ABSTRACT

Title of dissertation: DOMAIN ADAPTIVE
OBJECT RECOGNITION AND DETECTION

Fatemeh Mirrashed, Doctor of Philosophy, 2013

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science

Discriminative learning algorithms rely on the assumption that training and test data are drawn from the same marginal probability distribution. In real world applications, however, this assumption is often violated and results in a significant performance drop. We often have sufficient labeled training data from single or multiple "source" domains but wish to learn a classifier which performs well on a "target" domain with a different distribution and no labeled training data. In visual object detection, for example, where the goal is to locate the objects of interest in a given image, it may be infeasible to collect training data to model the enormous variety of possible combinations of pose, background, resolution, and lighting conditions affecting object appearance. Thus, we generally expect to encounter instances or domains at test time for which we have seen little or no training data.

To this end, we first propose a framework for domain adaptive object recognition and detection using Transfer Component Analysis, an unsupervised domain adaptation and dimensionality reduction technique. The idea is to obtain a transformation in feature space to a latent subspace that reduces the distance between the

source and target data distributions. We evaluate the effectiveness of this approach for vehicle detection using video frames from 50 different surveillance cameras.

Next, we explore the problem of extreme class imbalance present when performing fully unsupervised domain adaptation for object detection. The main challenge arises from the fact that images in unconstrained settings are mostly occupied by the background (negative class). Therefore, random sampling will not be effective in obtaining a sufficient number of positive samples from the target domain, which is required by any adaptation method. We propose a variation of co-learning technique that automatically constructs a more balanced set of samples from the target domain. We compare the performance of our technique with other approaches such as unbiased learning from multiple datasets and self-learning.

Finally, we propose a novel approach for unsupervised domain adaptation. Our method learns a set of binary attributes for classification that captures the structural information of the data distribution in the target domain itself. The key insight is finding attributes that are discriminative across categories and predictable across domains. We formulate our optimization problem to learn these attributes and the classifier jointly. We evaluate the performance of our method on a wide range of tasks including cross-domain object recognition and sentiment analysis on textual data both in inductive and transductive settings. We achieve a performance that significantly exceeds the state-of-the-art results on standard benchmarks. In many cases we reach the same-domain performance, the upper bound, in unsupervised domain adaptation scenarios.

DOMAIN ADAPTIVE OBJECT RECOGNITION AND DETECTION

by

Fatemeh Mirrashed

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2013

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Min Wu, Dean's Representative
Professor David W. Jacobs
Professor Ramani Duraiswami
Professor Amitabh Varshney

Dedication

To my parents, "Aghdas" and "Mirgholam",

who instilled in me the love of learning.

To my beloved "Bahman",

whose love and support made this possible.

To my beautiful "Kimia",

whose blessed birth marked the start of this journey.

Acknowledgments

To Be Completed

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Building visual models of objects robust to extrinsic[1] variations such as camera view angle, resolution, lighting, and blur has long been one of the challenges in computer vision. Generally, a discriminative or generative statistical model is learned by acquiring a large set of examples, extracting low-level features which encode shape, color, or texture from the segmented or cropped objects, and finally, training the model (usually a classifier) using the extracted features vectors. Applied to a test image, however, the trained model usually works only if the training set was representative of the test set, i.e., if the distribution over training examples roughly matches the distribution of the test data. Unfortunately, there are often cases when this implicit key assumption of learning algorithms is violated, resulting in a sharp performance drop.

There has been a recent growing interest in the machine learning community to develop effective mechanisms to transfer or adapt knowledge from one (source) domain to another related (target) domain. Taking advantage of these advancements, in Chapter 2 We propose a framework for adaptive object detection using Transfer Component Analysis, an unsupervised domain adaptation and dimensionality reduction technique. In Chapter 3 we focus on the challenge of obtaining a

[1] as opposed to intrinsic or intra-class variation of an object category with respect to different shapes, sizes, textures, colors, etc.

set of balanced samples from the target domain for a fully unsupervised domain adaptive object detection. And finally, in Chapter 4 we present a novel approach for domain adaptation for cross-domain classification task and evaluate our method on both visual and textual data. The following sections present an abstract of each of these upcoming chapters.

1.1 Domain Adaptive Object Detection

We study the use of domain adaptation and transfer learning techniques as part of a framework for adaptive object detection. Given labeled examples from the source domain and unlabeled examples from the target domain, we obtain a transformation to a latent subspace that reduces the distance between the source and target distributions while simultaneously preserving data properties. This enables standard classifiers to generalize directly to unseen examples from the target domain.

Unlike recent applications of domain adaptation work in computer vision, which generally focus on image classification, we apply our technique to vehicle detection in a challenging urban surveillance dataset, where the backgrounds, numbers, and poses of the objects of interest are all uncontrolled and vary highly. We demonstrate the performance of our approach with various amounts of supervision, including the fully unsupervised case.

