definition of *Shannon entropy* using a *probability distribution vector*, and then show how to measure the degree of consensus on the predictions that are made by a group of classifiers for an instance.

Definition 1 (Probability Distribution Vector). $\mathbf{p} \in \mathbb{R}_+^d$ *is a probability distribution vector if and only if* $\sum_{i=1}^d \mathbf{p}_{(i)} = 1$. *Each entry* $\mathbf{p}_{(i)}$ *of this vector represents the probability that this instance belongs to class* i.

Definition 2 (Shannon Entropy). *Assuming* $\mathbf{p} \in \mathbb{R}_+^d$ *is a probability distribution vector, then the Shannon entropy in* $\mathbf{p}$ *is defined as*

$$\mathbf{E}(\mathbf{p}) = \sum_{i=1}^{d} \mathbf{p}_{(i)} \log \frac{1}{\mathbf{p}_{(i)}}. \quad (3)$$

Given a group of m classifiers $\mathcal{H} = \{h^l\}_{l=1}^m$, each of which outputs a probability distribution vector $\mathbf{p}^l$ for an instance $\mathbf{x}$, the average probability distribution vector can be computed as:

$$\overline{\mathbf{p}} = \frac{\sum_{l=1}^m \mathbf{p}^l}{m}. \quad (4)$$

Then, using the Shannon entropy, we can measure the degree of consensus in these prediction results as shown in the examples of Table 1. For three-class classification problem, Table 1 records the probability distribution vectors of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ predicted by the classifiers h^1, h^2, and h^3, respectively, and their corresponding average probability distribution vectors. For the first instance $\mathbf{x}_1$, all the classifiers reach a perfect consensus that it belongs to class one with 100 percent probability. Therefore, the degree of consensus on these results reaches its maximum, while the entropy $\mathbf{E}(1,0,0)$ of the average distribution vector reaches its minimum for any three-entry distribution vectors. On the other hand, for the third instance $\mathbf{x}_3$, the three classifiers predict that it belongs to class one, two, and three, respectively with 100 percent probability. Thus, these prediction results totally disagree with each other, and their degree of consensus reaches its minimum. However, the entropy $\mathbf{E}(\frac{1}{3},\frac{1}{3},\frac{1}{3})$ of the average distribution vector reaches its maximum. Therefore, the negative of the entropy in the average probability distribution vector can be the consensus measure for the different prediction results. Formally, the definition of the entropy based consensus measure is:

Definition 3 (Entropy-Based Consensus Measure). *Given* m *probability distribution vectors* $\mathbf{p}^1, \ldots, \mathbf{p}^m$, *the consensus measure for these vectors is defined as*

$$\mathbf{C}_e(\mathbf{p}^1, \ldots, \mathbf{p}^m) = -\mathbf{E}(\overline{\mathbf{p}}), \quad (5)$$

where $\mathbf{E}$ *is the Shannon entropy in Definition 2 and* $\overline{\mathbf{p}}$ *is the average of these vectors defined by (4).*

Since we only consider the relative magnitude of two consensus measures, it is acceptable that the value of this consensus measure is negative. Thus, by this definition, the consensus degree for the prediction results of the second instance $\mathbf{x}_2$ is $-\mathbf{E}(0.7, 0.2, 0.1)$.

3 Problem Formulation and Consensus Regularization

In this section, we first formulate the problem of cross-domain learning from multiple source domains and then describe the principle of consensus regularization. Next, we analyze why and when this consensus regularization works for this problem formulation. Finally, we show how to exploit this principle for the logistic regression model.

3.1 Problem Formulation and Principle of Consensus Regularization

Let $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m$ be m $(\mathrm{m} > 1)$ source domains of labeled data, and the labeled data set from the lth source domain is represented by $\mathcal{D}_s^l = \{(\mathbf{x}_i^l, y_i^l)\}|_{i=1}^{n^l}$, where y_i^l is the label of $\mathbf{x}_i^l$ and n^l is the number of data in this domain.[1] The unlabeled target domain is denoted by $\mathcal{D}_t = \{(\mathbf{x}_i)\}|_{i=1}^n$, where n is the number of data objects in the target domain. Under the

1. Throughout this paper, the upper index of a letter denotes the index of source domains, while the lower index of a letter, if existing, denotes the index of the data.

assumption that the distributions of $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m, \mathcal{D}_t$ are different but related, we aim to train the classification models on these labeled source domains to better classify the unlabeled data in the target domain.

It is worth to mention that we assume the data distribution in the target domain is different from that of each source domains. There is a wide range of applications in which this assumption holds. Especially in the application that the source and target domain data are generated by different mechanism on different sites, this assumption can easily be satisfied. For example, when we aim to detect semantic concepts from the video data from different TV channels, we know that the data distributions of the same semantic concept from different TV channels are different. Therefore, we argue that the proposed method will be useful for these applications.

If we train m classifiers $h^1, \ldots, h^m$ locally, each of which is based only on the data from one source domain data, the ideal situation is that these m classifiers make a perfect consensus that they predict an instance from the target domain to be its ground truth with 100 percent confidence. However, since the distributions of $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m, \mathcal{D}_t$ are different, these initial m classifiers may disagree with each other to some degree on the prediction results of a certain instance. Thus, there is room to further maximize the consensus of these models on the prediction results of the data in the target domain. Therefore, we can incorporate this consensus measure into the standard framework of supervised learning as follows: This adaptive supervised learning framework with consensus regularization will output m models $h^1, \ldots, h^m$, which maximize the following equation:

$$\sum_{l=1}^{m} P(h^l \mid \mathcal{D}_s^l) + \theta \cdot Consensus(h^1, \ldots, h^m \mid \mathcal{D}_t), \quad (6)$$

where $P(h^l \mid \mathcal{D}_s^l)$ is the probability of the hypotheses h^l given the observed data set $\mathcal{D}_s^l$, $Consensus(h^1, \ldots, h^m | \mathcal{D}_t)$ is the consensus measure of these m models $h^1, \ldots, h^m$ on the prediction results of the data in the target domain $\mathcal{D}_t$, and θ is the trade-off parameter.

In the first term of (6), each model h^l is applied to its local source domain, while the second term in (6) is used as a bridge to link all these models, and realizes a mutual coupling optimization. In this way, each of these resultant models not only keeps the individuality of the corresponding local source domain, but also reveals the common nature of the target domain. Thus, this regularization framework maximizes not only the posteriori in each source domain, but also the consensus degree of these models.

Given a source domain data set $\mathcal{D}_s^l = \{(\mathbf{x}_i^l, y_i^l)\}|_{i=1}^{n^l}$ of *independent and identically-distributed* (IID) observations, the maximization of $P(h^l \mid \mathcal{D}_s^l)$ in the first term of (6) can be expanded further as follows:

$$\begin{aligned}\max P\big(h^l \mid \mathcal{D}_s^l\big) &= \max \frac{P(\mathcal{D}_s^l \mid h^l)P(h^l)}{P(\mathcal{D}_s^l)} = \max P\big(\mathcal{D}_s^l \mid h^l\big)P(h^l) \\ &= \max P(h^l) \cdot \prod_{i=1}^{n^l} P\big(y_i^l \mid \mathbf{x}_i; h^l\big) \\ &= \max\left(\log P(h^l) + \sum_{i=1}^{n^l} \log P\big(y_i^l \mid \mathbf{x}_i; h^l\big)\right). \end{aligned} \quad (7)$$

As to $Consensus(h^1, \ldots, h^m \mid \mathcal{D}_t)$, it is defined as the sum of the consensus measures of these m models on all the data in $\mathcal{D}_t$ and is shown as follows:

$$Consensus\big(h^1, \ldots, h^m \mid \mathcal{D}_t\big) = \sum_{i=1}^{n} \mathbf{C}_e\big(\mathbf{p}_i^1, \ldots, \mathbf{p}_i^m\big), \quad (8)$$

where $\mathbf{C}_e$ is the consensus measure in Definition 3, and $\mathbf{p}_i^l$ is the probability distribution vector predicted by the lth model h^l for the ith instance in the target domain $\mathcal{D}_t$. Indeed, this consensus measure has two benefits. One is to promote the degree of agreement on all the models, and the other is to minimize the entropy of the prediction results on the unlabeled data. Both will be further validated in Section 5.

3.2 Why Consensus Regularization?

In this section, we will theoretically show that maximizing agreement between any two classifiers could lead to the performance improvement of the individual classifiers.

To simplify the discussion, we consider a two-class classification problem with the labels 1 and -1. However, this analysis can be generalized to any d-class classification problems. We can train m models $h^1, \ldots, h^m$ for m source domains. Let Y be the target label, and the disagreement of any two individual models be $P(h^i \neq h^j)$ $(i, j \in \{1, \ldots, m\},\ i \neq j)$. Note that the symbol $P(\cdot)$ in this section is defined on the target domain. We also have the following three definitions:

Definition 4 (Nontrivial Classifier). *If a classifier h satisfies the condition*

$$P(h = u \mid Y = u) > P(h = \overline{u} \mid Y = u),$$

where $u \in \{-1, 1\}$ and $\overline{u}$ is the complement of u. Then, we call classifier h is a nontrivial classifier.

