2012 IEEE 12th International Conference on Data Mining

Adapting Component Analysis

Fatemeh Dorri
David R. Cheriton School of Computer Science,
University of Waterloo
Waterloo, ON, Canada N2L 3G1
fdorri@uwaterloo.ca

Ali Ghodsi
Department of Statistics and Actuarial Science,
University of Waterloo,
Waterloo, ON, Canada N2L 3G1
aghodsib@uwaterloo.ca

***Abstract*—A main problem in machine learning is to predict the response variables of a test set given the training data and its corresponding response variables. A predictive model can perform satisfactorily only if the training data is an appropriate representative of the test data. This is usually reflected in the assumption that the training data and the test data are drawn from the same underlying probability distribution. However, the assumption may not be correct in many applications for various reasons. We propose a method based on kernel distribution embedding and Hilbert-Schmidt Independence Criterion (HSIC) to address this problem. The proposed method explores a new representation of the data in a new feature space with two properties: (i) the distributions of the training and the test data sets are as close as possible in the new feature space, and (ii) the important structural information of the data is preserved. The algorithm can reduce the dimensionality of the data while it preserves the aforementioned properties and therefore it can be seen as a dimensionality reduction method as well. Our method has a closed-form solution and the experimental results show that it works well in practice.**

***Keywords*-domain adaptation; kernel embedding; Hilbert-Schmidt independence criterion;**

I. Introduction

In the realm of machine learning, a model is trained to predict the response variables of a test data set. The training procedure is usually based on minimizing a loss function over all samples of a training data set and their corresponding response variables. However, learning achieves its purpose only when the training data set is a suitable representative of the test data set; otherwise the method learns unrelated information, and therefore the efficiency of the prediction is not satisfactory.

In conventional predictive models, the statement that the training data is a suitable representative of the test data is reflected in the assumption that the underlying distributions of the training and test data sets are identical. But this assumption is not always valid. Different reasons may cause the underlying probability distributions of the training and test data sets to be different. The reason might be in the difficulty or uncontrollability in gathering data.

In last decades, attention has been focused on domain adaptation problem in machine learning [1] and it has been studied under different names [2] e.g. covariate shift [3], class imbalance [4], semi-supervised learning [5], multi-task learning [6] and sample selection bias [7], [8], but all these methods mostly tackle the problem by two approaches: re-weighting source instances or changing the representation space [2].

In re-weighting source instances, weights, $w_{tr}(\mathbf{x}, \mathbf{y})$, are assigned to the pre-defined loss function which is then minimized to learn the predictive model [1], [9]. In changing the representation approach, the data is embedded into a new feature space where the probability distributions of the training data and that of the test data in the new feature space are more similar [10]. There are common drawbacks among proposed methods in literature: (i) approximation of the underlying distributions makes solving the problem hard in high dimensional data sets, (ii) exploring a new representation of the data which is not necessarily linear, usually makes solving the problem computationally expensive, and (iii) some domain adaptation techniques are applicable only to restricted predictive models.

We propose a method that overcomes the above drawbacks based on kernel distribution embedding and Hilbert-Schmidt Independence Criterion. The proposed algorithm finds a new representation of the data in a new feature space such that the underlying probability distributions of the embedded training and test data sets are as close as possible and the important structural information of the data is also preserved for any further predictive analysis. These two constraints lead to a single optimization problem which has a closed-form solution. The algorithm has a good performance when the data is mapped to a lower dimensional space. So it can be used as a dimensionality reduction technique as well.

A. Notation

Let $\mathcal{X}$ and Φ denote random variables (i.e. input variables in the original feature space and the new feature space respectively). $\mathcal{Y}$ denotes its corresponding response variables (i.e. output variables like class labels). $P(\mathcal{X}, \mathcal{Y})$ is the underlying joint probability distributions of $\mathcal{X}$ and $\mathcal{Y}$. In domain adaptation problems, the underlying probability distributions of the training and test data sets denoted by $P_{tr}(\mathcal{X}, \mathcal{Y})$ and $P_{ts}(\mathcal{X}, \mathcal{Y})$ are different. $P_{tr}(\mathcal{X})$ and $P_{ts}(\mathcal{X})$ denote the true

1550-4786/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDM.2012.85

marginal probability distributions of $\mathcal{X}$ and $\mathcal{Y}$ in the training and test data sets. Similarly, $P_{tr}(\mathcal{Y}|\mathcal{X})$ and $P_{ts}(\mathcal{Y}|\mathcal{X})$ are used to show the true conditional probability distributions in the two domains.