1.2 Sampling For Fully Unsupervised Domain Adaptive Object Detection

We explore the problem of extreme class imbalance present when performing fully unsupervised domain adaptation for object detection. The main challenge arises from the fact that images in unconstrained settings are mostly occupied by the background (negative class). Therefore, random sampling will not typically result in a sufficient number of positive samples from the target domain, which is required by domain adaptation methods. Motivated by traditional semisupervised learning algorithms that aim for better classification using both labeled and unlabeled data, we propose a variation of co-learning technique that automatically constructs a more balanced set of samples from the target domain. We evaluate the effectiveness of our approach using a vehicle detection task in an urban surveillance dataset. Furthermore, we compare the performance of our technique with two other approachesone based on unbiased learning on multiple training data sets and the other on self-learning.

1.3 Domain Adaptive Classification: A Novel Approach

We propose an unsupervised domain adaptation method that exploits intrinsic compact structures of categories across different domains using binary attributes. Our method directly optimizes for classification in the target domain. The key insight is finding attributes that are discriminative across categories and predictable across domains. We achieve a performance that significantly exceeds the state-of-

the-art results on standard benchmarks. In fact, in many cases, our method reaches the same-domain performance, the upper bound, in unsupervised domain adaptation scenarios.

Chapter 2

Domain Adaptive Object Detection

2.1 Introduction

Learning algorithms rely on the assumption of similarity between the distribution of data in training and test sets. However in practice there are often slight or significant differences between these distributions. The differences could arise due to the cost of collecting large training data sets or the difficulties in obtaining training instances from a particular target domain. It is often infeasible to collect training data for the enormous variety of domains for the classification task in hand, therefore, in realistic applications we expect to encounter settings at the test time for which we have seen little or no training data.

Recently, the machine learning community and in particular the researchers in natural language processing have been seeking to develop effective mechanisms to transfer or adapt knowledge from one domain to another related domain [**?**, 1–5]. While these advances have also been applied by the computer vision community with promising results [6–9], object models are still being trained and tested on images consisting of only one object zoomed and cropped at the center of a relatively uniform background. As a result, in such experimental settings the general problem of object detection is reduced to that of image classification. While domain adaptation is a challenging problem for image classification, it becomes even more

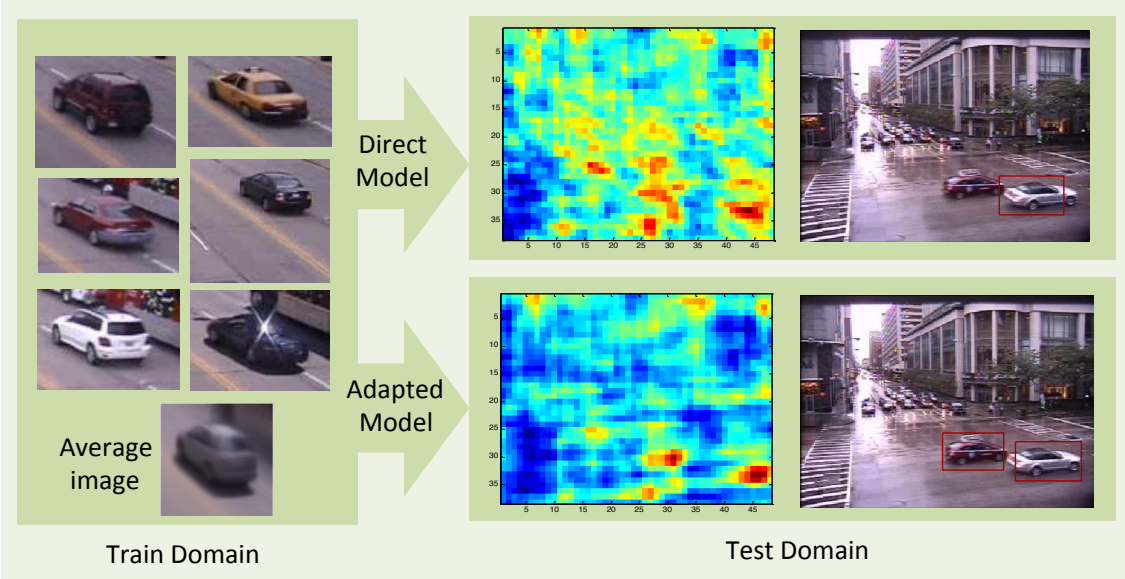


Figure 2.1: An example of the effects of domain change and our improved results after domain adaptation. If we directly apply the trained model to a new domain, the confidence map has multiple peaks, many of which do not correspond to vehicles. After domain adaptation, the highest peaks correspond to the two vehicles in the foreground.

challenging for object detection when target domain labels are unavailable and the majority of the image is occupied by the background class (a random sampling will not be sufficient for effective domain adaptation).

