An Unsupervised Domain Adaptation Algorithm Based on Canonical Correlation Analysis

Pan Xiao[1], Bo Du[1(✉)], and Xue Li[2]

[1] School of Computer, Wuhan University, Wuhan, Hubei, China
remoteking@whu.edu.cn
[2] LIESMARS, Wuhan University, Wuhan, Hubei, China

Abstract. This paper addresses the unsupervised domain adaptation problem, which is especially challenging as the target domain does not provide explicitly label information. To solve this problem, we develop a new algorithm based on canonical correlation analysis (CCA). Specifically, we first use CCA to project both domain data onto the correlation subspace. To exploit the target domain data further, we train an SVM classifier by the source domain to obtain the pre-label of the target domain. Considering that the label space between the source and target domain may be different or even disjoint, we introduce a class adaptation matrix to adapt them. An objective function taking all factors mentioned above into consideration is designed. Finally, we learn a classification matrix by iterative optimization. Empirical studies on benchmark tasks of action recognition demonstrate that our algorithm can improve classification accuracy significantly.

Keywords: Domain adaptation · Canonical correlation analysis
Unsupervised learning

1 Introduction

In computer vision, domain adaptation (DA) has become a very popular topic. It addresses the problem that we need to solve the same learning tasks across different domains [2,20]. Generally, we can divide domain adaptation into two parts: unsupervised DA in which target domain data are completely unlabeled, and semi-supervised DA where a small number of instances in the target domain are labeled. We focus on the unsupervised scenario, which is especially challenging as the target domain does not provide explicitly any information on how to optimize classifiers. The goal of unsupervised domain adaptation is to derive a classifier for the unlabeled target domain data by extracting the information that is invariant across source and target domains.

Canonical correlation analysis (CCA) is often used to deal with DA problems since it can obtain two projection matrices to maximize the correlation between two different domains [9]. The derived correlation subspace can preserve common features of both domains very well.

J. Yang et al. (Eds.): CCCV 2017, Part III, CCIS 773, pp. 26–37, 2017.
https://doi.org/10.1007/978-981-10-7305-2_3

In our work, an efficient unsupervised domain adaptation algorithm based on CCA is developed. Specifically, we first make use of CCA to derive the correlation subspace. In order to explore the target domain data further, we use the source domain data to train a SVM classifier and then obtain the pre-label of the target domain. Considering that the label space between source and target domain may be different or even disjoint, we introduce a class adaptation matrix to adapt them. Taking all factors mentioned above into consideration, we design an objective function. Finally, a fine classifier can be obtained by iterative optimization.

The rest of the paper is organized as follows. Section 2 first introduces the related work of DA and CCA. In Sect. 3, we discusses our proposed unsupervised domain adaptation algorithm based on canonical correlation analysis in detail. Section 4 shows the experimental results in a cross-domain action recognition dataset. The last section gives some conclusive discussions.

2 Related Work

We now review some state-of-the-art domain adaptation methods and the recent works related with deep learning are also discussed. Finally, we introduce the main idea of CCA.

2.1 Domain Adaptation Methods

Generally speaking, domain adaptation problems can be solved by instance-based and feature-based approaches.

The goal of instance-based approaches is to re-weight the source domain instances by making full use of the information of target domain. For example, Dai et al. [3] proposed an algorithm based on Adaboost, which can iteratively reinforce useful samples to help train classifiers. Shi et al. [21] attempted to find a new representation for the source domain, which can reduce the negative effect of misleading samples. In [11], a heuristic algorithm was developed to remove misleading instances of the source domain. Li et al. [13] proposed a framework that can iteratively learn a common space for both domain. Several methods [15,16,26–28] proposed by Wu et al. and Liu et al. can also help us solve the domain adaptation problem effectively.