The bold lower case alphabet, $\mathbf{x}$, is a d-dimensional sample. X_{tr} and X_{ts} denote the matrices of the training and test data set samples of size $d \times n_{tr}$ and $d \times n_{ts}$ respectively. n_{tr} and n_{ts} are the numbers of the samples in the training and test data sets respectively. $X_{d\times n} = [X_{tr}X_{ts}]$ is a matrix of n, $d-$dimensional samples where $n = n_{tr} + n_{ts}$. Φ is the new representation of the data in the new feature space, where Φ is defined to be $\Psi : X \rightarrow \Phi$ such that $\Phi := [\Phi_{tr}\Phi_{ts}]$. Φ_{tr} and Φ_{ts} denote the embedded training and test data sets in the new representation space.

II. Method

The main challenge of domain adaptation problem is that of the non-similarity of the joint probability distributions of the training and test data sets. Decomposing them as

$$p_{tr}(\mathcal{X}, \mathcal{Y}) = p_{tr}(\mathcal{X})p_{tr}(\mathcal{Y}|\mathcal{X})$$
$$p_{ts}(\mathcal{X}, \mathcal{Y}) = p_{ts}(\mathcal{X})p_{ts}(\mathcal{Y}|\mathcal{X}),$$

it is assumed that all the difference between the joint probability distributions of the training and test data sets is due to the difference between their marginal probability distributions and there exists a new representation of the data, Φ, such that the marginal probability distributions of embedded training and test data sets are similar which means $P_{tr}(\Phi) \approx P_{ts}(\Phi)$.

A. Minimizing the Distance Between Two Probability Distributions

The crucial criterion for solving domain adaptation problem is to make the discrepancy between probability distributions of the training and the test data sets as small as possible. Maximum Mean Discrepancy (MMD) is a non parametric measure of the distance between distributions of data sets. It is a metric representative of the distance between the means of those probability distributions as follows [11]

$$\textit{MMD}(\hat{P}_{tr}, \hat{P}_{ts}) = \|\mu_{X_{tr}}[\hat{P}_{tr}] - \mu_{X_{ts}}[\hat{P}_{ts}]\|_{\mathcal{H}} = \sup_{g\in\mathcal{F}, \|g\|_{\mathcal{H}}\leq 1} (\mathbf{E}_{X_{tr}\sim\hat{P}_{tr}} g(\mathbf{x}_{tr}) - \mathbf{E}_{X_{ts}\sim\hat{P}_{ts}} g(\mathbf{x}_{tr})), \quad (1)$$

where $\mathbf{E}_{\mathbf{x}\sim P}[g(\mathbf{x})]$ is the expectation value of the function $g(\mathbf{x})$ (where the samples are drawn from probability distribution P) . It has been also shown by Jegelka et al. [11] that MMD can be estimated empirically as

$$\|\mu_{X_{tr}}[\hat{P}_{tr}] - \mu_{X_{ts}}[\hat{P}_{ts}]\|^2_{\mathcal{H}} \quad \approx \quad tr(HL_MHL_\phi),$$

where L_Φ is a kernel over Φ, let's say $\Phi^T\Phi$, and L_M is a pre-defined kernel [11]. So the objective function is to

$$\textit{minimize} \quad tr(HL_MHL_\Phi) = tr(HL_MH\Phi^T\Phi). \quad (2)$$

A trivial solution of this is to collapse all the samples of each probability distribution to one point and then make those two points close to each other. But this new representation loses crucial information in data for the future predictive analysis. Therefore, this objective function by itself is not adequate and besides minimizing the distance between the aforementioned probability distributions, the new representation should also preserve the important data features that are needed for any post analysis.

