

Transfer Learning with Multiple Sources via Consensus Regularized Autoencoders

Fuzhen Zhuang¹, Xiaohu Cheng^{1,2}, Sinno Jialin Pan³,
Wenchao Yu^{1,2}, Qing He¹, and Zhongzhi Shi¹

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190, China

² University of Chinese Academy of Sciences, Beijing 100049, China

³ Institute for Infocomm Research, Singapore

{zhuangfz, chengxh, yuwc, heq, shizz}@ics.ict.ac.cn
sinnocat@gmail.com

Abstract. Knowledge transfer from multiple source domains to a target domain is crucial in transfer learning. Most existing methods are focused on learning weights for different domains based on the similarities between each source domain and the target domain or learning more precise classifiers from the source domain data jointly by maximizing their consensus of predictions on the target domain data. However, these methods only consider measuring similarities or building classifiers on the original data space, and fail to discover a more powerful feature representation of the data when transferring knowledge from multiple source domains to the target domain. In this paper, we propose a new framework for transfer learning with multiple source domains. Specifically, in the proposed framework, we adopt autoencoders to construct a feature mapping from an original instance to a hidden representation, and train multiple classifiers from the source domain data jointly by performing an entropy-based consensus regularizer on the predictions on the target domain. Based on the framework, a particular solution is proposed to learn the hidden representation and classifiers simultaneously. Experimental results on image and text real-world datasets demonstrate the effectiveness of our proposed method compared with state-of-the-art methods.

Keywords: Transfer Learning, Multiple Sources, Consensus Regularization, Feature Representation.

1 Introduction

Transfer learning or domain adaptation aims to extract common knowledge across domains such that a model trained on one domain can be adapted effectively to other domains [16]. In the past decade, a number of transfer learning methods have been proposed, most of which are focused on the 1vs1 transfer learning setting, where only one source domain and one target domain are assumed to be available when knowledge is transferred. However, in many real-world scenarios, given a target domain, there may be more than one source domain available for building classifiers. In this case, how to fully utilize multiple sources to ensure effective knowledge transfer is crucial.

So far, there are several works proposed for transfer learning with multiple source domains [8,27,7,4,9]. Most of them are focused on learning weights for different domains based on the similarities between each source domain and the target domain or learning more precise classifiers from the source domain data jointly by maximizing their consensus of predictions on the target domain data. For instance, Gao et al. [8] proposed a lazy ensemble method for multi-source transfer learning. Specifically, a number of supervised classifiers are trained from the source domains, then given an instance in the target domain, its local structure constructed in the source domains is used to estimate the weights for different source-domain classifiers to make predictions. Zhuang et al. [27] proposed a consensus regularization framework for multi-source transfer learning, where classifiers trained on multiple source domains are optimized jointly not only to achieve high prediction results on the corresponding domains, but also to make consistent predictions on target domain data. Similarly, Chattopadhyay et al. [4] introduced a transfer learning framework based on the multi-source domain adaptation methodology for detecting different stages of fatigue using surface electromyography signals. The works [7,4,9] need a few labeled data in the target domain, while in our work there are only labeled data in the source domains.

A common characteristic of most transfer learning methods with multiple domains is that knowledge transfer is performed on the original data space. However, in many applications, the supports of features of different domains may not be the same. In other words, there may exist domain-specific features in different domains, e.g., different product domains have their specific opinion words [1,14]. In this case, adapting models on the original data space may not be able to transfer knowledge effectively. Moreover, in many other applications, the data observed may be very complex, e.g., sensor signals. In this case, measuring similarity or dissimilarity between domains on the original data space may not be precise, which may limit the transferability across domains [13,15]. To address these issues, another branch of methods, which is referred to as the feature-based transfer learning approach, has been proposed in the 1vs1 transfer learning setting. The motivation of this approach is to learn a feature mapping or transformation to map the original data to a new feature space where the difference or distance between different domains can be reduced implicitly or explicitly.

Motivated by the idea of the feature-based methods in the 1vs1 transfer learning setting, in this paper, we propose an embedding-based framework for multi-source transfer learning. Specifically, in the proposed framework, we first adopt autoencoders [10] to construct a feature mapping to map an original instance to a hidden representation. Note that this mapping is shared by all the source and target domain data. We then train multiple classifiers on different source domain labeled data with the hidden representation jointly by introducing an entropy-based consensus regularizer on the predictions on the target domain data with the hidden representation. Based on the framework, a particular solution is proposed to learn the hidden representation and consensus regularized classifiers simultaneously. Different from the existing work proposed by Zhuang et al. [27], where a consensus regularizer is performing on the original data space, our model instead of a hidden feature space. We believe the great success of representation learning of autoencoders can lead to better transferability of our framework. As will be shown in the Experimental section, extensive experiments on image and text datasets verify our

hypotheses and demonstrate the superiority of our proposed framework over a variety of state-of-the-art methods.

2 Notations and Preliminaries

In this section, we first introduce some frequently used notations as presented in Table 1, and some preliminaries which will be used in our proposed framework.

