A novel transfer learning method based on common space mapping and weighted domain matching

Ru-Ze Liang  
King Abdullah University of Science and Technology, Saudi Arabia  
razeliang@outlook.com

Wei Xie  
Vanderbilt University, Nashville, TN 37235, United States  
wei.xie@vanderbilt.edu

Weizhi Li  
Suning Commerce R&D Center USA, Inc, Palo Alto, CA 94304, United States  
weizhili2014@gmail.com

Hongqi Wang  
School of Management of Harbin University of Science and Technology, Harbin 150000, China

Jim Jing-Yan Wang  
New York University Abu Dhabi, United Arab Emirates

Lisa Taylor  
Michigan State University, East Lansing, MI 48824, United States  
listaylor23857@yahoo.com

Abstract—In this paper, we propose a novel learning framework for the problem of domain transfer learning. We map the data of two domains to one single common space, and learn a classifier in this common space. Then we adapt the common classifier to the two domains by adding two adaptive functions to it respectively. In the common space, the target domain data points are weighted and matched to the target domain in term of distributions. The weighting terms of source domain data points and the target domain classification responses are also regularized by the local reconstruction coefficients. The novel transfer learning framework is evaluated over some benchmark cross-domain data sets, and it outperforms the existing state-of-the-art transfer learning methods.

Keywords—Transfer Learning; Linear Transformation; Distribution Matching; Weighted Mean

I. INTRODUCTION

Transfer learning has been a hot topic in the machine learning community. It aims to solve the classifier learning problem of a target domain which has limited label information with weak supervision information [1], with the help of a source domain which has sufficient labels. The problem of using two domains for the problem of one domain is that their distributions are significantly different. A lot of works have been proposed to learn from two domains with different distributions for the classification problem in the target domain [2], [3], [4], [5], [6], [7]. However, the performance of these works is not satisfactory. The shortages of these works paper are due to the ignorance of the label information of the target domain, the ignorance of the local connection of the data points of both source and target domain, or the ignorance of the the differences of the source domain data points for the learning problem of target domain.

In this paper, we propose a novel transfer learning approach to solve these problems. We map the data of two domains to one common space by linear transformation, and match the distribution of the two domains in this common space. In these common subspaces, we match the distributions by using the weighting factors of the source domain data points. We propose to minimize the classification errors of the data points of both the source and target domains to use the labels of the target domain. To do this, we learn a classifier in the common space by using the labels of data points of both domains, and then adapt the common classifier to the two domains by adding adaptive functions to the common classifier respectively. Moreover, we also propose to use local reconstruction information to regularize the learning of the weights of the source domain data points, and the classifier of the target domain. The learning problem is constructed by minimizing the objective function with regard to the parameters of the linear transformation matrix, the common classifier parameter and the adaptation parameters. We design an iterative learning algorithm to solve this problem.

II. PROPOSED TRANSFER LEARNING METHOD

A. Modeling

We suppose the source domain training set is \( S = \{ (x_1^1, y_1^1), \cdots, (x_{n_1}^1, y_{n_1}^1) \} \), where \( x_i^1 \in \mathbb{R}^m \) is the feature vector of \( m \) dimensions of the \( i \)-th data point, and \( y_i^1 \in \{ +1, -1 \} \) is its label. The target domain training set is \( T = \{ (x_1^2, y_1^2), \cdots, (x_{n_2}^2, y_{n_2}^2) \} \), where \( x_j^2 \in \mathbb{R}^m \) is the feature vector of the \( j \)-th data point, and \( y_j^2 \in \{ +1, -1 \} \) is its label. Only the first \( n_3 \) target domain data points are labeled. We map the data of both domains to a common space by a transformation matrix \( \Theta \in \mathbb{R}^{r \times m} \), \( y = \Theta x \). We present the distribution of the source domains in the common space as the weighted mean of the vectors of the data points,

\[
\mu_x^s = \frac{1}{n_1} \sum_{i=1}^{n_1} \Theta x_i^s \pi_i. \tag{1}
\]
where \( \pi_i \) is the weighting factor of the \( i \)-th data point. We also present the distribution of the target domain as the mean of its data points in the common space,

\[
\mu_i = \frac{1}{n_2} \sum_{j=1}^{n_2} \Theta x_j^i
\]  