We focus on domain adaptation applied to vehicle detection in urban surveillance videos, where the backgrounds, numbers, and poses of the objects of interest are all uncontrolled and vary highly. The detection and localization of vehicles in surveillance video, which is typically low resolution, is extremely difficult as it requires dealing with varying viewpoint, illumination conditions (e.g. sunlight, shadows, reflections, rain, snow),and traffic. These conditions are localized in space and

time, allowing us to model realistic domain changes by considering two cameras at different locations and points in time as the source and target domain. As we will demonstrate, the changes between some domains are sufficiently large that the classifier trained on the source domain performs extremely poorly. By applying recent domain adaptation techniques, we obtain significant improvements in these cases (see Fig 3.1 for an example result).

We use Transfer Component Analysis (TCA) [10], an unsupervised domain adaptation and dimensionality reduction technique, to learn a set of common transfer components underlying both domains such that, when projected onto this subspace, the difference in data distributions of two domains can be dramatically reduced while preserving data properties. Standard machine learning algorithms can then be used in this subspace to train classification or regression models across domains. While TCA obtains the transfer components by aligning distribution properties that are not class-aware, i.e., it does not guarantee that the same class in separate domains will project to the same coordinates in the shared subspace, we find that for our problem this alignment yields impressive results. Our contributions are the following:

- we extensively evaluate a domain adaptation technique, TCA, applied to vehicle detection on a challenging dataset
- we provide insights into what makes TCA perform so well on our dataset by comparing to basic machine learning techniques
- we propose an initial approach to selecting target samples for domain adap-

tation in a more general object detection setting (multiple object instances, objects are generally not centered, and the image consists of mostly background)

The remainder of this chapter is organized as follows. We review related literature in Section 2, followed by a detailed description of our proposed approach in Section 3. We describe the experiments and results in Section 4 and finally conclude in Section 5.

2.2 Related Work

Object category recognition and detection approaches that are invariant to view and other extrinsic changes have long been sought by researchers in computer vision [11]. Several methods address changes in view by learning separate appearance models for a small number of canonical poses corresponding to each object category [12, 13]. Other approaches employ parts-based models, which model variations in part appearance and inter-relationships over multiple views [14–17]. Recently, Gu and Ren [13] proposed a discriminative approach based on a mixture of templates, achieving the best performance on two different 3D object recognition datasets. Unfortunately, this performance gain is achieved at up to an order of magnitude higher cost–depending on the number of templates used–than a comparable view-specific method that employs a similar feature representation.

The problem of learning object models that can generalize to new views or domains is closely related to the problems of transfer learning [1] and domain adap-

tation [3], the two main groups of work that address the effects of domain change in machine learning. In general, a domain consists of the input data feature space and an associated probability distribution over it. If two domains are different, they may have different feature spaces or different marginal probability distributions. The problem of *domain adaptation* addresses domain changes, when the marginal distribution of the data in the training set (source domain) and the test set (target domain) are different but the tasks or conditional distributions of some additional variables, or labels, given the data are assumed to approximately the same. Domain adaptation techniques often work when the distribution of the two domains differ only slightly [5]. The problem of *transfer learning* addresses situations in which marginal distributions of data between the domains are the same but either the feature spaces or conditional distribution of the labels given data are different.

The natural language processing community has lately paid considerable attention to understanding and adapting to the effects of domain change. Daume et al [3] model the data distribution corresponding to source and target domains as a common shared component and a component that is specific to the individual domains. Under certain assumptions characterizing the domain shift, there have also been theoretical studies on the nature of classification error across new domains [18, 19]. Blitzer et al [20, 21] proposed a structural correspondence learning approach that selects some pivot features that would occur frequently in both domains. Ben-David et al [2] generalized the results of [20] 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. Insights along

this line of work were also provided by [4, 5]. Wang and Mahadevan [22] pose this problem as unsupervised manifold alignment, where source and target manifolds are aligned by preserving a notion of the neighborhood structure of the data points.

In visual object recognition, there is less consensus on the basic representation the data, so it is unclear how reasonable it is to make subsequent assumptions on the relevance of extracted features [20] and the transformations induced on them [22]. However, there have been recent efforts focusing on domain shift issues for 2D object recognition applications. For instance, Saenko et al [6] proposed a metric learning approach that can use labeled data for a few categories from the target domain to adapt unlabeled categories to domain change. Bergamo and Torresani [23] performed an empirical analysis of several variants of SVM for this problem. Lai and Fox [24] 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. Gopalan et al [9] take an incremental learning approach, following a geodesic path between the two domains modeled as points on a Grassmann manifold.

We extend recent work by applying a domain adaptation technique, TCA [10], to the problem of object detection. We study the effects of varying amounts of balanced target domain training samples, similar to the classification setting of [6–9], and we also explore the automatic acquisition of training data from the target domain, which is more applicable to the detection problem. In the detection setting, the class labels are unavailable, and the classes are highly imbalanced since the majority of windows in the image contain background and only a few are good examples of the object class.

