Transfer Learning with Joint Distribution Adaptation and Maximum Margin Criterion

Pengju Wang*, Luxi Lu, Jian Li and Wenya Gan

National Key Laboratory on Blind Signal Processing, Chengdu, China

*Email: 1023568010@qq.com

Abstract. Transfer learning aims to transfer knowledge from the labelled source domain data to build a good classifier for the target domain data which has few labels or none. Existing feature-based transfer learning methods seek to transform the data to a new feature space under which the distribution discrepancy is reduced. However, data in different classes may not be easy to be separated in the new feature representation. Therefore, a modified transfer learning algorithm with Joint Distribution Adaptation (JDA) and Maximum Margin Criterion (MMC) is put forward in this paper, which we call MMC-JDA for short. MMC-JDA is aimed at minimizing the distribution discrepancy between two domains and maximizing the separability of each class at the same time. Comparative experiments on 16 cross-domain public image datasets show that MMC-JDA is effective and performs better than several common transfer learning methods.

1. Introduction

There are two basic assumptions in traditional machine learning algorithms: 1) the training and testing data must have the same or similar distribution and be in the same feature space; 2) there are enough labelled samples to guarantee the classifier to be well-trained [1, 2]. However, these two assumptions may not hold in many practical applications. On the one hand, the unlabelled testing data may not have the same or similar distribution with the labelled training data. On the other hand, getting the labelled data is laborious and time-consuming. To solve the above inapplicable problems, transfer learning [1, 2], an important research direction of machine learning, has been widely studied in these years.

Most of the transfer learning algorithms are feature-based and try transforming the data to a new feature space under which the distributions of the two domains are similar, including the marginal distribution, the conditional distribution or both, but ignore the separability of each class [3, 4]. That is to say, data in different classes may not be easy to be separated in the new feature representation. Therefore, how to increase the separability of each class at the same time of distribution adaptation becomes an urgent problem to be solved.

In this paper, a modified transfer learning algorithm with Joint Distribution Adaptation (JDA) [4] and Maximum Margin Criterion (MMC) [5] is put forward. On the one hand, by referring to JDA, Maximum Mean Discrepancy (MMD) [6] is adopted to evaluate the discrepancy of marginal and conclusion distribution. On the other hand, MMC is used to measure the separability of the different classes. Finally, considering both of them, the optimization problem and learning algorithm of our approach are presented.
We use 16 cross-domain public image datasets to conduct comparative experiments. The experiments results show that MMC-JDA performs better than several common transfer learning methods.

2. Related work
The research on transfer learning is mainly focused on the learning methods which include two categories: instance-based and feature-based.

The similarities between the two categories are that both of them try to get labels for the target domain data. In other words, they have the same objective. The differences between the two categories are that the former methods do not need feature transformation while the latter methods do. And in the target domain, the former methods require a bit of labelled data while the latter methods require plenty of unlabelled data.

The representative work of the instance-based transfer learning methods is TrAdaboot method [7]. TrAdaboot method gradually adjust the weights of the samples in the source domain such that we can use the reweighted samples to train the classifier and get labels for the target domain. The representative works of the feature-based methods are Transfer Component Analysis (TCA) method [3] and Joint Distribution Adaptation (JDA) method [4]. TCA method reduces the discrepancy of the marginal distribution under a new feature representation while JDA method reduces the discrepancy of both the marginal and conditional distribution. Our work belongs to the feature-based transfer learning methods.

3. Proposed method
The details of our proposed method are introduced in this section. Firstly, we give the problem definition of our study. Then, we introduce JDA and MMC to construct the object of our problem. Finally, the optimization problem and learning algorithm of our approach are presented.

