Domain Adaptation for Object Recognition: An Unsupervised Approach*

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa
Center for Automation Research, University of Maryland, College Park, MD 20742 USA
{raghuram,liruonan,rama}@umiacs.umd.edu

Abstract

Adapting the classifier trained on a source domain to recognize instances from a new target domain is an important problem that is receiving recent attention. In this paper, we present one of the first studies on unsupervised domain adaptation in the context of object recognition, where we have labeled data only from the source domain (and therefore do not have correspondences between object categories across domains). Motivated by incremental learning, we create intermediate representations of data between the two domains by viewing the generative subspaces (of same dimension) created from these domains as points on the Grassmann manifold, and sampling points along the geodesic between them to obtain subspaces that provide a meaningful description of the underlying domain shift. We then obtain the projections of labeled source domain data onto these subspaces, from which a discriminative classifier is learnt to classify projected data from the target domain. We discuss extensions of our approach for semi-supervised adaptation, and for cases with multiple source and target domains, and report competitive results on standard datasets.

1. Introduction

In pattern classification problems, we are often confronted with situations where the data we have to train a classifier is 'different' from that presented during testing. Of the several schools of thought addressing this problem, two prominent ones are transfer learning (TL) [32], and domain adaptation (DA) [4]. These two strategies primarily differ on the assumptions of 'what' characteristics of data are changing between the training and testing conditions. Specifically, TL addresses the problem where the marginal distribution of data in the training set X (source domain) and the test set $\tilde{X}$ (target domain) are similar, while the conditional distributions of labels, $P(Y|X)$ and $P(\tilde{Y}|\tilde{X})$ with Y and $\tilde{Y}$ denoting labels in either domain, are different. On the other hand, DA pertains to the case where $P(Y|X) \approx P(\tilde{Y}|\tilde{X})$, but $P(X)$ significantly varies from $P(\tilde{X})$. This specific scenario occurs very naturally in unconstrained object recognition settings, where the domain shift can be due to change in pose, lighting, blur, and resolution, among others.

Understanding the effects of domain change has received substantial attention from the natural language processing community over the last few years (e.g. [4, 8, 16]). Although many fundamental questions still remain on the assumptions used to quantify a domain shift, there are several methods that have demonstrated improved performance under some domain variations. Given labeled samples from the source domain, these methods can be broadly classified into two groups depending on whether the target domain data has some labels or it is completely unlabeled. The former is referred to as semi-supervised DA, while the latter is called unsupervised DA. While semi-supervised DA is generally performed by utilizing the correspondence information obtained from labeled target domain data to learn the domain shifting transformation (e.g. [16]), unsupervised DA is based on the following strategies: (i) imposing certain assumptions on the class of transformations between domains [39], or (ii) assuming the availability of certain discriminative features that are common to both domains [8, 29].

In the context of object recognition, the problem of matching source and target data under some pre-specified transformations has been extensively studied. For instance, given appropriate representation of objects such as contours or appearance information, if it is desired to perform recognition invariant to similarity transformations, one can use Fourier descriptors [43], moment-based descriptors [25] or SIFT features [27]. Whereas in a broader setting where we do not know the exact class of transformations, the problem of addressing the domain changes has not received significant attention. Some recent efforts focus on semi-supervised DA [33, 7, 26]. However, with the ever-increasing availability of image/video data from diverse devices such as a digital SLR camera or a webcam, and image collections from the internet, it is not always reasonable to

*This work was supported by a MURI grant N00014-10-1-0934 from the Office of Naval Research.

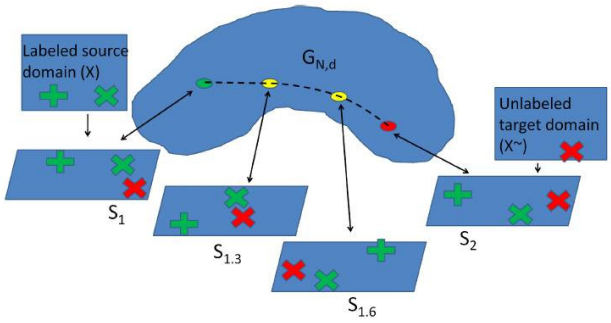


Figure 1. Say we have labeled data X from the source domain corresponding to two classes $+$ and $\times$, and unlabeled data $\tilde{X}$ from the target domain belonging to class $\times$. Instead of assuming some relevant features or transformations between the domains, we characterize the domain shift between X and $\tilde{X}$ by drawing motivation from incremental learning. By viewing the generative subspaces S_1 and S_2 of the source and target as points on a Grassmann manifold $G_{N,d}$ (green and red dots respectively), we first sample points along the geodesic between them (dashed lines) to obtain 'meaningful' intermediate subspaces (yellow dots). We then analyze projections of labeled $\times$, $+$ (green) and unlabeled $\times$ (red) onto these subspaces to perform classification. *(All figures are best viewed in color).*

assume the availability of labels in all domains. Specific example scenarios include, a robot trained on objects in indoor settings with the goal of recognizing them in outdoor unconstrained conditions, or when the user has few labeled data and lots of unlabeled data corresponding to same object categories, where one would want to generalize over all available data without requiring manual effort in labeling. Having said that, unsupervised DA is an inherently hard problem since we may not have any knowledge on how the domain change has affected the object categories.