B. Preserving the Important Features of the Data

The dependency of the original data and its new representation can be used as a measure that shows how well the structure and important features for predicting the response variables are preserved. HSIC [12] is considered as a measure for quantifying the dependency of two random variables.

Two random variables are independent iff the joint probability distribution of them is equal to the multiplication of their individual probability distributions. So the dependency between two random variables can be measured based on the distance between the above probability distributions [12] and this metric is estimated empirically as

$$\textit{HSIC}(X, \Phi) = (n-1)^{-2}tr(HK_XHL_\Phi), \quad (3)$$

where L_Φ is a kernel over Φ, let's say $\Phi^T\Phi$, and K_X is a valid kernel on the original data. The choice of the kernel implies the structure and important information that is desired to be preserved.

So a supplementary objective function is

$$\textit{maximize} \quad tr(HK_XHL_\Phi) = tr(HL_MH\Phi^T\Phi). \quad (4)$$

C. Adapting Component Analysis

Minimizing the distance between $P(\Phi_{tr})$ and $P(\Phi_{ts})$ and preserving the important features of X, are incorporated to establish a single optimization problem for solving domain adaptation problem in which its solution is an embedding of the data into a new feature space. The objective function of the proposed algorithm is defined as

$$\textit{maximize} \quad \frac{tr(HK_XHL_\Phi)}{tr(HL_MHL_\Phi)}, \quad (5)$$

where the denominator is the measure for the distance between the probability distributions of the training data set and that of the test data set. Minimizing this measure or equivalently, maximizing the inverse of it, makes this distance as small as possible. The numerator is the measure for estimating the dependency between the samples in the original space and their corresponding representations. Maximizing this measure, will preserve the structure and important information of the data depending on kernel, K_X. Rewriting the optimization problem in terms of Φ we have

$$\textit{maximize} \quad \frac{tr(HK_XH\Phi^T\Phi)}{tr(HL_MH\Phi^T\Phi)} = \frac{tr(\Phi HK_XH\Phi^T)}{tr(\Phi HL_MH\Phi^T)}. \quad (6)$$

The objective function in (6) is invariant with respect to any scaling of Φ, so Φ can be chosen such that the denominator $tr(\Phi H L_M H \Phi^T)$ is equal to one:

$$\begin{array}{ll} \textit{maximize} & tr(\Phi H K_X H \Phi^T) \\ \textit{subject to} & tr(\Phi H L_M H \Phi^T) = 1. \end{array} \tag{7}$$

This optimization problem is an instance of Rayleigh quotient and finding the optimal Φ is straight forward as it has a closed-form solution. This corresponds to an eigenvector estimation problem with Φ^T as a matrix of eigenvectors of $K_X^{-1} L_M$. The number of selected eigenvectors $d' \leq d$ is the dimensionality of the data in the new feature space. If d' is chosen to be less than d, Φ represents the data X in a lower dimensional space, which means this method not only handles domain adaptation problems but also can be used as a dimensionality reduction method.

The proposed domain adaptation algorithm called Adapting Component Analysis (ACA) exploits the response variables of the training data set in addition to the training and test sets to strengthen performance of the algorithm. These are encapsulated in kernel K_X. Once the appropriate representation is found, we can apply further predictive algorithms to the samples in the new feature space.

1) Choice of kernel K_X for classification task: Exploiting the response variables of the training data set (which could sometimes be easily accessible) is valuable information that can improve the efficiency of the algorithm. Using information of the response variables can be advantageous in finding a new representation of data that is more appropriate for the following predictive analysis.

ACA finds a shared feature space where the distance between training and test data set probability distributions are reduced while the structural properties of the data set are preserved. However, the new shared feature space does not need to keep the whole structure of the data unchanged. It is only needed to keep the informative features for predicting the response variables. ACA focused on classification task where the new representation is desired to preserve structure and important features of the data relevant to the classification predictive model in a supervised manner.

