

Subspace Distribution Alignment for Unsupervised Domain Adaptation

Baochen Sun
bsun@cs.uml.edu
Kate Saenko
saenko@cs.uml.edu

Computer Science Department
University of Massachusetts Lowell
Lowell, Massachusetts, US

Abstract

We propose a novel method for unsupervised domain adaptation. Traditional machine learning algorithms often fail to generalize to new input distributions, causing reduced accuracy. Domain adaptation attempts to compensate for the performance degradation by transferring and adapting source knowledge to target domain. Existing unsupervised methods project domains into a lower-dimensional space and attempt to align the subspace bases, effectively learning a mapping from source to target points or vice versa. However, they fail to take into account the difference of the two distributions in the subspaces, resulting in misalignment even after adaptation. We present a unified view of existing subspace mapping based methods and develop a generalized approach that also aligns the distributions as well as the subspace bases. We provide a detailed evaluation of our approach on benchmark datasets and show improved results over published approaches.

1 Introduction

Object recognition based on supervised machine learning methods has made great progress in recent years, with performance on common benchmarks improving at a dramatic pace. Availability of quality supervised training data has been key to this progress. Unfortunately, supervised image datasets are inherently biased [20], hampering generalization to novel test data.

The problem is that machine learning is very different from human learning. While humans can learn from very few labeled examples in one condition and generalize to novel conditions, traditional supervised machine learning requires extensive labeled data from each new condition. For example, a bicycle detector trained only on bicycles against a white background must relearn from new data to detect bikes in cluttered environments (see Figure 3 for examples). In this sense, supervised learning is “dumb,” as it only recognizes object images formed under the same conditions (pose, 3D shape, illumination, etc.) that were present in its training data.

Both theoretical [2, 9] and practical results [17, 20] show that the generalization error of supervised methods increases in proportion to the difference between the test and training input distributions. As pointed out by [9], even the state-of-the-art deep CNN features learned on a large labeled dataset of 1000 objects can only handle the smallest of distribution shifts. Clearly, addressing this issue is key to making recognition accurate enough to be deployable in real world applications.

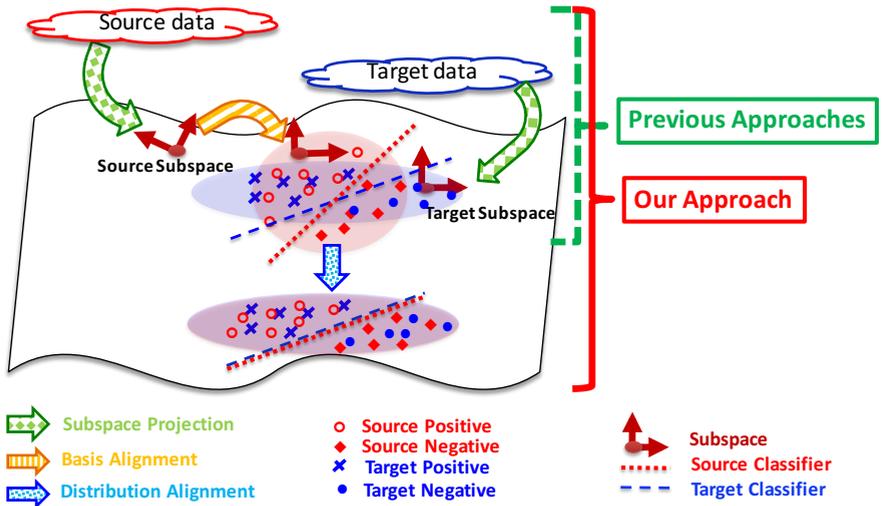


Figure 1: Existing approaches for domain adaptation via subspace mapping fail to properly align the source and target distributions. Our proposed Subspace Distribution Alignment (SDA) approach improves domain alignment by taking the difference of the source and target distributions into account. (Best viewed in color)

To compensate for the degradation in performance, many domain adaptation algorithms have been designed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], with most assuming some labeled examples in the target to learn the adaptation. In this paper, we focus on the unsupervised scenario where the target domain is unlabeled, but we have access to the target sample distribution.