Contributions: Instead of assuming some information on the transformation or features across domains, we propose a data-driven unsupervised approach that is primarily motivated by incremental learning. Since humans adapt (better) between extreme domains if they 'gradually' walk through the path between the domains (e.g. [34, 12]), we propose:

- Representing the generative subspaces of same dimension obtained from X and $\tilde{X}$ as points on the Grassmann manifold, and sample points along the geodesic between the two to obtain intermediate subspace representations that are consistent with the underlying geometry of the space spanned by these subspaces;
- We then utilize the information that these subspaces convey on the labeled X, and learn a discriminative classifier to predict the labels of $\tilde{X}$. Furthermore, we illustrate the capability of our method for handling multiple source and target domains, and in accommodating labeled data in the target, if any.

Organization of the paper: Section 2 reviews related work. Section 3 discusses the proposed method. Section 4 provides experimental details and comparisons with DA approaches for object recognition and natural language processing, and the paper is concluded in Section 5. Figure 1 illustrates the motivation behind our approach.

2. Related Work

One of the earliest works on semi-supervised domain adaptation was performed by Daumé III and Marcu [16] where they model the data distribution corresponding to source and target domains to consist of a common (shared) component and a component that is specific to the individual domains. This was followed by methods that combine co-training and domain adaptation using labels from either domains [36], and semi-supervised variants of the EM algorithm [14], label propagation[42] and SVM [18]. More recently, co-regularization approaches that work on augmented feature space to jointly model source and target domains [15], and transfer component analysis that projects the two domains onto the reproducing kernel Hilbert space to preserve some properties of domain-specific data distributions [31] have been proposed. Under certain assumptions characterizing the domain shift, there have also been theoretical studies on the nature of classification error across new domains [6, 4]. Along similar lines, there have been efforts focusing on domain shift issues for 2D object recognition applications. For instance, Saenko et al [33] proposed a metric learning approach that could use labeled data for few categories from the target domain to predict the domain change for unlabeled target categories. Bergamo and Torresani [7] performed an empirical analysis of several variants of SVM for this problem. Lai and Fox [26] performed object recognition from 3D point clouds by generalizing the small amount of labeled training data onto the pool of weakly labeled data obtained from the internet.

Unsupervised DA, on the other hand, is a harder problem since we do not have any labeled correspondence between the domains to estimate the transformation between them. Differing from the set of many greedy (and clustering-type) solutions for this problem [35, 23, 11], Blitzer et al [10, 9] proposed a structural correspondence learning approach that selects some 'pivot' features that would occur 'frequently' in both domains. Ben-David et al [5] generalized the results of [10] by presenting a theoretical analysis on the feature representation functions that should be used to minimize domain divergence, as well as classification error, under certain domain shift assumptions. More insights along this line of work was provided by [8, 29]. Another related method by Wang and Mahadevan [39] pose this problem in terms of unsupervised manifold alignment, where the manifolds on which the source and target domain lie are aligned by preserving a notion of the 'neighborhood structure' of the data points. All these methods primarily focus on natural language processing. However in visual object recognition, where we have still have relatively less

consensus on the basic representation to use for X and $\tilde{X}$, it is unclear how reasonable it is to make *subsequent* assumptions on the relevance of features extracted from X and $\tilde{X}$ [10] and the transformations induced on them [39].

3. Proposed Method

3.1. Motivation

Unlike existing methods that work with the information conveyed by the source and target domains *alone*, our methodology of addressing domain shift is inspired from incremental learning (that illustrates the benefits of adapting between extremes by gradually following the 'path' between them), and we attempt to identify 'potential' *intermediate domains* between the source and target and learn the information they convey about domain changes. In search of these novel domains, (i) we assume that we are given a N-dimensional representation of data from X and $\tilde{X}$, which depends on the user/ application, rather than relying on the existence of pivot features across domains [10], and (ii) we learn the 'path' between these two domains by exploiting the geometry of their underlying space, without making any assumptions on the domain shifting transformation (as in [39]). A formal problem statement is given below.