3.1. Problem definition
Given the source domain \( \mathcal{D}_s = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \) with whole labels and the target domain \( \mathcal{D}_t = \{x_{n+1}, x_{n+2}, ..., x_{n+m}\} \) with none labels, assume that the feature space \( \mathcal{X}_s = \mathcal{X}_t \), the label space \( \mathcal{Y}_s = \mathcal{Y}_t \) but the marginal probability distribution \( P(x_i) \neq P(x_j) \), the conditional probability distribution \( P(y | x_i) \neq P(y | x_j) \).

How to use the given conditions to build a classifier and get the labels for the target domain is the problem that transfer learning needs to solve.

3.2. Joint Distribution Adaptation
The objective of Joint Distribution Adaptation (JDA) is to adapt the distributions between domains, including the marginal distribution and conditional distribution [4]. To be more specific, it refers to minimizing the following discrepancy:

\[
D(\mathcal{D}_s, \mathcal{D}_t) \approx D(P(x_i), P(x_j)) + D(P(y_i | x_i), P(y_j | x_j))
\] (1)

which is composed of two parts, the former indicates the marginal distribution discrepancy while the latter indicates the conditional distribution discrepancy.

To calculate the distribution discrepancy in equation (1), Maximum Mean Discrepancy (MMD) [6] has been adopted, such that equation (1) can be represented as

\[
D(\mathcal{D}_s, \mathcal{D}_t) \approx \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} A^T x_i - \frac{1}{n_t} \sum_{j=n_s+1}^{n_s+m} A^T x_j \right\|^2 + \sum_{c=1}^{C} \left( \frac{1}{n_s^{(c)}} \sum_{x_i \in \mathcal{Y}_s^{(c)}} A^T x_i - \frac{1}{n_t^{(c)}} \sum_{x_j \in \mathcal{Y}_t^{(c)}} A^T x_j \right)^2
\] (2)
where $A$ denotes the transformation matrix, $c \in \{1, \ldots, C\}$ denote the labels of different classes, $D(c)_{ij}$ and $D_{st}^{(c)}$ denote the data falling into class $c$ in $D_{j}$ and $D_{s}$, respectively, $n_{s}$, $n_{c}$, $n_{s}^{(c)}$ and $n_{c}^{(c)}$ indicate the number of samples in $D_{j}$, $D_{s}$, $D_{s}^{(c)}$ and $D_{t}^{(c)}$, respectively. By taking advantage of the matrix trace, equation (2) can be represented as:

$$D(D_{j}, D_{t}) \approx \text{tr}(A^T X_{s} A) + \sum_{c=1}^{C} \text{tr}(A^T X_{c} A) = \sum_{c=0}^{C} \text{tr}(A^T X_{c} A)$$  (3)

where tr($\cdot$) indicates the trace of matrix, $X$ is composed of $x_{i}$ and $x_{j}$. $M_{c}$ are MMD matrices and can be computed as follows:

$$(M_{c})_{ij} = \begin{cases} 
\frac{1}{n_{s}^{(c)} n_{c}^{(c)}}, & x_{i}, x_{j} \in D_{s}^{(c)} \\
\frac{1}{n_{s}^{(c)} n_{c}^{(c)}}, & x_{i}, x_{j} \in D_{t}^{(c)} \\
-\frac{1}{n_{s}^{(c)} n_{c}^{(c)}}, & x_{i} \in D_{s}^{(c)}, x_{j} \in D_{t}^{(c)} \\
n_{s}^{(c)} n_{c}^{(c)}, & x_{j} \in D_{s}^{(c)}, x_{i} \in D_{t}^{(c)} \\
0, & \text{otherwise}
\end{cases}$$  (4)

For the case of $c = 0$, the parameters in equation (4) should be set as $n_{s}^{(0)} = n_{s}$, $D_{s}^{(0)} = D_{s}$, $D_{t}^{(0)} = D_{t}$, $n_{s}^{(0)} = n_{s}$ and $n_{c}^{(0)} = n_{c}$. By minimizing equation (3), the marginal and conditional distribution discrepancies can be both reduced in the new feature representation $Z=A^T X$.

