

Domain Adaptation by Aligning Locality Preserving Subspaces

Viresh Ranjan¹ Gaurav Harit² C. V. Jawahar¹

¹CVIT, IIT Hyderabad, India ²IIT Jodhpur, India

Abstract—The mismatch between the training data and the test data distributions is a challenging issue while designing many practical computer vision systems. In this paper, we propose an unsupervised domain adaptation technique to tackle this issue. We are interested in a domain adaptation scenario where source domain has large amount of labeled examples and the target domain has large amount of unlabeled examples. We align the source domain subspace with the target domain subspace in order to reduce the mismatch between the two distributions. We model the subspace using Locality Preserving Projections (LPP). Unlike previous subspace alignment approaches, we introduce a strategy to effectively utilize the training labels in order to learn discriminative subspaces. We validate our domain adaptation approach by testing it on two different domains, i.e. handwritten and printed digit images. We compare our approach with other existing approaches and show the superiority of our method.

I. INTRODUCTION

Dataset shift is a scenario when the training set and the test set do not follow the same underlying distribution [1]. It is a serious concern while designing computer vision algorithms for real world applications. For example, an OCR system trained on a few fonts might perform badly on a novel test font if the distribution of the characters in the test font is very different from that of the training fonts. The problem is far more challenging when one is interested in adapting a classifier trained on a printed data set to a handwritten character data set like MNIST [2]. In this paper, we address this specific domain adaptation problem.

We first conduct a toy experiment to motivate how the performance of the simple k-nearest neighbor based classifier degrades in the presence of dataset shift. For this experiment, we consider the task of classifying digit images in presence of dataset shift. The two domains we consider for the experiment are handwritten and printed digits. In order to classify test images from the target domain, we determine the nearest neighbors from the labeled source domain images based on their Euclidean distance to the test image. The test image is then assigned the majority label of the k nearest source domain images. Few of the test images and their corresponding nearest neighbors has been shown in Figure 1. We observe that for the majority of cases, the source domain samples would misclassify the target domain test image. Hence, in presence of dataset shift, a source domain classifier might perform badly on the target domain.

Machine learning algorithms which are designed with mechanisms for tackling the dataset shift are called Domain Adaptation (DA) algorithms. In most of the DA algorithms, following scenario is assumed:

Test Image(hw)	Nearest Neighbors(p)				Test Image(p)	Nearest Neighbors(hw)			

Fig. 1. Test images from target domain and the corresponding nearest neighbors from the source domain are shown. Acronym hw stands for handwritten domain and p stands for printed domain. The source and target domains are p and hw for the results on the left; hw and p for the results on the right. Nearest neighbor classifier performs poorly at both the cross domain classification tasks.

- 1) The training dataset (source domain) has plenty of labeled examples.
- 2) The learned model has to be tested on a test dataset (target domain) which may have a different distribution. Sufficient amount of unlabeled target domain data is available, apart from this few labeled examples from the target domain may be available while learning the model.

DA algorithms can be broadly categorized into two types: classifier adaptation based or feature adaptation based. Classifier based DA techniques such as [3] and [4] use plenty of labeled data from source domain and some labeled data from the target domain to learn a classifier which performs well on the target domain. Feature based DA approaches such as [5] and [6] try to reduce intraclass variations across the source and target domains. The feature based approaches can be further divided into two categories, semi-supervised or unsupervised, depending on whether few labeled examples from target domain are available or not. A review of different DA approaches for statistical classifiers can be found in [7].

Subspace based DA techniques such as [8], [9] and [10] are becoming a popular means of doing unsupervised DA. Recently

a subspace alignment based unsupervised DA approach has been presented in [11]. The central idea of their work is to align the source and target subspaces and then project all the data points to their respective aligned subspace before the classification. They model the source subspace by the eigenvectors obtained by doing PCA over the source domain and similarly for the target subspace. They align the source subspace with the target subspace by learning a transformation matrix. In their approach, however, label information present in the source domain is not being utilized while learning the subspaces. Also, PCA aims at maximizing the variance of the projected data but does nothing to preserve the local neighborhood inherent in the original space. Keeping in mind these two facts, using PCA for modeling subspaces might lead to less discriminative subspaces.