The purpose of feature-based approaches is to discover common latent features. For instance, a method integrating subspaces on the Grassman manifold was developed to learn a feature projection matrix for both domains in [7]. Zhang et al. [34] introduced a novel feature extraction algorithm, which can efficiently encode the discriminative information from limited training data and the sample distribution information from unlimited test data. In [5], a projection aligning subspaces of both domains was designed. The distributions of the feature space and the label space are considered in [8] to learn conditional transferable components. In [22–24], three subspaces extraction methods were proposed, which provides the new way to find the common subspaces of both domains. The method in

[19] attempted to project both domains into a Reproducing Kernel Hilbert Space (RKHS) and then obtain some transfer components based on Maximum Mean Discrepancy (MMD). In [30], the independence between the samples learned features and domain features is maximized to reduce the domains' discrepancy.

The discrepancies between domains [32] can be reduced through deep networks, which learn feature representation disentangling the factors of variations behind data [1]. Recent works have demonstrated that deep neural networks are powerful for learning transferable features [6,17,18,25]. Specifically, these methods embeds DA modules into deep networks to improve the performance, which mainly correct the shifts in marginal distributions, assuming conditional distributions remain unchanged after the marginal distribution adaptation. However, the recent research also finds that the features extracted in higher layers need to depend on the specific dataset [33].

2.2 Canonical Correlation Analysis

We briefly review canonical correlation analysis (CCA) as follows.

Suppose that $X^s = \{x_1^s, \ldots, x_n^s\} \in \mathbb{R}^{d_s \times n}$ and $X^t = \{x_1^t, \ldots, x_n^t\} \in \mathbb{R}^{d_t \times n}$ are source and target domain dataset respectively. n denotes the number of samples. CCA can obtain two projection vectors $u^s \in \mathbb{R}^{d_s}$ and $u^t \in \mathbb{R}^{d_t}$ to maximize the correlation coefficient ρ:

$$\max_{u^s, u^t} \rho = \frac{{u^s}^\top \sum_{st} u^t}{\sqrt{{u^s}^\top \sum_{ss} u^s}\sqrt{{u^t}^\top \sum_{tt} u^t}}, \tag{1}$$

where $\sum_{st} = X^s {X^t}^\top$, $\sum_{ss} = X^s {X^s}^\top$, $\sum_{tt} = X^t {X^t}^\top$, and $\rho \in [0, 1]$. According to [9], we can regard (1) as a generalized eigenvalue decomposition problem, there is

$$\sum\nolimits_{st} \left(\sum\nolimits_{tt}\right)^{-1} \sum\nolimits_{st}^\top u^s = \eta \sum\nolimits_{ss} u^s \tag{2}$$

Then, u^t can be calculated by $\sum_{tt}^{-1} \sum_{st}^\top u^s / \eta$ after u^s is obtained. To avoid overfitting and singularity problems, two terms $\lambda_s I$ and $\lambda_t I$ are added into $\sum_{ss}$ and $\sum_{tt}$ respectively. We have

$$\sum\nolimits_{st} \left(\sum\nolimits_{tt} + \lambda_t I\right)^{-1} \sum\nolimits_{st}^\top u^s = \eta \left(\sum\nolimits_{ss} + \lambda_s I\right) u^s \tag{3}$$

Generally speaking, we can obtain more than one pair of projection vectors $\{u_i^s\}_{i=1}^L$ and $\{u_i^t\}_{i=1}^L$. L denotes the dimensions of the CCA subspace. CCA can determine projection matrices $P_s = \{u_1^s, \ldots, u_d^s\} \in \mathbb{R}^{d_s \times L}$ and $P_t = \{u_1^t, \ldots, u_d^t\} \in \mathbb{R}^{d_t \times L}$, which can project the source and target domain data (X^s and X^t) onto the correlation subspace. Once the correlation subspace spanned by $\left\{u_i^{s,t}\right\}_{i=1}^L$ is derived, we can recognize the target domain data by the model trained from the source domain data.