3.2. Problem Description

Let $X = \{x_i\}_{i=1}^{N_1} \in \mathbb{R}^N$ denote data from the source domain pertaining to M categories or classes. Let $y_i \in \{1, 2, 3, ...M\}$ denote the label of x_i. We assume that the source domain is mostly labeled, i.e. $X = X_l \cup X_u$ where $X_l = \{x_{li}\}_{i=1}^{N_{l1}}$ has labels, say $\{y_{li}\}_{i=1}^{N_{l1}}$, and $X_u = \{x_{ui}\}_{i=1}^{N_{u1}}$ are unlabeled ($N_{l1} + N_{u1} = N_1$). We further assume that all categories have some labeled data. Let $\tilde{X} = \{\tilde{x}_i\}_{i=1}^{N_2} \in \mathbb{R}^N$ denote unlabeled data from the target domain corresponding to the same M categories. Since subspace models are highly prevalent in modeling data characteristics (e.g. [38]), we work with generative subspaces[1] corresponding to the source and target domain. Let S_1 and S_2 denote generative subspaces of dimension[2] $N \times d$ obtained by performing principal component analysis (PCA) [38] on X and $\tilde{X}$ respectively, where $d < N$. We now address two issues: (i) How to obtain the $N \times d$ intermediate subspaces $S_t, t \in \mathbb{R}, 1 < t < 2$, and (ii) How to utilize the information conveyed by these subspaces on the labeled data X_l to estimate the identity of unlabeled $\tilde{X}$?

[1] Since we do not have labeled data from the target domain, our starting point will be generative subspaces that characterize the global nature of the domains, rather than the discriminative ones.

[2] d refers to the number of eigenvectors of the PCA covariance matrix that have non-zero eigenvalues. We choose the value of d to be minimum of that of S_1 and S_2, and restrict its maximum value to be less than N to enable use of methods that'll be discussed soon. It is interesting to determine a better approach for doing this.

- Given two points S_1 and S_2 on the Grassmann manifold.
- Compute the $N \times N$ orthogonal completion Q of S_1.
- Compute the thin CS decomposition of $Q^T S_2$ given by $Q^T S_2 = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} = \begin{pmatrix} V_1 & 0 \\ 0 & \tilde{V}_2 \end{pmatrix} \begin{pmatrix} \Gamma(1) \\ -\Sigma(1) \end{pmatrix} V^T$
- Compute $\{\theta_i\}$ which are given by the arccos and arcsine of the diagonal elements of Γ and Σ respectively, i.e. $\gamma_i = \cos(\theta_i)$, $\sigma_i = \sin(\theta_i)$. Form the diagonal matrix Θ containing θ's as diagonal elements.
- Compute $A = \tilde{V}_2 \Theta V_1^T$.

Algorithm 1: Numerical computation of the velocity matrix: The inverse exponential map [20].

3.3. Generating Intermediate Subspaces

To obtain meaningful intermediate subspaces between S_1 and S_2, we require a set of tools that is consistent with the geometry of the space spanned by these $N \times d$ subspaces. The space of d-dimensional subspaces in $\mathbb{R}^N$ (containing the origin) can be identified with the Grassmann manifold $\mathbb{G}_{N,d}$. S_1 and S_2 are points on $\mathbb{G}_{N,d}$. Understanding the geometric properties of the Grassmann manifold has been the focus of works like [41, 19, 1], and these have been utilized in some vision problems with subspace constraints, e.g. [37, 21, 28, 22]. A compilation of statistical analysis methods on this manifold can be found in [13]. Since a full-fledged explanation of these methods is beyond the scope of this paper, we refer the interested readers to the papers mentioned above.

We now use some of these results pertaining to the geodesic paths, which are constant velocity curves on a manifold, to obtain intermediate subspaces. By viewing $\mathbb{G}_{N,d}$ as a quotient space of $SO(N)$, the geodesic path in $\mathbb{G}_{N,d}$ starting from S_1 is given by a one-parameter exponential flow [20]: $\Psi(t') = Q \exp(t'B)J$, where exp refers to the matrix exponential, and $Q \in SO(N)$ such that $Q^T S_1 = J$ and $J = \begin{bmatrix} I_d \\ 0_{N-d,d} \end{bmatrix}$. I_d is a $d \times d$ identity matrix, and B is a skew-symmetric, block-diagonal matrix of the form $B = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}, A \in \mathbb{R}^{(N-d)\times d}$, where the superscript T denotes matrix transpose, and the submatrix A specifies the direction and the speed of geodesic flow. Now to obtain the geodesic flow between S_1 and S_2, we compute the direction matrix A such that the geodesic along that direction, while starting from S_1, reaches S_2 in unit time. A is generally computed using inverse exponential mapping (Algorithm 1). Once we have A, we can use the expression for $\Psi(t')$ to obtain intermediate subspaces between S_1 and S_2 by varying the value of t' between 0 and 1. This is generally performed using the exponential map (Algorithm 2). Let S' refer to the collection of subspaces