Our current work is an unsupervised subspace alignment based DA approach, similar to [11]. We use Locality Preserving Projections (LPP) described by Niyogi and He [12] for modeling the subspaces. LPP builds an adjacency graph using neighborhood information from the data set. Once the adjacency graph is formed, LPP finds those projection directions which keep the connected points in the graph as close as possible. This technique preserves the local neighborhood information present in the original space. Note that while forming the adjacency graph, LPP uses closeness of points based on Euclidean distance. Hence, it does not utilize any label information while finding the projection directions. To effectively use the labels to obtain a discriminative subspace, we use a supervised version of the LPP for learning the source subspace. As labeled examples are not available in the target domain, we use the original version of the LPP for learning the target subspace. Once the source and the target subspaces are obtained, we align the two subspaces by learning a transformation. The data points are then projected to their respective subspace before doing classification. Following are the highlights of our subspace alignment based DA approach:

- 1) We use label information from the source domain while learning the subspaces. This results in basis vectors which are discriminative in nature and hence more suitable for the classification task.
- 2) We use LPP for modeling the subspaces. This preserves the local neighborhood of the data points from the original space to the projected subspace.
- 3) The subspaces can be learned directly by solving a generalized eigenvalue problem.
- 4) We introduce a dataset comprising of two domains for validating our DA approach. The dataset has sufficient number of examples for each category.

For validating our DA algorithm, we pick the task of classifying digit images in the presence of dataset shift. The two different domains we use in our experiments are printed digits and handwritten digits. Handwritten digits are obtained by randomly sampling a subset of images from the MNIST database [2]. For the printed domain, we create a dataset of printed digits consisting of 300 fonts. Now, our DA problem can be stated as: Given labeled digit images from one domain and unlabeled digit images from another domain, classify the unlabeled images.

II. RELATED WORK

A lot of work has been done in the natural language processing community regarding DA for various tasks such as sentiment classification, statistical machine translation, parts of speech tagging etc [13], [14], [15], [16], [17]. DA is also relevant for a number of Computer Vision tasks. There has been a lot of interest in DA techniques after the seminal work of Saenko *et al.* [5] in the area of object recognition. The authors present a supervised DA approach where corresponding image pairs across the two domains are used to learn a transformation to map points from the source domain closer to the same category points in the target domain. Kulis *et al.* [6] describe another supervised approach based on similar ideas. Gopalan *et al.* [8] present an unsupervised DA approach where source and target subspaces are points on a Grassmann manifold. They sample points along the geodesic between the source subspace and the target subspace to obtain intermediate subspaces. The data points are projected along all the intermediate subspaces to obtain a domain independent representation. Gong *et al.* [9] propose a geodesic flow kernel based approach and instead of sampling finite number of subspaces along the geodesic from source subspace to target subspace, they integrate over infinite number of intermediate subspaces. Ni *et al.* [10] present an unsupervised DA approach based on sparse coding where several dictionaries are learned starting from a dictionary learned from the source domain. Here each of the dictionary represents an intermediate domain between the source and the target subspace. Then the data points are projected over all the intermediate subspaces using these dictionaries. Qiu *et al.* [18] learn aligned dictionaries from multiple domains. This is a supervised approach as they use correspondence information across domains. Jhuo *et al.* [19] present a low rank reconstruction based DA strategy where source data points are transformed to an intermediate domain where they can be represented as a linear combination of the target domain data points. The intermediate representation is then used to transform the source domain data points to the target domain data points. In [20], [21], the difference between the source and target distributions is reduced by learning a latent feature representation. Yang *et al.* [3] learn a SVM classifier on source domain and adapt it for the target domain using some labeled data from the target domain.

III. DOMAIN ADAPTATION USING ALIGNMENT OF LOCALITY PRESERVING SUBSPACES

To obtain source and target domain subspaces, we use Locality Preserving Projection (LPP) [12]. In its original form, LPP does not use any label information. Hence to utilize the label information present in the source domain to obtain a discriminative subspace, we use a supervised version of the LPP. We describe LPP and a supervised version of LPP in Section III-A. This supervised version of LPP has been proposed in [22]. We refer to the supervised LPP as sLPP. Once the source domain and target domain subspaces are obtained using sLPP and LPP respectively, we align these two subspaces by learning a transformation matrix. The alignment technique has been discussed in Section III-B. In Section III-C, we describe our DA approach.