Table 1. The Notation and Denotation

$\mathcal{D}^{(i)}$	A data domain i
r	The number of source domains
m	The number of original features of a data domain
n_i	The number of instances of a data domain i
k	The number of hidden features
\mathbf{x}	An original instance
y	A class label
$\hat{\mathbf{x}}$	The reconstruction of \mathbf{x}
\mathbf{z}	An embedded instance
\mathbf{W}, \mathbf{b}	A weight matrix and bias vector of encoding
\mathbf{W}', \mathbf{b}'	A weight matrix and bias vector of decoding
$\boldsymbol{\theta}_i$	A vector of parameters of a classifier i
\top	The transposition of a matrix
\circ	The dot product of vectors or matrixes

2.1 Logistic Regression

In our proposed framework, we adopt logistic regression [6] as the base classifier. Note that the proposed framework is general, thus other types of classifiers can also be plugged into our framework. The goal of logistic regression is to estimate a conditional probability $P(y|\mathbf{x})$ in terms of a vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^{m \times 1}$ by solving the following maximization problem,

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^n \log \sigma(y_i \boldsymbol{\theta}^\top \mathbf{x}_i) - \lambda \boldsymbol{\theta}^\top \boldsymbol{\theta}, \quad (1)$$

over a set of labeled data $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$ is an input instance, y_i is its correspondingly discrete output, e.g., for binary classification $y_i \in \{-1, 1\}$, and $\sigma(u)$ is a sigmoid function defined as follows,

$$\sigma(u) = \frac{1}{1 + e^{-u}}. \quad (2)$$

The second term in (1) is a regularization term to avoid overfitting, where the trade-off parameter λ is a small positive constant. After $\boldsymbol{\theta}$ is estimated, the conditional probability of y given x can be computed by

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \sigma(y \boldsymbol{\theta}^\top \mathbf{x}), \quad (3)$$

which is used to classify target domain data, i.e., the predicted label of \mathbf{x} is $\max_y p(y|\mathbf{x}; \boldsymbol{\theta})$.

2.2 Autoencoders

An autoencoder first maps an input instance \mathbf{x} to a hidden representation \mathbf{z} through an encoding mapping:

$$\mathbf{z} = h(\mathbf{W}\mathbf{x} + \mathbf{b}),$$

where h is a nonlinear activation function, $\mathbf{W} \in \mathbb{R}^{k \times m}$ is a weight matrix, and $\mathbf{b} \in \mathbb{R}^{k \times 1}$ is a bias vector. The resulting latent representation \mathbf{z} is then mapped back to a reconstruction $\hat{\mathbf{x}}$ through a decoding mapping:

$$\hat{\mathbf{x}} = g(\mathbf{W}'\mathbf{z} + \mathbf{b}'),$$

where g is a nonlinear activation function, $\mathbf{W}' \in \mathbb{R}^{m \times k}$ is a weight matrix, and $\mathbf{b}' \in \mathbb{R}^{m \times 1}$ is a bias vector. Given a set of inputs $\{\mathbf{x}_i\}_{i=1}^n$, the parameters of an autoencoder are optimized by minimizing the reconstruction error as follows,

$$\min_{\mathbf{W}, \mathbf{b}, \mathbf{W}', \mathbf{b}'} = \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2. \quad (4)$$

Note that, in this paper we adopt the sigmoid function σ defined in (2), which is widely used in constructing autoencoders, as the nonlinear activation functions g and h for encoding and decoding respectively.

2.3 Consensus Measure

Given r classifiers in terms of their parameter vectors $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_r)$ and an instance \mathbf{x} , for a specific class c , we denote $(p_1(c), p_2(c), \dots, p_r(c))$ a vector of the predicted probabilities $P(y = c | \mathbf{x})$ of the r classifiers accordingly. Then the consensus measure of the predictions of the r classifiers on \mathbf{x} is given by

$$\psi(\mathbf{x}; \{\boldsymbol{\theta}_i\}_{i=1}^r) = - \sum_{c \in \mathcal{C}} \bar{p}(c) \log \frac{1}{\bar{p}(c)}, \quad (5)$$

where $\bar{p}(c) = \frac{1}{r} \sum_{i=1}^r p_i(c)$, and \mathcal{C} is the total set of classes. As shown in [27], maximizing (5) is equivalent to enforcing the r classifiers to make consistent predictions on \mathbf{x} as well as minimizing the entropy of the predictions of each classifier on \mathbf{x} . For binary classification, (5) can be rewritten as

$$\psi(\mathbf{x}; \{\boldsymbol{\theta}_i\}_{i=1}^r) = (\bar{p} - (1 - \bar{p}))^2 = (2\bar{p} - 1)^2. \quad (6)$$

Note that we say (5) and (6) are equivalent for binary classification in the sense that they have the same effect that: when maximizing them, the predictions on any instance from all the classifiers (from the different domains) are similar. Thus, their effects on making the prediction consensus are similar, though their value scales are not the same. In this paper, we focus on binary classification, thus adopt (6) as the consensus measure in the following section.

3 Consensus Regularized Autoencoders

3.1 Problem Formalization

Given r source domains $\mathcal{D}_S^{(1)}, \dots, \mathcal{D}_S^{(r)}$, where for each source domain $j \in \{1, \dots, r\}$, there are n_j labeled data, i.e., $\mathcal{D}_S^{(j)} = \{\mathbf{x}_{S_i}^{(j)}, y_{S_i}^{(j)}\}_{i=1}^{n_j}$, where $y_{S_i}^{(j)} \in \{-1, 1\}$, and a target domain \mathcal{D}_T without any labeled data, i.e., $\mathcal{D}_T = \{\mathbf{x}_{T_i}, y_{T_i}\}_{i=1}^n$, the goal is to train a classifier f to make precise predictions on \mathcal{D}_T or previously unseen instances in the target domain. Note that, in our transfer scenario there is not any labeled data in the target domain.