3 Our Method

Our approach mainly consists of four steps. Firstly, we use the CCA to find the source and target domain's projection matrices and then project both domain data onto the correlation subspace. The second step is to train a SVM classifier to obtain the pre-label matrix of the target domain data. Then, we introduce a sigmoid function to process dataset on the correlation subspace. Finally, by minimizing the norm of classification errors, we obtain a class adaptation matrix and a classification matrix simultaneously.

3.1 The Correlation Subspace

We denote $X_S = (x_1, x_2, \ldots, x_{N_S})^\top, x_i \in \mathbb{R}^d$ as the source domain data and $X_{Tu} = (x_1, x_2, \ldots, x_{N_{Tu}})^\top, x_i \in \mathbb{R}^d$ as the target domain data. Then we can use CCA mentioned above to find the projection matrices $P_S \in \mathbb{R}^{d \times L}$ and $P_{Tu} \in \mathbb{R}^{d \times L}$ for labeled source domain and unlabeled target domain data respectively. L denotes the dimension of the correlation subspace. Moreover, we denote $X_S^P \in \mathbb{R}^{N_S \times L}$ and $X_{Tu}^P \in \mathbb{R}^{N_{Tu} \times L}$ as data matrix of source and target domain projected onto the correlation subspace. Then, we have

$$X_S^P = X_S P_S \tag{4}$$

$$X_{Tu}^P = X_{Tu} P_{Tu} \tag{5}$$

3.2 The Pre-label of Target Domain

Let $Y_S = (y_1, y_2, \ldots, y_{N_S})^T \in \mathbb{R}^{N_S \times c}$ be the label matrix of source domain with c classes. In our algorithm, we propose to obtain the pre-label of target domain by training a SVM classifier on the CCA correlation subspace. And we denote $Y_{Tu} = (y_1, y_2, \ldots, y_{N_{Tu}})^T \in \mathbb{R}^{N_{Tu} \times c}$ as the pre-label matrix.

3.3 The Sigmoid Function

What's more, a sigmoid function $G(\cdot)$ is introduced to process both domain dataset on the correlation subspace. The role of $G(\cdot)$ is to preform a non-linear mapping, which can improve the generalization ability of our model further. Specifically, we have

$$R_S = G(X_S^P) = G(X_S P_S) \tag{6}$$

$$R_{Tu} = G(X_{Tu}^P) = G(X_{Tu} P_{Tu}) \tag{7}$$

3.4 The Classification Matrix and Class Adaptation Matrix

We first define a classification matrix $\beta \in \mathbb{R}^{L \times c}$. It aims to classify both domain data onto the right class as accurate as possible. That is to say, $R_S \beta$ and $R_{Tu} \beta$

should be similar to Y_S and Y_{Tu} respectively. Specifically, we define the objective function as

$$\min_{\beta} F(\beta) = \|\beta\|_{q,p} + C_S \|R_S\beta - Y_S\|_F^2 + C_{Tu} \|R_{Tu}\beta - Y_{Tu}\|_F^2 \tag{8}$$

where $\|\cdot\|_{q,p}$ and $\|\cdot\|_F^2$ are the $l_{q,p}$-norm and Frobenius norm respectively. C_S and C_{Tu} are the penalty coefficient for both domain data. Specifically, $\|\beta\|_{q,p}$ can be written as

$$\|\beta\|_{q,p} = (\sum_{i=1}^{m}(\sum_{j=1}^{n} |\beta_{ij}|^q)^{p/q})^{1/p} \tag{9}$$

$q \geq 2$ and $0 \leq p \leq 2$ are set to impose sparsity on β. It's difficult to solve the objective function when $p = 0$, therefore, we let $p = 1$. The classification accuracy will not be improved with larger q [10], so we set $q = 2$. Finally, the objective function can be described as

$$\min_{\beta} F(\beta) = \|\beta\|_{2,1} + C_S \|R_S\beta - Y_S\|_F^2 + C_{Tu} \|R_{Tu}\beta - Y_{Tu}\|_F^2 \tag{10}$$