2.3 Proposed Method

2.3.1 Formulation

Following the notation of Pan et al. [10], we define a domain to consist of a feature space and a distribution $P(X)$, defined over a set of examples $X = \{x_1, \ldots, x_n\}$ from the feature space. The examples in X have a corresponding set of labels $Y = \{y_1, \ldots, x_n\}$. While domains can differ both in the feature space and in the marginal distribution, we consider only the case where the feature space remains constant across domains. Given training features X_S and labels Y_S from the source domain and training features X_T from the target domain, our task is to learn a model that can predict the labels on new samples from the target domain.

While most domain adaptation methods assume that $P(X_S) \neq P(X_T)$ and that $P(Y_S|X_S) = P(Y_T|X_T)$, TCA [10] replaces the second assumption with a more realistic one, that probability $P(Y|X)$ may also change across domains, but that there exists a transformation ϕ such that $P(\phi(X_S)) \approx P(\phi(X_T))$ and $P(Y_S|\phi(X_S)) \approx P(Y_T|\phi(X_T))$. Based on these assumptions and given X_S and X_T, TCA obtains the transformation ϕ. A classifier can then be trained on transformed features $\phi(X_S)$ and labels Y_S and applied to transformed out-of-sample target features $\phi(X_T^o)$ to predict labels Y_T^o.

2.3.2 Transfer Component Analysis

Given training samples from two domains, X_S and X_T, TCA [10] obtains a transformation ϕ to a latent space that minimizes the distance between the trans-

formed distributions while preserving properties of both input feature spaces. This optimization is performed in a reproducing kernel Hilbert space (RKHS), under the assumption that ϕ is a feature map which defines a universal kernel. The distance between the transformed distributions is measured by the empirical estimate of Maximum Mean Discrepancy (MMD):

$$MMD(X_S, X_T) = ||\frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_{S_i}) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_{T_i})||^2,$$

where n_1 and n_2 are the number of samples in X_S and X_T, respectively, and the norm is the RKHS norm. Properties of the input feature spaces are preserved by maximizing the variance of the transformed data.

Instead of directly optimizing for the feature map ϕ, TCA first applies a parametric kernel (e.g., linear or RBF) to obtain the kernel matrix $K = [k(x_i, x_j)] \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ of the source and target training features, and then searches for a matrix $\widetilde{W} \in \mathbb{R}^{(n_1+n_2)\times m}$ that projects the empirical kernel map $K^{-1/2}K$ to an m-dimensional space $\widetilde{W}^T K^{-1/2}K$. Letting $W = K^{-1/2}\widetilde{W}$, the feature map ϕ induced by the kernel KWW^TK is thus optimized implicitly by the following constrained minimization:

$$\min_W \mathtt{tr}(W^TKLKW) + \mu\mathtt{tr}(W^TW)$$

$$\mathtt{s.t.}\ \ W^TKHKW = I.$$

Here, the MMD criterion is rewritten as $\mathtt{tr}(W^TKLKW)$, where $L_{ij} = 1/n_1^2$ if $x_i, x_j \in X_S$, $L_{ij} = 1/n_2^2$ if $x_i, x_j \in X_T$, and $L_{ij} = -1/(n_1n_2)$ otherwise. The variance is maximized by minimizing $\mathtt{tr}(W^TW)$ under the constraint that the projected data has unit covariance, $W^TKHKW = I$, where H is the centering matrix $H = I - 1/(n_1+n_2)\mathbf{1}\mathbf{1}^T$. The parameter μ controls the trade-off between minimizing the

distance between distributions and maximizing data variance. As Pan et al. [10] demonstrate, this optimization problem can be reformulated without constraints as

$$\max_{W} \mathtt{tr}((W^T(KLK + \mu I)W)^{-1}W^T KHKW).$$

This optimization problem is solved by obtaining the m leading eigenvectors of $(KLK+\mu I)^{-1}KHK$. A new sample x_o is mapped into the latent space by computing $W^T[k(x_1, x_o), \ldots, k(x_{n_1+n_2}, x_o)]^T$, where x_i are the training samples.

2.3.3 Unsupervised adaptation

For an object detector to adapt to a new domain using our proposed approach, a set of features from the target domain, X_T, is needed during the training stage. A straightforward unsupervised approach to obtaining such a set for a multi-scale sliding window detector would be to randomly select a number of windows that would be encountered during the detection process. However, this would yield a majority of windows from the negative class and only a few (most likely poorly localized) positive samples. A potential solution would be to introduce a small amount of supervision into the process. Since our proposed approach does not use class labels during the domain adaptation step, it is only important that the classes are balanced by the user somehow to prevent the joint latent space from being dominated solely by the target background class. While this may be an acceptable solution, especially if it is sufficient for the user to annotate a very low number of examples (in our experiments we show that very little supervision is necessary for significant improvements), we are also interested in studying the fully unsupervised

case.