Our proposed optimization problem for multi-source transfer learning is formulated as follows,

$$\min_{\Theta, \Theta', \{\theta_j\}} \mathcal{J} = \epsilon(\mathbf{x}_S, \hat{\mathbf{x}}_S, \mathbf{x}_T, \hat{\mathbf{x}}_T) + \gamma \Omega(\Theta, \Theta') + \alpha \ell(\mathbf{z}_S, y_S; \{\theta_j\}) - \beta \psi(\mathbf{z}_T; \{\theta_j\}), \quad (7)$$

where the first term in the objective is the reconstruction error of the source and target domain data, which can be written as follows,

$$\epsilon(\mathbf{x}_S, \hat{\mathbf{x}}_S, \mathbf{x}_T, \hat{\mathbf{x}}_T) = \sum_{j=1}^r \sum_{i=1}^{n_j} \|\mathbf{x}_{S_i} - \hat{\mathbf{x}}_{S_i}\|^2 + \sum_{i=1}^n \|\mathbf{x}_{T_i} - \hat{\mathbf{x}}_{T_i}\|^2,$$

and

$$\begin{aligned} \mathbf{z}_{S_i}^{(j)} &= \sigma(\mathbf{W} \mathbf{x}_{S_i}^{(j)} + \mathbf{b}), \quad \mathbf{z}_{T_i} = \sigma(\mathbf{W} \mathbf{x}_{T_i} + \mathbf{b}), \\ \hat{\mathbf{x}}_{S_i}^{(j)} &= \sigma(\mathbf{W}' \mathbf{z}_{S_i}^{(j)} + \mathbf{b}'), \quad \hat{\mathbf{x}}_{T_i} = \sigma(\mathbf{W}' \mathbf{z}_{T_i} + \mathbf{b}'). \end{aligned}$$

The second term in the objective is a regularization term on the parameters $\Theta = \{\mathbf{W}, \mathbf{b}\}$ and $\Theta' = \{\mathbf{W}', \mathbf{b}'\}$, which can be written as

$$\Omega(\Theta, \Theta') = (\|\mathbf{W}\|^2 + \|\mathbf{b}\|^2 + \|\mathbf{W}'\|^2 + \|\mathbf{b}'\|^2).$$

The third term in (7) is the total loss of each source classifiers over the corresponding source label data with the hidden representation, which can be written as

$$\ell(\mathbf{z}_S, y_S; \{\theta_j\}) = \sum_{j=1}^r \left(- \sum_{i=1}^{n_j} \log \sigma(y_{S_i}^{(j)} \theta_j^\top \mathbf{z}_{S_i}^{(j)}) + \lambda \theta_j^\top \theta_j \right),$$

where $\theta_j \in \mathbb{R}^{k \times 1}$. The last term in (7) is the consensus regularization terms of the predictions of the source classifiers on the target domain data, which can be written as

$$\psi(\mathbf{z}_T; \{\theta_j\}) = \sum_{i=1}^n \left\| 2 \frac{\sum_{j=1}^r \sigma(\theta_j^\top \mathbf{z}_{T_i})}{r} - 1 \right\|^2.$$

The trade-off parameters α, β, γ and λ are small positive contents to balance the effect of different terms to the overall objective (7).

3.2 A Particular Solution

The optimization problem (7) is an unconstrained optimization with five types of variables \mathbf{W} , \mathbf{b} , \mathbf{W}' , \mathbf{b}' and $\{\boldsymbol{\theta}_j\}$'s to be optimized, and does not have closed form solutions. To derive the solutions of the five types of variables, we propose to use gradient descent methods. To simplify the math expressions, we first introduce the following intermediate variables.

$$\begin{aligned} A_{S_i}^{(j)} &= (\hat{\mathbf{x}}_{S_i}^{(j)} - \mathbf{x}_{S_i}^{(j)}) \circ \hat{\mathbf{x}}_{S_i}^{(j)} \circ (1 - \hat{\mathbf{x}}_{S_i}^{(j)}), \\ A_{T_i} &= (\hat{\mathbf{x}}_{T_i} - \mathbf{x}_{T_i}) \circ \hat{\mathbf{x}}_{T_i} \circ (1 - \hat{\mathbf{x}}_{T_i}), \\ B_{S_i}^{(j)} &= \mathbf{z}_{S_i}^{(j)} \circ (1 - \mathbf{z}_{S_i}^{(j)}), \\ B_{T_i} &= \mathbf{z}_{T_i} \circ (1 - \mathbf{z}_{T_i}), \\ C_{T_i}^{(j)} &= \sigma(\boldsymbol{\theta}_j^\top \mathbf{z}_{T_i}) (1 - \sigma(\boldsymbol{\theta}_j^\top \mathbf{z}_{T_i})). \end{aligned}$$

Then, it can be shown that the partial derivatives of the objective \mathcal{J} in (7) with respect to \mathbf{W} , \mathbf{b} , \mathbf{W}' , \mathbf{b}' and $\{\boldsymbol{\theta}_j\}$'s can be computed as follows respectively,