In other words, we can restate the nontrivial condition as

$$P(h = u \mid Y = u) > 1/2 \quad \text{or} \quad P(h \neq u \mid Y = u) \leq 1/2.$$

Definition 5 (Nonperfect Classifier). *If a classifier h gives a prediction performance less than 100 percent in accuracy on the target domain, we call it nonperfect classifier.*

Definition 6 (Conditional Independent Classifiers). *The conditional independence of models $h^1, \ldots, h^m$ is shown as follows:*

$$P(h^i = u \mid h^j = v, Y = y) = P(h^i = u \mid Y = y), \quad (9)$$

where $u, v, y \in \{-1, 1\}$, $i, j \in \{1, \ldots, m\}$, and $i \neq j$.

For conditional independent, nonperfect, and nontrivial classifiers, we have the following theorem:

Theorem 1. *If the condition that conditional independent assumptions are satisfied, it holds that the disagreement is a strict upper bound on the misclassification errors of nontrivial and nonperfect classifiers.*

Proof. The classification error of h^i is

$$\begin{aligned}P(h^i \neq Y) &= P(h^i = 1, Y = -1) + P(h^i = -1, Y = 1)\\ &= P(h^i = 1, h^j = -1, Y = -1)\\ &\quad + P(h^i = 1, h^j = 1, Y = -1)\\ &\quad + P(h^i = -1, h^j = -1, Y = 1)\\ &\quad + P(h^i = -1, h^j = 1, Y = 1),\end{aligned}$$

and the disagreement between h^i and h^j is

$$\begin{aligned}P(h^i \neq h^j) &= P(h^i = 1, h^j = -1) + P(h^i = -1, h^j = 1)\\ &= P(h^i = 1, h^j = -1, Y = -1)\\ &\quad + P(h^i = 1, h^j = -1, Y = 1)\\ &\quad + P(h^i = -1, h^j = 1, Y = -1)\\ &\quad + P(h^i = -1, h^j = 1, Y = 1),\end{aligned}$$

where $i, j \in \{1, \ldots, m\}$, $i \neq j$.

To validate that $P(h^i \neq Y) < P(h^i \neq h^j)$, we only have to prove the following inequation:

$$\begin{aligned}&P(h^i = 1, h^j = 1, Y = -1) + P(h^i = -1, h^j = -1, Y = 1)\\ &< P(h^i = 1, h^j = -1, Y = 1) + P(h^i = -1, h^j = 1, Y = -1).\end{aligned} \tag{10}$$

According to (9) and the Bayes Principle, (10) can also be written as follows:

$$\begin{aligned}&P(h^i = 1 \mid Y = -1)P(h^j = 1, Y = -1)\\ &\quad + P(h^i = -1 \mid Y = 1)P(h^j = -1, Y = 1)\\ &< P(h^i = 1 \mid Y = 1)P(h^j = -1, Y = 1)\\ &\quad + P(h^i = -1 \mid Y = -1)P(h^j = 1, Y = -1).\end{aligned} \tag{11}$$

By Definitions 4 and 5, following (12)-(15) hold,

$$P(h^i = 1 \mid Y = -1) < P(h^i = -1 \mid Y = -1), \tag{12}$$

$$P(h^i = -1 \mid Y = 1) < P(h^i = 1 \mid Y = 1). \tag{13}$$

$$P(h^i = -1, Y = 1) > 0. \tag{14}$$

$$P(h^i = 1, Y = -1) > 0. \tag{15}$$

Therefore, (11) holds.

Finally, we have

$$P(h^i \neq Y) < P(h^i \neq h^j). \tag{16}$$

□

As mentioned above, we can obtain m resultant classifiers by maximizing the objective function (6), denoted as $f^1, \ldots, f^m$, respectively. Thus, we also have the following theorem:

Theorem 2. *If the condition that conditional independent assumptions are satisfied, it holds that the accuracy is the strict upper bound of the agreement for nontrivial and nonperfect classifiers.*

Proof. According to Theorem 1, the disagreement is the strict upper bound of the misclassification error for nontrivial, nonperfect classifiers, then

$$p(f^i \neq Y) < p(f^i \neq f^j). \tag{17}$$

Then, we can have

$$\begin{aligned}1 - p(f^i = Y) &< 1 - p(f^i = f^j)\\ p(f^i = Y) &> p(f^i = f^j).\end{aligned} \tag{18}$$

□

Both Theorems 1 and 2 show that the proposed consensus regularization framework can effectively reduce the classification errors and improve the learning performances.

3.3 Implementation of Consensus Regularization by Logistic Regression

Here, we show how to exploit the principle of consensus regularization for the logistic regression model.

According to the problem formulation in Section 3.1, this consensus regularization framework outputs m logistic models $\mathbf{w}^1, \ldots, \mathbf{w}^m$, which maximize:

$$\begin{aligned}g_e(\mathbf{w}^1, \ldots, \mathbf{w}^m) = &\sum_{l=1}^{m}\left(\sum_{i=1}^{n^l} \log P\left(y_i^l \mid \mathbf{x}_i^l; \mathbf{w}^l\right) - \frac{\lambda^l}{2}\mathbf{w}^{lT}\mathbf{w}^l\right)\\ &- \theta \cdot \sum_{i=1}^{n} \mathbf{E}\left(\frac{\sum_{l=1}^{m} P(y = -1|\mathbf{x}_i; \mathbf{w}^l)}{m}, \right.\\ &\left.\frac{\sum_{l=1}^{m} P(y = 1|\mathbf{x}_i; \mathbf{w}^l)}{m}\right),\end{aligned} \tag{19}$$

where the conditional probability P is the logistic function defined in (1), and $\mathbf{E}$ is the Shannon entropy. Note that this regularization framework can work for multiclass problems as well.

Due to the computing complexity in the entropy, for two-entry probability distribution vectors, the consensus measure in (5) can be simplified as:

$$\begin{aligned}\mathbf{C}_s(\mathbf{p}^1, \ldots, \mathbf{p}^m) &= \left(\overline{\mathbf{p}}_{(1)} - \overline{\mathbf{p}}_{(2)}\right)^2 = \left(\overline{\mathbf{p}}_{(1)} - \left(1 - \overline{\mathbf{p}}_{(1)}\right)\right)^2\\ &= \left(2\overline{\mathbf{p}}_{(1)} - 1\right)^2.\end{aligned} \tag{20}$$

It is clear that, when comparing the relative magnitude of two degrees of consensus, $\mathbf{C}_e$ and $\mathbf{C}_s$ are equivalent for two-entry probability distribution vectors in the sense that they always give the same answer. Thus, substitute the entropy-based consensus measure with the equivalent form $\mathbf{C}_s$, the new objective function is

$$\begin{aligned}g_s(\mathbf{w}^1, \ldots, \mathbf{w}^m) = &\sum_{l=1}^{m}\left(\sum_{i=1}^{n^l} \log P\left(y_i^l \mid \mathbf{x}_i^l; \mathbf{w}^l\right) - \frac{\lambda^l}{2}\mathbf{w}^{lT}\mathbf{w}^l\right)\\ &+ \theta \cdot \sum_{i=1}^{n}\left(2\frac{\sum_{l=1}^{m} P(y = 1 \mid \mathbf{x}_i; \mathbf{w}^l)}{m} - 1\right)^2,\end{aligned} \tag{21}$$

where the conditional probability P is the logistic function defined in (1).

To simplify the discussion, in this paper, we only describe this regularization framework in (21) for two-class classification problems, but it can be extended to multiclass problems using the framework in (19). Thus, the partial differential of the objective g_s is

$$\nabla_{\mathbf{w}^l}(g_s) = \frac{\partial g_s}{\partial \mathbf{w}^l} = \nabla^l_{sn}\big(\mathbf{w}^l, \mathcal{D}^l_s\big) + \nabla^l_{mn}\big(\mathbf{w}^1, \ldots, \mathbf{w}^m, \mathcal{D}_t\big), \quad (22)$$

where the function σ is defined in (1), and

$$\nabla^l_{sn}\big(\mathbf{w}^l, \mathcal{D}^l_s\big) = \frac{\partial g_s}{\partial \mathbf{w}^l} = \sum_{i=1}^{n^l} \big(1 - \sigma\big(y^l_i \mathbf{w}^{l^T} \mathbf{x}_i\big)\big) y^l_i \mathbf{x}^l_i - \lambda^l \mathbf{w}^l, \quad (23)$$

$$\nabla^l_{mn}\big(\mathbf{w}^1, \ldots, \mathbf{w}^m, \mathcal{D}_t\big) = \frac{4\theta}{m^2} \sum_{i=1}^{n} \left(2 \sum_{k=1}^{m} \sigma\big(\mathbf{w}^{k^T} \mathbf{x}_i\big) - m \right) \big(1 - \sigma\big(\mathbf{w}^{l^T} \mathbf{x}_i\big)\big) \sigma\big(\mathbf{w}^{l^T} \mathbf{x}_i\big) \mathbf{x}_i. \quad (24)$$

Though the objective function in (21) is neither concave nor convex, for given initial values, the local optimization solution can also be obtained by nonlinear optimization technique. In this study, we adopt the *conjugate gradient* method [8] as the optimization technique (the reason why we adopt conjugate gradient is described in the experimental section), and the initial models are set to the ones trained on each local source domain separately. The pseudocode of our method is shown in Algorithm 1. To solve the subproblem in Step 4 of Algorithm 1, traditional optimization techniques can be used. In our implementation, the function *fminunc* provided by Matlab is adopted for Step 4.

Algorithm 1. Centralized Version of Consensus Regularization by Conjugate Gradient Ascent

Input: The labeled data sets $\mathcal{D}^1_s, \ldots, \mathcal{D}^m_s$, the unlabeled data set $\mathcal{D}_t$, the element matrix $Q \in \mathbb{R}^{k \times k}$ where $k = |\mathbf{x}|$ is the dimension of the data in the source domain, the error threshold $\varepsilon > 0$, and the maximum iterating number max.

