Visual Domain Adaptation using Weighted Subspace Alignment

Shuo Chen , Fei Zhou *, Qingmin Liao

Visual Information PRoecssing Laboratory/Shenzhen Key Laboratory of Science and Technology, Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua University, China Shenzhen 518055, P.R. China

* corresponding author: flying.zhou@163.com

***Abstract*—Domain Adaptation (DA) has attracted a lot of attention in recent years. DA aims at overcoming the covariate shift in dataset and aligning multiple existing but partially related data collections. In this paper, we propose a new DA algorithm which aligns the weighted subspaces generated from source samples and target samples. The weighted subspaces of source samples are generated using weighted Principal Component Analysis (PCA). Specifically, the source samples closer to the target domain are given higher weights during the construction of subspaces, which is definitely beneficial for building an adaptable classifier. Subsequently, the weighted subspaces of source samples and the subspaces of target samples are aligned to achieve domain adaptation. Experimental results on standard datasets demonstrate the advantages of our approach over state-of-the-art DA approaches.**

***Index Terms*—Domain adaptation, Source domain, Target domain, Representation learning, Image classification**

I. Introduction

A major assumption in many machine learning algorithms is that the training and test data should have the same feature distribution. However, in many real-world applications, this assumption may not hold. In such cases, transfer learning can greatly improve the performance of learning. In recent years, transfer learning has emerged as a new learning framework to address this problem [1].

Domain Adaptation (DA) is a subproblem of transfer learning which using both source and target domains information to adapt automatically. DA is usually differentiated into two different scenarios based on the availability of labeling information from target domain [2]. In the first scenario, called semi-supervised DA, labeled target samples are available (e.g. [3]–[5]). More challengingly, in the second scenario, named unsupervised DA, labels of target samples are unavailable (e.g. [6]–[8]). In this paper we focused on the second scenario.

Earlier work in unsupervised DA mainly achieves adaptation in original feature space (e.g. [6], [9]). However, they cannot completely describe data so that these methods perform not well across domains. Recently, approaches based on subspaces are attractive in tackling unsupervised visual DA problems. Subspace-based methods for unsupervised DA are first proposed in [8]. The method of [8] uses incremental learning by gradually following the geodesic path between the source and target domain. Subsequently, the approach of [8] was extended to the infinite case, defining a new kernel equivalent to integrate over all common subspaces [10]. Furthermore, the source and target subspaces are aligned by learning a mapping function [11]. The solution of the corresponding optimization problem can be obtained in a simple closed form. A recent work in [12] relax the assumption by detecting a subset of labeled source data (landmarks) that could model the distribution of the data in the target domain well. Nevertheless, the method in [12] does not use the information from all the source samples available for training the classifier, as they use only landmarks and prune the rest. Moreover, all of the methods in [8], [10]–[12] have an underlying assumption that labeled source samples are of equal importance.

In this paper, we propose a weighted subspace alignment framework for DA by giving all the source samples different weights in the generation of subspaces. Source samples that are closer to the target domain are given higher weights. It is worth noticing that the idea of reweighting has been used in transfer learning [13]. However, the approach in [13] only reweight the source samples in building a classifier that performs better for the target samples, which does not solve the problem of different feature distribution of domains. Instead, in this paper, we align the source samples and target samples in feature space. However, it is not a good choice to align the original feature spaces since the dimension is high. Therefore, we generate the subspaces using the weight information and align the subspaces. We conduct an experiment for comparison and the results show that our algorithm has a better performance over state-of-the-art methods.

II. Proposed Algorithm

In this section, we introduce our new algorithm in detail. Our idea about this work is shown in Fig. 1. The shading of the color represents the importance of source samples where darker color means higher weight. The source samples that are closer to the target domain have deeper color, namely, higher weights. Then the weighted subspaces are generated leveraging the weight information. We suppose that there is a source domain and a target domain with different marginal distribution $\mathfrak{D}_S$ and $\mathfrak{D}_T$. Both source and target domain are lying in a D-dimensional space. There are m labeled samples from source domain constructing source matrix $\mathbf{S}$, where $\mathbf{S} = [\mathbf{s}_1 \ldots \mathbf{s}_i \ldots \mathbf{s}_m](\mathbf{s}_i \in \mathbb{R}^{D\times 1})$ and each column of $\mathbf{S}$ represents a source sample. Similarly, the target matrix $\mathbf{T}$ is composed of n target samples, where $\mathbf{T} = [\mathbf{t}_1 \ldots \mathbf{t}_j \ldots \mathbf{t}_n](\mathbf{t}_i \in \mathbb{R}^{D\times 1})$.