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \mathbf{W}} &= 2\mathbf{W}'^\top \left(\sum_{j=1}^r \sum_{i=1}^{n_j} A_{S_i}^{(j)} \circ B_{S_i}^{(j)} \mathbf{x}_{S_i}^{(j)\top} + \sum_{i=1}^n A_{T_i} \circ B_{T_i} \mathbf{x}_{T_i}^\top \right) \\ &\quad - \alpha \sum_{j=1}^r \sum_{i=1}^{n_j} \left(1 - \sigma(y_{S_i}^{(j)} \boldsymbol{\theta}_j^\top \mathbf{z}_{S_i}^{(j)}) \right) y_{S_i}^{(j)} \boldsymbol{\theta}_j \circ B_{S_i}^{(j)} \mathbf{x}_{S_i}^{(j)\top} \\ &\quad - \frac{4\beta}{r^2} \sum_{i=1}^n \left(\left(2 \sum_{j=1}^r \sigma(\boldsymbol{\theta}_j^\top \mathbf{z}_{T_i}) - r \right) \sum_{j=1}^r (C_{T_i}^{(j)} \boldsymbol{\theta}_j \circ B_{S_i}^{(j)} \mathbf{x}_{T_i}^\top) \right) \\ &\quad + 2\gamma \mathbf{W}, \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \mathbf{b}} &= 2\mathbf{W}'^\top \left(\sum_{j=1}^r \sum_{i=1}^{n_j} A_{S_i}^{(j)} \circ B_{S_i}^{(j)} + \sum_{i=1}^n A_{T_i} \circ B_{T_i} \right) \\ &\quad - \alpha \sum_{j=1}^r \sum_{i=1}^{n_j} \left(1 - \sigma(y_{S_i}^{(j)} \boldsymbol{\theta}_j^\top \mathbf{z}_{S_i}^{(j)}) \right) y_{S_i}^{(j)} \boldsymbol{\theta}_j \circ B_{S_i}^{(j)} \\ &\quad - \frac{4\beta}{r^2} \sum_{i=1}^n \left(\left(2 \sum_{j=1}^r \sigma(\boldsymbol{\theta}_j^\top \mathbf{z}_{T_i}) - r \right) \sum_{j=1}^r (C_{T_i}^{(j)} \boldsymbol{\theta}_j \circ B_{S_i}^{(j)}) \right) \\ &\quad + 2\gamma \mathbf{b}, \end{aligned} \tag{9}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}'} = \sum_{j=1}^r \sum_{i=1}^{n_j} 2A_{S_i}^{(j)} \mathbf{z}_{S_i}^{(j)\top} + \sum_{i=1}^n 2A_{T_i} \mathbf{z}_{T_i}^\top + 2\gamma \mathbf{W}', \tag{10}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b}'} = \sum_{j=1}^r \sum_{i=1}^{n_j} 2A_{S_i}^{(j)} + \sum_{i=1}^n 2A_{T_i} + 2\gamma \mathbf{b}', \quad (11)$$

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}_j} &= \alpha \left(- \sum_{i=1}^{n_j} \left(1 - \sigma(y_{S_i}^{(j)} \boldsymbol{\theta}_j^\top \mathbf{z}_{S_i}^{(j)}) \right) y_{S_i}^{(j)} \mathbf{z}_{S_i}^{(j)\top} + 2\lambda \boldsymbol{\theta}_j^\top \right) \\ &\quad - \frac{4\beta}{r^2} \sum_{i=1}^n \sum_{j=1}^r (2\sigma(\boldsymbol{\theta}_j^\top \mathbf{z}_{T_i}) - r) C_{T_i}^{(j)} \mathbf{z}_{T_i}^\top. \end{aligned} \quad (12)$$

Based on the above partial derivatives, with an initialization of \mathbf{W} , \mathbf{b} , \mathbf{W}' , \mathbf{b}' and $\{\boldsymbol{\theta}_j\}$'s, we can update them alternatively and iteratively by applying the following rules till the solutions are converged,

$$\begin{aligned} \mathbf{W} &\leftarrow \mathbf{W} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}}, & \mathbf{b} &\leftarrow \mathbf{b} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{b}}, \\ \mathbf{W}' &\leftarrow \mathbf{W}' - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}'}, & \mathbf{b}' &\leftarrow \mathbf{b}' - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{b}'}, \\ \boldsymbol{\theta}_j &\leftarrow \boldsymbol{\theta}_j - \eta \frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}_j}, \end{aligned} \quad (13)$$

where η is a learning rate. That is, in each iteration, we alternatively fix four of the five types of the variables and optimize the rest one.

3.3 Target Classifier Construction

After the solutions of \mathbf{W} , \mathbf{b} , \mathbf{W}' , \mathbf{b}' and $\{\boldsymbol{\theta}_j\}$'s are obtained, one can construct a classifier f_T in terms of θ_T for the target domain in two ways. One way is to construct the classifier combining all source classifiers $\{\boldsymbol{\theta}_j\}$'s based on a voting scheme. That is, for any instance \mathbf{x}_T from the target domain, which can be either from the observed unlabeled sample \mathcal{D}_T or unseen data sample, the classifier f_T make a prediction on it based on

$$f_T(\mathbf{x}_T) = \frac{1}{r} \sum_{j=1}^r \sigma(\boldsymbol{\theta}_j^\top (\sigma(\mathbf{W}\mathbf{x}_T + \mathbf{b}))).$$

Alternatively, another way to construct a target classifier is to first map instances from all the source domains to their corresponding hidden representations by $\mathbf{z}_{S_i}^{(j)} = \sigma(\mathbf{W}\mathbf{x}_{S_i}^{(j)} + \mathbf{b})$, and then apply standard classification algorithms, e.g., logistic regression or Support Vector Machine (SVM) [2], on the labeled data, $\{\mathbf{z}_{S_i}^{(j)}, y_{S_i}^{(j)}\}_{i=1, \dots, n_j}^{j=1, \dots, r}$, to train a unified classifier f_T in terms of a vector of parameter $\boldsymbol{\theta}_T$. For any instance \mathbf{x}_T from the target domain, one can first map it to an hidden representation by $\mathbf{z}_T = \sigma(\mathbf{W}\mathbf{x}_T + \mathbf{b})$, and then use $\boldsymbol{\theta}_T$ to make an prediction. In the sequel, we denote Consensus Regularized Autoencoders (CRA) for our proposed framework and the particular solution. The overall algorithm of CRA is summarized in Algorithm 1.