Rewriting K_X based on linear kernel, we have

$$K_X = \begin{bmatrix} X_{tr}^T \\ X_{ts}^T \end{bmatrix} [X_{tr}\ X_{ts}] = \begin{bmatrix} K_{X_{tr}X_{tr}} & K_{X_{tr}X_{ts}} \\ K_{X_{ts}X_{tr}} & K_{X_{ts}X_{ts}} \end{bmatrix},$$

where $K_{X_{tr}X_{tr}}$ and $K_{X_{ts}X_{ts}}$ capture the information of the structure of training and test data sets receptively. These two sub-matrices are important for learning or training of the predictive model and they should be preserved. But intuitively the relative structure of the training data set and the test data set need not to be necessarily fixed as domains are intended to get closer. So, this can help us modifying the structure of the data in a supervised manner with the known response variables of the training data set. So the matrix K_X can be changed to $\hat{K}_X$ where $\hat{K}_X$ is constructed based on the data and the known response variables. For example, $K_{X_{ts}X_{tr}}$ is initially representing the similarity of the training data set and the test data set samples. But in classification task, if two samples of the training data set are similar, based on their response variables, they should not be different from a test data set sample perspective. So the sub-matrices of $K_{X_{ts}X_{tr}}$ and $K_{X_{tr}X_{ts}}$ can be smoothed by a process which reduces the variation of the data in unrelated dimensions while it keeps the variation of the data along the directions which contain important information relevant to predicting the response variables. Therefore, those two sub-matrices, $K_{X_{ts}X_{tr}}$ and $K_{X_{tr}X_{ts}}$ are substituted with $\hat{K}_{X_{tr}X_{ts}} = K_{Y_{tr}} K_{X_{tr}X_{ts}}$ and $\hat{K}_{X_{ts}X_{tr}} = K_{X_{ts}X_{tr}} K_{Y_{tr}}$ where $K_{Y_{tr}}$ is a kernel on the response variables of the training data set X_{tr} which represents the similarity between the labels of the training data set samples and its main role is to even out the difference between similar samples. Based on this formulation, the sample of the training data set, $\mathbf{x}_i$ is changed to the weighted mean of its similar samples. The weight is proportional to the similarity of sample $\mathbf{x}_i$ and $\mathbf{x}_j$ (that is the (i,j)th entry of the kernel $K_{Y_{tr}}$). This makes the variation of similar samples smaller. Therefore, K_X is changed to

$$\hat{K}_X = \begin{bmatrix} K_{X_{tr}X_{tr}} & \hat{K}_{X_{tr}X_{ts}} \\ \hat{K}_{X_{ts}X_{tr}} & K_{X_{ts}X_{ts}} \end{bmatrix}. \tag{8}$$

III. Experimental Results

Domain adaptation problem has been studied extensively in past decades and several methods have been developed. In this paper, the proposed method is compared with MMDE [13] and CODA [14]. MMDE is chosen since a similar measure, MMD, is considered for estimating the distance between the probability distributions of the training and test sets. CODA is chosen as a recently developed method for solving domain adaptation problem.

A. The Kernel on the Response Variables

In the objective function defined in (6), K_X and L_M are assumed to be known. Then the kernel K_X has been changed to $\hat{K}_X$ in ACA. $\hat{K}_X = \begin{bmatrix} K_{X_{tr}X_{tr}} & \hat{K}_{X_{tr}X_{ts}} \\ \hat{K}_{X_{ts}X_{tr}} & K_{X_{ts}X_{ts}} \end{bmatrix}$ where $\hat{K}_{X_{ts}X_{tr}} = K_{X_{ts}X_{tr}}(K_{Y_{tr}})$ and $\hat{K}_{X_{tr}X_{ts}} = (K_{Y_{tr}})K_{X_{tr}X_{ts}}$. $K_{Y_{tr}}$ represents the similarity of the response variables of the training data set and can be constructed in various forms. Without loss of generality, we use delta kernel for Y. Multiplication of the kernel, $K_{Y_{tr}}$, with $K_{X_{ts}X_{tr}}$ and $K_{X_{tr}X_{ts}}$ is basically substituting each sample of the training data set by the weighted mean of its corresponding similar samples which makes the variation of the data along similar samples smaller.