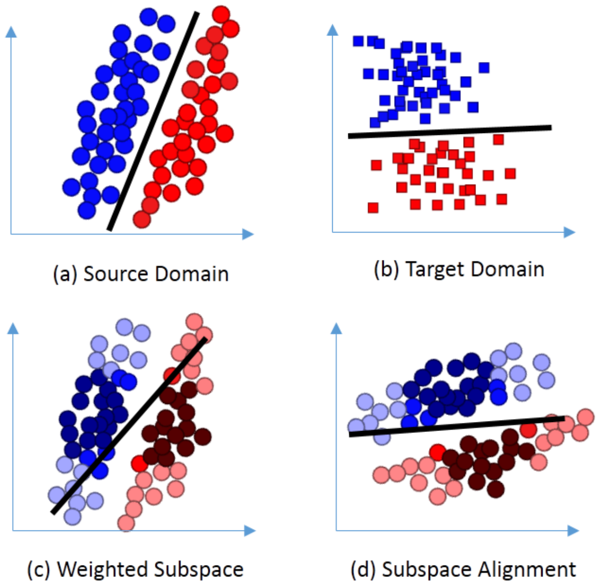

Fig. 1. Illustration of our idea (best shown in color). (a) original source domain. (b) original target domain. (c) weighted subspaces generated from source samples. (d) weighted subspaces after alignment.

A. *Weighted subspace generation*

Even though both the source and target data lie in the same D-dimensional space, they have been drawn according to different distributions. Consequently, rather than working on the original data themselves, it is suggested to handle more robust representations of the source and target domains and to learn the shift between these two domains. We use the subspaces to represent source domain and target domain and align the subspaces.

The first step of our algorithm is subspace generation. The aim of this step is to generate the subspace of source domain that are close to the target one, which is beneficial for the following alignment operation. Thus we reweight the source samples for the subspace generation. We give the source samples that are closer to the target domain higher weights. Let $\mathbf{w} = [w_1 \ldots w_m]^\mathsf{T}$, where $\mathbf{w} \in \mathbb{R}^{m\times 1}$ denote the weighting vector of source samples. The weight w_i of source sample $\mathbf{s}_i$ increases if its distribution is more similar to target domain, which can be expressed as:

$$\mathbf{w} \propto \frac{1}{\|\mathfrak{D}_\mathbf{S} - \mathfrak{D}_\mathbf{T}\|^2}. \tag{1}$$

To assign higher weights to the source samples that are closer to target domain , we need to design the strategy of weight assignment. We use the distance between source and target samples as the criterion. The distance between target sample $\mathbf{t}_j$ and source sample $\mathbf{s}_i$ is defined as follows:

$$d_{ji} = \|\mathbf{t}_j - \mathbf{s}_i\|_2^2. \tag{2}$$

We calculate the distance of every pair between target and source samples by Eq. (2) and get a distance matrix **DIST** (**DIST** $\in \mathbb{R}^{n\times m}$). The distance of target sample $\mathbf{t}_j$ and source sample $\mathbf{s}_i$ represent the disparity of samples from two domains. Then the minimum value of rows in matrix **DIST** is computed:

$$v_j = \min_{i=1\ldots m}\{d_{ji}\}, \tag{3}$$

where $\mathbf{v} = [v_1 \ldots v_j \ldots v_n]^\mathsf{T}$ ($\mathbf{v} \in \mathbb{R}^{n\times 1}$) contains the smallest distance between each sample from target domain to source domain. Then the weight of source sample $\mathbf{s}_i$ is calculated as follows:

$$w_i = w^{(0)} + \sum_{j=1}^{n} \mu(v_j - d_{ji}), \tag{4}$$

where $w^{(0)}$ is the initial weight (we set up as 1) and

$$\mu(x) = \begin{cases} 1, & x = 0 \\ 0, & x < 0. \end{cases}$$

Here the support region of function $\mu(x)$ is $(-\infty, 0]$ due to the definition of $\mathbf{v}$. From Eq. (4) we can see that if the distance d_{ji} equals the minimal distance of target instance $\mathbf{t}_j$, the weighting of source sample $\mathbf{s}_i$ will be increased. Through Eq. (4) we get the weighting vector $\mathbf{w}$ of the source samples.