Output: m classifiers $\mathbf{w}^1, \ldots, \mathbf{w}^m$.

1) Each source domain calculates the initial $\mathbf{w}^l_0$ by logistic regression based on its local data set $\mathcal{D}^l_s$
2) $k := 0$.
3) For $l = 1, \ldots, m$, compute the gradients $\nabla_{\mathbf{w}^l_k}(g_s)$ by (22), and set the searching directions as

$$\begin{cases} \mathbf{d}^l_0 = \nabla_{\mathbf{w}^l_0}(g_s) \\ \mathbf{d}^l_{k+1} = \nabla_{\mathbf{w}^l_{k+1}}(g_s) + \alpha_k \mathbf{d}^l_k \\ \alpha_k = -\nabla^T_{\mathbf{w}^l_{k+1}}(g_s) Q \mathbf{d}^l_k / \mathbf{d}^{l^T}_k Q \mathbf{d}^l_k \end{cases} \quad (25)$$

   If $\sum_{l=1}^m \|\nabla_{\mathbf{w}^l_k}(g_s)\| < \varepsilon$, then turn to Step 6.
4) Compute the best searching step $\gamma \geq 0$, which maximizes
$$u_s(\gamma) = g_s\big(\mathbf{w}^1_k + \gamma \mathbf{d}^1_k, \ldots, \mathbf{w}^m_k + \gamma \mathbf{d}^m_k\big). \quad (26)$$
   Then, for $l = 1, \ldots, m$, compute $\mathbf{w}^l_{k+1}$ by
$$\mathbf{w}^l_{k+1} = \mathbf{w}^l_k + \gamma \mathbf{d}^l_k. \quad (27)$$
5) $k := k + 1$. If $k \leq max$, then turn to Step 3.
6) Output $\mathbf{w}^1_k, \ldots, \mathbf{w}^m_k$.

4 Consensus Regularization in a Distributed Manner

In this section, we investigate how to extend this centralized consensus regularization method into a distributed learning algorithm, which can work in the situation where the source domains $\mathcal{D}^1_s, \ldots, \mathcal{D}^m_s$ and the target domain $\mathcal{D}_t$ are all distributed. In this distributed setting, the data nodes containing the source domain data are used as *slave nodes*, denoted by $sn^1, \ldots, sn^m$, and the data node containing the target domain data is used as the *master node*, denoted by mn.

Let us first revisit the partial differential of the objective g_s in (22), which consists of two parts. It is clear that the computation of the first term $\nabla^l_{sn}(\mathbf{w}^l, \mathcal{D}^l_s)$ needs only the local model $\mathbf{w}^l$ and the data set $\mathcal{D}^l_s$. Thus, it can be computed locally. The computation of the second term $\nabla^l_{mn}(\mathbf{w}^1, \ldots, \mathbf{w}^m, \mathcal{D}_t)$ involves all the models $\mathbf{w}^1, \ldots, \mathbf{w}^m$ and the target domain data set $\mathcal{D}_t$. Therefore, if the slave nodes sn^l $(l = 1, \ldots, m)$ sends $\mathbf{w}^l$ and ∇^l_{sn} to the master node mn, the master node can compute $\nabla_{\mathbf{w}^l}(g_s)$ by $\nabla_{\mathbf{w}^l}(g_s) = \nabla^l_{sn} + \nabla^l_{mn}$ in a straightforward manner.

As a result, if each round of the optimization process performs this synchronous communication of the statistic data between the slave nodes and the master node, the gradient $\nabla_{\mathbf{w}^l}(g_s)$ can be computed accurately. However, the maximization in the Step 4 of Algorithm 1 involves all the data from the source domains and the target domain, which is hard to be solved distributively. In order to extend Algorithm 1 to a distributed version, the searching step γ in it can be set to a constant. Then, the method of distributed consensus regularization is described in Algorithm 2, which is an approximation of Algorithm 1.

Algorithm 2. The Distributed Version of Consensus Regularization by Conjugate Gradient

Input: The labeled data sets $\mathcal{D}^1_s, \ldots, \mathcal{D}^m_s$ on the separated slave nodes $sn^1, \ldots, sn_m$, respectively, the unlabeled data set $\mathcal{D}_t$ on the master node mn, the error threshold $\varepsilon > 0$, the maximum iterating number max, and the step constant γ.

Output: m classifiers $\mathbf{w}^1, \ldots, \mathbf{w}^m$.

1) Each slave node sn^l $(l = 1, \ldots, m)$ calculates the initial $\mathbf{w}^l_0$ by logistic regression based on its local data set $\mathcal{D}^l_s$. Then they send this initial model $\mathbf{w}^l_0$ and the value of $\nabla^l_{sn}(\mathbf{w}^l_0, \mathcal{D}^l_s)$ $(l = 1, \ldots, m)$ to the master node mn.
2) $k := 0$.
3) The master node computes the gradients $\nabla_{\mathbf{w}^l_k}(g_s)$ $(l = 1, \ldots, m)$ for each model by (22), and sets the searching direction $\mathbf{d}^l_k$ as (25). If $\sum_{l=1}^m \|\nabla_{\mathbf{w}^l_k}(g_s)\| < \varepsilon$, then turn to Step 6; Otherwise, using the input constant γ, compute $\mathbf{w}^l_{k+1}$ as (27) for $l = 1, \ldots, m$.
4) The master node sends $\mathbf{w}^l_{k+1}$ $(l = 1, \ldots, m)$ to each slave node. Then each slave node computes $\nabla^l_{sn}(\mathbf{w}^l_{k+1}, \mathcal{D}^l_s)$ and sends it back to the master nodes.
5) $k := k + 1$. If $k \leq max$, then turn to Step 3.
6) Output $\mathbf{w}^1_k, \ldots, \mathbf{w}^m_k$.

In each round of Algorithm 2, each slave node sn^l sends a vector ∇^l_{sn} to the master node (in the first round, the slave node should also send the initial model to the master node), and the master node sends back the updated model. Therefore, if this process terminates after k iterations, the total communication overhead will be $(2k + 1) \sum_{l=1}^m |\mathbf{w}^l|$.

Fig. 2. Eight samples from the data sets *flower* and *traffic*.

TABLE 2
The Description of Image Data

flower	*flower.sunflower*	*flower.rose*	*flower.lotus*	*flower.tulip*
No. of samples	85	100	66	100
traffic	*traffic.aviation*	*traffic.bus*	*traffic.boat*	*traffic.dogsled*
No. of samples	100	100	100	100
No. of features	87	87	87	87

Note that this distributed process communicates only some summary statistics, such as ∇_{sn}^{l} $(l = 1, \ldots, m)$ and the classification models, without sending the raw source domain data. Therefore, this can also alleviate privacy concerns.

5 Experimental Evaluation

In this section, we empirically evaluate the performance of the proposed methods. In the experiments, we focus on the two-class classification problem. However, it is straightforward to extend the proposed methods for multiclass classification.

5.1 Data Preparation

The data preparation method is similar to the one in [3]. Since publicly available data are not originally designed for cross-domain learning from multiple source domains, we need to do some data preprocessing. It requires that the experimental data have at least a two-level hierarchical structure. In this paper, we assume $\mathcal{A}$ and $\mathcal{B}$ are two root categories in a data set, and $\mathcal{A}_1, \ldots, \mathcal{A}_4$ and $\mathcal{B}_1, \ldots, \mathcal{B}_4$ are the four sublevel categories of $\mathcal{A}$ and $\mathcal{B}$, respectively. These sublevel categories are used as three source domains and one target domain. Next, we construct the training and test data as follows: For $i = 1, \ldots, 4$, let $\mathcal{A}_{a_i}$ and $\mathcal{B}_{a_i}$ be the positive and negative instances in the ith domain $\mathcal{D}_{a_i} = \mathcal{A}_{a_i} \cup \mathcal{B}_{a_i}$, respectively; and $\mathcal{A}_{a_i}$ and $\mathcal{B}_{a_i}$ appear once and only once in these domains. In this way, the positive (negative) data from different domains are similar since they belong to the same top category $\mathcal{A}$ ($\mathcal{B}$), and the positive (negative) data from different domains are still different since they belong to different subcategories. Thus, these four domains have different but similar data distributions. We can then select any one of these four domains as the target domain, and the other three domains as the source domains. Therefore, given $\mathcal{A}_1, \ldots, \mathcal{A}_4$ and $\mathcal{B}_1, \ldots, \mathcal{B}_4$, we can construct 96 $(4 \cdot P_4^4)$ problem instances of three-source domains cross-domain learning problems. Note that, in this study, the number of the source domains for cross-domain learning is set to three.

5.1.1 Text Categorization

20 Newsgroup[2] is a benchmark data set for text categorization. We selected three top categories *sci*, *talk*, and *comp*, denoted by $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. The four subcategories in *sci* are *sci.crypt*, *sci.electronics*, *sci.med*, and *sci.space*, denoted by $\mathcal{A}_1, \ldots, \mathcal{A}_4$, respectively. The four subcategories in *talk* are *talk.politics.guns*, *talk.politics.mideast*, *talk.politics.misc*, and *talk.religion.misc*, denoted by $\mathcal{B}_1, \ldots, \mathcal{B}_4$, respectively. And the top category *comp* contains *comp.graphics*, *comp.os.ms-windows.misc*, *comp.sys.ibm.pc.hardware*, and *comp.sys.mac.hardware*, which are denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_4$, respectively. We can randomly select two top categories to construct 96 problem instances, but in this paper, we only list the evaluation results of two data sets *sci* versus *talk* and *comp* versus *talk*. The threshold of *Document Frequency* with the value of five is used to select the features.