In the absence of any supervision, we propose a scheme that relies on a detector trained on X_S and Y_S alone to extract positive and negative examples from the target distribution. Before performing domain adaptation using TCA, our scheme involves extracting the top and bottom scoring windows subject to some threshold (after non-maximal suppression), as the positive and negative samples to include in X_T. While a detector trained on the source domain alone would not be very accurate, we expect that regions of very high confidence are more likely to contain the object of interest than the regions of low confidence. While the detection rate may not be high, labels are not needed for the target training set, so labeling mistakes will not be very detrimental. Most importantly, we expect the resulting set of windows to be much more balanced than a random selection of samples.

2.4 Experiments and Results

2.4.1 Data Set Collection

2.4.1.1 Training set

We have collected more than 400 hours of video from 50 different traffic surveillance cameras, located in a large North American city, over a period of several months. We adopted a simple method to extract images of cars from these videos, for training our object detection models. We performed background subtraction and obtained the bounding boxes of foreground blobs in each video frame. We also

computed the motion direction of each foreground blob using optical flow. Vehicles are then extracted using a simple rule-based classifier which takes into account the size and motion direction of the foreground blobs. The range of acceptable values of the size and motion-direction are manually specified for each camera view. We manually removed the accumulated false positives. This simple procedure enabled us to collect a large number of images of vehicles(about 220000) in a variety of poses and illumination conditions, while requiring minimal supervision. We utilized the motion direction of each foreground blob for categorizing the images of vehicles of each camera viewpoint into a set of clusters. The clustering of images leads to categorization of the training images into a two level hierarchy, where the first level of categorization is according to the camera viewpoint and the second level is based on the motion-direction within each camera viewpoint. Since all the camera viewpoints are distinct, each leaf node of our hierarchy consists of training images of vehicles in a distinct pose. On average, each camera viewpoint has about two clusters, resulting in a total of about 99 clusters(leaf nodes of the hierarchy). These clusters which we refer to as domains cover an extremely diverse collection of vehicles in different poses, lightings and surroundings. Fig. 2.2 shows a few examples of the average images of the 99 training domains.

2.4.1.2 Test set

In order to evaluate our approach with respect to object detection, we annotated a set of 1616 frames collected from 21 out of the same 50 cameras that were

Figure 2.2: A few examples of the training domains presented here by their average images. Note the variations in pose and illumination across domains.

used for collecting the training data. From each camera viewpoint, frames were collected at different times of the day and contain large variations in illumination due to the changes in the direction of sunlight and the resulting reflections and shadows from buildings. Apart from the viewpoint which changes significantly across the cameras, the amount of traffic also varies. On average, each test image contains between one to three vehicles. Figure 2.3 shows a sample frame from each of the resulting 21 test domains.