A. Locality Preserving Subspaces

Given a dataset with m vectors x_1, x_2, \dots, x_m in R^n and their corresponding labels y_1, y_2, \dots, y_m , LPP finds a set of basis vectors A (each column of A is a basis vector) so that the neighborhood of each of the m points is preserved after the transformation $z_i = A^T x_i$. Note that LPP does not use the labels while finding the basis vectors. To obtain the transformation matrix A , first an adjacency graph $G = (V, E)$ with m nodes is formed. Nodes i and j of G are connected by an edge if the vectors x_i and x_j are close to one another. Here, x_i and x_j are considered to be close based on either of these two conditions :

- $\|x_i - x_j\|^2 < \epsilon$ where $\epsilon \in R$.
- x_i and x_j are among the k nearest neighbors of one another.

The edge strength W_{ij} between connected nodes i and j can be defined to be either $e^{-\frac{\|x_i - x_j\|^2}{t}}$ or simply 1, here $t \in R$. W_{ij} is assigned a value of 0 if the nodes i and j are not connected. The columns a of the matrix A can be found by solving the following generalized eigenvalue problem:

$$XLX^T a = \lambda DX^T a \quad (1)$$

where i^{th} column of X is x_i , D is a diagonal matrix such that $D_{ii} = \sum_j W_{ij}$ and $L = D - W$ is the Laplacian matrix.

Solutions of this equation are the eigen vectors that form the columns of the transformation matrix A .

a) Supervised LPP: Clearly LPP in its original form does not use any label information. Hence, if two vectors x_i and x_j belonging to different classes are close in original space R^n , their closeness would be preserved after the transformation also. Such scenarios would clearly have a negative impact on classification in the transformed space. To tackle this issue, we also consider the label of points x_i and x_j while forming the adjacency graph G . Hence the label aware closeness conditions can be given as:

- $\|x_i - x_j\|^2 < \epsilon$ and $y_i = y_j$; where $\epsilon \in R$.
- x_i and x_j are among the k nearest neighbors of one another and $y_i = y_j$.

here y_i and y_j are labels of x_i and x_j respectively. The remaining steps of sLPP are same as that of LPP. Clearly, sLPP would only preserve the intra class neighborhoods.

B. Aligning subspaces

Assume that both the source domain and the target domain data points lie in R^n . The m_s data points from source domain are arranged as column vectors of the $n \times m_s$ matrix X_s and similarly the m_t data points from the target domain are arranged in the $n \times m_t$ matrix X_t . The m_s dimensional column vector Y_s contains the labels of each of the source domain examples. Also, assume that the subspaces corresponding to the source domain and the target domain are known and each of the subspaces are represented using k basis vectors. Let the source subspace be represented by the $n \times k$ matrix A_s whose columns are the source domain basis vectors obtained by solving the generalized eigenvalue problem given in Equation 1.

Algorithm 1 Locality Preserving Subspace Alignment(LPSA)

Input: source vectors X_s , source labels Y_s , target vectors X_t , constant β
Output: transformed vectors Z_s, Z_t
 $A_s \leftarrow \text{sLPP}(X_s)$
 $A_t \leftarrow \text{LPP}(X_t)$
 $M \leftarrow A_t A_s^T (A_s A_s^T + \beta I)^{-1}$
 $Z_s^T \leftarrow A_s^T M^T X_s$
 $Z_t^T \leftarrow A_t^T X_t$

Similarly, the target subspace can be represented by the $n \times k$ matrix A_t whose columns are the target domain basis vectors. We want to find a transformation which aligns A_s with A_t . We model the transformation using a $n \times n$ matrix M . To obtain M , we minimize the following objective:

$$\|MA_s - A_t\|_F^2 + \beta \|M\|_F^2 \quad (2)$$

where the first term tries to align the two subspaces, the second term is a regularizer and β is a constant. Solution to this equation can be obtained in closed form as:

$$M = A_t A_s^T (A_s A_s^T + \beta I)^{-1} \quad (3)$$

here I is an identity matrix. MA_s is the transformed source domain subspace which is aligned with the target domain subspace.