We also introduce a class adaptation matrix $\Theta \in \mathbb{R}^{c\times c}$ to adapt in label space. This is because the label space between source and target domains may be different [29]. So label adaptation may help obtain a better classification model. To incorporate label adaptation into our method, we can redefine the objective function as

$$\begin{aligned}\min_{\beta,\Theta} \ F(\beta, \Theta) = \|\beta\|_{2,1} + C_S \|R_S\beta - Y_S\|_F^2 + \\ C_{Tu} \|R_{Tu}\beta - Y_{Tu} \circ \Theta\|_F^2 + \gamma \|\Theta - I\|_F^2\end{aligned} \tag{11}$$

$\|\Theta - I\|_F^2$ is a term to control the class distortion. And γ is the trade-off parameter. The symbol $\circ$ denotes a multiplication operator, which can perform label adaptation between domains. In [4], the importance of unlabeled data has been emphasized. It's believed that a large number of unlabeled target domain data containing meaningful information for classification may not be fully explored. We minimize the error between the $R_{Tu}\beta$ and $Y_{Tu} \circ \Theta$ to explore the unlabeled data further.

The problem in our method turns out how to find the optimal classification matrix β and class adaptation matrix Θ simultaneously.

3.5 Optimization Algorithm

We can obtain the solution for the objective function (11) easily since β and Θ is differentiable.

Firstly, by fixing $\Theta = I$, we can get the derivative of (11) with respect to β. And there is

$$\frac{\partial F(\beta, \Theta)}{\partial \beta} = 2Q\beta + 2C_S R_S^T(R_S\beta - Y_S) + 2C_{Tu}R_{Tu}^T(R_{Tu}\beta - Y_{Tu} \circ \Theta) \tag{12}$$

in which $Q \in \mathbb{R}^{L \times L}$ is a diagonal matrix. We can regard the i-th element of Q as

$$Q_{ii} = \frac{1}{2\left\|\beta_i\right\|_2} \tag{13}$$

in which β_i can be seen as the i-th row of β.

In our algorithm, to avoid $\beta_i = 0$, we incorporate a very small value $\epsilon > 0$ into (13). Specifically, we use $\left\|\beta_i\right\|_2 + \epsilon$ to update Q. So the Eq. (13) can be rewritten as follows

$$Q_{ii} = \frac{1}{2(\left\|\beta_i\right\|_2 + \epsilon)}, \epsilon > 0 \tag{14}$$

We can let the Eq. (12) be zero, namely $\frac{\partial F(\beta,\Theta)}{\partial \beta} = 0$, then the optimal β can be obtained, there is

$$\beta = (Q + C_S R_S^T R_S + C_{Tu} R_{Tu}^T R_{Tu})^{-1}(C_S R_S^T Y_S + C_{Tu} R_{Tu}^T Y_{Tu} \circ \Theta) \tag{15}$$

Second, according to the formula (15), we substitute the fixed β value into the objective function. The optimization problem (11) becomes

$$\min_{\Theta} F(\Theta) = C_{Tu}\left\|R_{Tu}\beta - Y_{Tu} \circ \Theta\right\|_F^2 + \gamma\left\|\Theta - I\right\|_F^2 \tag{16}$$

Then, we can obtain the derivative of (16) with respect to Θ. Specifically, we have

$$\frac{\partial F(\beta,\Theta)}{\partial \Theta} = -2C_{Tu}Y_{Tu}^T(R_{Tu}\beta - Y_{Tu} \circ \Theta) + 2\gamma(\Theta - I) \tag{17}$$

Similarly, by setting (17) to be zero, we have

$$\Theta = (C_{Tu}Y_{Tu}^T Y_{Tu} + \gamma I)^{-1}(C_{Tu}Y_{Tu}^T R_{Tu}\beta + \gamma I) \tag{18}$$

The result can be obtained by iteratively optimizing β and Θ. The optimization procedure of our model is summarized in Algorithm 1. T_{max} denotes the number of maximum iteration. In this paper, we set T_{max} to be 50. Once the number of iteration reach T_{max}, the iterative update procedure would be terminated.