Most of the existing unsupervised approaches have pursued adaptation by separately projecting the source and target distributions into a lower-dimensional manifold, and finding a transformation that brings the subspaces closer together. This process is illustrated in Figure 1. Geodesic methods [2, 10] find a path along the subspace manifold, and either project source and target onto points along that path [10], or find a closed-form linear map that projects source points to target [2]. Alternatively, the subspaces can be aligned by computing the linear map that minimizes the Frobenius norm of the difference between them, a method known as Subspace Alignment [10].

Intuitively, projecting data into a lower-dimensional subspace removes noisy dimensions and makes it easier to find the mapping. However, this approach ignores the overall distribution differences in the subspaces, even though the subspace bases are aligned. As can be seen in Figure 1, the source and target distributions in the aligned subspace can still be different due to variance differences among each dimension.

We propose a novel method called Subspace Distribution Alignment (SDA) based on the observation that aligning the distribution as well as the bases in the subspace may be beneficial. Our method can be seen as a generalization of the above subspace mapping based methods. We demonstrate the advantage of SDA, and propose two variants of it that

generalizes both the Subspace Alignment (SA) [10] and Geodesic Flow Kernel (GFK) [9] methods. We show that the infinite number of subspaces on the geodesic flow between source and target subspaces can be “hallucinated” by one “phantom” joint space where the distribution alignment is very straightforward. We evaluate our approach on standard domain adaptation benchmarks.

To summarize, our paper makes the following major contributions: (1) we propose a new method for domain adaptation which outperforms state-of-the-art methods on benchmark datasets; (2) we unify two state-of-the-art domain adaptation methods in our framework; (3) we propose a novel view of “hallucinating” infinite number of subspaces in one joint space.

2 Related work

Domain adaptation, or covariate shift, is a fundamental problem in machine learning, and has attracted a lot of attention in the machine learning and computer vision community, e.g. [9, 14] (see [13] for a comprehensive overview).

Domain adaptation methods for visual data attempt to learn classifiers on a labeled source domain and transfer it to a target domain. Early visual adaptation methods were applied to domain shift in video, including work by Duan et al. [8], who proposed to adapt video concept classifiers (e.g. *person*, *office*) between news videos collected from different news channels. [16, 17] applied domain adaptation ideas to object category classification in still images.

There are two settings for visual domain adaptation: (1) **unsupervised domain adaptation** where there are no labeled examples available in the target domain; and (2) **semi-supervised domain adaptation** where there are a few labeled examples in the target domain. Most existing algorithms operate in the semi-supervised setting [10, 9, 9, 10, 12, 15, 16, 17, 18, 21]. However, in real world applications, unlabeled target data is often much more abundant and labeled examples are very limited, so the question of how to utilize the unlabeled target data is more important for practical visual domain adaptation. Thus, in this paper, we focus on the unsupervised scenario.

The most related unsupervised approaches to ours are geodesic manifold adaptation [10], a precursor to Geodesic Flow Kernel (GFK) [9], Subspace Alignment (SA) [10], and Fast Adaptation [19]. However, as described in the next section, our method is more general and both SA and GFK can be unified in our framework. Fast Adaptation [19] proposed an adaptive version of Linear Discriminant Analysis (LDA) by incorporating both the source and target covariance structure. Since it is limited to LDA and does not use subspace projection, it is also a special case of our method.

3 Subspace Distribution Alignment (SDA)

We follow a common strategy of first projecting the source and target data into respective lower-dimensional subspaces, then finding a mapping between the subspaces [9, 9, 10].

