Mind the Gap: Subspace based Hierarchical Domain Adaptation

Anant Raj
Department of Electrical Engineering
IIT Kanpur
Kanpur, India 208016
anantraj@iitk.ac.in

Vinay P. Namboodiri
Department of Computer Science and Engineering
IIT Kanpur
Kanpur, India 208016
vinaypn@iitk.ac.in

Tinne Tuytelaars
PSI-VISICS/ESAT
K.U. Leuven
Heverlee, Belgium 3001
tinne.tuytelaars@esat.kuleuven.be

Abstract

Domain adaptation techniques aim at adapting a classifier learnt on a source domain to work on the target domain. Exploiting the subspaces spanned by features of the source and target domains respectively is one approach that has been investigated towards solving this problem. These techniques normally assume the existence of a single subspace for the entire source / target domain. In this work, we consider the hierarchical organization of the data and consider multiple subspaces for the source and target domain based on the hierarchy. We evaluate different subspace based domain adaptation techniques under this setting and observe that using different subspaces based on the hierarchy yields consistent improvement over a non-hierarchical baseline.

1 Introduction

While evaluating unseen test instances on a classifier trained over a set of labelled training instances, there is a standard assumption that test instances and training instances follow the same distribution. However, many real world scenarios violate this assumption. Think of a case where someone wants to classify the images taken with his low quality phone camera for which he doesn't have labels available. Can the person classify those images using the classifier which was trained on some publicly available dataset like ImageNet or Flickr ? The obvious answer is no. Many studies have shown that if the test instances are not sampled from the same distribution as the training instances then the performance of the classifier significantly diminishes [4, 14, 15]. This problem of domain shift is also extensively studied in the field of natural language processing and speech processing [6, 7]. To address this challenge, methods have been suggested to adapt a domain *(Source Domain)* with respect to the other domain *(Target Domain)* so that a classifier trained on *Source Domain* data also contains the property of *Target Domain* data. One can distinguish two settings in the domain adaptation literature: (1) *the unsupervised setting* when the target domain is completely unlabeled and (2) *the semi-supervised setting* when the target domain is partially labeled. In both settings, the source domain is fully labelled. In this work we focus on the unsupervised setting that is more challenging one. A promising line of work to solve this problem is by subspace based domain adaptation [1, 2, 3]. However, none of the above approaches takes the semantic (dis)similarity of the category classes into account. Classes which are semantically similar have a very different distribution than classes which are semantically different. Based on this observation, we advocate that it's better to align the subspaces separately rather than considering the whole target data distribution at once.

To address this challenge we propose a new method of *step-wise subspace alignment* for domain adaptation. Step-wise subspace alignment here indicates that we first align the subspaces for a set of larger clusters or group of semantically similar categories and then for the categories within the clusters.

We evaluate the effectiveness of proposed approach on a standard dataset having classes arranged according to their semantics in the hierarchy. However, the proposed approach can also be effective for the case when the hierarchy is not available. In such scenario similar categories can be clustered together in unsupervised way.

2 Related Work

As mentioned in section 1 domain adaptation is widely studied in many fields including Natural Language Processing, Speech Processing and Computer Vision [4, 6, 9]. A survey on recent advances in domain adaptation in natural language processing and computer vision can be found in [5, 8, 13]. Subspace based approaches are most popular for solving the visual domain shift problem [1, 2, 3]. The same principal lies behind these approaches. They first determine separate subspaces for source and target data and then project the data onto these subspaces and/or a set of intermediate sampled subspaces with the aim of making the feature point domain invariant. In [3], a method is proposed to sample subspaces along the geodesic between source and target subspace on the Grassmann manifold. Once sampling is done then features are projected onto those sampled subspaces and a classifier is trained on the projected features. In [2], the geodesic flow kernel is proposed to capture the incremental details in subspaces between source and target subspace along the geodesic. Instead of using intermediate subspaces, [1] proposes to learn a transformation to directly align the source subspace to the target subspace.

Only few works have looked at the use of hierarchies in the context of domain adaptation. In [10], Nguyen *et al.* propose to adapt a hierarchy of features to exploit the richness of visual data. The intent behind this work is similar to our work, in that semantic closeness and context information are exploited to boost domain adaptation performance. Taking this idea forward a recent work on hierarchical adaptive structural SVM for domain adaptation has been proposed in [11]. They organize multiple target domains into a hierarchical structure (tree) and adapt the source model to them jointly. Others have used statistical methods for hierarchical domain adaptation, e.g. in [12] a hierarchical Bayesian prior is used to solve the domain shift problem in natural language and speech processing. However, the previous works have assumed a single common subspace between source and target, while our approach makes use of the hierarchical structure among the different classes to learn separate subspaces.