### 3.3. Maximum Margin Criterion

The objective of Maximum Margin Criterion (MMC) is to find a linear mapping under which the different classes can be effectively separated [5]. Separability can be measured by two quantities: the within-class scatter matrix $S_{w}$ and the between-class scatter matrix $S_{b}$. $S_{w}$ indicates how big are the differences of the samples in the same class (should be small) and $S_{b}$ indicates how far are the different classes apart (should be large). They are computed as follows:

$$S_{w} = \sum_{c=1}^{C} \sum_{x \in X_{c}} (x - \bar{x}_{c})(x - \bar{x}_{c})^{T}$$  (5)

$$S_{b} = \sum_{c=1}^{C} n_{c} (x_{c} - \bar{x})(x_{c} - \bar{x})^{T}$$  (6)

where $c \in \{1, \ldots, C\}$ denote the labels of different classes, $n_{c}$ indicates the number of samples falling into class $c$, $\bar{x}$ and $\bar{x}_{c}$ denote the sample mean for all samples and class $c$, respectively. Considering both of $S_{b}$ and $S_{w}$, MMC function is defined as

$$J(A) = \text{tr}(A^T (S_{b} - S_{w}) A)$$  (7)

where $A$ denotes the transformation matrix. By maximizing equation (7), the separability of the data in different classes can be increased in the new feature representation $Z=A^T X$. 

3
3.4. Our approach: MMC-JDA

To solve the transfer learning problem, our method MMC-JDA minimizes the distribution discrepancy between two domains and maximizes the separability of each class at the same time. Specifically, the optimization function of MMC-JDA can be derived by combining equation (3) and equation (7) as follows:

\[
\min_{A} \text{tr} \left( A^T \left( X \sum_{c=0}^{C} M_c X^T + \mu(S_u - S_b) \right) A \right) + \lambda \|A\|_F^2 \\
\text{s.t.} \quad A^T X X^T A = I, \quad \lambda \geq 0, \mu \geq 0
\]

where \( \lambda \) denotes the regularization parameter to make sure that the optimization function is well-defined, \( \mu \) is a factor to leverage the importance of MMC. It is obvious that MMC-JDA is equivalent to JDA in the case of \( \mu = 0 \).

Moreover, in the constraint condition, \( I \in \mathbb{R}^{(n_t + n_i)(n_t + n_i)} \) is the unit matrix, and \( H = I - I/(n_t + n_i) \cdot I \), denoting the center matrix. The constraint condition \( A^T X X^T A = I \) is appended to avoid the trivial solution (\( A = 0 \)), such that the transform patterns do not collapse to one point and will be preserved as in PCA [8].

Kernelization: For nonlinear problems, we can consider using a kernel mapping \( \psi: x \mapsto \psi(x) \) and kernel matrix \( K = \psi(X)^T \psi(X) \in \mathbb{R}^{(n_t + n_i)(n_t + n_i)} \), such as linear or sigmoid kernel matrix. According to Representer theorem [9], the kernel optimization function can be formulated as

\[
\min_{A} \text{tr} \left( A^T \left( K \sum_{c=0}^{C} K_c + \mu(K_u - K_b) \right) A \right) + \lambda \|A\|_F^2 \\
\text{s.t.} \quad A^T K K A = I, \quad \lambda \geq 0, \mu \geq 0
\]

where \( A \in \mathbb{R}^{n \times k} \) denotes the transformation matrix. The definition of \( K_u \) and \( K_b \) is trivial in this paper and can be referred in the literature [10].

