A Simple Approach for Unsupervised Domain Adaptation

Xifeng Guo, Wei Chen, Jianping Yin
College of Computer, National University of Defense Technology, China
guoxifeng1990@163.com, cwintardis@gmail.com, jpyin@nudt.edu.cn

***Abstract*—Domain adaptation (DA) aims to eliminate the difference between the distribution of labeled source domain on which a classifier is trained and that of unlabeled or partly labeled target domain to which the classifier is to be applied. Compared with the semi-supervised domain adaptation where some labeled data from target domain is utilized to help train the classifier, the unsupervised domain adaptation where no labels can be seen from the target domain is without doubt more challenging. Most published approaches suffer from high complexity of designment or implementation. In this paper, we propose a simple method for unsupervised domain adaptation which minimizes domain shift by projecting each instance from source and target domains into a common feature space using a linear kernel function. Our method is extremely simple without hyper-parameters (it can be implemented in two lines of Matlab code) but still outperforms the state-of-the-art domain adaptation approaches on standard benchmark datasets.**

I. Introduction

Traditional machine learning algorithms perform well only when the training dataset and testing dataset share the same distribution. However, this assumption can be challenged in real world where the distribution of training data always differs from that of testing data (see Fig. 1). Or when the labels of instances in one domain are invisible or limited and we want to train a machine learning model (e.g. a classifier) on sufficient labeled data at hand but from a different domain, there will exist the distribution difference. Numerous articles[1], [2], [3] have proven that the test error of supervised methods generally increases in proportion to the difference between the distributions of training and testing samples. Addressing domain shift is undoubtedly critical for successfully applying machine learning methods in real world applications.

Many approaches have been developed to take care of domain shift, known as domain adaptation (DA) algorithms. But most of them assume that some labeled examples in the target domain are provided to learn the proper adapted model. Daumé III [4] proposes a supervised domain adaptation approach notable for its extreme simplicity: it merely augments the features by copying the original feature, then trains a classifier on the new features from both domains. The method is "frustratingly easy" yet still effective. However, it cannot handle the scenario where the target domain is unlabeled, which requires more challenging unsupervised domain adaptation approaches. While the published unsupervised domain adaptation methods suffer from either complicated designment or unbearable time complexity. The most recent "frustratingly easy" unsupervised domain adaptation is the Correlation Alignment (CORAL) [5].

Fig. 1. Example for domain shift. The bags from domain 1 do not have any background while those from domain 2 have complex background. The classifier trained on one domain will suffer from performance degradation when it is tested on the other domain

It minimizes the difference between distributions of source and target features by aligning the second-order statistics, namely, the covariance. But a hyper-parameter is required to guarantee the stability of the algorithm.

In this paper, we propose a simple unsupervised domain adaptation method inspired by feature augmentation strategy [4]. Instead of directly copying the features, we first augment the data matrix by mixing all instances from source and target domains then use a linear kernel function to get the new representation of each instance. That is to say, we take the similarities between instance $\mathbf{x}_i$ and all instances from mixed domain (source + target) as the projected feature of $\mathbf{x}_i$. Our method is simple and efficient as the only computation is matrix multiplication or linear transformation. The source code of our method can be publicly available[1]. Furthermore, the proposed strategy has the potential to be extended to multi-source domain adaptation scenarios.

The remaining of this paper is organized as follows. Section II reviews notable domain adaptation solutions. Our proposed unsupervised domain adaptation method is introduced in Section III and experimental results are presented in Section IV. We finish with conclusion in Section V.

[1]https://github.com/XifengGuo/Easy_DA_code.git

II. Related work

A variety of domain adaptation approaches have been proposed in the literature, categorized into supervised, semi-supervised and unsupervised domain adaptation.