C. DA by aligning subspaces

In [11], an unsupervised domain adaptation technique is presented where the source and the target subspaces are aligned and the samples are then projected to their respective subspaces. In unsupervised scenario for domain adaptation, we have plenty of labeled data available in the source domain whereas only unlabeled data is available in the target domain. However, such an approach does not utilize any label information present in the source domain. Our unsupervised domain adaptation method, described below, uses labeled data from the source domain as well as unlabeled data from the target domain for learning the source and target subspaces respectively. The authors used eigenvectors induced by doing a PCA as the basis vectors of the subspaces. Although the eigenvectors obtained by PCA maximizes the overall variance of the data, they do not preserve the local neighborhood of the data points. Hence we use LPP for obtaining the source and target subspaces.

b) DA by aligning LPP subspaces: The goal of our DA approach is to use such subspaces where neighborhood of data points in the original space is preserved in the transformed space and also to utilize the label information present in the source domain while learning the source subspace. We describe our algorithm for doing these in Algorithm 1. Let X_s be the $n \times m_s$ matrix containing the source domain examples, where n is the dimension of each example and there are m_s such examples. Let Y_s be a m_s dimensional column vector containing the labels of the source examples. Also, X_t contains the target domain examples. The Algorithm 1 takes as input the source domain points X_s , target domain data points X_t and the labels of source domain data points Y_s and outputs the data vectors in the respective aligned subspaces, i.e. Z_s and Z_t . In order to utilize the source labels, the algorithm uses sLPP to learn the source subspace A_s . Target subspace A_t is learned

by LPP. Once A_s and A_t are obtained, the two subspaces are aligned using the technique mentioned in Section III-B. The source and target domain data points are now projected over the respective aligned subspaces represented by MA_s and A_t respectively as $Z_s^T = (MA_s)^T X_s$ and $Z_t^T = A_t^T X_t$.

D. Discussion

Most of the subspace based domain adaptation techniques, for example [8], [9], [11] and [10] are unsupervised in nature. A majority of these techniques ([8], [9], [10]) share a common theme wherein they try to obtain the representation of the data points across the intermediate subspaces between the source and the target subspace. This helps in obtaining a domain invariant representation of the data points. The work of [11] is different from these approaches as they do not obtain the intermediate representations of data points, but rather align the source and target domain subspaces and subsequently each data point is projected to a single subspace. All these approaches do not utilize the label information present in the source domain. Hence the subspaces over which they project the data points may not be discriminative enough for the classification task. Our approach, however, utilizes the source domain labels and finds such a source subspace which preserves the intra-class neighborhoods. Hence the source subspace in our approach is discriminative in nature. We find the target subspace and align it with the source subspace. Our approach preserves the geometry of data points from both the source and the target domains and also utilizes the label information from the target obtain to obtain discriminative subspaces which are suitable for classification.

IV. DATASET AND EXPERIMENTS

In this section, we give details about the datasets used for the experiments and the features used for representing the images. We also validate our domain adaptation technique by doing nearest neighbor based classification experiments and compare our approach with related approaches.

A. Dataset and Representation

For our experiments, we use digit images (0 – 9) from two domains, i.e. printed and handwritten. Handwritten digits are obtained by randomly sampling 300 images of each of the digits from the MNIST database. These images are equally subdivided into three sets, i.e. Train, Test and Validation set. All the images are binarized using the thresholding technique given in [23]. For printed digits, we obtained 300 different fonts from the internet and generated binary images of the digits in each of the fonts. To keep the image size same across both the domains, all synthetic images were generated in the same size as the images in the MNIST database, i.e. 28×28 . Again, the synthetic dataset was equally subdivided into Train, Test and Validation sets.

For image representation, we use histogram of oriented gradient features (HOG) described by Dalal *et al.* [24]. The reason for using gradient features instead of raw pixel representation is that these features have been shown to give better results in handwritten digits classification task [25]. HOG features are obtained by dividing an image into square cells and computing a histogram of edge orientations for the pixels within each

TABLE I. TABLE SHOWS CLASSIFICATION ACCURACIES FOR CROSS DOMAIN EXPERIMENTS. DECREASE IN PERFORMANCE WHEN THE SOURCE AND THE TARGET DOMAINS ARE DIFFERENT (NON-DIAGONAL ENTRIES IN THE TABLE) IS CLEARLY VISIBLE FROM THE TABLE.