3.5. Learning algorithm

Based on the constrained optimization theory, we denote \( \Phi = (\phi_1, \phi_2, ..., \phi_k) \) as the Lagrange multiplier. Thus, the Lagrange function for equation (8) can be derived as

\[
L = \text{tr} \left( A^T \left( X \sum_{c=0}^{C} M_c X^T + \mu(S_u - S_b) \right) A \right) + \lambda \|A\|_F^2 + \text{tr}((I - A^T X X^T A)\Phi)
\]

By setting \( \frac{\partial L}{\partial A} = 0 \), the optimization function in equation (8) can be inferred to a generalized eigendecomposition problem as follows:

\[
X \sum_{c=0}^{C} M_c X^T + \mu(S_u - S_b) + \lambda I \right) A = X X^T A \Phi
\]

Finally, we can obtain the optimal transformation matrix \( A \) by solving equation (11) and selecting its \( k \) smallest eigenvectors, where at most \( n_t + n_i - 1 \) eigenvectors can be extracted. The kernel optimization problem in equation (9) can be solved in the same way.

It is noteworthy that, the samples in target domain \( D_t \) have none labels, so we cannot evaluate the conditional distribution \( P(y_t | x_t) \), the between-class scatter matrix \( S_b \) and the within-class scatter
matrix $S_w$. Thus the classifier trained on $D_t$ can be used to predict the pseudo labels on $D_s$. The pseudo labels may not be credible, so we update them iteratively. Algorithm 1 shows a complete process of our method MMC-JDA.

Algorithm 1. MMC-JDA

**Input:** Data $X$, labels $y_s$, subspace dimension $k$, regularization parameter $\lambda$, iteration number $T$  
**Output:** Classifier $f$.

1. **begin**
2. Set $t=1$, calculate matrix $M_0$ by equation (4), set $\{M_i = 0\}^C_{i=1}$ and $S_b = S_w = 0$.
3. **repeat**
4. Solve equation (11) for its $k$ smallest eigenvectors to build the transformation matrix $A$.
5. Train the classifier $f$ on $\{A^T x_i, y_i\}^n_{i=1}$.
6. Predict the pseudo labels on $D_t$: $\{\hat{y}_j = f(A^T x_j)\}^{n_t}_{j=1}$.
7. Update matrices $\{M_i = 0\}^C_{i=0}$ by equation (4), update $S_b$ and $S_w$ by equation (5) and equation (6), update $t = t + 1$.
8. **until** $t = T$
9. **return** classifier $f$.

4. Experiments

The performance of MMC-JDA is tested through comparative experiments in this section.

4.1. Datasets

The experiments are conducted on five frequently-used public image databases: USPS (U), MNIST (M), COIL20 (CO1 and CO2), Caltech (C) and Office, where USPT and MNIST are standard image datasets for digit recognition, while COIL20, Office and Caltech are standard image datasets for object recognition. Office dataset can be further divided into three subdomains: Amazon (A), Webcam (W) and DSLR (D).

In the comparative experiments, we adopt USPS + MNIST and COIL20 datasets released by literature [4], and Office + Caltech datasets released by literature [11]. The detailed information of these datasets is introduced in table 1.

| Datasets  | Type  | Subsets | #Classes | #Samples | #Features |
|----------|-------|---------|----------|----------|----------|
| USPT     | Digit | U       | 10       | 1,800    | 256      |
| MNIST    | Digit | M       | 10       | 2,000    | 256      |
| COIL20   | Object| CO1, CO2| 20       | 1,440    | 1,024    |
| Caltech  | Object| C       | 10       | 1,123    | 800      |
| Office   | Object| A, W, D | 10       | 1,410    | 800      |

4.2. Experiments details

In the experiments, six baseline methods are chosen for comparison. Principal Component Analysis (PCA) [8] is chosen as the traditional dimension reduction method while Geodesic Flow Kernel (GFK) [11], Transfer Subspace Learning (TSL) [12], Transfer Component Analysis (TCA) [3] and Joint Distribution Adaptation (JDA) [4] are four common transfer learning algorithms. 1 Nearest Neighbor
classifier (1NN) is the basic classifier and will be used after the processing of PCA and the other transfer learning algorithms.