Supervised domain adaptation techniques use the labeled source data and labeled target data to minimize domain shift. Note that unlabeled target data still may exist but remain unused. Pan et al. [6] propose a new dimensionality reduction method called maximum mean discrepancy embedding (MMDE) for domain adaptation,aiming to learn a shared latent space where distance between distributions can be reduced with the data variance preserved. The work in [1] investigates domain adaptation by metric learning techniques, which learn a transformation that minimizes the effect of domain induced changes. The feature augmented method [4] is notable for extremely simplicity. Given an instance $\mathbf{x}$, it defines the augmented feature $\tilde{\mathbf{x}} = (\mathbf{x}; \mathbf{x}; \mathbf{0})$ for instance in source domain and $\tilde{\mathbf{x}} = (\mathbf{x}; \mathbf{0}; \mathbf{x})$ for instance in target domain. Then a classifier can be trained on the augmented features.

Semi-supervised domain adaptation methods assume that some labels are available in target domain and unlabeled target data is also explored to assist the adaptation. Duan et al. [7] utilize the unlabeled target data to more precisely measure the data distribution mismatch between the source and target domains based on the maximum mean discrepancy. Method in [8] develops a subspace co-regularized method for multilingual text classification problem. It minimizes the training error on the labeled data in each language meanwhile penalizes the distance between the subspaces of the two languages of both labeled and unlabeled documents.

While unsupervised domain adaptation approaches do not use the labels but the structure of target samples to eliminate domain shift. The re-weighting techniques [9], [10] for unsupervised domain adaptation aim to minimize the distribution difference by giving each instance from source domain a weight. Recent state-of-the-art unsupervised approaches [11], [12], [13], [14] choose to project the source and target distributions into a lower-dimensional manifold, and finding a transformation that brings the subspaces closer together. Geodesic method [11] finds a path along the subspace manifold and at last finds a closed form linear map that projects source points to target. Subspace alignment method [12] computes the linear map that minimizes the Frobenius norm of the difference between the subspaces (e.g. obtained by PCA) of source and target domains. However, these approaches only align the bases of the subspaces, not the distribution of the projected data and require expensive computation for subspace projection and hyper-parameter selection. Most recently, Sun et al. [5] propose a simple unsupervised adaptation by aligning the covariances of source and target domains but still require a hyper-parameter to ensure the stability.

Our work belongs to the most challenging unsupervised domain adaptation. It linearly projects the instances from source and target domains to a shared feature space. The proposed method is free of hyper-parameter and easy to implement.

III. Proposed approach for unsupervised domain adaptation

We present an extraordinarily simple unsupervised domain adaptation method which projects source and target data into a shared common feature space using a linear kernel function. Then a SVM classifier is trained on the new representation of source data and tested on projected target data. Especially, when a linear kernel SVM classifier is applied we can derive a simple but graceful kernel.

A. Notation and formulation

In this paper, the sample $\mathbf{x}$ will be represented by a row vector. Focusing on multi-class classification task, we denote the source data by $D_S = \{\mathbf{x}_i\}_1^m$, $\mathbf{x} \in \mathbb{R}^d$ with labels $Y_S = \{y_i\}_1^m$, $y \in \{1, 2, \ldots, L\}$, and target data by $D_T = \{\mathbf{u}_j\}_1^n$, $\mathbf{u} \in \mathbb{R}^d$, where $\mathbf{x}$ and $\mathbf{u}$ are d dimensional feature representations. To be convenient, the augmented domain containing all data from source and target domain is denoted by $D_S \cup D_T$.

Inspired by augmented feature technique [4], we extract information from both domains to represent each instance. To be specific, we let the similarities between instance $\mathbf{x}_i$ and all instances in $D_S \cup D_T$ be the new feature representation of $\mathbf{x}_i$, denoted by $\hat{\mathbf{x}}_i = (\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{i(m+n)})$, i.e.

$$\hat{x}_{ij} = \text{sim}(\mathbf{x}_i, \mathbf{z}_j), \mathbf{z}_j \in D_S \cup D_T. \tag{1}$$

The new source data projected into the common feature space is denoted by $X_S = \{\hat{\mathbf{x}}_i\}_1^m$, $\hat{\mathbf{x}} \in \mathbb{R}^{m+n}$. Analogously, the new target data is $X_T = \{\hat{\mathbf{u}}_j\}_1^n$, $\hat{\mathbf{u}} \in \mathbb{R}^{m+n}$. Then a SVM classifier can be trained on X_S and applied to X_T. Notice that the dimensionality of projected feature has changed from d to $m+n$. If $m+n \gg d$, the time complexity to train the classifier can be unbearable. We can deal with this problem with a kernel SVM model which will described in next subsection.