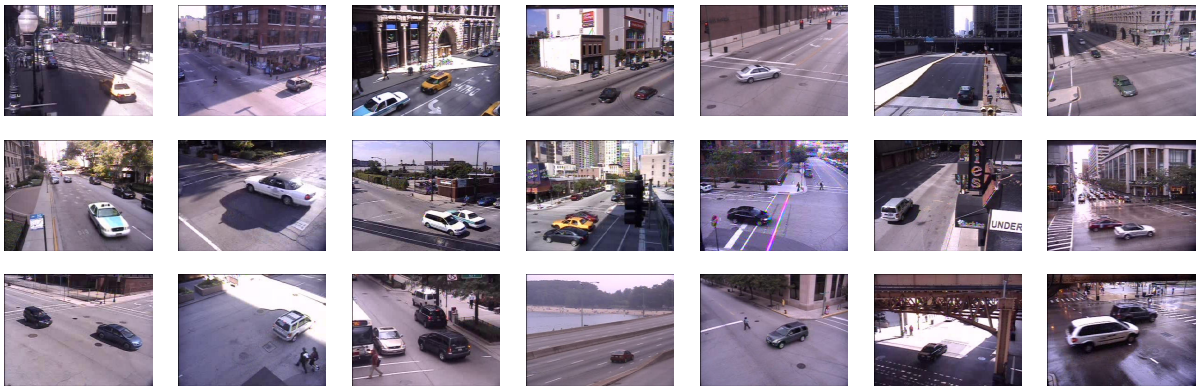


Figure 2.3: A sample frame from each of the 21 test domains

2.4.2 Image Classification

Since our sliding-window detection approach applies a binary classifier at each window location, we first evaluate the performance of TCA by conducting binary classification on our training set of car and background images from 99 domains. For these experiments, the classification performance is measured by average precision. We used HOG features (as implemented in [12]) with a dimension of 55,648 to represent images and an SVM with linear kernel (*LibLinear* [**?**]) as the classifier. For the baseline method, we trained the classifier on images from only one of the 99 domains (source domain), and tested it on all the images of all the domains without any adaptation. For the cases where the source and target domains were the same, the images were split into half for training and testing. We perform the same procedure for our proposed method but instead trained and tested the classifier on feature vectors projected onto the latent subspace learned by TCA, using a linear kernel and $\mu = 1$ for all experiments. The dimension of the subspace (m) was set to 15 for all the experiments. This selection was done based on the results of a set of pilot experiments in which we varied m from 5 to 500 and observed that classification performance started to degrade for $m < 10$. As shown in Fig. 2.4 even with the decreasing number of unlabeled samples of the target domain from 50% down to only 10 random samples, the adapted classifier can still outperform the baseline in majority of cases. Once the number of target samples is decreased to 2, we are no longer able to improve performance.[1] Of particular note is that our proposed

[1]Note that when only 2 or 10 samples are chosen from the target domain, we repeat the experiment 20 and 10 times, respectively, to remove the effects of selecting a few bad target

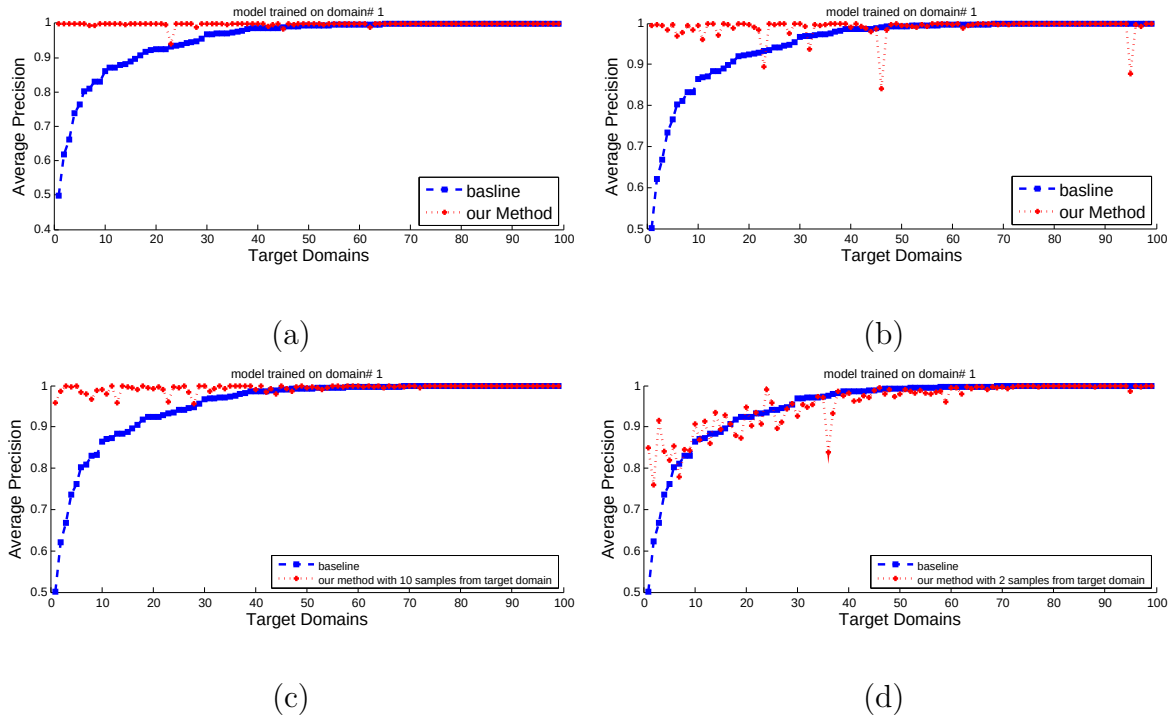


Figure 2.4: A comparison of performance of the baseline classifiers with the adapted ones. To simplify visualization, the results have been sorted by the average precisions of the baseline classifiers. Adaptation by (a) 50%, (b) 10%, (c) 10, and (d) 2 samples from the target domains.

domain adaptation approach is able to drastically improve results even when the baseline performance is close to chance, often improving performance close to an average precision of 1.

2.4.2.1 Comparison with Principal Component Analysis (PCA)

We compare to a number of alternatives based on Principal Component Analysis (PCA) to gain some insight into what contributes to the impressive performance

samples.