(2)

Naturally we hope the distributions of the two domains can be as close to each other as possible. So we propose to minimize the squared \( \ell_2 \) norm distance between them with regard to both \( \Theta \) and \( \pi \),

\[
\min_{\Theta, \pi} \frac{1}{2} \| \mu^\pi - \mu \|^2_2
= \frac{1}{2} \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \Theta x_i^s \pi_i - \frac{1}{n_2} \sum_{j=1}^{n_2} \Theta x_j^t \right\|^2_2.
\]  

(3)

We design a linear classifier in the common space,

\[
g(x) = w^\top y = w^\top \Theta x,
\]  

(4)

where \( w \in \mathbb{R}^r \) is the parameter vector of the common classifier. Then we adapt it to two domains by adding adaptive functions to the common classifier, and obtain the source domain classifier \( f \), and the target domain classifier \( g \).

\[
f(x^t) = g(x^t) + \Delta_s(x^t) = w^\top \Theta x^t + u^\top x^t + b^t, \quad \text{and} \quad h(x^s) = g(x^s) + \Delta_t(x^s) = w^\top \Theta x^s + v^\top x^s + b^s,
\]  

where \( \Delta_s(x^t) = u^\top x^s \) is the source domain adaptive function, \( u \in \mathbb{R}^m \) is its parameter vector of the adaptation function, and \( b^t \) and \( b^s \) are bias terms. where \( \Delta_t(x^s) = v^\top x^t \) is the target domain adaptive function, and \( v \in \mathbb{R}^m \) is its parameter vector. To measure the classification errors of the two classifiers over the training set, we use the popular hinge loss, and minimize it to learn the parameters,

\[
\min_{\Theta, w, u, v, \pi} \left\{ \sum_{i=1}^{n_1} \pi_i \max(0, 1 - y^s_i f(x^t_i)) + \sum_{j=1}^{n_3} \max(0, 1 - y^t_i h(x^s_j)) \right\}.
\]  

(6)

In this classification error minimization problem, we also use the source domain data point weighting factors to weight the classification error terms.

We denote the neighborhood set of the \( i \)-th source data point as \( \mathcal{N}_i^s \), and the reconstruction coefficients of \( \mathcal{N}_i^s \) are solved by the following minimization problem,

\[
\min_{\omega_{ik}, k \in \mathcal{N}_i^s} \left\| x_i^s - \sum_{k \in \mathcal{N}_i^s} \omega_{ik} x_k^s \right\|^2_2
\]  

s.t. \( \sum_{k \in \mathcal{N}_i^s} \omega_{ik} = 1, \omega_{ik} \geq 0, \forall k \in \mathcal{N}_i^s \)

\]  

(7)

where \( \omega_{ik}, k \in \mathcal{N}_i^s \) are the coefficients for reconstruction of \( x_i^s \) from the neighbors in \( \mathcal{N}_i^s \). Then we use them to regularize the learning of the source domain weighting factors,

\[
\min_{\pi} \frac{1}{2} \sum_{i=1}^{n_1} \left( \sum_{k \in \mathcal{N}_i^s} \omega_{ik} \pi_k \right)^2.
\]  

(8)

Similarly we also have the neighborhood reconstruction coefficients for the target domain data set, and we use them to regularize the classification responses,

\[
\min_{w, u, v, \pi} \sum_{j=1}^{n_2} \left( h(x^s_j) - \sum_{k' \in \mathcal{N}_j^t} \omega'_{jk'} h(x'_{k'}) \right)^2.
\]  

(9)

The overall minimization problem for the transfer learning framework is the combination of problems of (3), (6), (8), (9), and squared \( \ell_2 \) norms of classifier parameter vectors to avoid over-fitting.