Algorithm 1. Consensus Regularized Autoencoders (CRA)

Input: Given r source domains $\mathcal{D}_S^{(1)}, \dots, \mathcal{D}_S^{(r)}$, where $\mathcal{D}_S^{(j)} = \{\mathbf{x}_{S_i}^{(j)}, \mathbf{y}_{S_i}^{(j)}\}_{i=1}^{n_j}$, a target domain $\mathcal{D}_T = \{\mathbf{x}_{T_i}\}_{i=1}^n$, trade-off parameters $\alpha, \beta, \gamma, \lambda$, and the number of hidden features k .

Output: A classifier on the target domain.

1. Initialize $\mathbf{W}, \mathbf{b}, \mathbf{W}',$ and \mathbf{b}' by performing an autoencoder algorithm on instances of all the domains, and train $\{\theta_j\}$'s on the corresponding domain data independently.
2. Fix $\{\theta_j\}$'s, update $\mathbf{W}, \mathbf{b}, \mathbf{W}',$ and \mathbf{b}' alternatively based on the update rules in (13) and the corresponding derivatives in (8), (9), (10) and (11).
3. Fix $\mathbf{W}, \mathbf{b}, \mathbf{W}',$ and \mathbf{b}' , update $\{\theta_j\}$'s based on the update rules in (13) and the corresponding derivative in (12).
4. If the solutions are converged, construct a target classifier as described in Section 3.3, otherwise, go to Step 2.

Table 2. Description of the image dataset

	<i>flower</i>				<i>traffic</i>			
	<i>sunflower</i>	<i>rose</i>	<i>lotus</i>	<i>tulip</i>	<i>aviation</i>	<i>bus</i>	<i>boat</i>	<i>dogsled</i>
No. of instance	85	100	66	100	100	100	100	100

4 Experiments

In this section, we conduct extensive experiments on two real-world datasets to systematically evaluate the effectiveness of our proposed method for multi-source transfer learning.

4.1 Datasets

Image Dataset. We conduct experiments on the image dataset of multi-source transfer learning problems used in [27]. The dataset contains two main categories, *flower* and *traffic*, selected from the COREL collection¹. Each main category further contains four subcategories. The *flower* category can be further classified into *sunflower*, *rose*, *lotus* and *tulip*, while the *traffic* category can be further classified into *aviation*, *bus*, *boat* and *dogsled*. Figure 1 shows one example of each subcategory respectively. Following the same preprocessing proposed in [27], we randomly select one subcategory from *flower* and one subcategory from *traffic* to construct a domain, thus can construct 24 (4!) different groups of domains, where each group contains 4 different domains and each subcategory appears once and only once in each group. In each group, we then randomly select one domain as the target domain, and the rest 3 domains as the source domains. Finally, we can construct 96 (4 × 4!) multi-source (3 source domains) image classification problems. Each image is represented by 87 features, which include 36 features are based on color histogram [25] and 51 features are based on SILBP texture histogram [19]. The description of the image dataset is summarized in Table 2.

¹ <http://archive.ics.uci.edu/ml/datasets/Corel+Image+Features>



Fig. 1. Examples of the eight subcategories of the dataset

Sentiment Dataset. We use the Multi-domain sentiment benchmark dataset generated by [1] for experiments. The dataset contains reviews of 4 types of products, books, dvd, electronics, and kitchen, crawled from Amazon.com. Each product review is annotated as positive or negative based on its overall sentiment polarity. Each type of products is considered as a domain, and each domain contains 2,000 reviews, of which 1,000 are positive and the other 1,000 are negative. Each review is represented as a vector of 3126 word features. Following similar preprocessing used in [1], we randomly select one of the 4 domains as the target domain, and the rest 3 domains as the source domains. Therefore, we can conduct four multi-source sentiment classification problems.

4.2 Baseline Methods and Implementation Details

Baseline Methods. We compare our proposed method CRA with various baseline methods, including the standard logistic regression (LR) and SVM without transfer learning, an embedding method based on autoencoders (EAER) [23], a dimensionality reduction method for 1vs1 transfer learning problems, Transfer Component Analysis (TCA) [15], the Centralized Consensus Regularization (CCR_3) [27] for multi-source transfer learning problems on the original data space, and a recently proposed 1vs1 transfer learning method based on autoencoders, marginalized Stacked Denoising Autoencoders (mSDA) [5].