B. Toy Classification Example

We first test our proposed algorithm on a toy example. The training and test data set consists of 100 and 200 samples

of the two dimensional data drawn from multivariate normal distributions which their means are, $\mu_{tr} = (-1, 3)$ and $\mu_{ts} = (2, 1)$ respectively and their covariance matrices are similar which is $\sigma = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix}$.

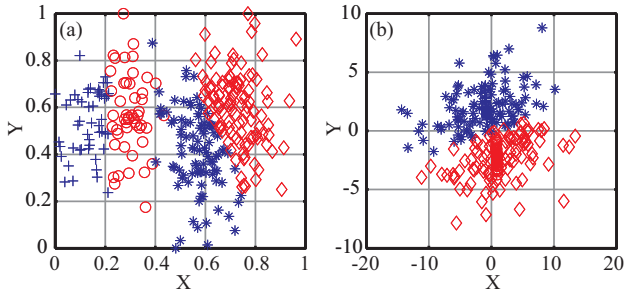

Figure 1: (a) 2-dimensional data in the original space. Circles, ∘, and crosses, × are two classes of the training data set and diamonds,◇, and stars, ∗, are two classes of the test data set. (b) The new representation of the data in the new feature space based on ACA.

The training and test data sets are categorized in two classes. The samples of each set belong to the first class if their first feature values are smaller than their corresponding mean and they belong to the second class if they are larger.

Fig. 1-a demonstrates the data in the original feature space and Fig. 1-b is the embedded data where ACA algorithms is applied. As it can be seen, the distance between the embedded training and test data set distributions is reduced. Consequently, the new training data samples are better representatives of the test data set for classification. 1-nearest neighbor (1-NN) has been used as the classifier through all experiments and The result of the classifier on the embedded data is compared with MMDE and CODA. The original data is also classified without any changes using 1-NN and the corresponding error rate is considered as the baseline. The error rate is the mean of the number of misclassified samples. As it is shown in Table I, ACA provides significant improvement in the error rate of the classification process.

Table I: The error rate comparison for different algorithms and the baseline on the toy example

DIFFERENT ALGORITHMS	BASELINE	ACA	MMDE	CODA
ERROR RATE	55%	8%	57%	53%

C. Real World Data Sets

In this section, we test the proposed method, ACA, on different real world data sets of images, text and biological data bases.

The First data set which is a collection of images, is MNIST handwritten digits [15]. This data set consists of 8-bit gray scale images of the digits between "0" and "9". Seven different data sets called Dig-1 to Dig-7 are generated from MNIST data set. Domain adaptation problem is defined as if a classifier is trained on two digits of the training set, while it is tested on two different digits of the test set. The training and test digits of the Dig-1 to Dig-7 are shown in Table II. These sets are randomly chosen among all possible cases. The number of the training and test samples are 300 and 500 respectively for all of the data sets in Table II.

The error rate of different algorithms are represented in Table II. It shows that the error rate in ACA is on average decreased to less than 10% approximately which is considerably less than the error rate of MMDE and CODA. The dimensionality of the output data in ACA is set to 2 through all experiments in this paper. The new representation of Dig-1 in 2-dimensional space is depicted in Fig. 2 which is significantly better for classification.

Table II: Different data set generated from MNIST database and and the error rate comparison for different algorithms.