4 Experimental Results

4.1 Experimental Setting

Dataset. The Inria Xmas Motion Acquisition Sequences (IXMAS)[1] records 11 actions. Each action can be seen as a category. There are 12 actors involved in this action shooting and they perform each action three times. Therefore, 396 instances are captured by one camera in total. As seen from Fig. 1, five cameras (domains) are used to capture the actions simultaneously. To extract features from each image, we follow the procedure in [14]. Finally, each image can be regarded as a vector of 1000 dimensions. This dataset aims to set a standard for human action recognition.

[1] http://4drepository.inrialpes.fr/public/viewgroup/6.

Algorithm 1. Domain Adaptation Based on Canonical Correlation Analysis

```
Input: Source domain data $X_S$, Target domain data $X_{Tu}$, Source domain label $Y_S$,
    The number of maximum iteration $T_{max}$
Output: Classification matrix $\beta^t$, Class adaptation matrix $\Theta^t$
Calculate $P_S$ and $P_{Tu}$ based on canonical correlation analysis;
Calculate $X_S^P$ and $X_{Tu}^P$ using (4) and (5);
Obtain the pre-label $Y_{Tu}$ of target domain data by training a SVM classifier on the
    CCA correlation subspace;
Calculate $R_S$ and $R_{Tu}$ using (6) and (7);
$t \leftarrow 1$
$Q^t \leftarrow I_{L\times L}$
$\Theta^t \leftarrow I_{c\times c}$
while not converged ($t < T_{max}$) do
    Calculate the classification matrix $\beta^t$ using (15);
    Update $\Theta^{t+1}$ using (18);
    Update $Q^{t+1}$ using (14);
    $t \leftarrow t + 1$
end while
```

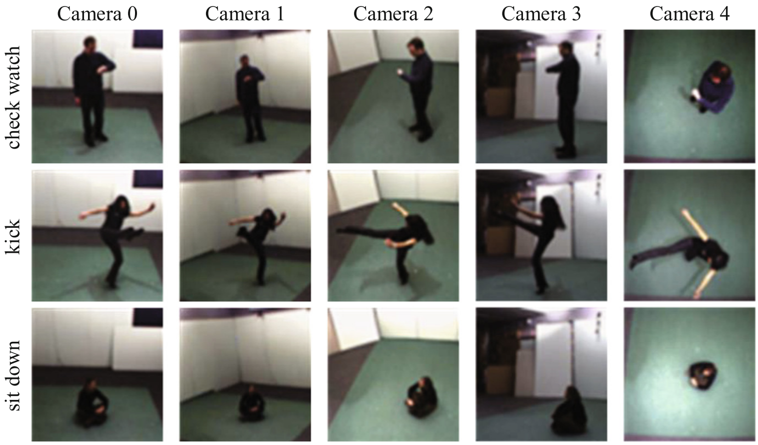

Fig. 1. Example actions of the IXMAS dataset. Each row represents an action at five different views.

Implementation Details. We follow the operation in [31] to obtain the CCA projection matrices for both domains. Specifically, two thirds of domains' samples in each catagory are selected. And the training set consists of 30 labeled samples per category in source domain and all unlabeled samples in target domain. The test set consists of all unlabeled target domain data. Then we follow the procedure mentioned in Sect. 3 to train a classifier and get the classification accuracies. The above procedure is repeated ten times. We give the average classification accuracy in Table 1.