- Given a point on the Grassmann manifold S_1 and a tangent vector $B = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}$.
- Compute the $N \times N$ orthogonal completion Q of S_1.
- Compute the compact SVD of the direction matrix $A = \tilde{V}_2 \Theta V_1$.
- Compute the diagonal matrices $\Gamma(t')$ and $\Sigma(t')$ such that $\gamma_i(t') = \cos(t'\theta_i)$ and $\sigma_i(t') = \sin(t'\theta_i)$, where θ's are the diagonal elements of Θ.
- Compute $\Psi(t') = Q \begin{pmatrix} V_1\Gamma(t') \\ -\tilde{V}_2\Sigma(t') \end{pmatrix}$, for various values of $t' \in [0, 1]$.

Algorithm 2: Algorithm for computing the exponential map, and sampling along the geodesic [20].

$S_t, t \in \mathbb{R}, 1 \leq t \leq 2$, which includes S_1, S_2 and all intermediate subspaces. Let N' denote the total number of such subspaces.

3.4. Performing Recognition Under Domain Shift

We now model the information conveyed by S' on X and $\tilde{X}$ to perform recognition across domain change. We basically approach this stage by projecting X and $\tilde{X}$ onto S', and looking for correlations between them (by using the labels available from X). Let x'_{li} denote the $dN' \times 1$ vector formed by concatenating the projection of x_{li} onto all subspaces contained in S'. We now train a discriminative classifier $D(X'_l, Y'_l)$, where X'_l is the $dN' \times N_{l1}$ data matrix (with $x'_{li}, i = 1\ to\ N_{l1}$ forming the columns), and Y'_l is the corresponding $N_{l1} \times 1$ label vector (whose i^{th} row corresponds to y_{li}), and infer identity of $dN' \times 1$ vectors corresponding to projected target data $\tilde{x}'_i$. We use the method of partial least squares[3] (PLS) [40] to construct D since dN' is generally several magnitudes higher than N_{l1}, in which case PLS provides flexibility in choosing the dimension of the final subspace unlike other discriminant analysis methods such as LDA [3]. We outline the operating principle behind PLS in the Appendix.

3.5. Extensions

3.5.1 Semi-supervised Domain Adaptation

We now consider cases where there are some labels in the target domain. Let $\tilde{X} = \tilde{X}_l \cup \tilde{X}_u$ where $\tilde{X}_l = \{\tilde{x}_{li}\}_{i=1}^{N_{l2}}$ has labels, say $\{\tilde{y}_{li}\}_{i=1}^{N_{l2}}$, and $\tilde{X}_u = \{\tilde{x}_{ui}\}_{i=1}^{N_{u2}}$ is unlabeled ($N_{l2} + N_{u2} = N_2$). We now use a $dN' \times (N_{l1} + N_{l2})$ data matrix (whose columns correspond to the projections of labeled data from both domains onto S') and the corresponding $(N_{l1} + N_{l2}) \times 1$ label vector to build the classifier D, and infer the labels of $\tilde{x}_{ui}, i = 1\ to\ N_{u2}$.

[3] Alternately, one can choose any other method for the steps involving PCA, and PLS.

1. Given a set of k points $\{q_i\}$ on the manifold.
2. Let μ_0 be an initial estimate of the Karcher mean, usually obtained by picking one element of $\{q_i\}$ at random. Set j = 0.
3. For each $i = 1, .., k$, compute the inverse exponential map ν_i of q_i about the current estimate of the mean i.e. $\nu_i = \exp_{\mu_j}^{-1}(q_i)$.
4. Compute the average tangent vector $\bar{\nu} = \frac{1}{k}\sum_{i=1}^{k} \nu_i$.
5. If $\|\bar{\nu}\|$ is small, then stop. Else, move μ_j in the average tangent direction using $\mu_{j+1} = \exp_{\mu_j}(\epsilon\bar{\nu})$, where $\epsilon > 0$ is small step size, typically 0.5.
6. Set $j = j + 1$ and return to Step 3. Continue till μ_j does not change anymore or till maximum iterations are exceeded.

Algorithm 3: Algorithm to compute the sample Karcher mean [13].

3.5.2 Adaptation Across Multiple Domains

There can also be scenarios where we have multiple domains in source and target [30, 17]. One way of dealing with k_1 source domains and k_2 target domains is to create generative subspaces $S_{11}, S_{12}, .., S_{1k_1}$ corresponding to the source, and $S_{21}, S_{22}, ..., S_{2k_2}$ for the target. From this we can compute the mean of source subspaces, say $\bar{S}_1$, and the mean for target $\bar{S}_2$. A popular method for defining the mean of points on a manifold was proposed by Karcher [24]. A technique to obtain the Karcher mean is given in Algorithm 3. We then create intermediate subspaces between $\bar{S}_1$ and $\bar{S}_2$, and learn the classifier D to infer target labels as before.