NAME	TRAIN	TEST	BASELINE	ACA	MMDE	CODA
DIG-1	0, 1	3, 4	45.60%	**4.20%**	12.00%	17.20%
DIG-2	5, 7	2, 9	18.20%	**9.0%**	20.80%	18.00%
DIG-3	3, 4	1, 6	17.00%	**7.01%**	33.10%	28.20%
DIG-4	2, 8	3, 9	46.40%	**7.1%**	45.20%	41.60%
DIG-5	6, 9	5, 7	27.60%	**12.60%**	34.40%	21.60%
DIG-6	1, 3	3, 6	40.90%	**3.6%**	37.20%	10.8%
DIG-7	8, 4	3, 2	43.40%	**13.60%**	38.4%	26.8%

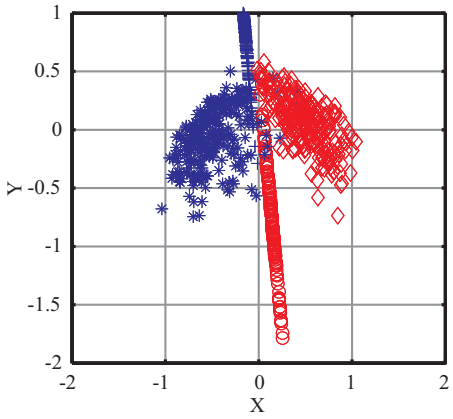

Figure 2: The 2-dimensional representation of Dig-1 data set based on ACA. Circles, ∘, and crosses, × are two classes of the training data set. Diamonds,◇, and stars, ∗, are two classes of the test data set based on labels.

The second database, the 20 Newsgroups data set, consists of about 20,000 newsgroup text documents which are categorized into four groups based on similar topics. The data set is binary occurrence of the data for 100 words across 16242 postings [16]. In order to have a domain adaptation problem, three data sets are generated such that "Newsgroup-1" data

set consists of 1000 randomly selected postings from groups 1 and 2 as the training data set, and 2000 randomly selected postings from groups 3 and 4 as the test data set. Similarly, "Newsgroup-2" and "Newsgroup-3" data sets have the same number of postings randomly selected from groups $1, 3$ and $1, 4$ in their training data sets and $2, 4$ and $2, 3$ in their test data sets respectively. In each of the artificially generated data set, the task is to classify the postings of the test data set while the algorithm is learned based on the training data set.

Our proposed method, ACA, is compared with baseline, MMDE and CODA on the Newsgroup-1, Newsgroup-2 and Newsgroup-3 data sets in Table III. The error rate is the average error over 10 trials where in each trial the samples are randomly chosen from the original data set. As it is shown in Table III, the error rate has been decreased from almost 50% to approximately $25\% - 30\%$ for ACA. ACA outperforms the other methods except in the second database which is Newsgroup-2. For Newsgroup-2 the CODA has a slightly better error rate, and that could be partly because the 2-dimensional representation of the data is not appropriate in this case or, because CODA is initially designed to solve domain adaptation problem that are characterized by missing features, and this is often the case in natural language processing while our algorithm is not developed for a specific type of data.

To further test the performance of the proposed algorithm, we run a set of classification experiments on several UCI data sets [17] in which they are biased artificially based on the so-called simple bias procedure[18]. Wine, German Credit, India diabetes and Ionosphere are the data sets from UCI archive [17] where their "Biasing Ratio" which is defined in the biasing procedure is 80%. The error rate of different methods on these data sets are also shown on Table III. It shows that ACA outperforms the other methods in these data sets as well.

Table III: Test result on various data sets by different methods. 1-nearest neighbour has been used for classification.

DATA SET	BASELINE	ACA	MMDE	CODA
NEWSGROUP-1	49.75%	**28.9%**	40.67%	32.20%
NEWSGROUP-2	43.35%	29.75%	38.2%	**25.19%**
NEWSGROUP-3	41.6%	**23.2%**	40.03%	37.1%
WINE	39.44%	**30.99%**	48.26%	31.98%
GERMAN CREDIT	41.50%	**30.06%**	40.62%	32.48%
INDIA DIABETES	42.13%	**38.01%**	40.71%	40.35%
IONOSPHERE	24.61%	22.29%	26.71%	**20.50%**

The Breast Cancer dataset from the UCI Archive [17] is a biological data set. The data includes 699 examples of benign (positive label) and malignant (negative label)samples. This is a binary classification problem from 9 initial features. The experiment is repeated with different *Biasing Ratio*s equal to 70%, 80% and 90%.