Thereafter, the weighted subspace of source samples, is generated through weighted PCA. The weighted PCA operation first calculates the weighted covariance matrix through

$$\mathbf{C} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{s}_i - \overline{s})^\mathsf{T}\omega_i(\mathbf{s}_i - \overline{s}), \tag{5}$$

where $\overline{s}$ is the weighted mean and calculated through $\overline{s} = \sum_{i=1}^{m}\omega_i\mathbf{s}_i / \sum_{i=1}^{m}\omega_i$. Then we perform eigen-decomposition on $\mathbf{C}$ and select eigenvectors corresponding with the d largest eigenvalues. These eigenvectors are used as bases of the source subspaces, denoted as $\mathbf{X}_S \in \mathbb{R}^{D\times d}$. And the bases of the target subspaces are generated through PCA, denoted as $\mathbf{X}_T \in \mathbb{R}^{D\times d}$. By following the theoretical analysis of [11], the optimal value of d is able to be determined. In [11], an upper bound of d is inspired by concentration inequalities on eigenvectors. In this paper, we also use this bound to tune the number of dimensions d in PCA.

B. *Weighted subspace alignment*

After getting the weighted subspace, we design the subspace alignment correspondingly.

Subspace alignment aims to find a linear transformation $\mathbf{M}$ to best map the source eigenvectors to the target eigenvectors. $\mathbf{M}$ is learned by minimizing the following Frobenius norm:

$$F(\mathbf{M}) = \|\mathbf{X}_S\mathbf{M} - \mathbf{X}_T\|_F^2, \tag{6}$$

$$\mathbf{M} = argmin_{\mathbf{M}}(F(\mathbf{M})). \tag{7}$$

Since $\mathbf{X_S}$ and $\mathbf{X_T}$ are generated from the first d eigenvectors, they are orthonormal ($\mathbf{X}_S{}^{\mathsf{T}}\mathbf{X}_S = \mathbf{I}_d$ and $\mathbf{X}_T{}^{\mathsf{T}}\mathbf{X}_T = \mathbf{I}_d$ where $\mathbf{I}_d$ is the identity matrix of size d). Because the Frobenius norm is invariant to orthonormal operations, Eq. (7) can be written as follows:

$$F(\mathbf{M}) = \left\| \mathbf{X}_S{}^{\mathsf{T}}\mathbf{X}_S\mathbf{M} - \mathbf{X}_S{}^{\mathsf{T}}\mathbf{X}_T \right\|_F^2. \tag{8}$$

From Eq. (8), the optimal $\mathbf{M}$ can be obtained as:

$$\mathbf{M} = \mathbf{X}_S{}^{\mathsf{T}}\mathbf{X}_T \tag{9}$$

Matrix $\mathbf{M}$ transforms samples from the source subspace coordinate system into the target subspace coordinate system. The transformation $\mathbf{M}$ can be used to bring the reweighted source samples into the same subspace as the projected target samples, by computing $\mathbf{X}_S\mathbf{M}$. It means that the source samples are projected into the target aligned source subspace $\mathbf{S}_a$ via $\mathbf{S}\mathbf{X}_S\mathbf{M}$. Meanwhile, the target samples are projected into the target subspace $\mathbf{T}_a$ by $\mathbf{T}\mathbf{X}_T$. Finally, a nearest neighbor classifier is learned from this d-dimensional space. The proposed algorithm can be concluded as pseudo-code in Algorithm 1.

Algorithm 1: Weighted Subspace Alignment DA algorithm

Input : Source Data $\mathbf{S}$, Target data $\mathbf{T}$, Source labels $\mathbf{l}_S$, Subspace dimension d
Output: Predicted target labels $\mathbf{l}_T$
$\mathbf{DIST} \leftarrow (\mathbf{T}, \mathbf{S})$;
$\mathbf{v} \leftarrow min(\mathbf{DIST})$;
$\mathbf{w} \leftarrow \mu(\mathbf{v} - \mathbf{DIST})$;
$\mathbf{X}_S \leftarrow weighted\ PCA(\mathbf{S}, \mathbf{w}, d)$;
$\mathbf{X}_T \leftarrow PCA(\mathbf{T}, d)$;
$\mathbf{M} \leftarrow \mathbf{X}_S{}^{\mathsf{T}}\mathbf{X}_T$;
$\mathbf{S}_a \leftarrow \mathbf{S}\mathbf{W}\mathbf{X}_S\mathbf{M}$;
$\mathbf{T}_a \leftarrow \mathbf{T}\mathbf{X}_T$;
$\mathbf{l}_T \leftarrow classifier(\mathbf{S}_a, \mathbf{T}_a, \mathbf{l}_S)$;