B. Algorithm

It seems that the similarity measure in Eq. (1) is critical to our domain adaptation method. Even though the choice can be Euclidean distance, Mahalanobis distance, linear kernel function, Gaussian kernel function or any other similarity metric, the resulting performance has no much difference. To be simplified, the linear kernel function $K(\mathbf{a}, \mathbf{b}) = \mathbf{a}\mathbf{b}^\top$ is utilized to measure the similarity between instances $\mathbf{a}$ and $\mathbf{b}$. So Eq. (1) is replaced by

$$\hat{x}_{ij} = \mathbf{x}_i \mathbf{z}_j^\top, \mathbf{z}_j \in D_S \cup D_T. \tag{2}$$

And the projected source and target data are

$$\begin{aligned} X_S &= D_S \cdot (D_S \cup D_T)^\top, \\ X_T &= D_T \cdot (D_S \cup D_T)^\top. \end{aligned} \tag{3}$$

Until now, our domain adaptation task has been fulfilled, as presented in Algorithm 1. The following procedure should be training and testing classifier on X_S and X_T respectively. However, the projected data may encounter high dimensionality problem because the number of samples in source and

target domain, $m+n$, can be very large. We propose to use a linear kernel SVM trained on X_S to overcome this problem. Then the kernel matrix of X_S is

$$\begin{aligned} X_S X_S^\top &= D_S(D_S \cup D_T)^\top \left(D_S(D_S \cup D_T)^\top\right)^\top \\ &= D_S \begin{bmatrix} D_S^\top & D_T^\top \end{bmatrix} \begin{bmatrix} D_S \\ D_T \end{bmatrix} D_S^\top \\ &= D_S \left(D_S^\top D_S + D_T^\top D_T\right) D_S^\top. \end{aligned} \tag{4}$$

When test on X_T, the kernel matrix is

$$X_T X_S^\top = D_T \left(D_S^\top D_S + D_T^\top D_T\right) D_S^\top. \tag{5}$$

Let $A = D_S^\top D_S + D_T^\top D_T$, then A is apparently positive semi-definite matrix and is of size $d \times d$. In this way, we can bypass X_S and X_T and directly feed $D_S A D_S^\top$ into a SVM training model (e.g. libsvm with kernel type $t = 4$). There is no doubt that the testing results of these two strategies are consistent, but the efficiency of the latter overwhelms the former when $m + n \gg d$.

Algorithm 1: Easy unsupervised domain adaptation.

Input: Source domain D_S;
Target domain D_T.
Output: Projected source data X_S;
Projected target data X_T.

$X_S = D_S \cdot (D_S \cup D_T)^\top$;
$X_T = D_T \cdot (D_S \cup D_T)^\top$;

C. Relation to existing methods

As discussed in last subsection, we can establish a positive semi-definite matrix A to imply the linear transformation for original instance. Deploy eigenvalue decomposition on A to get $A = U\Lambda U^\top = (U\Lambda^{\frac{1}{2}})(U\Lambda^{\frac{1}{2}})^\top$ where U is formed by eigenvectors and Λ by corresponding eigenvalues of A. So our kernel function is

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{x}_i A \mathbf{x}_j^\top \\ &= (\mathbf{x}_i U\Lambda^{\frac{1}{2}})(\mathbf{x}_j U\Lambda^{\frac{1}{2}})^\top \\ &= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle. \end{aligned} \tag{6}$$

And thus the feature mapping function is $\Phi(\mathbf{x}) = \mathbf{x} \cdot (U\Lambda^{\frac{1}{2}})$ which means that the common feature space is reached by a linear projection for each instance from original domains.

The **Subspace Alignment (SA)** method [12] actually also aims to find a transformation matrix A_{sa} then trains a SVM with $D_S A_{sa} D_S^\top$ just like what we do. However, their $A_{sa} = (P_{1:k} P_{1:k}^\top)(Q_{1:k} Q_{1:k}^\top)$, where $P_{1:k}$ and $Q_{1:k}$ are the first k principal components of source data D_S and target data D_T respectively, requires selecting the dimensionality k of the subspace.