5.1.2 Image Classification

Two top categories *flower* and *traffic* from the COREL collection[3] are constructed. The four subcategories in *flower* are *flower.sunflower*, *flower.rose*, *flower.lotus*, and *flower.tulip*. The four subcategories in *traffic* are *traffic.aviation*, *traffic.bus*, *traffic.boat*, and *traffic.dogsled*. Eight samples from the image data sets are shown in Fig. 2 and the detailed description is shown in Table 2. For each image, we extract 87-dimension features, including 36-dimension color histogram [9] and 51-dimension SILBP texture histogram [10].

2. http://people.csail.mit.edu/jrennie/20Newsgroups/.
3. http://wang.ist.psu.edu/docs/related.shtml.

5.2 Benchmark Methods and Evaluation Metrics

5.2.1 Benchmark Classification Methods

We evaluate the proposed multisource cross-domain learning algorithms in two scenarios: distributed and centralized.

Distributed approach. In this approach, the algorithm is implemented in a distributed manner. The simplest distributed approach is Distributed Ensemble (DE), where a classifier is trained based on the local data on each source domain. The other distributed approach is Distributed Consensus Regularization (DCR) described in Algorithm 2. In both DE and DCR, the prediction is made by the ensemble method through majority voting with equal weights.

Centralized approach. In this approach, all the data from the source domains and the target domain are accumulated and processed at a centralized node. In this case, the simplest method is Centralized Training (CT) which trains a global classifier on all the data. Meanwhile, if all the data from the source domains are put together as one labeled data set, Centralized Consensus Regularization (CCR) in Algorithm 1 with $m = 1$, denoted by CCR_1, can be used. The cross-domain learning method CoCC [3] and the semisupervised techniques TSVM [11] as well as SGT [12] can also be applied to this situation. On the other hand, in order to explicitly consider that the centralized data are from three different source domains, CCR with $m = 3$, denoted by CCR_3, is also adopted.

In summary, our proposed method DCR, CCR_1, and CCR_3 are compared with DE, CT, CoCC, TSVM, and SGT. Note that when the parameter θ in the objective function (21) is set to 0, the consensus regularization is useless. Thus, at this time the individual classifiers are trained only on their local source domain data respectively, then DCR is equivalent to DE, and CCR_1 is equivalent to CT. Also, DCR with a small step constant achieves the same accuracy performance as CCR_3 at the cost of some communication overhead. Therefore, these two algorithms DCR and CCR_3 are denoted by CCR_3 only.

Details of implementation. Some initial experiments show that several popular nonlinear optimization techniques output similar models for the proposed optimization problem and conjugate gradient is the fastest one. Therefore, we use conjugate gradient in this paper. After some preliminary test, we find that λ^l is not very sensitive in the value range [10, 300], so the λ^l in the objective function (21) is set to 145 (for $l = 1, \ldots, m$).[4] And the value range of θ is [0, 0.25]. In the algorithms of consensus regularization, the maximal iterating number max is set to 200, and the error threshold ε is set to 0.1. Before conducting the optimization for consensus regularization, Logistic Regression[5] is performed on each source domain to obtain the initial values of the model for further optimization. The parameters of CoCC, TSVM, and SGT are the same as those in [3].

5.2.2 Evaluation Metrics

The performance of the comparison methods is evaluated by accuracy. Let c be the function which maps each instance to its true class label, and f be the function which maps each instance to its prediction label given by the classifier. Accuracy is defined as

$$a = \frac{|\{\mathbf{d}|\mathbf{d} \in \mathcal{D}_t \wedge c(\mathbf{d}) = f(\mathbf{d})\}|}{|\mathcal{D}_t|}. \tag{28}$$

We also measure the consensus degree of multiple classifiers on the target domain as follows: Let h be the function which maps each instance to the probability distribution vector predicted by the classifier. This consensus degree of m classifiers $h_1, \ldots, h_m$ is defined as

$$c = \frac{\sum_{\mathbf{d} \in \mathcal{D}_t} \sqrt{\mathbf{C}_s(h_1(\mathbf{d}), \ldots, h_m(\mathbf{d}))}}{|\mathcal{D}_t|}, \tag{29}$$

where $\mathbf{C}_s$ is defined in (20). It is clear that these m classifiers reach perfect consensus when c reaches its maximal value 1.

5.3 Summary of the Problems

To highlight the significance of the experiments performed in this section, we will outline the problems to be answered as follows:

- In general, to validate the superiority and effectiveness of the proposed algorithm, we compared it with the benchmark methods in Section 5.4.1, including: 1) Comparison of CCR_3, DE, and CT and 2) Comparison of CCR_3, TSVM, SGT, and CoCC (Section 5.4).
- Second, we analyze the reasons why our algorithm can obtain a significant improvement by benchmark methods for the cross-domain learning task, and conjecture that there are three sources of the gain: 1) In the multiple source domains paradigm, exploiting distribution differences among the source domains is very important to boost the performance of the proposed approach. 2) Note that the consensus regularization not only has the effect of minimizing entropy but also the effect of maximizing consensus. Therefore, these are the other two sources of the gain (Section 5.5).
- Third, it is worth mentioning that our consensus regularization framework is actually an inductive algorithm, so we investigate the generalization ability of the output classifiers and how many unlabeled samples are sufficient for the optimization process (Section 5.6).
- Finally, we studied the convergence property of the proposed algorithm. Fast convergence is desired for an iterative algorithm (Section 5.7).

5.4 Performance Comparison

5.4.1 Comparison of CCR_3, DE, and CT

We have three data sets described in Section 5.1, and 96 problem instances are constructed for each data set. For each problem instance, we record the values of accuracy and consensus for the resultant classifiers of algorithm CCR_3 on different values of θ. Table 3 gives an example of these measures for one of these problem instances on data *sci versus talk*. Due to the space limitation, we cannot list all the 96×3 tables. However, the properties in these tables are similar, as can be seen later in this section.

For each θ, Algorithm CCR_3 outputs three classifiers on the corresponding three source domains. These classifiers are tested on their own source domains and the target

4. For the image data, λ^l is set to 0.05 after some preliminary parameter tuning experiments.

5. http://research.microsoft.com/~minka/papers/logreg/.

TABLE 3
The Accuracy (Percent) and Consensus Measure in an Example Problem

θ	The classifier on $\mathcal{D}_s^1$		The classifier on $\mathcal{D}_s^2$		The classifier on $\mathcal{D}_s^3$		Consensus	CCR_3	CCR_1
	Acc. on $\mathcal{D}_s^1$	Acc. on $\mathcal{D}_t$	Acc. on $\mathcal{D}_s^2$	Acc. on $\mathcal{D}_t$	Acc. on $\mathcal{D}_s^3$	Acc. on $\mathcal{D}_t$		Acc. on $\mathcal{D}_t$	Acc. on $\mathcal{D}_t$
0	100	71.10	99.95	55.75	100	72.10	0.2775	73.94	71.99
0.05	100	92.14	99.95	90.61	99.94	93.20	0.6459	93.46	74.47
0.1	100	93.14	99.95	92.67	99.94	92.93	0.7385	93.30	76.11
0.15	100	93.83	99.95	93.46	99.89	93.88	0.7861	93.72	77.90
0.2	100	93.25	99.95	92.99	99.89	93.20	0.8146	93.25	80.43
0.25	100	93.35	99.95	92.99	99.89	93.14	0.8349	93.20	81.12

Source domains: $\mathcal{D}_s^1(\mathcal{A}_2,\mathcal{B}_4)$, $\mathcal{D}_s^2(\mathcal{A}_1,\mathcal{B}_2)$, $\mathcal{D}_s^3(\mathcal{A}_4,\mathcal{B}_3)$; Target domain: $\mathcal{D}_t(\mathcal{A}_3,\mathcal{B}_1)$

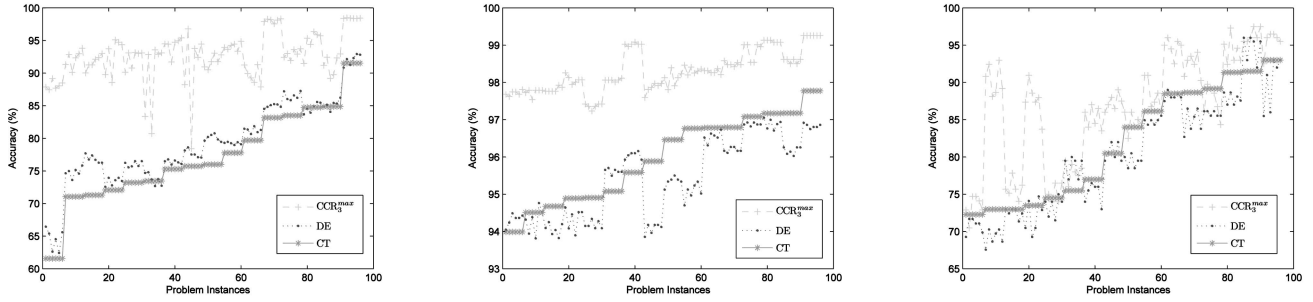


Fig. 3. The performance comparison on three data sets. (a) CCR_3^{max} versus DE and CT on *sci versus talk*. (b) CCR_3^{max} versus DE and CT on *comp versus talk*. (c) CCR_3^{max} versus DE and CT on *flower versus traffic.*