4. Experiments

We first compare our method with existing approaches for 2D object recognition [33, 7], and empirically demonstrate the benefits of creating intermediate domains. In this process, we also test the performance of the semi-supervised extension of our algorithm, and for cases with more than one source or target domains. Finally, we provide comparisons with unsupervised DA approaches on natural language processing tasks.

4.1. Comparison with Metric Learning Approach [33]

We used the dataset of [33] that has 31 different object categories collected under three domain settings: images from $amazon$, $dslr\ camera$, and $webcam$. There are 4652 images in total, with the object types belonging to backpack, bike, notebook, stapler etc. The $amazon$ domain has a average of 90 instances for each category, whereas $DSLR$ and $webcam$ have roughly around 30 instances for a cate-

(a)

Domain		Metric learning [33] (*semi-supervised*) Classification (%) (mean)		Ours Classification (%) (mean±std. deviation)	
Source	Target	asymm	symm	Un-supervised	Semi-supervised
webcam	dslr	25	27	19±1.2	37±2.3
dslr	webcam	30	31	26±0.8	36±1.1
amazon	webcam	48	44	39±2.0	57±3.5

(b)

Domain		Metric learning [33] (*semi-supervised*) Classification % (mean)		Ours Classification (%) (mean±std. deviation)	
Source	Target	asymm	symm	Un-supervised	Semi-supervised
webcam	dslr	53	49	42±0.6	59±3.1

Table 1. Comparison of classification performance with [33]. (a) with labels for all target domain categories. (b) with labels only for partial target categories. asymm and symm are two variants proposed by [33].

Figure 2. Sample retrieval results from our unsupervised method on the dataset of [33]. Left column: query image from the target domain. Columns 2 to 6: Top 5 closest matches from the source domain. Source/ target combination for rows 1 to 5 are as follows: $dslr/amazon$, $webcam/dslr$, $dslr/webcam$, $webcam/amazon$, $amazon/webcam$.

gory. The domain shift is caused by several factors including change in resolution, pose, lighting etc.

We followed the protocol of [33] in extracting image features to represent the objects. We resized all images to 300×300 and converted them into grayscale. SURF features [2] were then extracted, with the blob response threshold set at 1000. The 64-dimensional SURF features were then collected from the images, and a codebook of size 800 was generated by k-means clustering on a random subset of $amazon$ database (after vector quantization). Then the images from all domains are represented by a 800 bin histogram corresponding to the codebook. This forms our data representation for X and $\tilde{X}$, with $N = 800$. From this we learnt the subspaces corresponding to source and target, and chose the subspace dimension d to be the lower of the two (and less than N). The value of d was set between 185 and 200 for different experiments on this dataset. We experimentally fixed the number of intermediate subspaces to 8 (i.e. $N' = 10$), and the PLS dimensions p to 30 (please refer to the Appendix on how we obtain p-dimensional vectors using PLS).

We report results on two experimental settings, (i) with labeled data available in both source and target domains - 3 labels per category in target for $amazon/webcam/dslr$, and 8 per category in source domain for $webcam/dslr$, and 20 for $amazon$; and (ii) labeled data is available in both domains only for the first half of categories, whereas the last 16 categories has labels only in the source domain. For the first setting, we determine the identity of all unlabeled data from the target domain, whereas for the second setting we determine the labels of unlabeled target data from the last 16 categories. For both experiments, we report the results

of our method in unsupervised setting (where we do not use labels from target, even if available) and semi-supervised setting (where the target labels are used) in Tables 1(a) and 1(b) respectively. The performance accuracy (number of correctly classified instances over total test data from target) is reported over 20 different trials corresponding to different labeled data across source and target domains. It can be seen that although our unsupervised adaptation results are slightly lower than that of [33] (which is reasonable since we throw away all correspondence information, while [33] uses them), our semi-supervised extension offers better performance improvement. Also note that the result in Table 1(b) is better than the corresponding category of Table 1(a) since the former is a 16 way classification, while the later is a 31-way classification. Some retrieval results from our unsupervised approach, corresponding to different source and target domain combinations, are presented in Figure 2.