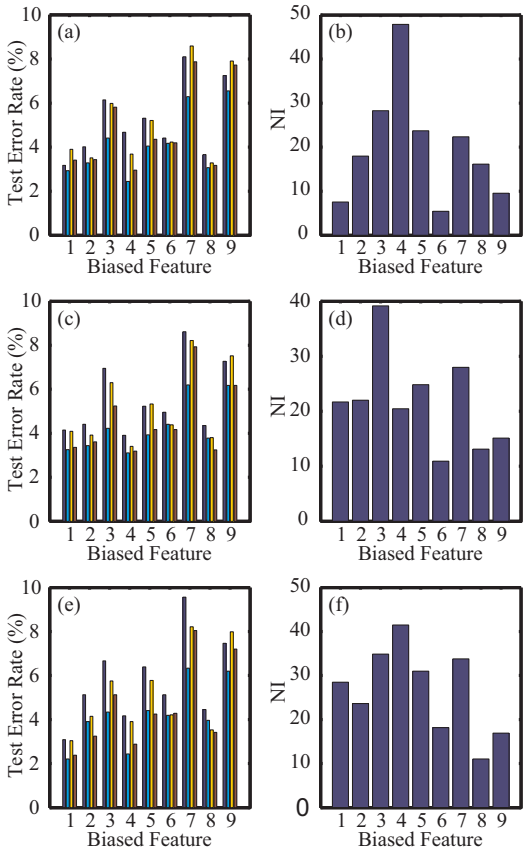

Figure 3: (a), (c) and (e) are the error rate comparison in different algorithms on Breast Cancer data set with *Biasing Ratio* of 70%, 80% and 90% respectively. The X-axis is showing the features which the biasing process is based on. (b), (d) and (f) are the Normalized Improvement of ACA with respect to the baselines of (a), (c) and (e) respectively. The bars from left to right correspond to Baseline, ACA, MMDE and CODA.

The performance of the ACA is compared with the baseline, MMDE and CODA in Fig. 3. The X-axis is the feature number that the biasing procedure is based on it. We repeat the experiment with different *Biasing Ratio*s. All the results are depicted in left column of Fig. 3. As can be seen, ACA has better performance compared with the other methods.

Another parameter for showing the efficiency of a method is Normalized Improvement (NI) which quantifies how much algorithm A outperforms with respect to the algorithm B. This parameter is estimated as

$$NI = \frac{|Error_A - Error_B|}{Error_A}. \tag{9}$$

On the right column of Fig. 3 the Normalized Improvement of ACA with respect to the baseline is shown. As can be seen after adapting the domains of training and test data sets,

the performance is improved approximately up to 50% with respect to the baseline in some features.

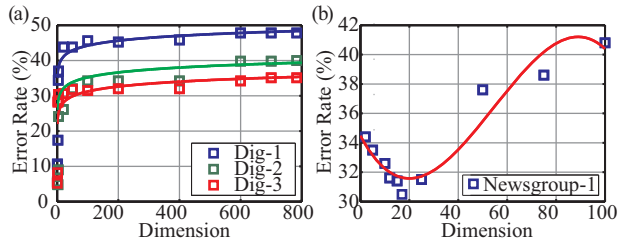

Figure 4: (a) The error rate changes of ACA vs. different dimensions on Dig-1, Dig-2 and Dig-3 data sets. (b) The error rate changes of ACA vs. different dimensions on Newsgroup-2 data set.

As it is mentioned earlier, ACA can also be used as a dimensionality reduction technique. We run the ACA algorithm on different data sets. For Dig-1, Dig-2 and Dig-3 data sets, the error rates versus the output dimension which varies from 1 to 784 is depicted in Fig. 4(a). 784 is the dimensionality of the data in original space. The error rate is minimum in low dimensional space. The changes of the error rate along different dimensions of the Newsgroup-2 is also demonstrated in Fig. 4(b). The error rate is minimum when the dimension of the data is about 15-25 in this case. As can be seen the algorithm has a good performance in low dimensions. So ACA can be considered as a dimensionality reduction technique as well. The appropriate dimension in each data set can be calculated by cross validation in practice.