Similarly, the **Geodesic Flow Kernel (GFK)** method [11] ends up with a transformation A_g studied, although it is derived from analyzing lots of manifold subspaces. This method succeeds to calculate A_g in a closed form but still suffers from the inconvenience of determining the dimensionality of subspaces.

The **Correlation Alignment (CORAL)** approach [5] is notable for simplicity and effectiveness. It accomplishes domain adaptation task by aligning the covariances of source and target domains. The corresponding transformation matrix is

$$A_c = \left(\mathrm{cov}(D_S) + \delta I\right)^{-\frac{1}{2}} \left(\mathrm{cov}(D_T) + \delta I\right)^{-\frac{1}{2}},$$

where a parameter δ needs to be determined. Furthermore, when D_S and D_T are normalized to zero-mean, our transformation $A = d(\mathrm{cov}(D_S) + \mathrm{cov}(D_T))$ where d is the dimensionality of the sample in D_S or D_T. We can notice that both CORAL method and ours explore the fusion information of covariances of source and target domains. But apparently our method is more simple and efficient.

IV. Experiments

We evaluate our method on object recognition task using standard benchmarks and protocols [1], [11], [12], [5]. In all experiments we assume the target domain is unlabeled.

A. Datasets

We use the standard Office [1] and extended Office-Caltech10 [11] datasets as benchmarks. The Office dataset consists of images from *Webcam* (denoted by **W**), *DSLR* images (denoted by **D**) and *Amazon* images (denoted by **A**). The *Caltech10* images are denoted by **C**. So the Office-Caltech10 dataset contains totally 4 domains, each of which contains 10 categories of office objects (back_pack, bike, calculator, headphones, keyboard, laptop_computer, monitor, mouse, mug and projector). Therefore 12 domain adaptation problems can be conducted. We denote a domain adaptation problem by the notation $S \rightarrow T$, namely, **A**→**C** (train a classifier on Amazon and test on Caltech10), **C**→**W**, **W**→**A**, and so on. We use the image representations provided by [11] for Office-Caltech10 dataset (SURF features encoded with a visual dictionary of 800 words).

B. Experimental setup

We compare the proposed domain adaptation approach with three recent published domain adaptation methods (GFK [11], SA [12] and CORAL [5]) and no adaptation (NA) baseline. GFK (Geodesic Flow Kernel) and SA (Subspace Alignment) are manifold based methods that project the source and target distributions into a lower-dimensional manifold. The GFK method uses the kernel trick to integrate over an infinite number of subspaces along the subspace manifold. SA aligns the source and target subspaces by minimizing the Frobenius norm of their difference. The CORAL (Correlation Alignment) eliminates the distribution difference by aligning the covariances of source and target domains. The baseline NA directly employs the classifier trained on original source domain to test the target domain.

We use the standard random-sampling protocol and fully-transductive protocol as in [11], [12], [5] to evaluate the performances of domain adaptation methods. The random-sampling

TABLE I
ACCURACIES OF ALL 12 DOMAIN SHIFTS ON THE OFFICE-CALTECH10 DATASET UNDER RANDOM-SAMPLING PROTOCOL

	A→ C	A→ D	A→ W	C→ A	C→ D	C→ W	D→ A	D→ C	D→ W	W→ A	W→ C	W→ D	AVG
NA	35.7	34.5	25.4	42.9	39.2	31.6	35.4	31.3	71.7	32.3	25.6	78.4	40.3
SA [12]	40.0	38.3	**40.0**	47.3	39.4	40.4	35.8	34.7	67.6	36.8	32.0	66.5	43.2
GFK [11]	40.4	**39.6**	38.4	**48.6**	42.5	40.8	39.0	33.4	72.7	34.5	31.3	74.9	44.7
CORAL [5]	40.1	36.4	38.0	47.1	37.7	39.4	37.5	33.7	80.5	**37.9**	**34.3**	84.5	45.6
Ours	**40.5**	38.0	39.7	48.0	**43.1**	**41.8**	**39.1**	**34.9**	**80.8**	36.9	32.9	**84.6**	**46.7**