\[
\min_{\Theta, w, u, v, \pi} \left\{ \sum_{i=1}^{n_1} \pi_i \max(0, 1 - y^s_i f(x^t_i)) + \sum_{j=1}^{n_3} \max(0, 1 - y^t_i h(x^s_j)) + C_1 \left( \| u \|^2_2 + \| v \|^2_2 \right)
+ C_2 \left( \sum_{i=1}^{n_1} \pi_i - \sum_{k \in \mathcal{N}_i^s} \omega_{ik} \pi_k \right)^2
+ \sum_{j=1}^{n_2} \left( h(x^s_j) - \sum_{k' \in \mathcal{N}_j^t} \omega'_{jk'} h(x'_{k'}) \right)^2 \right\}
+ \frac{C_3}{2} \left( \sum_{i=1}^{n_1} \Theta x_i^s \pi_i - \sum_{j=1}^{n_2} \Theta x_j^t \right)^2.
\]  

(10)

In this minimization problem, we impose \( \Theta \) to be orthogonal, impose a lower bound and a upper bound for \( \pi \), and an additional constraint to \( \pi \), so that the summation of all the elements of \( \pi \) is \( n_1 \).

**B. Optimization**

We rewrite the source domain and target domain classifiers as a linear function of the input feature vectors,

\[
f(x^t) = \phi^\top x^s + b^t, \quad \text{where} \quad \phi = \Theta^\top w + u, \quad \text{and} \quad h(x^s) = \varphi^\top x^s + b^s, \quad \text{where} \quad \varphi = \Theta^\top w + v.
\]  

(11)
Then we have the following minimization problem,

\[
\min_{\Theta, w, \phi, \varphi} \left\{ \sum_{i=1}^{n_1} \pi_i \max(0, 1 - y_i^*(\phi^T x_i^* + b^*)) + \sum_{j=1}^{n_3} \max(0, 1 - y_j^i(\varphi^T x_j^i + b^i)) + \frac{C_1}{2} \left( \|\phi - \Theta^T w\|_2^2 + \|\varphi - \Theta^T w\|_2^2 \right) + C_2 \left( \sum_{i=1}^{n_1} \pi_i - \sum_{k \in N_i^s} \omega_{i,k} \pi_k \right)^2 + \sum_{j=1}^{n_2} \left( \varphi^T x_j^i - \sum_{k \in N_j^s} \omega_{j,k} \varphi^T x_k^i \right)^2 \right\}
\]

\[+ \frac{C_3}{2} \left( \sum_{i=1}^{n_1} \Theta x_i^i \pi_i - \frac{1}{n_2} \sum_{j=1}^{n_2} \Theta x_j^i \right)^2 \]

\[s.t. \Theta^T = I_r, \quad \Theta \leq 1, \quad \text{and} \quad \pi^T 1 = n_1. \]  

(12)

To solve this problem, we use the iterative optimization method to update the variables one by one.

1) Solving \( w \) and \( \Theta \): We first solve \( w \) by setting the derivative of objective with regard to \( w \) to zero, and we have

\[ w = \frac{1}{2} \Theta (\phi + \varphi). \]  

(13)

Then we substitute it to (12), and consider the optimization of \( \Theta \), we have

\[
\min_{\Theta} \quad \text{Tr} \left[ \Theta \left( -\frac{C_1}{4}(\phi + \varphi)(\phi + \varphi)^T + \frac{C_3}{2} \left( \frac{1}{n_1} \sum_{i=1}^{n_1} x_i^i \pi_i \right) - \frac{1}{n_2} \sum_{j=1}^{n_2} x_j^i \right) \left( \frac{1}{n_1} \sum_{i=1}^{n_1} x_i^i \pi_i - \frac{1}{n_2} \sum_{j=1}^{n_2} x_j^i \right)^T \right] \]

\[s.t. \Theta^T = I_r, \]  

(14)

This problem can be easily solve by the eigen-decomposition method.

2) Updating \( \phi \) and \( \varphi \): To update both \( \phi \) and \( \varphi \), we consider the following minimization problem,

\[
\min_{\phi, \varphi} \left\{ Q(\phi, \varphi) = \sum_{i=1}^{n_1} \max(0, 1 - y_i^*(\phi^T x_i^* + b^*)) \pi_i + \sum_{j=1}^{n_3} \max(0, 1 - y_j^i(\varphi^T x_j^i + b^i)) + \frac{C_1}{2} \left( \|\phi - \Theta^T w\|_2^2 + \|\varphi - \Theta^T w\|_2^2 \right) + C_2 \left( \sum_{i=1}^{n_1} \varphi^T x_j^i - \sum_{k \in N_j^s} \omega_{j,k} \varphi^T x_k^i \right)^2 \right\}
\]

(15)

To solve this problem, we use the sub-gradient algorithm to update \( \phi \) and \( \varphi \), \( \phi \leftarrow \rho \nabla Q_\phi \) and \( \varphi \leftarrow \varphi - \rho \nabla Q_\varphi \).