IV. Conclusion

We have presented a domain adaptation algorithm in which the data samples are transferred to a new feature space. The new representation of the data is explored such that the training and the test data sets in the new feature space are as close as possible while the important structural information of the data is preserved. In order to solve this problem and satisfy the aforementioned properties, we have defined a fast optimization problem such that its solution is known to be eigenvectors of a given matrix. Our experimental results show that the algorithm performs well in practice and has a good efficiency in lower dimensions, so it can be used as a dimensionality reduction technique.

References

[1] Huang, J. Smola, A. J., Gretton, A., Borgwardt, K. M. and Scholkopf, B. Correcting Sample Selection bias by Unlabeled Data. In Scholkopf, J. P. and Hoffman, T. (ed.), *Advances in Neural Information Processing System*, chapter 19, pp. 601–608. MIT Press, Cambridge, MA, 2007.

[2] Jiang, J. A literature Survey on Domain Adaptation of Statistical Classifier. 2008. ($http://sifaka.cs.uiuc.edu/jiang4/domain-adaptation/survey/da-survey.pdf$)

[3] Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[4] Japkowicz, N. and Stephen, S. The Classic Imbalance Problem: A Systematic Study. *Intelligent Data Analysis*, 6(5):429–450, 2002.

[5] Chapelle, O., Scholkopf, B., and Zien, A., editors. Semi-Supervised Learning. *MIT Press*, 2006.

[6] Micchelli, C. A., and Pontil, M. Kernels for multi-task learning. *In Saul, L. K., Weiss, Y., and Bottou, L., editors, Advances in Neural Information Processing Systems 17* pp. 921–928, MIT Press, Cambridge, Massachusetts, USA, 2005.

[7] Heckman, J. J. Sample Selection Bias as a Specification Error. *Econometrica*, 47(1):153–161, 1979.

[8] Zadrozny, B. Learning and Evaluating Classifiers Under Sample Selection Bias. In *Proceedings of the 21th Annual International Conference on Machine Learning (ICML 2004)*, pp. 114–121, Banff, CA, 2004.

[9] Chan, Y. S., Ng, H. T. Estimating Class Priors in Domain Adaptation for Word Sense Disambigution. In *Proceedings of the 21th International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 89–96, Sydney, Australia, 2006.

[10] Daume III, H., Kumar, A. and Saha, A. Co-regularization Based Semi-supervised Domain Adaptation. In *Proceedings of the Conference on Neural Information Processing Systems(NIPS 2010)*, pp. 1–9, Vancouver, Canada, 2010.

[11] Jegelka, S., Gretton, A., Scholkopf, B., Sriperumbudur, B. K., and Luxburg., U. V. Generalized clustering via kernel embeddings. *Proceedings of the 32nd annual German conference on Advances in artificial intelligence (KI 09)*, pp. 144–52, 2009.

[12] Gretton, A., Bousquet, O., Smola, A. J., and Scholkopf, B. Measuring Statistical Dependence Between Hilbert-Schmidt Norms . *Proceedings of Algorithmic Learning Theory (ALT)*, 227:63–77, 2005.

[13] Pan, S. J., Kwok, J. T., and Yang, Q. Transfer Learning via Dimensionality Reduction. In *Proceedings of AAAI*, pp. 677–682, Chicago, Illinois, USA, 2008.

[14] Chen, C., Weinberger, K. Q., and Blitzer, J. C. Co-training for domain adaptation. In *J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, Advances in Neural Information Processing Systems 24 (NIPS-24)*, pp. 2456-2464. 2011.

[15] http://yann.lecun.com/exdb/mnist/

[16] http://www.cs.nyu.edu/ roweis/data.html

[17] http://archive.ics.uci.edu/ml

[18] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schoelkopf, B., Covariate Shift by Kernel Mean Matching In *Dataset Shift in Machine Learning, Covariate Shift and Local Learning by Distribution Matching, J. Quionero-Candela and M. Sugiyama and A. Schwaighofer and N. Lawrence (Eds.)*, MIT Press, Cambridge, MA pp.131–160, 2008.