Suppose the source data has points $D_s = [x_{s,1}, \dots]$, $x_{s,i} \in \mathbb{R}^N$, with labels $L_s = [y_1, \dots]$, while the target data has examples $D_t = [x_{t,1}, \dots]$ without label information. S_s and S_t are the d dimensional source and target subspaces respectively, which could be computed by PCA or another transformation as illustrated in Section 3.2. Our goal is to project the source training points D_s into target space. Suppose T is a matrix that transforms the source subspace to target subspace (or vice versa). After this transformation T , S_s^T projects the transformed source data back to the target space. Thus, the resulting mapping is:

$$M_s = S_s T S_t^T \quad (1)$$

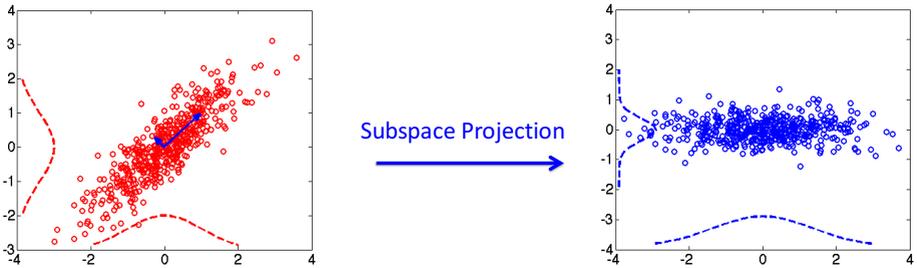


Figure 2: An illustration of why standard feature normalization fails to align distributions in subspace methods. Even though the original data are normalized to have zero mean and unit variance (left figure), subspace projection (e.g., PCA) changes the variance of the dimensions in the new basis (right figure). Thus, the variance along the bases for the source subspace is likely to be different to those of the target subspace. (Best viewed in color)

After this mapping, the classifier learned on $D_s M_s$ could be used on the target data directly. Usually T is learned in such a way that the mapped source subspace is as close to the target subspace as possible.

The intuition behind our approach is that although the above transformation might align the subspaces (the bases of the subspaces), it might not fully align the data distributions in the subspaces. We use the first- and second-order statistics, namely the mean and the variance, to describe a distribution in this paper. Other statistics (i.e. median) or even certain probability distributions (i.e. normal distribution, beta distribution, etc.) could be used as well. For example, suppose the source distribution is normal and the target distribution is beta, then the distribution alignment would be transforming the source normal distribution to target beta distribution.

Normalizing the original input data does not necessarily solve this issue. As illustrated in Figure 2, even if the original data are normalized to have zero mean and unit variance, subspace projection (i.e. PCA) changes the variance in the subspace. Thus, the variance along the bases for the source subspace is likely to be different from that of the target subspace.

Subspace alignment also does not address the problem of distribution alignment. Consider the case in Figure 1, where original source data and target data were projected into subspaces and we assume the subspace transformation T fully aligned these two subspace bases. However, the distributions are still different enough to cause reduced performance of source-trained classifiers since these distribution difference affects the decision boundary. We want the source distribution and target distribution in the subspaces to be aligned as well.

We propose Subspace Distribution Alignment (SDA), which aligns the source and target data in the subspaces, namely, it aligns the distributions as well as the subspace bases. SDA constructs the mapping between source and target points to be

$$M_s = S_s T A S_t^T \quad (2)$$

where the difference with the previous approaches is the inclusion of the A matrix. A is used to align the distributions in the subspaces. As mentioned before, we use mean and variance to describe a distribution in this paper. Since means are typically zero after data preprocessing (i.e. normalization) and are not affected by subspace projection, there is no need to align them. As covariance is the general case of variance in multidimensional space, A could be simply constructed from the covariance matrices of the source and target data

in the subspaces. Suppose W_s and W_t are the square roots of the covariance matrices of the source and target data in the subspaces respectively. These two matrices could be diagonal if the subspace bases are the principal components, as in Section 3.1, however, they may not be diagonal, as in Section 3.2. Then A is $W_s^{-1}W_t$ since W_s^{-1} transforms the source data into zero mean and unit covariance while W_t adds the covariance of target data as illustrated in Figure 1. Thus, A transforms the statistics of source subspace distribution to the target ones.

Note that W_s^{-1} is widely used in, e.g., [8, 11, 19], to decorrelate or whiten the data. However, the use of W_s^{-1} in the mainstream approaches is quite different from this paper. They use it to remove correlation (whitening) of input features, such as HOG. In our paper it aligns the distributions, as the features in the subspace are usually uncorrelated after subspace projection (i.e. PCA). As illustrated in Figure 2, A is also quite different from normalization. Other forms of the A matrix could be used as well based on the chosen distribution statistics.