4.2. Comparison with Semi-supervised SVM's [7]

We then used the data of [7] that has two domains: the target domain with images from Caltech256 that has 256 object categories, and the source domain corresponding to the weakly labeled results of those categories obtained from $Bing$ image search. We used the classeme features to represent the images. Each image was represented by a 2625-dimensional binary vector, which models several semantic attributes of the image [7]. We followed the protocol of [7] and present results on classifying the unlabelled target data under two experimental settings, (i) by fixing the number of labeled samples from the source domain and varying the labeled samples from target (starting from one), and (ii) doing the reverse by fixing the number of labeled target data, and varying the labeled samples from source. We also consider another operating point of no labeled data from the target and source domains respectively (corresponding to the above two settings) to perform unsupervised DA. It can be seen from Figures 3(a) and 3(b) that our method gives better performance overall, with the gain in accuracy increasing with the number of labeled data. The performance is measured using the percentage of correctly classified unlabeled samples from the target, averaged across several trials on choosing different labeled samples.

4.3. Studying the information conveyed by intermediate subspaces, and multi-domain adaptation

We now empirically study the information we gain by creating the intermediate domains. We use the data of [33, 7] where we evaluate the performance of our algorithm (unsupervised case) across different values[4] of N' ranging from 2 to 15. The same experimental setup of Sec 4.1 and 4.2 was followed. $N' = 2$ denotes no intermediate subspace, and we use the information conveyed by S_1 and S_2 alone. This provides a baseline for our method. As seen in Figure 3(c), all values of $N' > 2$ offers better performance than $N' = 2$. Although this result is data-dependent, we see that we gain some information from these new domains.

[4]All these runs correspond to $p = 30$, which was empirically found to give the best performance.

Domain		Ours Classification (%) (mean±std. deviation)	
Source	Target	Un-supervised	Semi-supervised
amazon, dslr	webcam	31±1.6	52±2.5
amazon, webcam	dslr	25±0.4	39±1.1
dslr, webcam	amazon	15±0.4	28±0.8
webcam	amazon, dslr	28±1.9	42±2.8
dslr	amazon, webcam	35±1.7	46±2.3
amazon	dslr, webcam	22±0.2	32±0.9

Table 2. Performance comparison across multiple domains in source or target, using the data from [33].

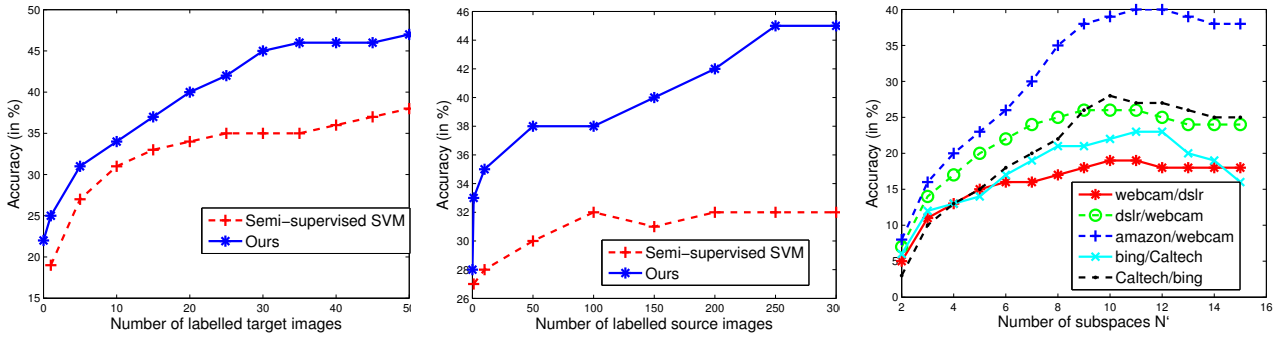
Domain		Method Classification (%)		
Target	Source	[10]	[9]	Ours
B	D,E,K	76.8,75.4,66.1	79.7,75.4,68.6	78.2,76.3,74.2
D	B,E,K	74.0,74.3,75.4	75.8,76.2,76.9	76.1,75.8,79.1
E	B,D,K	77.5,74.1,83.7	75.9,74.1,86.8	81.2,76.2,87.6
K	B,D,E	78.7,79.4,84.4	78.9,81.4,85.9	78.1,82.0,89.7

Table 3. Performance comparison with some unsupervised DA approaches on language processing tasks [9]. Key: B-books, D-DVD, E-electronics, and K-kitchen appliances. Each row corresponds to a target domain, and three separate source domains.

We then experimented with the data of [33] when there are multiple domains in source or target. We created six different possibilities, three cases with two source domain and one target domain, and the other three with two target domains and one source domain. The experimental setup outlined in Sec 4.1 was followed, where we consider the case with labels for all target categories. We provide the classification accuracy of our unsupervised and semi-supervised variants in Table 2. Although we do not have a baseline to compare with, one possible relation with the results in Table 1(a) is for the case where the target domain is $webcam$ and the source domains contain $dslr$ and $amazon$. It can be seen that the joint source adaptation results lie somewhere in between single source domain cases.