Note that the methods EAER and TCA only map original data to a latent space, where a classifier needs to be further specified for final classification problems. Here, we consider LR or SVM as the base classifier for EAER and TCA. Moreover, except for CCR_3 , all the other baselines are not proposed for multi-source transfer learning problems, to conduct experiments with multiple source domains, we can either apply them on each pair of a source domain and the target domain or apply them on the pair of a *unified* source domain which simply combines all source domains and the target domain to learn a target classifier. For each of these baselines, i.e., LR, SVM, EAER, TCA, and mSDA, we report the mean, the maximum as well as the minimum accuracies of their corresponding target classifiers based on pairwise domains. For our proposed CRA, as we discussed, there are two ways to construct a target classifier. One is a voting-based combination of the multiple learned source classifiers, the other is to learn

Table 3. Average results (in %) on the 96 multi-source image classification problems

	LR	SVM	LR		SVM		mSDA	CCR ₃	CRA _v	LR	SVM
			EAER	TCA	EAER	TCA				CRA _u	CRA _u
Max	83.9	81.7	83.2	84.2	85.6	85.2	83.1	87.5	89.2	89.4	88.9
Min	65.0	56.0	62.3	66.8	71.3	69.8	64.6	83.5			
Mean	76.1	69.6	74.9	77.0	79.4	79.1	73.5	85.9			

a unified classifier from hidden representations of all source domains. We denote CRA_v and CRA_u the target classifiers built in these two ways respectively.

Implementation Details. For the trade-off parameters in CRA, the settings are listed as follows, $\alpha = 1$, $\beta = 0.5$, $k = 10$, $\gamma = 0.0001$, $\lambda = 1$ for the image dataset, and $\alpha = 100$, $\beta = 20$, $k = 80$, $\gamma = 0.0001$, $\lambda = 1$ for sentiment dataset. In experiments, we also study the parameter sensitivity of the parameters. For the parameters in TCA, EAER and mSDA, we carefully tune the number of dimensions k , and report the best results (e.g., in TCA, k varies from 10 to 80 with interval 10 for image data). We set the parameters of CCR₃ as the those published in [27], in which the parameter θ controlling the importance of consensus is sampled from $[0.05, 0.25]$ with interval 0.05. Thus the three values of minimum, mean and maximum for CCR₃ are also reported.

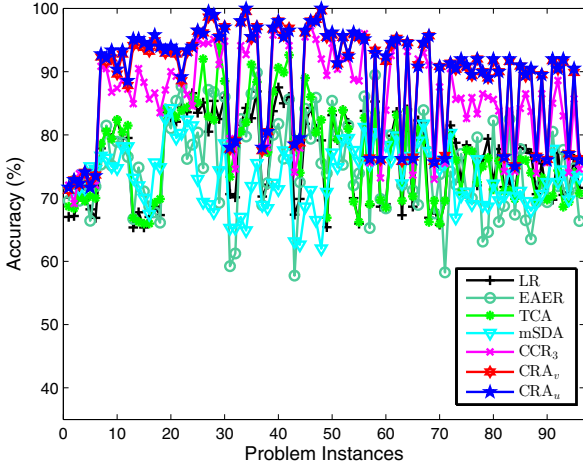
4.3 Experimental Results

Results on Image Data. We show the detailed mean accuracies of 96 image classification problems in Figure 2, and their average results in Table 3. From these results, we have some attractive observations: 1) CRA is significantly better than the traditional machine learning algorithms LR and SVM, which validate the effectiveness of the proposed transfer learning framework. 2) CRA outperforms TCA, EAER and mSDA, which shows that CRA can benefit from discovering a more powerful feature representation and incorporating consensus regularization from multiple source domains. 3) CRA performs better than CCR₃, which indicates the superiority of representation learning of autoencoders. Furthermore, the t -test with 95% confidence shows that CRA is significantly better than all the compared baselines.

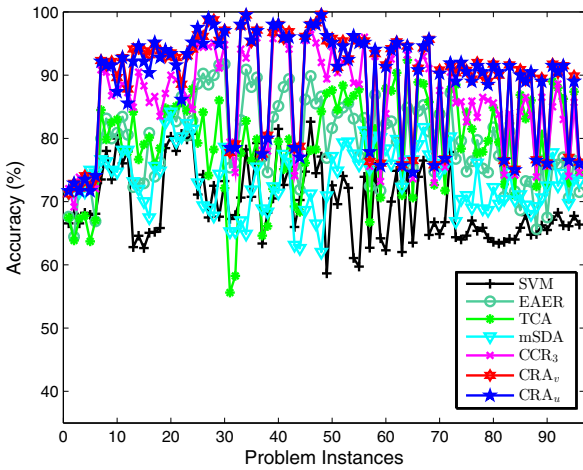
Results on Sentiment Data. To further verify the effectiveness of the proposed framework CRA, we also make comparisons of all algorithms on sentiment classification problems. The detailed results are recorded in Table 4. Except mSDA is slightly better than CRA according to the maximum accuracies, CRA outperforms all the baselines. These results again validate the effectiveness of CRA, which can take full advantage of autoencoders and consensus regularization from multiple source domains simultaneously in a unified optimization framework.

4.4 Parameter Sensitivity

Here, we also investigate the parameter influence of three important trade-off parameters on image data, i.e., the relative importance of incorporating labeled information



(a) The mean accuracies of 96 multi-source image classification problems using LR as the base classifier



(b) The mean accuracies of 96 multi-source image classification problems using SVM as the base classifier

Fig. 2. The mean accuracies of 96 multi-source image classification problems

from source domains, the effect of considering consensus regularization and the number of hidden nodes for autoencoder. When we consider one parameter, the rest parameters are fixed. α and β are sampled from the value set $\{0.01, 0.1, 0.5, 1, 5, 10, 50, 100\}$, and k is sampled from the value set $\{5, 10, 20, 30, 40, 50, 60, 70, 80\}$. Six problems are randomly selected from 96 ones, and all the results of CRA_v are shown in Figure 3. We find that CRA is not sensitive to the number of hidden nodes k from Figure 3(c), so we