In the comparative study, we set the common parameters of different methods to be fixed. We set the regularization parameter $\lambda = 1$, the subspace dimension $k = 100$ and the iteration number $T = 10$. For our method, factor $\mu$ is searched in $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. For the methods based on kernel function, linear kernel i.e. $K(x_i, x_j) = x_i^T x_j$ is used. The classification accuracy on the data of target domain is used to evaluate the performance, which is defined as follows:

$$\text{Accuracy} = \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbb{1}(f(x_j) = y_j)$$

(12)

4.3. Experiment results

4.3.1. Classification accuracy.

For all datasets, we follow literature [4] to conduct 16 comparative experiments. The classification accuracy of the above seven methods is shown in table 2, in which the bold results denote the best performance of the corresponding datasets.

| Datasets     | INN   | PCA   | GFK   | TSL   | TCA   | JDA   | MMC-JDA |
|--------------|-------|-------|-------|-------|-------|-------|---------|
| U→M          | 44.70 | 44.95 | 46.45 | 53.75 | 52.20 | 58.40 | 60.10   |
| M→U          | 65.94 | 66.22 | 67.22 | 66.06 | 54.17 | 66.11 | 72.83   |
| CO1→CO2      | 83.61 | 84.72 | 72.50 | 88.06 | 88.47 | 92.64 | 93.75   |
| CO2→CO1      | 82.78 | 84.03 | 74.17 | 87.92 | 86.11 | 93.75 | 93.75   |
| C→A          | 23.70 | 36.95 | 41.02 | 44.47 | 38.20 | 44.78 | 47.60   |
| A→C          | 26.00 | 34.73 | 40.25 | 37.58 | 37.76 | 39.36 | 41.59   |
| C→W          | 25.76 | 32.54 | 40.68 | 34.24 | 38.64 | 41.69 | 45.76   |
| W→C          | 19.86 | 26.36 | 30.72 | 29.83 | 29.30 | 31.17 | 32.50   |
| C→D          | 25.48 | 38.22 | 38.85 | 43.31 | 41.40 | 45.22 | 49.68   |
| D→C          | 26.27 | 29.65 | 30.28 | 28.50 | 31.70 | 31.52 | 32.86   |
| A→W          | 29.83 | 35.59 | 38.98 | 33.90 | 37.63 | 37.97 | 41.02   |
| W→A          | 22.96 | 31.00 | 29.75 | 30.27 | 30.06 | 32.78 | 33.40   |
| A→D          | 25.48 | 27.39 | 36.31 | 26.11 | 33.12 | 39.47 | 39.49   |
| D→A          | 28.50 | 32.05 | 32.05 | 27.56 | 32.15 | 33.09 | 33.72   |
| W→D          | 59.24 | 77.07 | 80.89 | 87.26 | 87.26 | 89.17 | 89.81   |
| D→W          | 63.39 | 75.93 | 75.59 | 85.42 | 86.10 | 89.49 | 91.86   |
| Average      | 40.84 | 47.34 | 48.48 | 50.27 | 50.27 | 54.16 | 56.23   |

Based on table 2, the following conclusions can be drawn: 1) Transfer learning methods perform better than traditional machine learning methods. This demonstrates the effectiveness of transfer learning. 2) MMC-JDA achieves better performance than the other six methods. The average classification accuracy on 16 cross-domain datasets is 56.23%, which improves by 2.13% compared to JDA method. The reason is that MMC-JDA considers the distribution adaptation as well as the separability of each class, while other transfer learning methods only consider the distribution adaptation.
4.3.2. Effectiveness verification.
To verify the effectiveness of our method, TCA, JDA and MMC-JDA are run on dataset U→M. Then we compare the optimization function value calculated using equation (9) and the classification accuracy of them. Figure 1(a), figure 1(b) show the results of the three methods with increasing iteration.

From figure 1, we can observe: 1) Both the optimization function value and classification accuracy begin to converge after about eight iterations. Thus, we recommend setting the iteration number larger than eight but not too large for better performance. 2) JDA achieves better performance than TCA because it adapts the marginal distributions as well as the conditional distributions while TCA only adapts the marginal distributions. 3) MMC-JDA outperforms JDA and TCA because it not only adapts the distributions but also increases the separability of each class.