III. Experiments

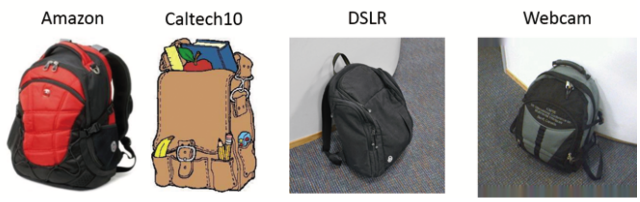

Fig. 2. Examples of Amazon, Caltech10, DSLR, and Webcam.

We evaluate our algorithm in the context of object recognition using standard datasets in comparison with the state-of-the-art DA methods in [10]–[12].

TABLE I
Accuracy (%) on target domains with unsupervised DA using a NN classifier ($\mathcal{A}$: Amazon, $\mathcal{C}$: Caltech10, $\mathcal{W}$: Webcam, and $\mathcal{D}$: DSLR)

Method	NN	GFK	LM	SA	Ours
$\mathcal{A} \to \mathcal{C}$	26.00	**40.25**	40.16	39.80	39.98
$\mathcal{A} \to \mathcal{D}$	25.48	36.31	**40.76**	36.94	38.85
$\mathcal{A} \to \mathcal{W}$	29.83	38.98	38.98	37.63	**39.32**
$\mathcal{C} \to \mathcal{A}$	23.70	41.02	41.75	42.07	**42.17**
$\mathcal{C} \to \mathcal{D}$	25.48	38.85	39.49	45.86	**46.49**
$\mathcal{C} \to \mathcal{W}$	25.76	40.68	37.97	32.20	**42.03**
$\mathcal{D} \to \mathcal{A}$	28.50	32.05	30.97	34.24	**35.38**
$\mathcal{D} \to \mathcal{C}$	26.27	30.28	31.34	32.50	**34.37**
$\mathcal{D} \to \mathcal{W}$	63.39	75.59	84.75	88.47	**89.49**
$\mathcal{W} \to \mathcal{A}$	22.96	29.75	31.00	**34.34**	33.19
$\mathcal{W} \to \mathcal{C}$	19.86	30.72	29.21	28.76	**30.89**
$\mathcal{W} \to \mathcal{D}$	59.24	80.89	83.44	88.54	**90.44**
Average	31.37	42.95	44.15	45.11	**46.88**

A. Data preparation

We use the Office dataset [6] and Caltech10 [10] dataset (extracting 10 image categories from Caltech256 [14]) that contain four domains altogether. The Office Dataset contains three object domains: Amazon (images downloaded from amazon.com), Webcam (images taken using a webcam) and DSLR (images taken from a Digital Single-Lens Reflex camera). Following the previous work [10], image representations are extracted as SURF features [15]. SURF features have been shown to be highly repeatable and robust to noise, displacement, geometric and photometric transformations. SURF features are extracted and quantized into an 800-bin histogram with codebooks from Amazon. Then the histograms are standardized by z-score. Thus we have four domains, $\mathcal{A}$ (Amazon), $\mathcal{C}$ (Caltech10), $\mathcal{D}$ (DSLR) and $\mathcal{W}$ (Webcam) leading to 12 DA problems, for example, $\mathcal{A} \to \mathcal{C}$. Fig. 2 shows several image examples of the four domains.