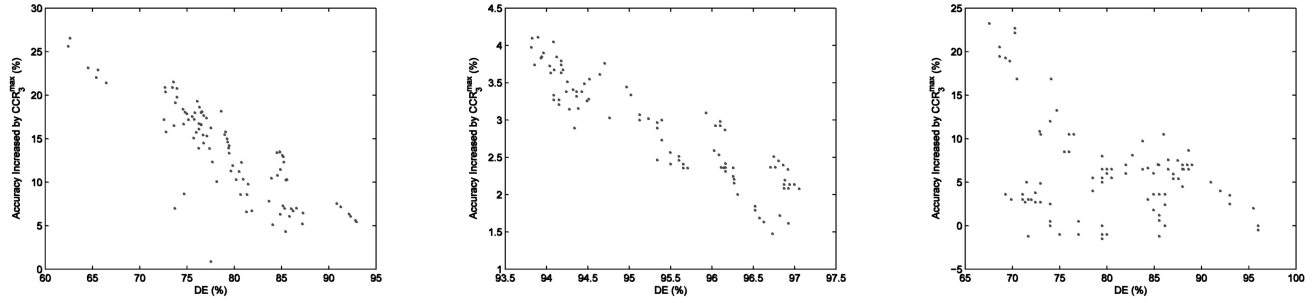


Fig. 4. The relationship between DE versus CCR_3^{max} on three data sets. (a) DE versus CCR_3^{max} on *sci versus talk*. (b) DE versus CCR_3^{max} on *comp versus talk*. (c) DE versus CCR_3^{max} on *flower versus traffic*.

domain, and the results are recorded from the second to seventh column of Table 3. Then, the eighth column records the consensus measure of three classifiers. Finally, the accuracy performances of CCR_3 and CCR_1 are shown in the 9th and 10th column of this table. As mentioned above, when $\theta = 0$ (ignoring the consensus regularizer), the accuracy for CCR_3 (73.94 at the first row and ninth column of Table 3) is that of DE, and the accuracy for CCR_1 (71.99 at the first row and 10th column of Table 3) is that of CT.

The results in Table 3 show that: 1) When $\theta \neq 0$, CCR_3 always outperforms DE and CT. 2) When $\theta \neq 0$, the performances of the local classifiers tested on their own source domains are stable (always near 100 percent). Additionally, under this situation, the performances of the local classifiers tested on the target domain increase significantly. For example, the performance of the classifier on $\mathcal{D}_s^2$ increases from 55.75 percent to more than 90 percent when it is applied to the target domain. 3) When θ increases, the consensus measure of the resultant three classifiers increases. When this consensus measure reaches some extent, the classifiers always output the same results for an instance. So the performances of these classifiers tested on the target domain are almost equal to that of CCR_3 when $\theta \neq 0$.

To further validate this on the other 95 tables, for each of these tables, we measure four values: 1) the performance of DE; 2) the performance of CT; 3) the average performance of CCR_3 when θ is sampled in [0.05, 0.25], denoted by $\overline{\mathrm{CCR}_3}$; and 4) the best performance of CCR_3 when θ is sampled in [0.05, 0.25], denoted by CCR_3^{max}; For each of the four numbers, we can average its values on all the 96 problem instances. These results are shown in Figs. 3 and 4, and Table 4. In Fig. 3, the 96 problem

TABLE 4
Average Values (Percent) on 96 Problem Instances

Data set	CCR_3^{max}	$\overline{\mathrm{CCR}_3}$	DE	CT
sci vs. talk	92.66	90.42	79.21	77.19
comp vs. talk	98.29	98.08	95.43	95.97
flower vs. traffic	87.54	85.89	80.50	81.97

TABLE 5
The Data Description for the Performance Comparison among TSVM, SGT, CoCC, and CCR_3

Data Set	$\mathcal{D}_s$		$\mathcal{D}_t$
	$\mathcal{D}_s^1$	$\mathcal{D}_s^2$	
comp vs. sci	comp.graphics sci.crypt	comp.os.ms-windows.misc sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x, sci.med, sci.space
rec vs. talk	rec.autos talk.politics.guns	rec.motorcycles talk.politics.misc	rec.sport.baseball, talk.religion.misc rec.sport.hockey, talk.politics.mideast
rec vs. sci	rec.autos sci.space	rec.sport.baseball sci.med	rec.motorcycles, rec.sport.hockey sci.crypt, sci.electronics
sci vs. talk	sci.electronics talk.religion.misc	sci.med talk.politics.misc	sci.crypt, talk.politics.guns sci.space, talk.politics.mideast

instances are sorted by the increasing order of the performances by CT.

Fig. 3 shows that CCR_3^{max} almost outperforms DE and CT on all 96 problem instances, which prove the effectiveness of the proposed algorithm. In Figs. 4a and 4b, the x-axis represents the accuracy of DE while the y-axis represents the performance difference between CCR_3^{max} and DE. Figs. 4a and 4b show that this performance improvement decreases when the performance of DE increases. The reason is that if the accuracy of DE is high, these original classifiers usually output the same right results, and the consensus measure of the classifiers is big. In this case, the room for further increasing this consensus measure is very limited, and thus the improvement by consensus regularization is small. Although the size of samples in image data sets is small, Fig. 4c also reveals the similar trend as Figs. 4a and 4b.

Table 4 lists the average values of the four accuracy measures over the 96 problem instances on three data sets. Note that the same performance of CCR_3^{max} can be achieved by DCR in a distributed manner. Compared with DE, the maximal accuracy of DCR increases from 79.21 to 92.66.

5.4.2 Comparison of CCR_3, $TSVM$, SGT, and $CoCC$

To compare CCR_3 with CoCC, TSVM, and SGT, we select the data sets in [3] which can be modified to fit our problem setting. We divide the single source domain of the original problem into multiple source domains. Four data sets, which are described in Table 5, are selected for this comparison. We measure the performance of CCR_3 on these problems by the two values $\overline{\mathrm{CCR}_3}$ and CCR_3^{max}. The experimental results in Table 6 show that both $\overline{\mathrm{CCR}_3}$ and CCR_3^{max} outperform TSVM and SGT on these four data sets.

TABLE 6
The Performance Comparison Results (Percent) among TSVM, SGT, CoCC, and CCR_3

Data Set	TSVM	SGT	CoCC	$\overline{\mathrm{CCR}_3}$	CCR_3^{max}
comp vs. sci	81.7	72.1	87.0	91.4	93.1
rec vs. talk	96.0	90.9	96.5	97.8	98.0
rec vs. sci	93.8	93.8	94.5	96.3	96.7
sci vs. talk	89.2	91.7	94.6	92.8	93.6

Except that CoCC slightly outperforms CCR_3 on the fourth data set, both $\overline{\mathrm{CCR}_3}$ and CCR_3^{max} are better than CoCC on the other three data sets.

5.5 Source of the Performance Gain

In this section, we investigate the key reasons why the consensus regularization can lead to a better learning performance.

5.5.1 Exploiting the Distribution Differences among the Source Domains

For the consensus regularization framework, the distribution differences among the source domains are very important. However, it is very difficult to measure these differences in a precise way. To simplify the discussion, we will focus on two levels of distribution differences. The first level is that every source domain has a specific distribution (as depicted in Section 5.1), and the other situation we consider is that the source domains have slightly different distributions. If we merge the m source domains into one source domain $\mathcal{D}_s$ and then randomly divide it into m (here, $m = 3$) source domains, denoted as $\mathcal{SD}_s^1, \ldots, \mathcal{SD}_s^m$, the distribution differences of the source domains $\mathcal{SD}_s^1, \ldots, \mathcal{SD}_s^m$ are much smaller than that of the source domains $\mathcal{D}_s^1, \ldots, \mathcal{D}_s^m$. We reperformed the experiments on the source domains $\mathcal{SD}_s^1, \ldots, \mathcal{SD}_s^m$, and also recorded the four values of each table, denoted as SDE, SCT, SCCR_3^{max}, and $\overline{\mathrm{SCCR}_3}$ (Note that $\mathrm{SCT} = \mathrm{CT}$). Fig. 5 shows the results on both the original source domains and the randomly generated source domains. As can be seen, the performance improvements on the original source domains are much better than the one on the randomly generated source domains. The t-test with 95 percent confidence also shows that the accuracy improvement difference $((\mathrm{CCR}_3^{max} - \mathrm{DE}) - (\mathrm{SCCR}_3^{max} - \mathrm{SDE}))$ is statistically significant. The results validate the effectiveness of exploiting the distribution differences among the source domains to increase the learning performance.