4.3.3. Parameter Analysis.
There are three major parameters in the proposed method: regularization parameter $\lambda$, factor $\mu$, and subspace dimension $k$. To analyze the effect of three parameters on the performance of MMC-JDA, the control variable method is adopted while running MMC-JDA on four representative cross-domain datasets: U→M, CO1→CO2, C→D and A→C.

The default values of the three parameters are set to $\lambda = 1$, $\mu = 0.01$, $k = 100$. The tests are run on the condition that one variable among the three parameters is varying while the other two variables are fixed. The test ranges are $\lambda \in \{0.0001, 0.001, 0.01, 0.1, 1, 10\}$, $\mu \in \{0.0001, 0.001, 0.01, 0.1, 1, 10\}$ and $k \in \{20, 40, \ldots, 180, 200\}$. We use linear kernel and set the iteration number $T = 10$. The classification accuracy under different values of the parameters is plotted in figure 2.

From figure 2, we can observe: 1) The classification accuracy increases as the regularization parameter $\lambda$ increases, but the accuracy decreases when $\lambda$ is too large. 2) The classification accuracy increases as the factor $\mu$ increases, but the accuracy decreases when $\mu$ is too large. 3) The classification accuracy increases as the subspace dimension $k$ increases, but the accuracy decreases when $k$ is too large.
Based on figure 2, we can observe that the curves plotted on different datasets show the similar trends in each figure and that choosing $\lambda \in [0.1,1]$, $\mu \in [0.0001,0.01]$ and $k \in [100,200]$ can achieve better performance.

5. Conclusion
In this paper, a modified transfer learning algorithm with Joint Distribution Adaptation (JDA) and Maximum Margin Criterion (MMC) is proposed. Compared to other transfer learning methods, the proposed method MMC-JDA not only adapts the distributions but also increases the separability of each class. Comparative experiments results show that MMC-JDA is effective and performs better than several common transfer learning methods. In the future, the research on the optimization function of transfer learning will be continued.

References
[1] Weiss K, Khoshgoftaar T M and Wang DD 2016 A survey of transfer learning Journal of Big Data
[2] Pan S J and Yang Q 2010 A Survey on Transfer Learning IEEE TKDE 22 pp 1345–59
[3] Pan S J, Tsang I W, Kwok J T, and Yang Q 2011 Domain adaptation via transfer component analysis TNN vol 22 pp 199–210
[4] Long MS, Wang JM, Ding GG, Sun JG, and Yu P S 2013 Transfer feature learning with joint distribution adaptation Proceedings of ICCV pp 2200–07
[5] Wu J, Zhou JJ and Zhu JH 2011 Radar target recognition method based on nonparametric maximum margin criterion Control and Decision 26 pp 1835–39
[6] Gretton A, Borgwardt K M, Rasch M J, Schölkopf B, and Smola A J 2006 A kernel method for the two-sample problem Proceedings of NIPS
[7] Dai WY, Yang Q, Xue GR and Yu Y 2007 Boosting for transfer learning Proc. of the 24th Int’l Conf. on Machine Learning (San Francisco: Morgan Kaufmann Publishers) pp 193-200
[8] Jolliffe I T 2002 Principal component analysis (New York: Springer)
[9] Schölkopf B and Smola A J 2002 Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond (Cambridge: MIT Press)
[10] Zhou ZH 2016 Machine Learning (Beijing: Qsinghua University Press)
[11] Gong B, Shi Y, Sha F and Grauman K 2012 Geodesic flow kernel for unsupervised domain adaptation Proceedings of CVPR
[12] Si S, Tao D, and Geng B 2010 Bregman divergence-based regularization for transfer subspace learning IEEE TKDE 22 pp 929-42