B. Classification results

We compare our approach with one baseline and three state-of-the-art DA methods for image recognition problems: 1-Nearest neighbor classifier (NN), Geodesic flow kernel (GFK) [10], Landmarks (LM) [12], and Subspace alignment (SA) [11]. As suggested by [10], NN classifier is chosen as the base classifier since it does not require tuning cross-validation parameters. Table I provides the quantitative results of the proposed methods in comparison with the state-of-the-art unsupervised DA approaches. Bold indicates the best result for each domain split. For a better explanation, the results are also visualized in Fig. 3. Note that, without the operation of adaptation, the standard NN classifier can only achieve an average classification accuracy of 31.37 %. Furthermore, it can be observed that our method has the best performance in 9 out of 12 DA subproblems. And in the other 3 DA tasks our algorithm has the second best performance. The average classification accuracy of our algorithm is **46.88%**, gaining a significant performance improvement of **2%** compared to methods including NN, GFK, LM, and SA. In GFK, the subspace dimension d should be small enough to ensure that

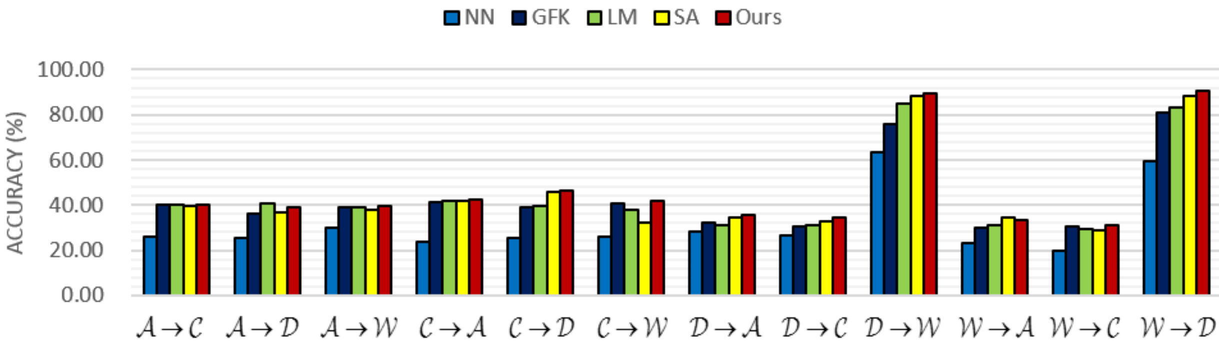

Fig. 3. Recognition accuracy (%) on 12 cross-domain datasets. (Best viewed in color)

different subspaces can transit smoothly along the geodesic flow. However, the low-dimensional subspaces can not represent input data accurately in the case of high-dimensional while the subspace dimension d in our algorithm has an theoretical upper bound which is flexible to represent the input data. Instead of selecting only a subset of source data as in LM, our algorithm utilize information from all source samples. Consequently, our algorithm performs better than LM. Moreover, the difference of accuracy between SA and our method comes from the fact that the former takes all source samples as equal, while the later reweights the source samples.

IV. Conclusion

In this paper, we present a new DA algorithm aligning the subspace generated on the reweighted samples . Through the operation of weighted subspace alignment, the source samples which distribute more similarly to the target domain are given higher weights. Then the subspaces of reweighted source samples and target samples are aligned for the purpose of adaptation. The proposed algorithm achieves excellent performance on several image classification datasets. Experiment results show that our algorithm outperforms state-of-the-art DA methods. The unsupervised DA methods mentioned in this paper are all restricted to a single source domain, while there maybe exists multiple unknown domains in source data. In future work, we plan to exploit latent domains and investigate the influence of our algorithm on multi-domains.

V. Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No.61271393 and 61301183 and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen under Grant No.JCYJ20140417115840272, ZDSYS20140509172959974 and JCYJ20150331151358138.

References

[1] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[2] V. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE, Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, May 2015.

[3] H. Daumé, A. Kumar, and A. Saha, "Frustratingly easy semi-supervised domain adaptation," *ACL Workshop on Domain Adaptation for NLP*, 2010.

[4] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 668–675.

[5] T. Yao, Y. Pan, C.-w. Ngo, H. Li, and T. Mei, "Semi-supervised Domain Adaptation with Subspace Learning for Visual Recognition," in *CVPR*, 2015.

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

[7] A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," in *Advances in Neural Information Processing Systems*, 2010, pp. 478–486.

[8] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 999–1006.

[9] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 1785–1792.

[10] 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)*. IEEE, 2012, pp. 2066–2073.

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

[12] B. Gong, K. Grauman, and F. Sha, "Learning kernels for unsupervised domain adaptation with applications to visual object recognition," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 3–27, 2014.

[13] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 193–200.

[14] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*. Springer, 2006, pp. 404–417.