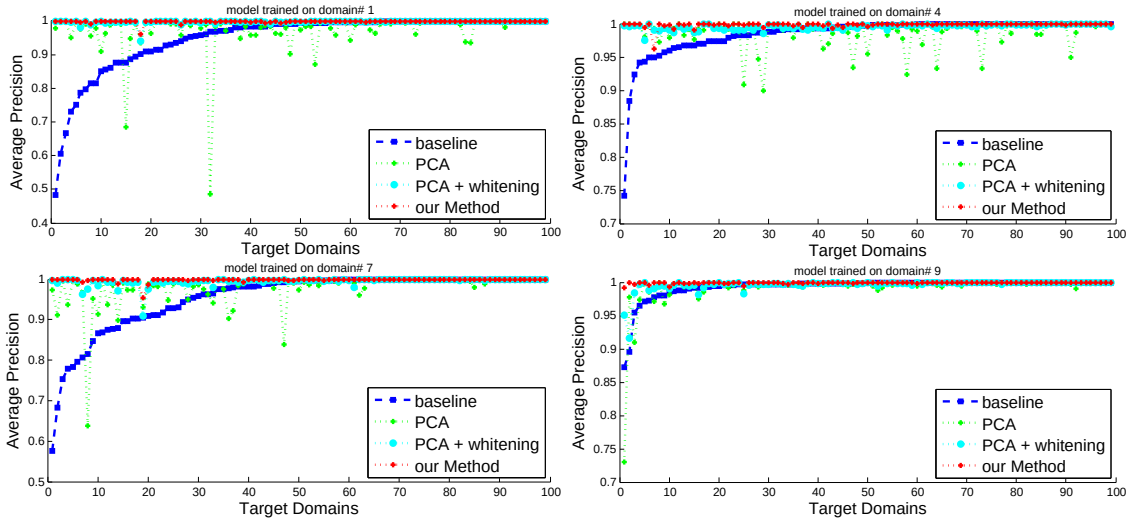


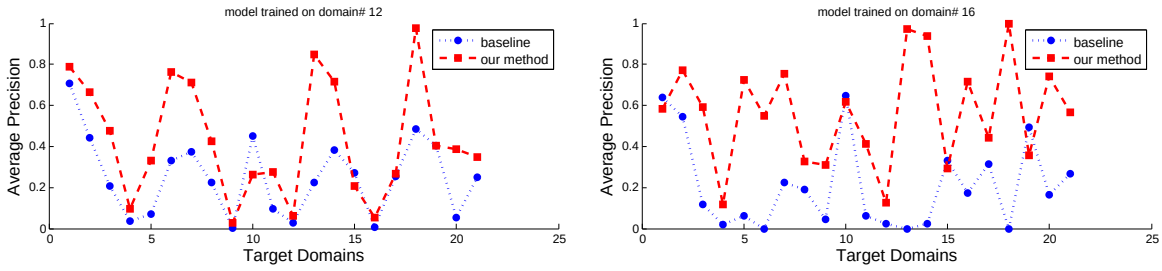
Figure 2.5: Performance comparison of TCA with PCA and PCA plus whitening on four test domains. By performing PCA and then whitening the projected data, we are able to match much (though not all) of the performance improvement of TCA.

in the classification task, especially in the cases where domains are so different from each other that directly applying the classifier trained on the source domain alone yields classification rates close to chance. As described in section 4.3, the effect of TCA is threefold: 1) the means in the RKHS are close to each other, 2) data variance is maximized, and 3) the dimensionality of the input data is reduced prior to classifier training. Since PCA obtains a subspace in which the variance of the projected data is maximized, it produces two of the three effects of TCA. In addition, we note that if the MMD criterion is removed from the TCA optimization, the result is that only variance is maximized (as for PCA), but that the formulation ensures that the *projected* data has unit covariance, whereas the standard implementation of PCA yields orthonormal projection vectors instead. To eliminate this

difference, we *whiten* [**?**] the PCA projected data as a post-processing step to ensure that its covariance is also a unit matrix. Figure 2.5 shows the performance of our baseline approach, TCA, PCA, and PCA with whitening as a post-processing step. Interestingly, performing PCA provides an improvement in performance, but at the cost of some negative transfer in some easy cases. The whitening post-processing step mimicks the results of TCA very closely (although TCA still outperforms), removing most spurious negative transfer cases. While we also performed experiments (not shown) where the means of the source and target distributions were removed to align them exactly, we did not observe an improvement in performance as we did for PCA and PCA with whitening. These preliminary results lead us to believe that it is the combination of dimensionality reduction and whitening which contribute most to the improved adaptation to domain change.

2.4.3 Object Detection

Here we present the results of running the classifiers trained as described in section 2.4.2, at multiple scales and in a sliding window detection fashion on our test data set. For our semi-supervised approach, we use 100 positive and negative samples from the target domain for domain adaptation. We perform experiments by applying each of the 99 source domains to each of the 21 target domains, yielding 99×21 possible testing scenarios. Figure 2.6 shows the performance graphs for two examples of these experiments while table 2.4.3 shows the overall results per each target domain by averaging over all the iterations of the source domains.



(a) (b)

Figure 2.6: Comparison of performance between baseline detectors and the semi-supervised adapted ones. It showcases two examples of the 99 graphs resulted from the 99×21 testing scenarios.