Table 4. Detailed and average results (in %) on the 4 multi-source sentiment classification problems

Tasks		LR	SVM	LR		SVM		mSDA	CCR ₃	CRA _v	LR	SVM
				EAER	TCA	EAER	TCA				CRA _u	CRA _u
<i>tar.book</i>	Max	79.3	78.4	67.8	68.5	73.0	66.2	82.3	78.6	79.2	79.2	79.1
	Min	71.0	71.5	57.0	58.9	69.3	59.3	77.6	78.2			
	Mean	75.7	74.9	63.0	64.2	70.9	62.8	79.9	78.4			
<i>tar.kitchen</i>	Max	85.6	85.4	78.9	75.2	77.5	73.1	84.7	86.1	85.9	86.3	85.8
	Min	76.4	74.9	71.0	64.2	75.9	64.7	81.4	85.6			
	Mean	81.0	80.5	76.6	69.4	76.7	68.7	83.5	85.9			
<i>tar.elec.</i>	Max	83.9	83.1	74.2	72.9	72.8	70.5	85.2	79.3	84.1	84.7	82.4
	Min	73.5	73.0	68.5	60.7	69.4	59.4	74.4	75.4			
	Mean	78.7	78.9	70.8	67.1	71.2	65.2	81.0	75.6			
<i>tar.dvd</i>	Max	79.7	79.5	69.5	68.5	70.8	67.4	82.3	80.2	80.6	81.1	80.8
	Min	73.6	72.2	56.5	61.4	67.7	61.3	78.2	79.7			
	Mean	77.0	75.9	65.1	65.2	69.0	64.3	80.3	80.1			
<i>Average</i>	Max	82.1	81.6	72.6	71.3	73.5	69.3	83.7	81.1	82.5	82.8	82.0
	Min	73.6	72.9	63.2	61.3	70.6	61.2	77.9	79.7			
	Mean	78.1	77.5	68.9	66.5	72.0	65.3	81.2	80.5			

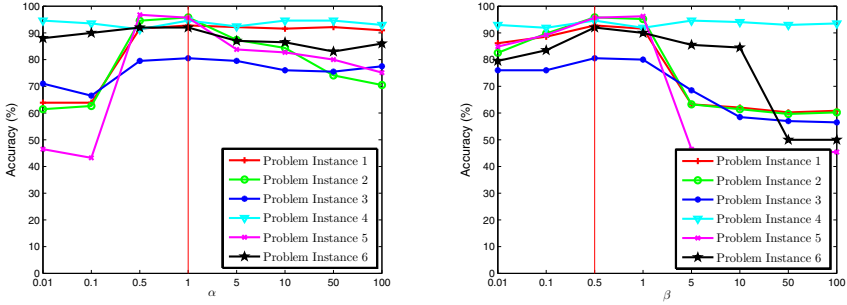
set $k = 10$ in the experiments for high efficiency. In Figure 3(a), CRA gets very low performance when the value of α is small, which indicates the importance of labeled information from source domains. Also in Figure 3(b), it is observed that the setting of large value of β will lead to over-fitting and degrade the performance of CRA. According to these insights, we set $\alpha = 1$, $\beta = 0.5$ and $k = 10$ in this paper to achieve good and stable results.

5 Related Work

In this section, we survey some previous works which are closely related to our work, including transfer learning and autoencoder.

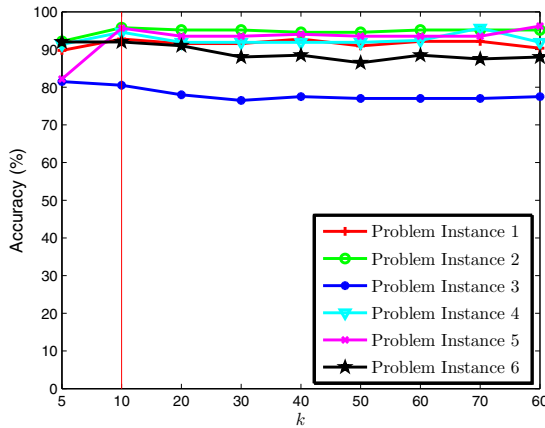
5.1 Embedding with Autoencoder

Autoencoders are primarily seen as a dimensionality reduction technique and thus use a bottleneck, namely the lower dimensional hidden layer of autoencoder, to learn a compressed representation which is represented by the hidden layer [3,10]. Currently variants of autoencoders have been investigated. Sparse autoencoders [17] use the idea of introducing a form of sparsity regularization to restrict the capacity of hidden units. Denoising autoencoders [21,22] learn to reconstruct the clean input from a artificially corrupted input and capture the structure of the input distribution. Sparse coding [12] can be viewed as a kind of autoencoder that uses a linear decoder tends to favor learning over-complete representations. These are often called regularized autoencoders, where some regularization terms are proposed to improve the data reconstruction performance.



(a) The Parameter Influence of α

(b) The Parameter Influence of β



(c) The Parameter Influence of k

Fig. 3. The Study of Parameter Influence on CRA

Contractive autoencoders [18], which shares a similar motivation with Denoising autoencoders, learn robust representations by adding an analytic contractive penalty term to the basic autoencoder. Marginalized Stacked Denoising Autoencoders (mSDA) [5] can be seen as the first try to use autoencoding technique for domain adaptation. However they have not considered consensus regularization from multiple sources.