5.5.2 Effects of Consensus Regularization and Entropy Minimization

As mentioned in Section 3.1, the regularization framework proposed in this paper has two effects—consensus regularization and entropy minimization. We compare the performance of CCR_3^{max}, CCR_1^{max}, and DE (CT is similar to DE). Algorithm CCR_1^{max} is regarded to only have entropy

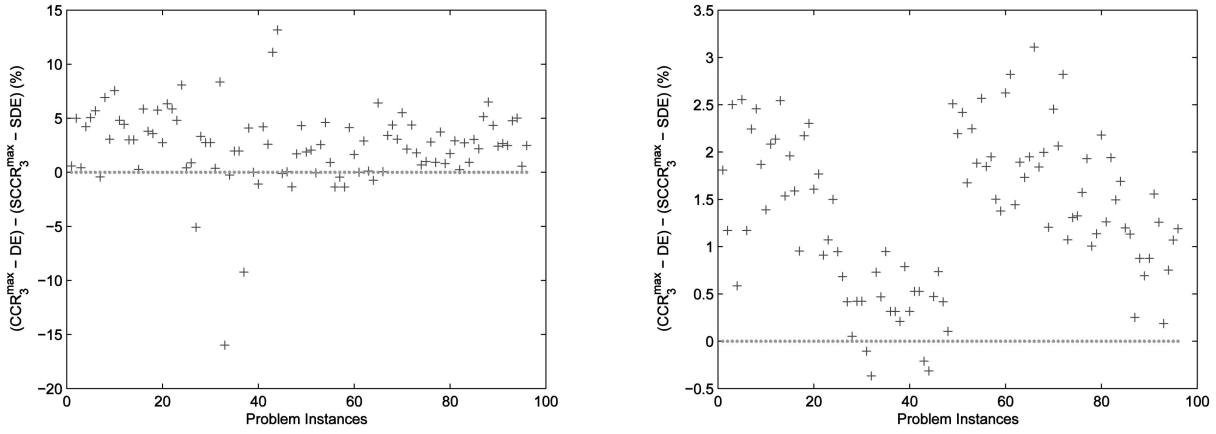


Fig. 5. Exploiting the distribution differences among the dource domains. (a) Problem Instances versus $(\mathrm{CCR}_3^{max} - \mathrm{DE}) - (\mathrm{SCCR}_3^{max} - \mathrm{SDE})$ on *sci versus talk*. (b) Problem Instances versus $(\mathrm{CCR}_3^{max} - \mathrm{DE}) - (\mathrm{SCCR}_3^{max} - \mathrm{SDE})$ on *comp versus talk*.

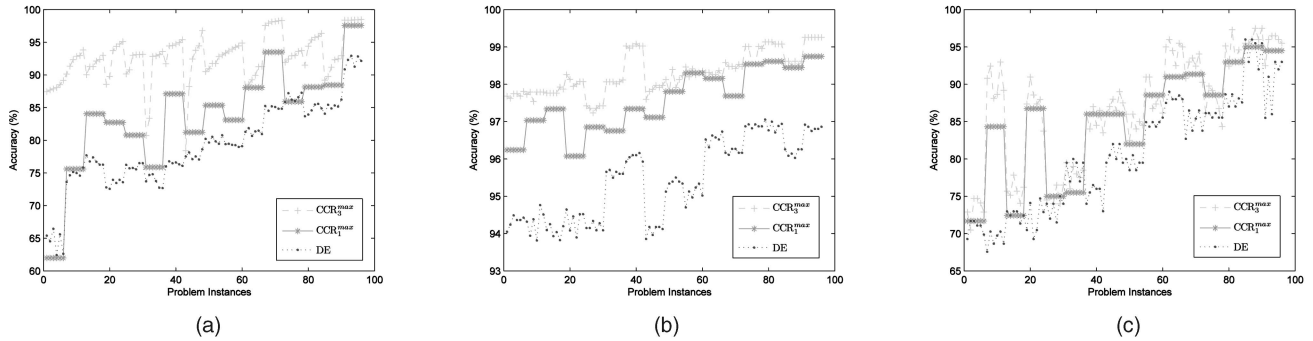


Fig. 6. The performance comparison on three data sets. (a) CCR_3^{max} versus CCR_1^{max} and DE on *sci versus talk*. (b) CCR_3^{max} versus CCR_1^{max} and DE on *comp versus talk*. (c) CCR_3^{max} versus CCR_1^{max} and DE on *flower versus traffic*.

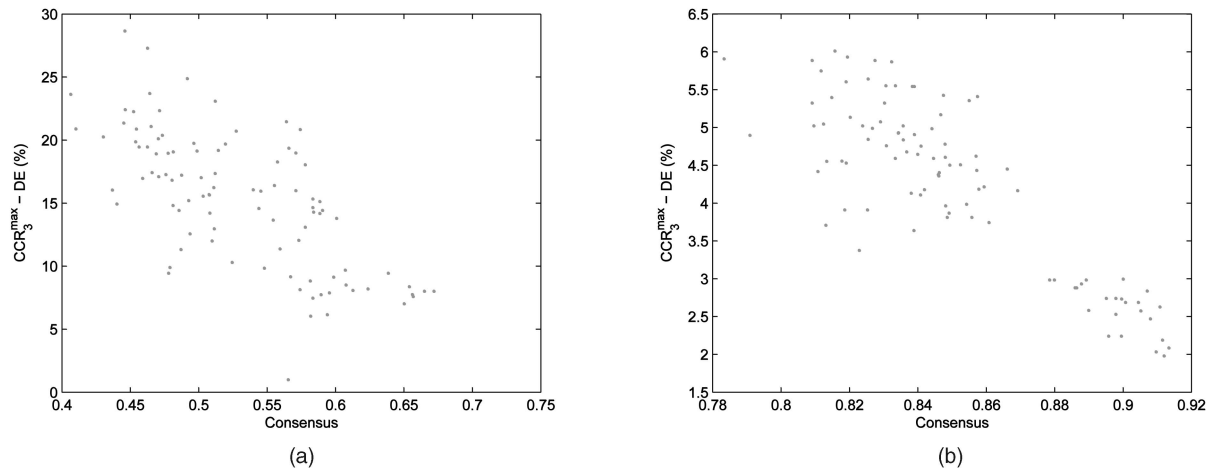
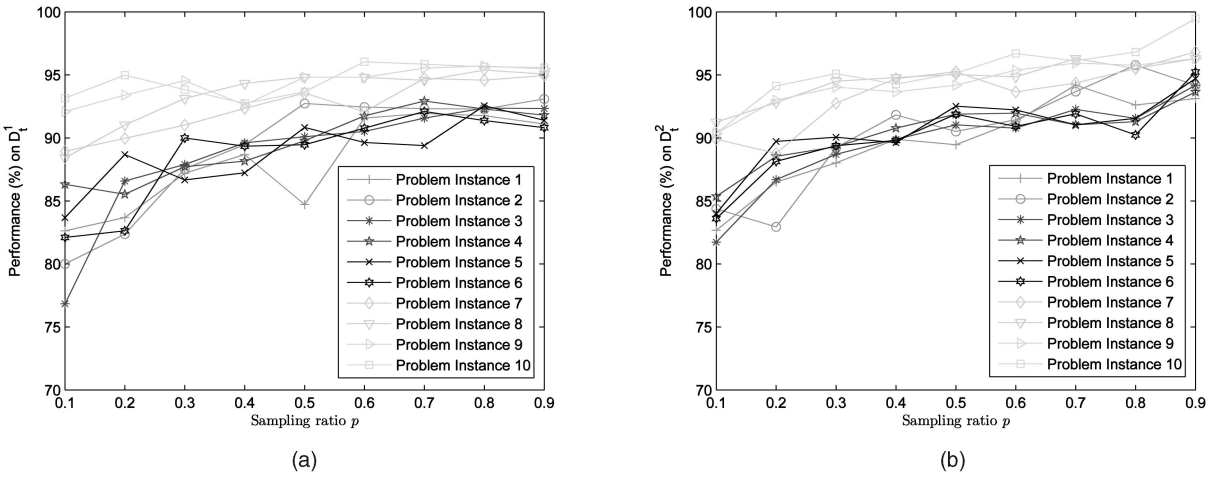


Fig. 7. The relationship between consensus and accuracy improvement. (a) Consensus versus $\mathrm{CCR}_3^{max} - \mathrm{DE}$ on *sci versus talk*. (b) Consensus versus $\mathrm{CCR}_3^{max} - \mathrm{DE}$ on *comp versus talk*.

minimization factor not the effect of consensus regularization, because there is only one classifier in the optimization process. All the results on three data sets are shown in Fig. 6. In the figure, we can find that: 1) CCR_1^{max} is better than DE. This validates the effect of entropy minimization and 2) CCR_3^{max} is better than CCR_1^{max}. This shows that our consensus regularization framework not only exploits the entropy minimization but also the consensus maximization of classifiers when being applied to multiple source domains.

5.5.3 Relationship between Consensus and Performance Improvement

To emphasize the theme of this work, we investigate how the consensus influences the performance improvement. The consensus (defined in (29)) of classifiers predicting on the target domain, and the accuracy improvement after optimization are recorded and shown in Fig. 7. Note that the consensus affects the classifiers trained on the source domains rather than the optimized output classifiers. In



(a) (b)

Fig. 8. Generalization ability affected by sampling ratio p. (a) Performance of CCR_3^{max} on *sci versus talk*. (b) Performance of CCR_3^{max} on *sci versus talk*.