In the following, we provide two variants of SDA. The first one aligns the source and target data between two subspaces (source subspace and target subspace) while the second one aligns them between an infinite number of subspaces (all the subspaces on the geodesic flow). We show that SA [10] and GFK [9] are special cases of our generalized method.

3.1 Distribution Alignment between Two Subspaces

A common approach is using the source principal components P_s and target principal components P_t learned by PCA directly as the source subspace S_s and target subspace S_t respectively. One way to obtain T is to find a matrix that maps S_s to be as close as possible to S_t in the sense of minimizing the Frobenius matrix norm [10], resulting in $T_{TS} = S_s^T S_t$. Since T_{TS} aligns the bases of S_s and S_t , the distribution alignment matrix A should also be applied between these two subspaces. Suppose E_s and E_t are the eigenvalues corresponding to S_s and S_t . Then we could set $W_s = E_s^{\frac{1}{2}}$ and $W_t = E_t^{\frac{1}{2}}$ as E_s and E_t are the variances of the orthogonal principal components. This is equivalent to using the square roots of the covariance matrix as the cross correlations are zero after PCA projection and avoids redundant calculation since E_s and E_t are returned by PCA as well. Thus, we assign $A_{TS} = W_s^{-1}W_t = E_s^{-\frac{1}{2}} E_t^{\frac{1}{2}}$. The final mapping of Subspace Distribution Alignment between Two Subspaces (SDA-TS) is:

$$M_s = S_s T_{TS} A_{TS} S_t^T = S_s (S_s^T S_t) (E_s^{-\frac{1}{2}} E_t^{\frac{1}{2}}) S_t^T \quad (3)$$

Note that, by setting A_{TS} to the identity matrix, we obtain SA as a special case. The pseudocode of SDA-TS is illustrated in Algorithm 1.

3.2 Distribution Alignment between Infinite Subspaces

As shown in [9], the kernel trick can be applied to integrate over an **infinite** number of subspaces on the geodesic flow path from the source to the target domain. At first glance, it seems impossible to align distributions between such infinite subspaces. However, in the following, we will show that the distribution alignment can be done in one “phantom” joint space computed as part of the geodesic flow kernel.

The geodesic flow $\Phi(t)$ is constructed based on source principal components P_s and target principal components P_t for $t \in [0, 1]$ where $\Phi(0) = P_s$ and $\Phi(1) = P_t$. The variable t can be thought of as a time step that specifies the amount of flow. Thus, at time $t = 0$, the flow is just a projection onto the source subspace, at time $t = 1$, it is the projection onto the target subspace, with all intermediate values projecting onto intermediate subspaces that interpolate between them. In particular, $\Phi(t) = \Omega^T \begin{bmatrix} \Gamma(t) \\ -\Sigma(t) \end{bmatrix}$ where $\Omega^T = [P_s U_1 \quad R_s U_2]$. R_s

Algorithm 1 SDA-TS for Domain Adaptation

Input: Source Data D_s , Target Data D_t
Source Labels L_s , and Subspace Dimension d
Output: Target Labels L_t
 $S_s, E_s \leftarrow SVDS(D_s, d)$ % source principal components and source subspace
 $S_t, E_t \leftarrow SVDS(D_t, d)$ % target principal components and target subspace
 $T_{TS} = S_s S_t^T$ % basis alignment matrix
 $W_s = E_s^{\frac{1}{2}}$
 $W_t = E_t^{\frac{1}{2}}$
 $A_{TS} = W_s^{-1} W_t$ % distribution alignment matrix
 $M_s \leftarrow S_s T_{TS} A_{TS} S_t^T$ % final mapping
 $L_t \leftarrow \text{Classifier}(D_s, D_t, M_s, L_s)$

is the orthogonal complement of P_s and U_1, U_2 are derived using generalized singular value decomposition (GSVD) of $P_s^T P_t$ and $R_s^T P_t$. Both $\Gamma(t)$ and $\Sigma(t)$ are diagonal matrices with elements equal to the cosine and sine of the principal angles θ between P_s and P_t .