3 Background

The proposed approach builds up on the previously proposed subspace based methods [1, 2, 3]. One could learn the domain shift between source and target data on the original features itself. However this would be sub-optimal and involve significantly modifying the classifiers. Therefore, it is more common to learn it on a more robust representation of the data by first selecting d dominating eigenvectors obtained using *principal component analysis*. These d eigenvectors work as the basis vectors for the *source* and *target* subspaces. The source and target features are then projected on the subspaces. Two recently proposed state-of-the-art subspace based domain adaptation methods [1, 2] used in this paper are discussed in 3.1 and 3.2

3.1 Subspace Alignment

Subspace alignment based domain adaptation method consists of learning a transformation matrix M that maps the source subspace to the target one [1]. The mathematical formulation to this problem is given by

$$F(M) = \|X_S M - X_T\|_F^2 \qquad M^* = \operatorname*{argmin}_M (F(M)). \qquad (1)$$

X_s and X_t are matrices containing the d most important eigenvectors for source and target respectively. M is a transformation matrix from the source subspace X_S to target subspace X_T and $\|.\|_F$ is the *Frobenius norm*. The solution of *eq.* 1 is $M^* = X_S' X_T$ and hence for the target aligned source coordinate system we get $X_a = X_S X_S' X_T$.

3.2 Geodesic Flow Kernel

The geodesic flow kernel based domain adaptation method constructs an infinite-dimensional feature space that carries the information of incremental change from source to target domain data [2]. A key step in this method is to determine the geodesic curve between the two subspaces and to construct the geodesic flow kernel. If X_S and X_T are source and target subspaces having the same dimension then these two subspaces are separate points on a Grassmann manifold which is also a *Riemannian*

Manifold. Let R_S be the orthogonal complement to X_S. From the property of *Riemannian Manifold*, flow from X_S towards X_T can be calculated as:

$$\phi(t) = X_S U_1 \Gamma(t) - R_S U_2 \Sigma(t) \text{ where } X_S^T X_T = U_1 \Gamma V^T, R_S^T X_T = -U_2 \Sigma V^T.$$

Based on the decomposition of the source subspace X_S and its orthogonal complement R_S, we can obtain a geodesic flow kernel matrix G that is given by

$$\begin{bmatrix} G \end{bmatrix} = \begin{bmatrix} P_S U_1 & R_S U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & \Lambda_2 \\ \Lambda_2 & \Lambda_3 \end{bmatrix} \begin{bmatrix} U_1^T P_S^T \\ U_2^T R_S^T \end{bmatrix} \text{ where } \Lambda s \text{ are diagonal matrices that depend on}$$
the principal angles.

Once we obtain the geodesic flow matrix G, we can relate labeled samples x_i from the source subspace X_i and unlabeled samples x_j from the target subspace X_j by using the distance metric $x_i^T [G] x_j$.

4   Our Approach

In this section we describe how the methods explained in section 3 are adapted for hierarchical domain adaptation. Instead of using the same subspace throughout, we postulate that better results can be obtained by using different subspaces for different levels of the hierarchy. Indeed, the more specific subspaces spanned by instances of categories of a certain branch of our tree (corresponding to similar categories), can be expected to better fit the data and therefore better model the domain shift. For the source domain, these subspaces can easily be obtained. For the target domain, however, no class labels are available as we are working in the unsupervised setting. Therefore, the exact subspaces cannot be computed. We circumvent this problem by first predicting the parent class label for each instance, using the global subspaces and applying domain adaptation at the level of the root node. We then use these predicted parent class labels to compute the next level of subspaces. This results in a two step approach, as summarized in the algorithm below. In hierarchical subspace

---

Algorithm 1 Subspace Based Hierarchical Domain Adaptation

1: **procedure** HIERARCHICAL DOMAIN ADAPTATION(Source Data S,Target Data T)
2:      $X_{S_{root}} \leftarrow PCA(S)$ and $X_{T_{root}} \leftarrow PCA(T)$
3:      $SubspaceAlign(X_{S_{root}}, X_{T_{root}})$ or $GFK(X_{S_{root}}, X_{T_{root}})$
4:      $ClassifyParent$(Target Data T)
5:      $i \leftarrow 0$
6:      **while** $i < No. \ of \ Parents$ **do**
7:          $X_{S_i} \leftarrow PCA(S_i)$ $\triangleright S_i$s are labeled data points (from i^{th} parent)
8:          $X_{T_i} \leftarrow PCA(T_i)$ $\triangleright T_i$s are data points classified as i^{th} parent by the root classifier
9:          $SubspaceAlign(X_{S_i}, X_{T_i})$ or $GFK(X_{S_i}, X_{T_i})$
10:          $ClassifyChild$(Target Data T_i)
11:      **return** $Accuracy$