TABLE II
ACCURACIES OF ALL 12 DOMAIN SHIFTS ON THE OFFICE-CALTECH10 DATASET UNDER FULLY-TRANSDUCTIVE PROTOCOL

	A→ C	A→ D	A→ W	C→ A	C→ D	C→ W	D→ A	D→ C	D→ W	W→ A	W→ C	W→ D	AVG
NA	41.7	**44.6**	31.9	53.1	**47.8**	41.7	26.2	26.4	52.5	27.6	21.2	78.3	41.1
SA [12]	42.0	40.8	41.4	51.1	44.6	39.0	38.1	34.2	70.8	36.2	31.2	71.3	45.1
GFK [11]	43.9	41.4	41.4	55.2	42.7	42.0	40.3	35.3	74.2	34.3	28.9	79.6	46.6
CORAL [5]	**45.1**	39.5	**44.4**	52.1	45.9	46.4	37.7	33.8	**84.7**	36.0	**33.7**	86.6	48.8
Ours	45.0	38.2	41.0	**56.2**	**47.8**	**48.8**	**40.4**	**36.5**	83.4	**36.8**	32.1	**87.3**	**49.5**

protocol randomly selects 20 images for each category to form the training set when the source domain is **A**, **C** or **W**, and 8 images for source domain **D**. Then the training set adapted by domain adaptation approaches is used to train a SVM (libsvm) classifier and then all instances in target domain are tested by the trained classifier. For each domain adaptation problem we repeat the experiment 20 times and report the average classification accuracy. The fully-transductive protocol trains and tests the classifier on all instances of source domain and that of target domain.

C. Results

The experimental results of domain adaptation methods on the Office-Caltech10 dataset using random-sampling protocol are shown in Table I. The results of methods GFK [11], SA [12] and CORAL [5] are reported using the codes provided by the corresponding authors rather than directly using the results in their papers. Our approach outperforms the others on 7 domain adaptation problems as well as in terms of average accuracy. Note that our method is free of hyper-parameters and thus much more efficient. The results conducted under fully-transductive protocol, as shown in Table II, also demonstrate the effectiveness of our method. By comparing Table II and I, we can say that the performance difference between NA and other methods is smaller as more source data is used. This may be because when more training data is used, the intra-class difference is getting larger and the classifier needs to focus more on the "essence" of an object. This is also reflected on the observation that the NA method achieves the highest accuracy on domain adaptation problems **A**→**D** and **C**→**D** in Table II.

V. CONCLUSION

In this paper, we proposed an extremely easy domain adaptation method. It projects all instances in source and target domains into a common feature space by using a linear kernel function. The domain shift is eliminated by fusing the covariances of source and target domains. While neither complex procedure nor hyper-parameter is needed, our approach can be implemented easily. Despite of the simplicity, our approach is also effective, which is demonstrated by extensive experiments on standard benchmarks. The future work is to extend our method to multi-source domain adaptation.

ACKNOWLEDGEMENT

This work was financially supported by the National Natural Science Foundation of China (Project no. 60970034, 61170287 and 61232016).

REFERENCES

[1] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 213–226.

[2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1521–1528.

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning (ICML)*, 2014, pp. 647–655.

[4] H. Daumé III, "Frustratingly easy domain adaptation," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007, pp. 256–263.

[5] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[6] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2008, pp. 677–682.

[7] L. Duan, D. Xu, I.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 9, pp. 1667–1680, 2012.

[8] Y. Guo and M. Xiao, "Cross language text classification via subspace co-regularized multi-view learning," in *International Conference on Machine Learning (ICML)*, 2012.

[9] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems (NIPS)*, 2006, pp. 601–608.

[10] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 7, 2007, pp. 264–271.

[11] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2066–2073.

[12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2960–2967.

[13] M. Long, J. Wang, G. Ding, J. Sun, and P. Yu, "Transfer joint matching for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1410–1417.

[14] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista, "Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3846–3854.