We can think of Ω^T as a joint space that is constructed from both the source principal components P_s and target principal components P_t . As shown above, each geodesic flow projection is created by first performing a fixed projection Ω^T to map points to this joint space, followed by a rotation $\begin{bmatrix} \Gamma(t) \\ -\Sigma(t) \end{bmatrix}$ to “hallucinate” the intermediate subspace. The last step depends only on the angles θ between P_s and P_t . Since Ω^T is independent of θ , the kernel trick is applied to the rotation $\begin{bmatrix} \Gamma(t) \\ -\Sigma(t) \end{bmatrix}$ alone. It integrates over an infinite number

of principal angles θ to obtain the final transformation $T_{IS} = \begin{bmatrix} \Lambda_1 & \Lambda_2 \\ \Lambda_2 & \Lambda_3 \end{bmatrix}$. Here, $\Lambda_{1,2,3}$ are diagonal matrices based on θ where the diagonal elements of $\Lambda_{1,2,3}$ are $\lambda_{1i} = 1 + \frac{\sin(2\theta_i)}{2\theta_i}$, $\lambda_{2i} = \frac{\cos(2\theta_i) - 1}{2\theta_i}$, and $\lambda_{3i} = 1 - \frac{\sin(2\theta_i)}{2\theta_i}$. Based on this definition of geodesic flow, T_{IS} plays the role of hallucinating and then integrating over an infinite number of intermediate subspaces.

As Ω^T maps points to the same joint space (constructed from both P_s and P_t), we can apply the distribution alignment A_{IS} in this space. Let $\hat{D}_s = D_s \Omega^T$ and $\hat{D}_t = D_t \Omega^T$ be the projected source and target data. Their corresponding covariance matrices are $\Sigma_s = \text{cov}(\hat{D}_s)$ and $\Sigma_t = \text{cov}(\hat{D}_t)$. Thus, $A_{IS} = W_s^{-1} W_t = \Sigma_s^{-\frac{1}{2}} \Sigma_t^{\frac{1}{2}}$. The resulting mapping of Subspace Distribution Alignment between Infinite Subspaces (SDA-IS) is:

$$M_s = S_s T_{IS} A_{IS} S_t^T = [P_s U_1 \quad R_s U_2] \begin{bmatrix} \Lambda_1 & \Lambda_2 \\ \Lambda_2 & \Lambda_3 \end{bmatrix} A_{IS} [P_s U_1 \quad R_s U_2]^T \quad (4)$$

By setting A_{IS} to the identity matrix, we obtain GFK as a special case. The pseudocode of SDA-IS is illustrated in Algorithm 2.

4 Experiments

In this paper, we evaluate our methods in the context of object recognition. We use the standard datasets and protocols of [10, 9, 11, 12, 13]. We compare our methods to their special cases as well as a baseline method that does not perform any adaptation.

Algorithm 2 SDA-IS for Domain Adaptation

Input: Source Data D_s , Target Data D_t

 Source Labels L_s , and Subspace Dimension d
Output: Target Labels L_t
 $P_s \leftarrow SVDS(D_s, d)$

% source principal components

 $P_t \leftarrow SVDS(D_t, d)$

% target principal components

 $R_s \leftarrow NULL(P_s^T)$
 $U_1, U_2 \leftarrow GSVD(P_s^T P_t, R_s^T P_t)$
 $\Omega^T \leftarrow [P_s U_1 \ R_s U_2]$

% joint space

 $T_{IS} \leftarrow \text{Kernel Trick}$

% hallucinating and integrating over infinite subspaces

 $\hat{D}_s \leftarrow D_s \Omega^T$
 $\hat{D}_t \leftarrow D_t \Omega^T$
 $W_s \leftarrow COV(\hat{D}_s)^{\frac{1}{2}}$
 $W_t \leftarrow COV(\hat{D}_t)^{\frac{1}{2}}$
 $A_{IS} = W_s^{-1} W_t$

% distribution alignment matrix

 $M_s \leftarrow S_s T_{IS} A_{IS} S_t^T$

% final mapping

 $L_t \leftarrow \text{Classifier}(D_s, D_t, M_s, L_s)$



Figure 3: Sample images from the Office-Caltech10 dataset.