	Handwritten	Printed
Handwritten	89.0	48.8
Printed	70.0	87.1

TABLE II. CLASSIFICATION ACCURACIES FOR DA EXPERIMENTS ARE SHOWN. HERE, HW IS ACRONYM FOR HANDWRITTEN DOMAIN, NA REFERS TO NO ADAPTATION SCENARIO, PCA (SOURCE) REFERS TO PROJECTING ALL THE SAMPLES TO THE SOURCE DOMAIN PCA SUBSPACE, SIMILARLY PCA (TARGET) REFERS TO PROJECTING ALL THE SAMPLES TO TARGET DOMAIN PCA SUBSPACE. PCA COMBINED REFERS TO PROJECTING TO THE COMBINED SOURCE AND TARGET PCA SUBSPACE. THE TABLE SHOWS RESULTS RESULTS FOR THE CASE WHEN THE SOURCE AND TARGET DOMAINS ARE HANDWRITTEN AND PRINTED RESPECTIVELY.

Source	Target	Method	Accuracy
HW	Printed	NA	48.8
HW	Printed	PCA (source)	55.9
HW	Printed	PCA (target)	56.5
HW	Printed	PCA (combined)	56.5
HW	Printed	[11]	57.0
HW	Printed	Ours	64.8

cell. We used the cell size 8 for our experiment. The HOG feature of each cell was concatenated to obtain a column vector representation for each image. All the feature vectors were normalized to have zero mean and unit variance.

B. Experiments

As described in Section IV-A, data from both the domains has been divided equally into Train, Test and Validation sets. For all the subsequent experiments, Train set examples are used as labeled source domain examples and Test set examples are used as unlabeled target domain examples. The optimal values for subspace dimension and β are learned by doing classification on the validation set. We conduct a cross domain classification experiment to observe how the classification accuracy of a nearest neighbor based classifier decreases in presence of dataset shift. In this experiment, we classify target domain samples using labeled examples from the source domain. We present the results in Table I. We observe that accuracy is high when training and test set belong to same domain. Also, the classification accuracy decreases when training and test sets are from different domains. In Table II, we compare our subspace based DA approach with existing techniques for the cross domain classification task. For the no adaptation case, we directly classify the target domain test points using labeled examples from the source domain.

We also show results for the trivial projection cases when samples from both of the domains are projected to a common subspace. We consider three common subspaces, source PCA subspace obtained by doing PCA for the source domain data points, target PCA subspace obtained by doing PCA of target

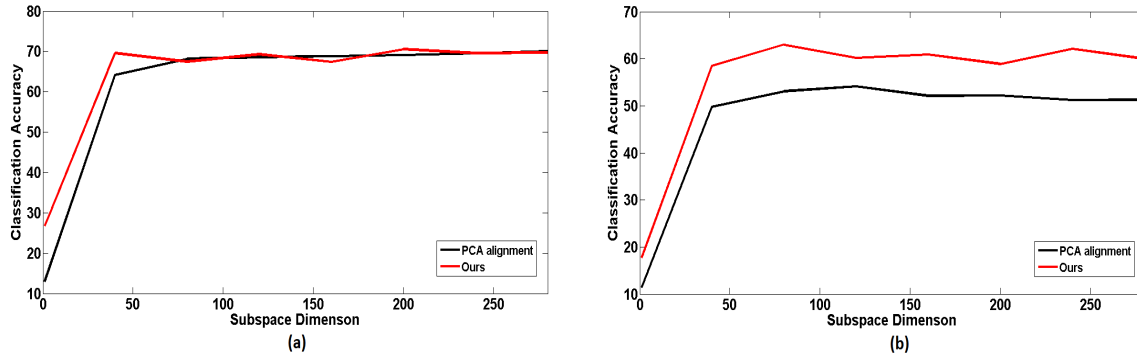


Fig. 2. The plots for classification accuracy as a function of subspace dimension has been shown. Here, we compare our subspace alignment approach with the PCA subspace alignment approach proposed by [11]. Plot (a) corresponds to the case when source domain is Printed, target domain is handwritten. For Plot (b), source domain is handwritten and target domain is printed.