Table 1. The classification accuracies and standard errors (%) for all methods on the IXMAS dataset

Domains	SVM	SA	TCA	GFK	MIDA	Ours
cam0→cam1	15.3±1.1	63.8±1.0	48.3±1.3	49.4±1.4	59.8±0.5	**81.3±0.8**
cam0→cam2	12.1±0.9	72.1±1.3	54.1±4.4	55.2±6.8	58.0±1.3	**86.4±1.3**
cam0→cam3	7.3±0.8	72.6±1.3	55.4±4.7	55.5±9.0	57.7±1.4	**91.9±1.0**
cam0→cam4	10.7±0.7	73.0±1.1	45.3±3.8	51.2±6.0	54.3±2.8	**92.5±1.3**
cam1→cam0	7.5±0.8	71.6±0.9	26.6±2.0	59.0±1.7	53.5±1.6	**89.0±0.8**
cam1→cam2	13.3±1.3	68.6±1.1	66.7±2.1	53.5±1.5	67.6±1.7	**84.9±1.3**
cam1→cam3	7.1±0.8	72.1±0.9	59.5±2.8	60.7±11.3	57.1±1.7	**90.9±0.9**
cam1→cam4	10.9±0.6	70.5±2.1	44.1±4.7	61.9±3.2	51.8±2.1	**92.0±1.0**
cam2→cam0	16.8±2.1	72.3±1.2	47.8±6.7	61.7±5.2	60.0±1.0	**88.7±1.4**
cam2→cam1	12.3±0.6	71.8±0.6	29.3±3.1	59.1±7.6	57.2±1.8	**84.7±0.8**
cam2→cam3	11.1±0.7	68.8±1.1	68.7±1.2	50.5±11.4	67.2±1.6	**91.6±0.8**
cam2→cam4	8.0±0.7	72.9±1.5	51.2±5.1	60.6±3.7	51.1±3.7	**88.1±1.1**
cam3→cam0	5.3±1.0	74.4±2.0	68.3±3.6	56.3±2.0	59.5±1.2	**84.0±1.1**
cam3→cam1	6.9±1.0	72.8±1.0	63.3±2.5	52.1±7.1	58.8±1.3	**90.6±1.0**
cam3→cam2	12.4±0.8	76.5±2.6	66.8±3.1	65.5±2.7	59.6±0.9	**82.5±0.9**
cam3→cam4	10.4±0.9	68.6±1.6	47.2±3.2	47.3±7.8	67.7±2.0	**87.5±1.2**
cam4→cam0	18.5±1.3	63.9±0.8	60.7±1.2	55.8±3.8	63.9±1.3	**86.5±0.7**
cam4→cam1	12.5±1.4	71.0±1.6	42.0±4.6	66.1±0.9	57.5±1.3	**89.4±1.0**
cam4→cam2	6.7±0.8	74.4±1.4	65.6±3.2	59.0±6.0	58.1±1.7	**90.4±0.9**
cam4→cam3	9.6±1.3	76.5±1.9	65.4±3.2	65.4±12.8	59.5±0.9	**89.0±0.6**
Average	10.7±1.0	71.4±1.4	53.8±3.3	57.3±5.6	59.0±1.6	**88.1±1.0**

4.2 Comparison Methods

We compare our framework with a baseline and several classic unsupervised domain adaptation methods.

SVM [12]. We regard SVM as the baseline. SVM has become a classic method to solve classification problems. To solve the DA problem, we use the original features in both domains directly. Specifically, We build a prediction model based on the source domain data and then classify instances in the target domain. Since SVM is not developed for DA problem, the final result on target domain may be the worst when compared with other methods.

Subspace Alignment (SA) [5]. This algorithm is very simple. It learns PCA subspaces of both domains at first. And then a linear mapping aligning the PCA subspaces is derived. After that, we can build models based on the source domain to classify the target domain data on the common subspace.

Transfer Component Analysis (TCA) [19]. This algorithm is designed according to maximum mean discrepancy (MMD), which can measure the distance between two distributions. By minimizing MMD, a projection matrix narrowing the distance between both domains can be obtain. This method can also map both domain data into a kernel space. In our experiments, Gaussian RBF kernels are taken.