4.4. Comparison with unsupervised approaches on non-visual domain data

We now compare our approach with other unsupervised DA approaches that have been proposed for natural language processing tasks. We used the dataset of [9] that performs adaptation for *sentiment classification*. The dataset has product reviews from amazon.com for four different domains: books, DVD, electronics and kitchen appliances. Each review has a rating from 0 to 5, a reviewer name and



(a) (b) (c)

Figure 3. (a), (b): Performance comparison with [7]. (a) Number of labeled source data = 300. (b) Number of labeled target data = 10. Semi-supervised SVM refers to the top performing SVM variant proposed in [7]. *Please note that our method also has an unsupervised working point (at position 0 on the horizontal axis).* (c) Empirically studying the effect of N' on data from [33, 7]. Naming pattern refers to source/ target domain. Accuracy for $N' > 2$ is more than that for $N' = 2$, which says that the intermediate subspaces do provide some useful information. However, since larger values of N' need not always translate into better classification (e.g. Bing/Caltech curve), it is interesting to formally study the optimal value of N'.

location, review text, among others. Reviews with rating more than 3 were classified as positive, and those less than 3 were classified negative. The goal here is to see whether the process of learning positive/ negative reviews from one domain, is applicable to another domain. We followed the experimental setup of [9], where the data representation for X and $\tilde{X}$ are unigram and bigram features extracted from the reviews. Each domain had 1000 positive and negative examples each, and the data for each domain was split into a training set (source domain) of 1600 instances and a test set (target domain, with hidden labels) of 400 instances. The classification accuracies with different settings of source and target domain are given in Table 3. We see that our method performs better overall, even though we do not identify pivot features from the bigram/unigram data features (as done by the other two methods). This experiment also illustrates the utility of our method for domain adaptation across general, non-visual domains.

5. Conclusion

We have proposed a data driven approach for unsupervised domain adaptation, by drawing inspirations from incremental learning. Differing from existing methods that make assumptions on transformations or feature distributions across domains, we investigated the information conveyed by 'potential' intermediate domains on the unknown domain shift. Although the tools used to create these novel domains are consistent with the underlying geometry of data, the absence of labeled target data does not allow us to guarantee that these domains would 'physically' correspond to the 'actual' domain transformation. Therefore to enable a better understanding of unsupervised domain adaptation, the following broad problems are of interest: (i) utilizing generic priors on possible domain shifts to create and traverse physically meaningful intermediate domains, and (ii) exploring data representations beyond linear subspaces, with some desirable domain invariant properties that could accommodate potentially different data dimensionality across domains.

Appendix: PLS

Let $\mathbb{X} \in R^m$ denote an m-dimensional space of feature vectors and similarly let $\mathbb{Y} \in R$ be a 1-dimensional space representing the class labels. Let the number of samples (training patches) be n. PLS decomposes the zero-mean matrix $\mathbf{X}$ $(n \times m)$ and zero-mean vector $\mathbf{y}$ $(n \times 1)$ into

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \quad (1)$$

$$\mathbf{y} = \mathbf{U}\mathbf{q}^T + \mathbf{f} \quad (2)$$

where $\mathbf{T}$ and $\mathbf{U}$ are $(n \times p)$ matrices containing p extracted latent vectors, the $(m \times p)$ matrix $\mathbf{P}$ and the $(1 \times p)$ vector $\mathbf{q}$ represent the loadings and the $(n \times m)$ matrix $\mathbf{E}$ and the $(n \times 1)$ vector $\mathbf{f}$ are the residuals. The PLS method, using the nonlinear iterative partial least squares (NIPALS) algorithm, constructs a set of weight vectors (or projection vectors) $W = \{w_1, w_2, ..., w_p\}$ such that

$$[cov(t_i, u_i)]^2 = \max_{|\mathbf{w}_i|=1} [cov(\mathbf{X}\mathbf{w}_i, \mathbf{y})]^2 \quad (3)$$

where $\mathbf{t}_i$ is the i^{th} column of matrix $\mathbf{T}$,$\mathbf{u}_i$ the i^{th} column of matrix $\mathbf{U}$ and $cov(t_i, u_i)$ is the sample covariance between latent vectors $\mathbf{t}_i$ and $\mathbf{u}_i$. After the extraction of the latent vectors $\mathbf{t}_i$ and $\mathbf{u}_i$, the matrix $\mathbf{X}$ and vector $\mathbf{y}$ are deflated by subtracting their rank-one approximations based on $\mathbf{t}_i$ and $\mathbf{u}_i$. This process is repeated until the desired number of latent vectors had been extracted. The dimensionality reduction is performed by projecting the feature vector $\mathbf{v}_i$, extracted from a i^{th} detection window, onto the weight

vectors $W = \{w_1, w_2, .., w_p\}$, obtaining the latent vector $\mathbf{z}_i$ $(1 \times p)$ as a result. Such vectors obtained from the unlabeled target domain are compared with vectors from the labeled source domain to perform classification. We used the nearest neighbor classifier with l_2 norm as the distance measure for this purpose, although any other classifier such as an SVM can be used. The same approach can also be used to infer the labels of unlabeled data from the source domain.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Appl. Math.*, 80:199–220, February 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *CVIU*, 110:346–359, June 2008.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE TPAMI*, 19:711 –720, July 1997.