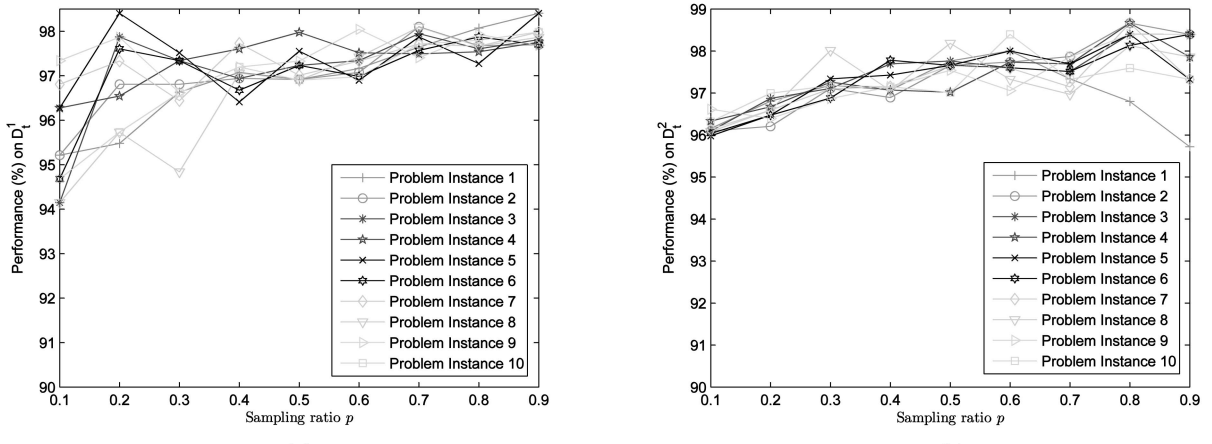
Fig. 7, the x-axis represents consensus of classifiers while the y-axis represents the performance improvement of CCR_3^{max} compared with DE. Fig. 7 shows that the accuracy improvement decreases when the value of consensus increases, which indicates that the more consistent of classifiers predicting on target domain, the smaller of performance improvement room. To some extend, this requirement of classifier diversity is identical to the boosting algorithm, such as Bagging, Adaboost [13], [14], and so on.

5.6 Inductive Setting of Consensus Regularization

It is worth mentioning that our consensus regularization is an inductive algorithm and can output classifiers for the unseen test data. In this section, we evaluate the generalization ability of the output classifiers on text data sets *sci versus talk* and *comp versus talk*. Specifically, we randomly sample (without replacement) ratio p of the data in the target domain $\mathcal{D}_t$ to form a new data set $\mathcal{D}_t^1$, and the left data in $\mathcal{D}_t$ to form the other data set $\mathcal{D}_t^2$. Then, the unlabeled data in $\mathcal{D}_t^1$ are used for the classifier optimization in the training process, and the unlabeled data in $\mathcal{D}_t^2$ are used to test the generalization ability of the classifiers output by the training process. We also test the accuracy of these classifiers on $\mathcal{D}_t^1$. Additionally, we test the generalization ability under different values of p, and sample the ratio p with interval 0.1 in the bound $[0.1, 0.9]$. In each data set, the first 10 problem instances are selected, and the results are shown in Figs. 8 and 9. From the results, we can find that: 1) The increase of the unlabeled data in $\mathcal{D}_t^1$ used for training improves the generalization ability of the output classifiers. 2) When $p \geq 0.6$ the accuracy of the output classifiers on $\mathcal{D}_t^2$ is almost the same as the results when all the unlabeled target domain data are used for training in Section 5.4.1. These results show that the proposed algorithm has a good generalization ability when more than 60 percent of target domain data are used for optimization process.

5.7 Algorithm Convergence

We check the convergence property of CCR_3 on six randomly selected problem instances on the data set *sci versus talk*. These results are shown in Fig. 10, where the x-axis represents the number of iterations and the y-axis represents the accuracy performance. It can be observed that, for each problem instance, the accuracy performance increases as the number of iterations and almost converges after 30 iterations. This indicates that the proposed algorithm converges very quickly.



(a) (b)

Fig. 9. Generalization ability affected by sampling ratio p. (a) Performance of CCR_3^{max} on *comp versus talk*. (b) Performance of CCR_3^{max} on *comp versus talk*.

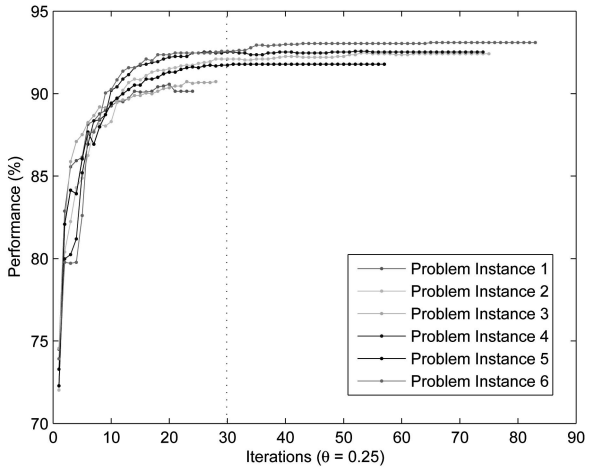


Fig. 10. Iterations versus the performance of CCR_3 on *sci versus talk*.

6 Related Work

In this section, we introduce some related work in the fields of transfer learning (cross-domain learning), semisupervised classification, and multiview learning.

Transfer Learning aims to solve the fundamental problem of mismatched distributions between the training and testing data. It is also referred to as *Cross-Domain Learning* or *Domain Adaptation*, which adapts the knowledge from source domains (*in-domain*, *auxiliary-domain*) to a target domain (*out-of-domain*, *primary-domain*). In general, previous works in this area can be grouped into two categories. The first category is under the assumption that there are some labeled data from the target domain with different distribution. For instance, Liao et al. [15] estimated the degree of mismatch of each instance in the source domain with the whole target domain, and incorporated this information into logistic regression. Also, Dai et al. [2] extended boosting-based learning algorithms to transfer learning, in which the source domain data with very different distribution are less weighted for data sampling. They also analyzed the theoretical effectiveness of this algorithm using the *Probability Approximately Correct* (PAC) theory. In addition, Yang et al. [6] studied the problem of transform the existing classifier from the source domain to a target domain in an incremental way. The principle behind this transformation is that the difference between the classifiers before and after adaption should be as small as possible. This work involves the data from multiple source domains, but it does not consider the distribution difference among these source domains. Duan et al. [16] proposed a Domain Adaptation Machine (DAM) that learns from multiple sources via auxiliary classifiers. In this approach, they induce a new target classifier which shares similar decision value with the auxiliary classifiers (trained from the source domains) on unlabeled target domain and minimizes the empirical error on labeled target domain, while we refine the existing auxiliary classifiers during the optimization. All the above works need some labeled data from target domain, however, labeled data are often very difficult and expensive to obtain.

In the second category, for the problem that the data from the target domain are totally unlabeled, Ling et al. [17] developed a new spectral classification algorithm that optimizes an objective function to seek for the maximal consistency between the supervised information from the source domain and the intrinsic structure of the target domain. Gao et al. [18] proposed a heterogenous source consensus learning framework that efficiently negotiates multiple heterogenous sources/models and internal structure (unsupervised clustering) of target domain. However, these methods need sufficient test data to represent the intrinsic structure. In addition, Jiang [19] developed a two-phase feature selection framework for domain adaptation. Although this method considered multiple source domains, it was different from our consensus regularization. In their approach, they first selected the features called general features among all domains, and then learned a general classifier by emphasizing those general features. Second, they made use of unlabeled data from target domain to pick up features that are specifically useful for the target domain. Finally, Gao et al. [20] proposed a multiple model local structure mapping scheme that attaches different weights to the models by the local manifold structure of testing samples. Also, Xing et al. [4] proposed a transductive learning algorithm for this problem. Their method performs a two-phase label propagation, which is based on the adjacent matrix of the data. However, the methods in [4], [20] cannot output the classifier for new unlabeled data. The proposed method in this paper falls into this category of transfer learning.

Moreover, none of the above existing works consider the problem of transfer learning from multiple source domains to a target domain in a distributed manner. These methods were not developed for the situation that the training data for transfer learning are geographically distributed. Also, we explicitly leverage the distribution differences among the source domains and the target domain in our model to further improve the learning performance in the target domain.

Semisupervised Classification uses a large amount of unlabeled data, together with the labeled data, to achieve better prediction on unlabeled data. Different from transfer learning, the labeled and unlabeled data in semisupervised learning are from the same distribution. To the best of our knowledge, the most related work in this area is semisupervised learning by entropy minimization [21]. To compare with this method in the same problem setting, we assume that the labeled data and unlabeled data consist of a source domain and target domain, respectively. The regularization framework in [21] is recognized as an instance of the objective (19) with $m = 1$ (m is the number of source domains). In other words, our regularization approach is more general and includes this method as a special case. Furthermore, our consensus regularization method is designed in a distributed manner and has the effect of consensus regularization.

Multiview Learning is a new and natural, but nonstandard learning problem, where the data are represented by multiple independent sets of features. As an example, Yarowsky [22] as well as Blum and Mitchell [23] noticed that having multiple representations can improve the performance of semisupervised classification. Sindhwani et al. [24] proposed a coregularization approach, but this regularization term do not have the effect of entropy minimization, which is utilized in our work. In addition, Dasgupta et al. [25] and Abney [26], [27] provided PAC bounds on the error of co-Training in terms of the

disagreement rate of hypotheses on unlabeled data in two independent views. This inspires the principle of consensus maximization, which says that by minimizing the disagreement rate on unlabeled data, the error rate can be minimized. Our work utilizes this principle to another problem setting: transfer learning from multiple domains. From the point of view of data partition, the difference between multiview learning and multidomain learning is that the data are vertically partitioned for multiview learning while they are horizontally partitioned for multidomain learning.