Geodesic Flow Subspaces (GFK) [7]. This method applies the Grassman manifold to solve DA problems. First of all, the PCA or PLSA subspaces of both domains are computed. Then the subspaces are embedded into the Grassman manifold. And we can use the subspaces to obtain super-vertors by transforming the original features. Finally, low dimensional feature vectors are derived and we can train a prediction model on them.

Maximum Independence Domain Adaptation (MIDA) [30]. MIDA introduces Hilbert-Schmidt independence criterion to adapt different domains. Specifically, in order to reduce the difference across domains, we can try to obtain the maximum of the independence between the learned features and the sample features.

4.3 Parameter Tuning

In our method, there are totally four parameters including C_S, C_{Tu}, ϵ and γ. Generally speaking, it is not appropriate for an algorithm to tune the four parameters at the same time. Actually, there is no need to tune all of them. We can find the optimal solution by freezing two parameters. To be specific, we set $\epsilon = 1$ and $\gamma = 0.1$. Then we search for the best values of Cs and C_{Tu} within the ranges $\left\{4^0, 4^1, 4^2, 4^3, 4^4, 4^5, 4^6\right\}$ and $\left\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\right\}$ respectively. Finally, the best performance of our model is reported.

For SVM and other four state-of-the-art DA methods, we follow the procedures in corresponding paper to tune parameters and then report the best classification results.

4.4 Experimental Results and Comparisons

The classification accuracies and standard errors are summarized in Table 1. Cam0-cam5 represent different domains. Specifically, the form A→B states that A is the source domain and B is the target domain. For example, cam0→cam1 represents that images captured by cam0 are used as the source domain and images captured by cam1 are regarded as the target domain. The classification accuracy of SVM can be seen from the second column of Table 1. And the results of the classic unsupervised DA methods are shown in the third to the sixth column. The last column is the result of our proposed method. Totally, 20 domain pairs are given and we bold the best results for each pair. From Table 1, we can conclude that

– The classification model trained by SVM doesn't perform well. As can be seen from the table, average accuracy is around 11% and most of the results are no more than 15%. In real applications, such a model is useless.

- We can obtain better prediction models by training classifiers based on those classic unsupervised DA methods (SA, TCA, GFK, MIDA). The average classification accuracy for each method is above 50%. It's worth noting that the result of SA is highest (71.4%) compared to TCA, GFK and MIDA. That is to say, SA is more suitable to deal with IXMAS dataset.
- The classification result can be improved further by our model. Specifically, the average accuracy of our proposed algorithm is 88.1%. The result is good enough since it is improved around 77% points compared with SVM.

5 Conclusion

A new unsupervised domain adaptation algorithm based on canonical correlation analysis is proposed in this paper. Our method shows competitive performance when compared with some state-of-the-art methods, e.g. SVM, SA, TCA, GFK, MIDA.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grants U1536204, 60473023, 61471274, and 41431175, China Postdoctoral Science Foundation under Grant No. 2015M580753.

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
2. Blitzer, J., Kakade, S., Foster, D.P.: Domain adaptation with coupled subspaces, pp. 173–181 (2011)
3. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: International Conference on Machine Learning, pp. 193–200 (2007)
4. Duan, L., Xu, D., Tsang, I.W.: Domain adaptation from multiple sources: a domain-dependent regularization approach. IEEE Trans. Neural Netw. Learn. Syst. **23**(3), 504 (2012)
5. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: International Conference on Computer Vision, pp. 2960–2967 (2013)
6. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189 (2015)
7. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: Computer Vision and Pattern Recognition, pp. 2066–2073 (2012)
8. Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., Schölkopf, B.: Domain adaptation with conditional transferable components, pp. 2839–2848 (2016)
9. Hardoon, D.R., Szedmak, S.R., Shawe-Taylor, J.R.: Canonical correlation analysis: an overview with application to learning methods. Neural Comput. **16**(12), 2639–2664 (2004)
10. Hou, C., Nie, F., Li, X., Yi, D., Wu, Y.: Joint embedding learning and sparse regression: a framework for unsupervised feature selection. IEEE Trans. Cybern. **44**(6), 793–804 (2014)