TABLE III. CLASSIFICATION ACCURACIES FOR DA EXPERIMENTS ARE SHOWN. HERE, HW IS ACRONYM FOR HANDWRITTEN DOMAIN, NA REFERS TO NO ADAPTATION SCENARIO, PCA (SOURCE) REFERS TO PROJECTING ALL THE SAMPLES TO THE SOURCE DOMAIN PCA SUBSPACE, SIMILARLY PCA (TARGET) REFERS TO PROJECTING ALL THE SAMPLES TO TARGET DOMAIN PCA SUBSPACE. PCA COMBINED REFERS TO PROJECTING TO THE COMBINED SOURCE AND TARGET PCA SUBSPACE. THE TABLE SHOWS RESULTS RESULTS FOR THE CASE WHEN THE SOURCE AND TARGET DOMAINS ARE PRINTED AND HANDWRITTEN RESPECTIVELY.

Source	Target	Method	Accuracy
Printed	HW	NA	70.0
Printed	HW	PCA (source)	68.1
Printed	HW	PCA (target)	68.9
Printed	HW	PCA (combined)	70.2
Printed	HW	[11]	70.6
Printed	HW	Ours	73.2

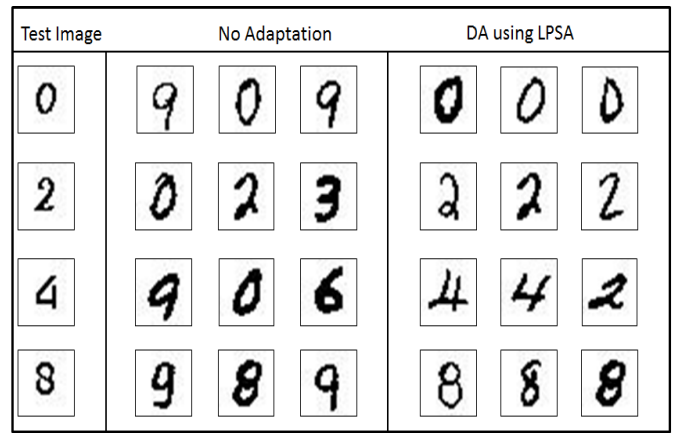


Fig. 4. Qualitative comparison of nearest neighbor classifier for No Adaptation case with our DA approach. Here LPSA is acronym for locality preserving subspace alignment. The source and target domains are handwritten and printed respectively. The figure shows test image from target domain and corresponding nearest neighbors from source domain.

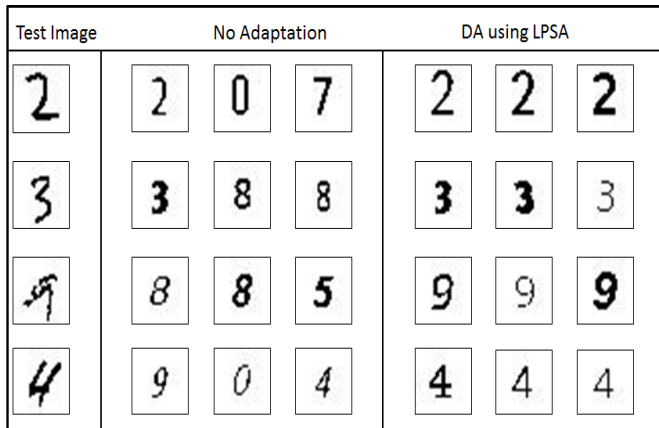


Fig. 3. Qualitative comparison of nearest neighbor classifier for No Adaptation case with our DA approach. Here LPSA is acronym for locality preserving subspace alignment. The source and target domains are printed and handwritten respectively. The figure shows test image from target domain and corresponding nearest neighbors from source domain.

data points and combined PCA subspace obtained by doing PCA of samples from both the domains. We also compare our approach with the recent subspace alignment approach of [11]. We observe that our approach outperforms all the other approaches. We also observe that improvement because of our DA approach is significantly better in the case when the source and target domains are handwritten and printed respectively.