7 Conclusions

In this paper, we study the problem of cross-domain learning from multiple source domains to a target domain. Specifically, for the case that data from multiple source domains and the target domain are semantically related, but have different data distributions, we propose a consensus regularization framework to exploit the distribution differences and learn the knowledge among training data from multiple source domains to boost the learning performance in a target domain. In this framework, we design a distributed learning algorithm, in which a local classifier is trained in each source domain by considering both local data and the prediction consensus with the classifiers from other source domains. To combine the learning results from multiple sources, only summary statistics, rather than the whole data, are required to transfer among these distributed domains. This greatly reduces the communication cost and mitigates the privacy concerns. In addition, we provide a theoretical analysis on the key advantages of the proposed consensus regularization method. As demonstrated in the experimental results, the consensus regularization learning method can effectively improve the learning performance in the target domain by leveraging the knowledge learned from multiple source domains.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 60675010, 60933004, 60975039), 863 National High-Tech Program (No. 2007AA01Z132), National Basic Research Priorities Programme (No.2007CB311004), and National Science and Technology Support Plan (No. 2006BAC08B06). Also, this research was supported in part by the US National Science Foundation (NSF) via grant number CNS 0831186 and the Rutgers Seed Funding for Collaborative Computing Research. Finally, the authors are grateful to the anonymous referees for their constructive comments on the paper. A preliminary version of this work has been published in the *Proceedings of the 17th ACM Conference on Information and Knowledge Mining (CIKM)* [1]. Ping Luo was the corresponding author for this paper.

References

[1] P. Luo, F.Z. Zhuang, H. Xiong, Y.H. Xiong, and Q. He, "Transfer Learning from Multiple Source Domains via Consensus Regularization," *Proc. 17th ACM Conf. Information and Knowledge Mining (CIKM),* pp. 103-112, 2008.

[2] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for Transfer Learning," *Proc. 24th Int'l Conf. Machine Learning (ICML),* pp. 193-200, 2007.

[3] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Co-Clustering Based Classification for Out-of-Domain Documents," *Proc. 13th ACM SIGKDD,* pp. 210-219, 2007.

[4] D. Xing, W. Dai, G. Xue, and Y. Yu, "Bridged Refinement for Transfer Learning," *Proc. 11th European Conf. Practice of Knowledge Discovery in Databases (PKDD),* pp. 324-335, 2007.

[5] A. Smeaton and P. Over, "TRECVID: Benchmarking the Effectiveness of Information Retrieval Tasks on Digital Video," *Proc. Image and Video Retrieval,* pp. 451-456, 2003.

[6] J. Yang, R. Yan, and A.G. Hauptmann, "Cross-Domain Video Concept Detection Using Adaptive SVMs," *Proc. 15th Int'l Conf. Multimedia,* pp. 188-197, 2007.

[7] D. Hosmer and S. Lemeshow, *Applied Logistic Regression.* Wiley, 2000.

[8] A. Ruszczynski, *Nonlinear Optimization.* Princeton Univ. Press, 2006.

[9] L. Zhang, "The Research on Human-Computer Cooperation in Content-Based Image Retrieval," PhD thesis, Tsinghua Univ., Beijing, 2001 (in Chinese).

[10] Z.P. Shi, F. Ye, Q. He, and Z.Z. Shi, "Symmetrical Invariant LBP Texture Descriptor and Application for Image Retrieval," *Proc. Congress on Image and Signal Processing,* pp. 825-829, 2008.

[11] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning (ICML),* pp. 200-209, 1999.

[12] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," *Proc. 20th Int'l Conf. Machine Learning (ICML),* pp. 290-297, 2003.

[13] B. Leskes and L. Torenvliet, "The Value of Agreement, A New Boosting Algorithm," *J. Computer and System Sciences,* vol. 74, no. 4, pp. 557-586, 2005.

[14] T.G. Dietterich, "Ensemble Methods in Machine Learning," *Lecture Notes in Computer Science,* vol. 1857, pp. 1-15, Springer, 2000.

[15] X. Liao, Y. Xue, and L. Carin, "Logistic Regression with an Auxiliary Data Source," *Proc. 22nd Int'l Conf. Machine Learning (ICML),* pp. 505-512, 2005.

[16] L. Duan, I.W. Tsang, D. Xu, and T.S. Chua, "Domain Adaptation from Multiple Sources via Auxiliary Classifiers," *Proc. 26th Int'l Conf. Machine Learning (ICML),* pp. 289-296, 2009.

[17] X. Ling, W.Y. Dai, G.R. Xue, Q. Yang, and Y. Yu, "Spectral Domain-Transfer Learning," *Proc. 14th ACM SIGKDD,* pp. 488-496, 2008.

[18] J. Gao, W. Fan, Y.Z. Sun, and J.W. Han, "Heterogeneous Source Consensus Learning via Decision Propagation and Negotiation," *Proc. 15th ACM SIGKDD,* 2009.

[19] J. Jiang, "Domain Adaptation in Natural Language Processing," PhD thesis, Dept. of Computer Science, Graduate College of the Univ. of Illinois at Urbana-Champaign, 2008.

[20] J. Gao, W. Fan, J. Jiang, and J.W. Han, "Knowledge Transfer via Multiple Model Local Structure Mapping," *Proc. 14th ACM SIGKDD,* pp. 283-291, 2008.

[21] Y. Grandvalet and Y. Bengio, "Semi-Supervised Learning by Entropy Minimization," *Proc. 19th Conf. Neural Information Processing Systems (NIPS),* pp. 529-536, 2005.

[22] D. Yarowsky, "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods," *Proc. 33rd Ann. Meeting of the Assoc. for Computational Linguistics (ACL),* pp. 189-196, 1995.

[23] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory,* pp. 92-100, 1998.

[24] V. Sindhwani, P. Niyogi, and M. Belkin, "A Co-Regularization Approach to Semi-Supervised Learning with Multiple Views," *Proc. 22nd Int'l Conf. Machine Learning (ICML) Workshop Learning with Multiple Views,* pp. 74-79, 2005.

[25] S. Dasgupta, M.L. Littman, and D.A. McAllester, "PAC Generalization Bounds for Co-Training," *Proc. 15th Conf. Neural Information Processing Systems (NIPS),* pp. 375-382, 2001.

[26] S. Abney, "Bootstrapping," *Proc. 40th Ann. Meeting of the Assoc. for Computational Linguistics (ACL),* 2002.

[27] S. Abney, "Understanding the Yarowsky Algorithm," *Computational Linguistics,* vol. 30, no. 3, pp. 365-395, 2004.

Fuzhen Zhuang is a PhD candidate in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include machine learning, data mining, distributed classification and clustering, and natural language processing.

Ping Luo received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a research scientist in the Hewlett-Packard Labs, China. His general area of research is knowledge discovery and machine learning. He has published several papers in some prestigious refereed journals and conference proceedings, such as the *IEEE Transactions on Information Theory*, *IEEE Transactions on Knowledge and Data Engineering*, *Journal of Parallel and Distributed Computing*, the ACM SIGKDD, and the ACM CIKM. He is the recipient of the President's Exceptional Student Award, Institute of Computing Technology, CAS, in 2007. He is a member of the IEEE Computer Society and the ACM.

Hui Xiong received the BE degree from the University of Science and Technology of China, the MS degree from the National University of Singapore, and the PhD degree from the University of Minnesota. He is currently an associate professor in the Management Science and Information Systems Department at Rutgers University. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published more than 60 technical papers in peer-reviewed journals and conference proceedings. He is a coeditor of *Clustering and Information Retrieval* (Kluwer Academic Publishers, 2003) and a co-editor-in-chief of *Encyclopedia of GIS* (Springer, 2008). He is an associate editor of the *Knowledge and Information Systems* journal and has served regularly in the organization committees and the program committees of a number of international conferences and workshops. He was the recipient of the 2008 IBM ESA Innovation Award, the 2009 Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence, the 2007 Junior Faculty Teaching Excellence Award, and the 2008 Junior Faculty Research Award at the Rutgers Business School. He is a senior member of the IEEE, and a member of the ACM.

Yuhong Xiong received the BS degree in electronic engineering from Tsinghua University in Beijing, China, and the PhD degree in electrical engineering and computer sciences from UC Berkeley. He is a senior research scientist and project manager at Hewlett-Packard Labs, China. His current research interests include Web mining, information extraction, human-computer collaboration, and information management.

Qing He received the BS degree in mathematics from Hebei Normal University, Shijiazhang, P.R. China, and the MS degree in mathematics from Zhengzhou University, P.R. China, in 1985 and 1987, respectively. He received the PhD degree in fuzzy mathematics and artificial intelligence from Beijing Normal University, P.R. China, in 2000. He is a professor in the Institute of Computing Technology, Chinese Academy of Science (CAS), and he is a professor at the Graduate University of Chinese Academy of Science (GUCAS). Since 1987 to 1997, he has been with Hebei University of Science and Technology. He is currently a doctoral tutor at the Institute of Computing and Technology, CAS. His interests include data mining, machine learning, classification, and fuzzy clustering.

Zhongzhi Shi is a professor in the Institute of Computing Technology, CAS, leading the Research Group of Intelligent Science. His research interests include intelligence science, multiagent systems, Semantic Web, machine learning, and neural computing. He has won a second-Grade National Award at Science and Technology Progress of China in 2002, two second-Grade Awards at Science and Technology Progress of the Chinese Academy of Sciences in 1998 and 2001, respectively. He is a senior member of the IEEE, a member of the AAAI and the ACM, chair for the WG 12.2 of IFIP. He serves as vice president for Chinese Association of Artificial Intelligence.