5.2 Transfer Learning

Recent years have witnessed numerous research in transfer learning [16]. Here we only list some closely related works, i.e., transfer embedding and subspace learning (or learning on topic level). Pan et al. [13] proposed a dimensionality reduction approach to find out such latent feature space that supervised learning algorithms can be applied to train classification models and obtain satisfying results. After that, they also proposed a transfer component analysis (TCA) algorithm to learn some transfer components across domains [15]. Si et al. [20] developed a transfer subspace learning framework, which can

be applicable to various dimensionality reduction algorithms and minimize the Bregman divergence between the distribution of training data and testing data in the selected subspace. Zhuang et al. [26] exploited the stable associations between word topics and document classes as the bridge for knowledge transfer. Zhang et al. [24] proposed to match data distributions in the Hilbert space, which can be formulated as aligning kernel matrices across domains when given a pre-defined empirical kernel map. However, these works are all in the 1vs1 transfer learning setting. Compared to the previous work learning from multiple sources [27] on the original data space, we focus on the representation learning of autoencoders for transfer learning. For cross-domain activity recognition, Hu et al. [11] developed a bridge between the activities in two domains by learning a similarity function via Web search, under the condition that the sensor readings are from the same feature space. However, they assumed some labeled target domain data are available in their model.

To sum up, we propose a unsupervised transfer framework via consensus regularized autoencoders, which takes full advantage of autoencoders and consensus regularization from multiple sources. And finally, the extensive experiments demonstrate its effectiveness.

6 Conclusions

In this paper, we study the transfer learning framework from multiple source domains via consensus regularized autoencoders. In this framework, the well known representation learning technique autoencoder is incorporated, and the consensus prediction on target domain data given by classifiers trained from multiple source domains is considered. Then we formalize the autoencoders and consensus regularization into a unified optimization framework. Finally, a series of experiments on image and text data are conducted to validate the effectiveness of our framework.

We assume all the source domains play the same important role in this paper. It would be interesting to assign different weights to different source domains and investigate their importance in the future work.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 61175052, 61203297, 61035003), National High-tech R&D Program of China (863 Program) (No.2014AA012205, 2013AA01A606, 2012AA011003).

References

1. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th ACL (2007)
2. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th AWCLT (1992)
3. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics* (1988)
4. Chattopadhyay, R., Ye, J.P., Panchanathan, S., Fan, W., Davidson, I.: Multi-source domain adaptation and its application to early detection of fatigue. In: Proceedings of the 17th ACM SIGKDD, pp. 717–725. ACM (2011)

5. Chen, M.M., Xu, Z.X., Weinberger, K., Sha, F.: Marginalized denoising autoencoders for domain adaptation. In: Proceedings of the 29th ICML (2012)
6. David, H., Stanley, L.: Applied Logistic Regression. Wiley, New York (2000)
7. Duan, L., Tsang, I.W., Xu, D., Chua, T.S.: Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th ICML (2009)
8. Gao, J., Fan, W., Jiang, J., Han, J.W.: Knowledge transfer via multiple model local structure mapping. In: Proceedings of the 14th ACM SIGKDD (2008)
9. Ge, L., Gao, J., Zhang, A.D.: Oms-tl: a framework of online multiple source transfer learning. In: Proceedings of the 22nd ACM CIKM, pp. 2423–2428. ACM (2013)
10. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and helmholtz free energy. In: Advances in NIPS (1994)
11. Hu, D.H., Zheng, V.W., Yang, Q.: Cross-domain activity recognition via transfer learning. *Pervasive and Mobile Computing* 7(3), 344–358 (2011)
12. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research* (1997)
13. Pan, S.J., Kwok, J.T., Yang, Q.: Transfer learning via dimensionality reduction. In: Proceedings of the 23rd AAAI (2008)
14. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th WWW (2010)
15. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. *IEEE TNN* (2011)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE TKDE* (2010)
17. Poultney, C., Chopra, S., Cun, Y.L.: Efficient learning of sparse representations with an energy-based model. In: Advances in NIPS (2006)
18. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: Proceedings of the 28th ICML (2011)
19. Shi, Z.P., Ye, F., He, Q., Shi, Z.Z.: Symmetrical invariant lbr texture descriptor and application for image retrieval. In: Congress on Image and Signal Processing (2008)
20. Si, S., Tao, D.C., Geng, B.: Bregman divergence-based regularization for transfer subspace learning. *IEEE TKDE* (2010)
21. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th ICML (2008)
22. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR* (2010)
23. Yu, W., Zeng, G., Luo, P., Zhuang, F., He, Q., Shi, Z.: Embedding with autoencoder regularization. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) *ECML PKDD 2013, Part III*. LNCS, vol. 8190, pp. 208–223. Springer, Heidelberg (2013)
24. Zhang, K., Zheng, V., Wang, Q.J., Kwok, J., Yang, Q., Marsic, I.: Covariate shift in hilbert space: A solution via surrogate kernels. In: Proceedings of The 30th ICML, pp. 388–395 (2013)
25. Zhang, L.: The Research on Human-computer Cooperation in Content-based Image Retrieval. Ph.D. thesis, Tsinghua University, Beijing (2001) (in Chinese)
26. Zhuang, F.Z., Luo, P., Xiong, H., He, Q., Xiong, Y.H., Shi, Z.Z.: Exploiting associations between word clusters and document classes for cross-domain text categorization. *Statistical Analysis and Data Mining* (2011)
27. Zhuang, F.Z., Luo, P., Xiong, H., Xiong, Y.H., He, Q., Shi, Z.Z.: Cross-domain learning from multiple sources: A consensus regularization perspective. *IEEE TKDE* (2010)