4.1 Datasets

We use the same Office-Caltech10 dataset as GFK [9] and SA [10], to allow direct comparison. It contains 10 object categories (backpack, bike, calculator, headphones, keyboard, laptop computer, monitor, mouse, mug, and projector) in four image domains: *Webcam*, *DSLR*, *Amazon*, and *Caltech256*. We additionally conduct experiments on the original Office31 dataset [11], which contains 31 object categories (the same 10 categories of Office-Caltech10 plus 21 additional categories) in three domains: *Webcam*, *DSLR*, and *Amazon*. Figure 3 shows example images from the Office-Caltech10 dataset.

We use the standard SURF features released by the dataset authors [9, 11]. In these features, the images were encoded with 800-bin bag-of-words histograms and normalized to zero mean and unit standard deviation in each dimension. Since there are four domains for Office-Caltech10 dataset, there are **12** experiment settings, namely, A-C (train classifier on (A)mazon, test on (C)altech), A-D (train on (A)mazon, test on (D)SLR), A-W, and so on. For the Office31 dataset, there are **6** experiment settings as there are only three domains.

4.2 Setup

We compare our methods SDA-TS and SDA-IS to their special cases SA and GFK respectively. Our baseline is NA (no adaptation), where we use the classifier trained on the original

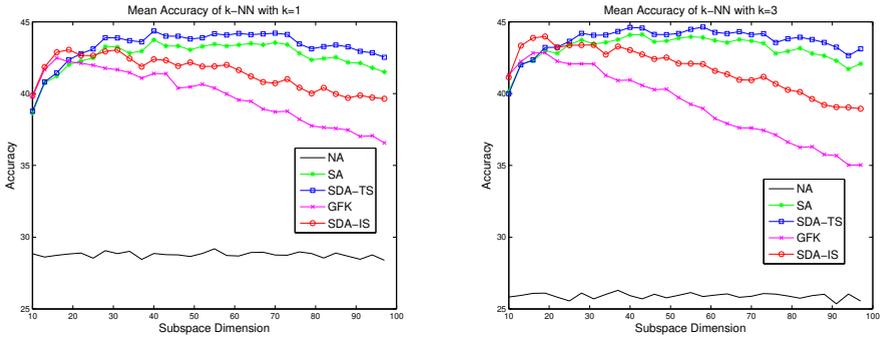


Figure 4: Mean accuracy across all **12** experiment settings (domain shifts) of the k -NN Classifier on the **Office-Caltech10** dataset. Both our methods SDA-IS and SDA-TS outperform GFK and SA consistently. The x-axis shows the dimensionality of the subspace. For NA, as it uses the original feature space without subspace projection, we just average over 20 randomized trials at each subspace dimension. **Left:** k -NN Classifier with $k=1$; **Right:** k -NN Classifier with $k=3$. (Best viewed in color)

source domain directly on the target domain. For SA and GFK, we use the source code provided by the authors. We follow the protocol of [10, 9] and use k -NN (with $k = 1$) as base classifier. To assess stability with respect to the k parameter, we also set k to 3.

One key parameter of SA and GFK is the subspace dimension d . As shown in [10, 9] and our own experiments, the optimal d is always between 10 and 100 for these datasets. d is usually set by cross-validation [10] or some distance measure [9]. Since we want to show that our methods outperform their special cases consistently, we run experiments on all the subspace dimensions from 10 to 100.

For each of the 12 or 6 domain shifts, we conduct experiments using the same setting as [10, 9]. Briefly, for each subspace dimension per domain shift, we conduct experiments in 20 randomized trials and get the average accuracy. In each trial, we randomly sample a certain number (20 for *Amazon*, *Caltech*, and *Webcam*; 8 for *DSLR* as there are only 8 images per category in the *DSLR* domain) of labelled images in the source domain as training set, and use the unlabelled data in the target domain as the test set. The final mean accuracy is averaged over the 12 or 6 domain shifts.