---

alignment we learn different *metric M* at different levels of hierarchy independently. Without loss of generality, we consider here hierarchies with only two levels, i.e. composed of a root node, a set of parent nodes (each corresponding to a set of similar categories) and a set of child nodes or leaf nodes (corresponding to the different categories). Hence the mathematical formulation of our approach is governed by *eq.* 2 and 3.

$$F(M_{root}) = \|X_{S_{root}} M_{root} - X_{T_{root}}\|_F^2 \qquad M_{root}^* = \underset{M_{root}}{\operatorname{argmin}}(F(M_{root})) \qquad (2)$$

$$\forall i \in parent, F(M_i) = \|X_{S_i} M_i - X_{T_i}\|_F^2 \qquad M_i^* = \underset{M_i}{\operatorname{argmin}}(F(M_i)) \qquad (3)$$

Here M_{root}^* is the transformation matrix learned at the topmost level of hierarchy to differentiate between the parents. Each parent category consists of several similar child categories. $X_{S_{root}}$ and $X_{T_{root}}$ are the source and target subspaces considering all the source and target data. M_i^* is the transformation matrix learned at the second level of hierarchy to distinguish between the children of parent i. X_{S_i} and X_{T_i} are source and target subspaces for categories that belong to parent i. Hence X_{S_i} and X_{T_i} are obtained using only the data points that belong to a child category of parent i. Solutions of the *eq.* 2 and 3 are similar to *eq.* 1.

In hierarchical geodesic flow kernel we compute different kernel matrices at different levels of hierarchy for categorization at a specific level. For classifying between the parent classes, kernel matrix G_{root} is computed considering the source and target subspaces generated by all the data. For classifying between the children of a specific parent category i we compute kernel matrix G_i considering the subspace obtained from the children classes of parent i. Similarity between two data points depends on the hierarchy level at which prediction is performed.
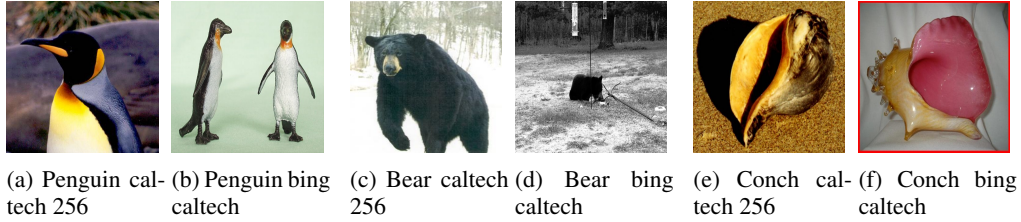
| (a) Penguin cal-tech 256 | (b) Penguin bing caltech | (c) Bear caltech 256 | (d) Bear bing caltech | (e) Conch cal-tech 256 | (f) Conch bing caltech |

Figure 1: Image samples taken from caltech-256 and bing caltech to show domain shift

5 Experiments

In this section we evaluate our results on a part of hierarchy taken from the caltech-256 and bing-caltech [17, 18]. We show our experimental results on the animal hierarchy consisting of the following three parent nodes: aquatic, terrestrial and avian animals and each parent consists of several child categories. For each image in the dataset we compute $4096-$ dimensional convolutional neural network based features obtained using Decaf [16] by first resizing the full image to the desired input size. Note that the dataset has not been augmented with any virtual examples by flipping or random cropping. In this paper we have used K-NN as our classifier as this has also been similarly used in [1, 2]. The rank of the domain is decided using the procedure given in [2] and based on this procedure we fix the dimensionality of the subspaces for both root and parent subspaces as 53. We first show the result without applying any domain adaptation algorithm on the source and target data to show that there exists a non-negligible domain shift between these two datasets. This is shown in table 1 in column "Base Accuracy". The results in table 1 show that the hierarchy based subspace alignment consistently improves the results. We also evaluate the similarity between subspaces of source and target domain by taking dot product ($trace(A' * B)$) at various levels of hierarchy to analyse our approach. This result is provided in table 2. As can be seen from the table 2, the maximum similarity is observed to be between the relevant subspaces in source and target. The low values we obtain off the main diagonal indicate the subspaces for the different parent nodes are quite different from one another and different from the root subspace.