In Figure 2, we compare the performance of our subspace alignment approach with the approach of [11] as the subspace dimension is increased. We observe that for lower dimensional subspaces, both the approaches perform badly. For higher dimensional subspaces (around 50 and above), performance of both the approaches improve significantly. We observe that for the case when the source domain is handwritten and the target domain is printed, our method consistently outperforms [11] by a good margin. In the other case, when the source and target domains are printed and handwritten respectively, although our method outperforms [11], the difference between the two is not as prominent as the previous case. In Figure 3, we compare the qualitative results for no adaptation case with our subspace alignment approach for the cross domain nearest neighbor based classification task. In this figure, the source and target domains are printed and handwritten respectively. For the experiment, a test image is picked from the target domain and samples from source domain are sorted based on their distance to the test image. We can clearly observe the improvement in the results because of our approach. Although the source and target domain samples look visually very different from one another, our subspace alignment algorithm transforms the

samples so that the intra-class variations across the domains is minimized. In Figure 4, we repeat the previous experiment taking the source and target domains as handwritten and printed respectively. Here also, our approach performs much better than the no adaptation case.

V. CONCLUSION

We presented an unsupervised DA strategy for classification in the presence of dataset shift. We have shown the application of our strategy for the task of digits classification, however, the approach is general and can be used for other tasks. Our approach not only learns subspaces by utilizing the source domain labels but also preserves the local neighborhood of data points. Hence the subspaces are discriminative in nature. We show the superiority of our approach over other existing approaches by showing significant improvement in classification over the other methods.

REFERENCES

- [1] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.
- [2] Y. LeCun and C. Cortes, “The MNIST database of handwritten digits,” 1998.
- [3] J. Yang, R. Yan, and A. G. Hauptmann, “Cross-domain video concept detection using adaptive svms,” in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 188–197.
- [4] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, “Domain transfer svm for video concept detection,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2009, pp. 1375–1381.
- [5] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *ECCV*. Springer, 2010, pp. 213–226.
- [6] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2011, pp. 1785–1792.
- [7] J. Jiang, “A literature survey on domain adaptation of statistical classifiers,” URL: <http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey>, 2008.
- [8] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in *Computer Vision (ICCV), IEEE International Conference on*. IEEE, 2011, pp. 999–1006.
- [9] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, pp. 2066–2073.
- [10] J. Ni, Q. Qiu, and R. Chellappa, “Subspace interpolation via dictionary learning for unsupervised domain adaptation,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2013, pp. 692–699.
- [11] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars *et al.*, “Unsupervised visual domain adaptation using subspace alignment,” *Proceedings of ICCV*, 2013.
- [12] X. He and P. Niyogi, “Locality preserving projections,” in *NIPS*, 2003, pp. 234–241.
- [13] G. Foster, C. Goutte, and R. Kuhn, “Discriminative instance weighting for domain adaptation in statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 451–459.
- [14] J. Jiang and C. Zhai, “Instance weighting for domain adaptation in NLP,” in *ACL*, 2007, p. 22.
- [15] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2006, pp. 120–128.
- [16] H. Daumé III, “Frustratingly easy domain adaptation,” in *ACL*, 2007.
- [17] X. Glorot, A. Bordes, and Y. Bengio, “Domain adaptation for large-scale sentiment classification: A deep learning approach,” in *Proceedings of the 28th International Conference on Machine Learning*, pp. 513–520.
- [18] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, “Domain adaptive dictionary learning,” in *ECCV*. Springer, 2012, pp. 631–645.
- [19] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, “Robust visual domain adaptation with low-rank reconstruction,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2012, pp. 2168–2175.
- [20] S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,” in *AAAI*, 2008, pp. 677–682.
- [21] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *Neural Networks, IEEE Transactions on*, pp. 199–210, 2011.
- [22] D. Cai, X. He, J. Han, and H.-J. Zhang, “Orthogonal laplacianfaces for face recognition,” *Image Processing, IEEE Transactions on*, pp. 3608–3614, 2006.
- [23] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, pp. 23–27, 1975.
- [24] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition. CVPR. IEEE Computer Society Conference on*. IEEE, 2005, pp. 886–893.
- [25] S. Maji and J. Malik, “Fast and accurate digit classification,” *EECS Department, University of California, Berkeley, Tech. Rep.*, 2009.