For NA, as it uses the original features, we just average over 20 randomized trials without doing any subspace projection.

4.3 Results

Figure 4 and Figure 5 show the mean accuracy across all the domain shifts of k -NN for Office-Caltech10 and Office31 respectively. From the results we can see that **both** our methods SDA-TS and SDA-IS outperform their special cases SA and GFK consistently on **both** datasets. This is true for nearest neighbor parameters $k=1$ and $k=3$, with similar overall trends. All methods outperform the baseline NA by a large margin. The improvement of SDA-IS over GFK is larger than the improvement of SDA-TS over SA. One possible explanation is that the joint space Ω^T of GFK contains more dimensions (same dimension as the original source and target data) than the source subspace S_s and target subspace S_t of SA. Thus, there is more room for improvement. We also note that the performance of infinite-subspace methods tends to decline faster with larger subspace dimension.

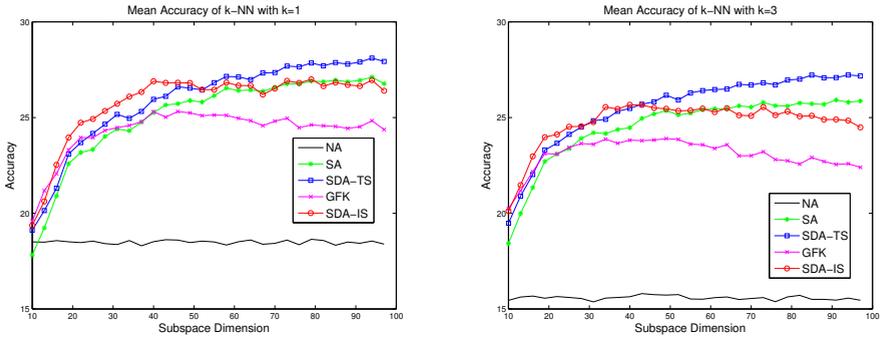


Figure 5: Mean accuracy across all **6** experiment settings (domain shifts) of the k -NN Classifier on the **Office31** dataset. Both our methods SDA-IS and SDA-TS outperform GFK and SA consistently. The x-axis shows the dimensionality of the subspace. For NA, as it uses the original feature space without subspace projection, we just average over 20 randomized trials at each subspace dimension. **Left:** k -NN Classifier with $k=1$; **Right:** k -NN Classifier with $k=3$. (Best viewed in color)

5 Conclusion

In this paper, we proposed a new method for unsupervised domain adaptation. Our approach incorporates distribution alignment into subspace adaptation. Two recent subspace adaptation methods, Geodesic Flow Kernel [9] and Subspace Alignment [7], can be obtained as special cases of our approach. By taking into account the difference of the two distributions in the source and target subspaces, our method further reduces data mismatch, leading to improved results. Extensive experimental results on standard benchmarks demonstrate the advantage of our approach over published methods.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by NSF Awards IIS-1451244 and IIS-1212928.

References

- [1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011.
- [2] S. Ben-david, J. Blitzer, K. Crammer, and O. Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.
- [3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.
- [4] H. Daume III. Frustratingly easy domain adaptation. In *ACL*, 2007.
- [5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

- [6] Lixin Duan, Ivor W. Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, 2009.
- [7] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.
- [8] Daniel Goehring, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Interactive adaptation of real-time object detectors. In *ICRA*, 2014.
- [9] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.
- [10] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.
- [11] Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*. 2012.
- [12] J. Hoffman, K. Saenko, B. Kulis, and T. Darrell. Discovering latent domains for multi-source domain adaptation. In *ECCV*, 2012.
- [13] J. Jiang. A literature survey on domain adaptation of statistical classifiers. http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/.
- [14] J. Jiang and C. X. Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, 2007.
- [15] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.
- [16] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.
- [17] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*. 2010.
- [18] Ming Shao, Dmitry Kit, and Yun Fu. Generalized transfer subspace learning through low-rank constraint. *IJCV*, 2014.
- [19] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *BMVC*, 2014.
- [20] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.
- [21] J. Yang, R. Yan, and A. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *ICDM Workshops*, 2007.