[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, May 2010.

[5] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *NIPS*, pages 137–145, December 2007.

[6] S. Ben-David, T. Luu, T. Lu, and D. Pal. Impossibility theorems for domain adaptation. In *AISTATS*, pages 129–136, Sardinia, Italy, May 2010.

[7] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach. In *NIPS*, pages 181–189. Vancouver, Canada, December 2010.

[8] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *NIPS*, pages 129–136, Vancouver, Canada, December 2008.

[9] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pages 440–447, Prague, Czech Republic, June 2007.

[10] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, pages 120–128, Sydney, Australia, July 2006.

[11] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE TPAMI*, 32:770 –787, May 2010.

[12] S. Chalup. Incremental learning in biological and machine learning systems. *International Journal of Neural Systems*, 12:447–466, June 2002.

[13] Y. Chikuse. *Statistics on special manifolds*, volume 174 of *Lecture Notes in Statistics*. Springer Verlag, 2003.

[14] W. Dai, G. Xue, Q. Yang, and Y. Yu. Transferring naive Bayes classifiers for text classification. In *National Conference on Artificial Intelligence*, pages 540–543, Vancouver, Canada, July 2007.

[15] H. Daumé III, A. Kumar, and A. Saha. Co-regularization based semi-supervised domain adaptation. In *NIPS*, pages 478–486. Vancouver, Canada, December 2010.

[16] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, May 2006.

[17] M. Dredze and K. Crammer. Online methods for multi-domain learning and adaptation. In *EMNLP*, pages 689–697, Honolulu, HI, USA, October 2008.

[18] L. Duan, I. Tsang, D. Xu, and T. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, pages 289–296, Montreal, Canada, June 2009.

[19] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Application*, 20:303–353, April 1999.

[20] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *IEEE Workshop on Statistical Signal Processing*, pages 315 – 318, St. Louis, MO, USA, September 2003.

[21] J. Hamm and D. D. Lee. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *ICML*, pages 376–383, Helsinki, Finland, July 2008.

[22] M. Harandi, S. Shirazi, C. Sanderson, and B. Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR*, pages 2705–2512, Colorado Springs, CO, USA, June 2011.

[23] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429–449, October 2002.

[24] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30:509–541, May 1977.

[25] A. Khotanzad and Y. Hong. Invariant image recognition by zernike moments. *IEEE TPAMI*, 12:489–497, May 1990.

[26] K. Lai and D. Fox. Object recognition in 3D point clouds using Web data and domain adaptation. *The International Journal of Robotics Research*, 29:1019–1028, August 2010.

[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, November 2004.

[28] Y. M. Lui and J. R. Beveridge. Grassmann registration manifolds for face recognition. In *ECCV*, pages 44–57, Marseille, France, October 2008.

[29] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *Arxiv preprint arXiv:0902.3430*, 2009.

[30] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, pages 1041–1048. Vancouver, Canada, December 2009.

[31] S. J. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Networks*, 22:199 –210, February 2011.

[32] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowledge and Data Engineering*, 22:1345 –1359, October 2010.

[33] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, Heraklion, Greece, September 2010.

[34] J. Schlimmer and R. Granger. Incremental learning from noisy data. *Machine learning*, 1:317–354, March 1986.

[35] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, October 2000.

[36] G. Tur. Co-adaptation:Adaptive co-training for semi-supervised learning. In *ICASSP*, pages 3721–3724, Taipei, Taiwan, April 2009.

[37] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In *CVPR*, pages 1–8, Anchorage, AK, USA, June 2008.

[38] M. Turk and A. Pentland. Face recognition using eigenfaces. In *CVPR*, pages 586 –591, Maui, HI, USA, June 1991.

[39] C. Wang and S. Mahadevan. Manifold alignment without correspondence. In *IJCAI*, pages 1273–1278, Pasadena, CA, USA, July 2009.

[40] H. Wold. *Partial Least Squares*, volume 6. Encyclopedia of Statistical Sciences, 1985.

[41] Y. Wong. Differential geometry of Grassmann manifolds. *Proceedings of the National Academy of Science*, 57:589–594, March 1967.

[42] D. Xing, W. Dai, G. Xue, and Y. Yu. Bridged refinement for transfer learning. *PKDD*, pages 324–335, September 2007.

[43] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computers*, C-21:269 –281, March 1972.