11. Jiang, J., Zhai, C.X.: Instance weighting for domain adaptation in NLP. In: Meeting of the Association of Computational Linguistics, pp. 264–271 (2007)
12. Joachims, T.: Making large-scale SVM learning practical, pp. 499–526 (1998)
13. Li, X., Zhang, L., Du, B., Zhang, L., Shi, Q.: Iterative reweighting heterogeneous transfer learning framework for supervised remote sensing image classification. IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens. **PP**(99), 1–14 (2017)
14. Liu, J., Shah, M., Kuipers, B., Savarese, S.: Cross-view action recognition via view knowledge transfer. In: Computer Vision and Pattern Recognition, pp. 3209–3216 (2011)
15. Liu, T., Tao, D., Song, M., Maybank, S.J.: Algorithm-dependent generalization bounds for multi-task learning. IEEE Trans. Pattern Anal. Mach. Intell. **39**(2), 227–241 (2017)
16. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. IEEE Trans. Pattern Anal. Mach. Intell. **38**(3), 447–461 (2016)
17. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105 (2015)
18. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: Advances in Neural Information Processing Systems, pp. 136–144 (2016)
19. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2011)
20. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
21. Shi, Q., Du, B., Zhang, L.: Domain adaptation for remote sensing image classification: a low-rank reconstruction and instance weighting label propagation inspired algorithm. IEEE Trans. Geosci. Remote Sens. **53**(10), 5677–5689 (2015)
22. Tao, D., Li, X., Wu, X., Maybank, S.J.: General tensor discriminant analysis and Gabor features for gait recognition. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1700 (2007)
23. Tao, D., Li, X., Wu, X., Maybank, S.J.: Geometric mean for subspace selection. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 260–274 (2009)
24. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **28**(7), 1088–1099 (2006)
25. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: maximizing for domain invariance (2014). arXiv preprint arXiv:1412.3474
26. Wu, J., Cai, Z., Zeng, S., Zhu, X.: Artificial immune system for attribute weighted naive Bayes classification. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2013)
27. Wu, J., Hong, Z., Pan, S., Zhu, X., Cai, Z., Zhang, C.: Multi-graph-view learning for graph classification. In: 2014 IEEE International Conference on Data Mining (ICDM), pp. 590–599. IEEE (2014)
28. Wu, J., Pan, S., Zhu, X., Zhang, C., Wu, X.: Positive and unlabeled multi-graph learning. IEEE Trans. Cybern. **47**(4), 818–829 (2017)
29. Xiang, E.W., Pan, S.J., Pan, W., Su, J., Yang, Q.: Source-selection-free transfer learning. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2011, Barcelona, Catalonia, Spain, July, pp. 2355–2360 (2011)

30. Yan, K., Kou, L., Zhang, D.: Domain adaptation via maximum independence of domain features (2016)
31. Yeh, Y.R., Huang, C.H., Wang, Y.C.: Heterogeneous domain adaptation and classification by exploiting the correlation subspace. IEEE Trans. Image Process. **23**(5), 2009–2018 (2014). A Publication of the IEEE Signal Processing Society
32. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems, pp. 3320–3328 (2014)
33. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Eprint Arxiv, vol. 27, pp. 3320–3328 (2014)
34. Zhang, L., Zhang, L., Tao, D., Huang, X.: Sparse transfer manifold embedding for hyperspectral target detection. IEEE Trans. Geosci. Remote Sens. **52**(2), 1030–1043 (2014)