Source Dataset	Target Dataset	Method	Base Accuracy	Accuracy (without Hierarchy)	Accuracy (with Hierarchy)
Caltech-256	Caltech-Bing	GFK	24.41	39.96	**40.41**
Caltech-Bing	Caltech-256	GFK	21.11	45.23	**49.67**
Caltech-256	Caltech-Bing	SA	24.41	39.24	**40.78**
Caltech-Bing	Caltech-256	SA	21.11	44.12	**48.36**

Table 1: Result for hierarchical domain adaptation on animal hierarchy of caltech-256 and bing-caltech. Here GFK represents Geodesic flow kernel and SA represents subspace alignment

	Root(Target)	Avian(Target)	Terrestrial(Target)	Aquatic(Target)
Root(Source)	**3.96**	1.35	0.26	2.05
Avian(Source)	3.11	**3.74**	-0.10	0.18
Terrestrial(Source)	1.44	2.62	**3.92**	0.76
Aquatic(Source)	1.22	0.43	1.42	**3.16**

Table 2: Similarity Matrix for Source Subspace and Target Subspace considering each Hierarchy level separately. Here caltech-256 is considered as source and bing-caltech as target.

6 Conclusion

In this paper, we have considered a hierarchical subspace based domain adaptation approach. Based on the evaluation we observe that considering different domain adaptation subspaces specific to the individual category level can indeed aid the domain adaptation. In future, we would like to evaluate the effect of restricting the subspaces to groups of classes which need not be obtained strictly based on hierarchy which would generalize the approach to any source and target domains that are not hierarchically labeled.

References

[1] Fernando, Basura, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. "Unsupervised visual domain adaptation using subspace alignment." In Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 2960-2967. IEEE, 2013

[2] Gong, Boqing, Yuan Shi, Fei Sha, and Kristen Grauman. "Geodesic flow kernel for unsupervised domain adaptation." In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2066-2073. IEEE, 2012.

[3] Gopalan, Raghuraman, Ruonan Li, and Rama Chellappa. "Domain adaptation for object recognition: An unsupervised approach." In Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 999-1006. IEEE, 2011.

[4] Torralba, Antonio, and Alexei A. Efros. "Unbiased look at dataset bias." In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1521-1528. IEEE, 2011.

[5] Margolis, Anna. "A literature review of domain adaptation with unlabeled data." Tec. Report (2011): 1-42.

[6] Uribe, Diego. "Domain adaptation in sentiment classification." In Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on, pp. 857-860. IEEE, 2010.

[7] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach." In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 513-520. 2011.

[8] Ben-David, Shai, John Blitzer, Koby Crammer, and Fernando Pereira. "Analysis of representations for domain adaptation." Advances in neural information processing systems 19 (2007): 137.

[9] Leggetter, Christopher J., and Philip C. Woodland. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models." Computer Speech & Language 9.2 (1995): 171-185.

[10] Nguyen, Hien V., Huy Tho Ho, Vishal M. Patel, and Rama Chellappa. "Joint hierarchical domain adaptation and feature learning." IEEE Transactions on Pattern Analysis and Machine Intelligence, submitted (2013).

[11] Xu, Jiaolong, Sebastian Ramos, David Vzquez, and Antonio M. Lpez. "Hierarchical Adaptive Structural SVM for Domain Adaptation." arXiv preprint arXiv:1408.5400 (2014).

[12] Finkel, Jenny Rose, and Christopher D. Manning. "Hierarchical bayesian domain adaptation." Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009.

[13] Patel, Vishal M., Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. "Visual Domain Adaptation: An Overview of Recent Advances."

[14] Khosla, Aditya, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. "Undoing the damage of dataset bias." In Computer VisionECCV 2012, pp. 158-171. Springer Berlin Heidelberg, 2012.

[15] Perronnin, Florent, Jorge Snchez, and Yan Liu. "Large-scale image categorization with explicit data embedding." In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2297-2304. IEEE, 2010.

[16] Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition." In Proceedings of The 31st International Conference on Machine Learning, pp. 647-655. 2014.

[17] Griffin, Gregory, Alex Holub, and Pietro Perona. "Caltech-256 object category dataset." (2007).

[18] Bergamo, Alessandro, and Lorenzo Torresani. "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach." In Advances in Neural Information Processing Systems, pp. 181-189. 2010.