3) Updating \( \pi \): To solve \( \pi \), we have the following minimization problem,

\[
\min_{\pi} \left\{ \sum_{i=1}^{n_1} \max(0, 1 - y_i^*(\phi^T x_i^* + b^*)) \pi_i + C_2 \left( \sum_{i=1}^{n_1} \pi_i - \sum_{k \in N_i^s} \omega_{i,k} \pi_k \right)^2 \right\}
\]

\[+ C_3 \left( \sum_{i=1}^{n_1} \Theta x_i^i \pi_i - \frac{1}{n_2} \sum_{j=1}^{n_2} \Theta x_j^i \right)^2 \]

\[s.t. \quad 0 \leq \pi \leq \delta 1, \quad \text{and} \quad \pi^T 1 = n_1. \]  

(16)

This problem is a linear constrained quadratic programming problem, and we solve it by using the active set algorithm [8].

### III. Experiments

#### A. Data Sets

In the experiments, we use three benchmark data sets, which are the 20-Newsgroup corpus data set, the Amazon review data set, and the Spam email data set. 20-Newsgroup corpus data set is a data set of newspaper documents. It contains documents of 20 classes. The classes are organized in a hierarchical structure. For a class, it usually have two or more sub-classes. For example, in the class of car, there are two sub-classes, which are motorcycle and auto. To split this data set to source domain and target domain, for one class, we keep one sub-class in the source domain, while put the other sub-class to the target domain. We follow the splitting of source and target domain of NG14 data set of [2]. In this data set, there are 6 classes, and for each class, one sub-class is in the source domain, and another sub-class is in the target domain. For each domain, the number of data points is 2,400. The bag-of-word features of each document are used as original features. Amazon review data set is
a data set of reviews of products. It contains reviews of three types of products, which are books, DVD and Music. The reviews belong to two classes, which are positive and negative. We treat the review of books as source domain, and that of DVD as target domain. For each domain, we have 2,000 positive reviews and 2,000 reviews. Again, we use the bag-of-words features as the features of reviews. Spam email data set is a set of emails of different individuals. In this data set, there are emails of three different individuals’ inboxes, and we treat each individual as a domain. In each individual’s inbox, there are 2,500 emails, and the emails are classified to two different classes, which are normal email and spam email. We also randomly choose one individual as a source domain, and another one as a target domain.

B. Results

In the experiments, we use the 10-fold cross validation. For each data set, we use each domain as a target domain in turns, and randomly choose another domain as a source domain. The classification accuracies of the compared methods over three benchmark data sets are reported in Table I. The proposed method outperforms all the compared methods over three benchmark data sets. In the experiments over the 20-Newsgroup data set, the proposed method outperforms the other methods significantly.

IV. Conclusions

In this paper, we proposed a novel transfer learning method. Instead of learning a common representation and classifier directly for both source and target domains, we proposed to learn common space and classifier, and then adapt it to source and target domains. We proposed to weight the source domain data points in the subspaces to match the distributions of the two domains, and to regularize the weighting factors of the source domain data points and the classification responses of the target domain data points by the local reconstruction coefficients. The minimization problem of our method is based on these features, and we solve it by an iterative algorithm. Experiments show its advantages over some other methods. In the future, we will extend the proposed algorithm to various applications, such as multimedia technology [9], [10], [11], [12], [13], medical imaging [14], [15], [16], [17], [18], bioinformatics [19], [20], [21], [22], material science [23], [24], high-performance computing [25], [26], [27], malicious websites detection [28], [29], [30], biometrics [31], [32], etc. We will also consider using some other models to represent and construction the classifier, such as Bayesian network [33], [34], [35].

REFERENCES