Domain	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
baseline	.37	.46	.44	.09	.24	.21	.37	.12	.07	**.35**	.25	.07	.38	.31	.14	.12	.17	.43	**.30**	.29	.22
ours	**.60**	**.55**	**.63**	**.10**	**.42**	**.49**	**.58**	**.22**	**.1**	.29	**.37**	.07	**.80**	**.67**	.19	**.26**	**.29**	**.83**	.27	**.53**	**.33**

Table 2.1: Comparison of overall performance between baseline detectors and the semi-supervised adapted ones. The numbers in 2nd and 3rd rows show the detection performances (in average precision) for each target domain, averaged over all the iterations of the source domains per each target domain

Ave. Prec.-baseline detector	0.26
Ave. Prec.-detector with semi-supervised adaptation	0.41
Average performance improvement	61.28%

Table 2.2: Averaging of the detection results with semi-supervised adaptation over all the target domains

For our proposed method of unsupervised adaptation, where a balanced set of target samples are obtained automatically, we select a subset of testing scenarios where the performance increase by our semi-supervised adaptation approach, in which the target samples are obtained manually, is most pronounced. We focus on these examples because we expect them to be the most difficult ones for our unsupervised approach, since it relies on first applying the baseline algorithm (which is not adapted to the new domain), and it is for these cases that the baseline algorithm is performing the worst.

A limited number of positive and negative samples from the target domain (1-12 depending on results of the baseline detections) were automatically acquired by running the baseline detector on a few frames of the target domain. The most and least confident predictions from the baseline detectors were used as positive and negative samples of the target domain. As Figure 2.7 shows, while the performance improvement obtained by unsupervised adaption (green curve) is lower than that of semi-supervised method (red curve), it still outperforms the baseline detector (blue curve) in the majority of cases.

The difference in performance between the unsupervised and semi-supervised approaches can be a result of two factors: 1) poor quality positive and negative samples, and 2) fewer positive and negative samples from the target domain. To further investigate whether the degradation is a result of the reduced numbers or the poor quality of the samples from the target domain, we repeat the detection experiments with semi-supervised adaptation but restricted the semi-supervised approach to use the same exact numbers of target samples as the ones obtained in the unsupervised

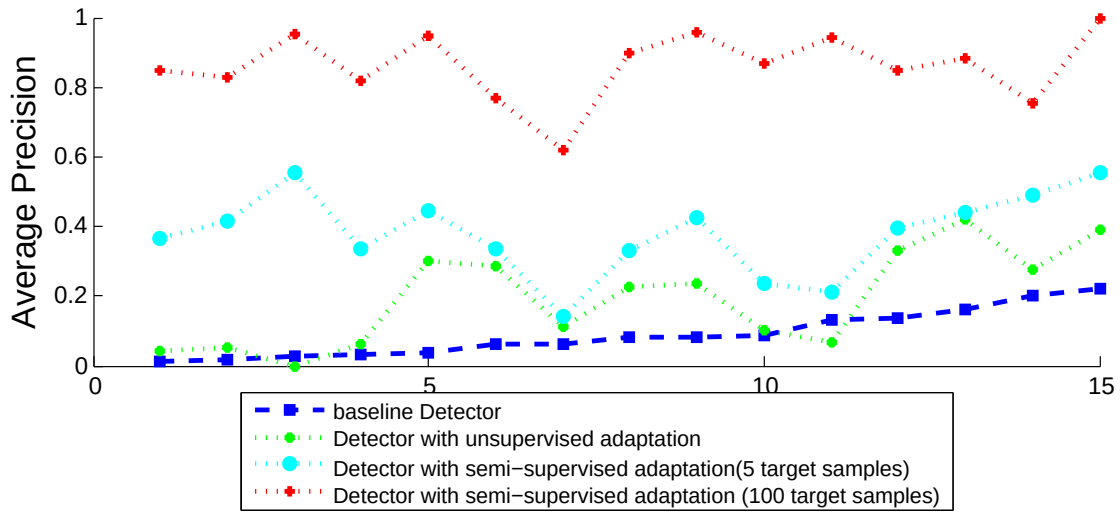


Figure 2.7: Comparison of different approaches of domain adaptation for detection

Ave. Prec.-baseline detector	0.09
Ave. Prec.-detector with unsupervised adaptation (1 to 12 target samples)	0.17
Ave. Prec.-detector with semi-supervised adaptation(1 to 12 target samples)	0.38

Table 2.3: Averaging of the results presented in Figure 2.7 over all the target domains

approach. Comparing the restricted sample semi-supervised approach (cyan curve) to the unsupervised approach (green curve) in Figure 2.7, we observe that when the baseline classifier (blue curve) performs very poorly on the target domain (left side of the graph), the automatically obtained samples are too noisy for our adaptation method to work. However, it is very promising that our unsupervised approach begins to match the performance of the semi-supervised approach at a relatively low baseline average precision.

2.5 Conclusion

We have presented and evaluated an approach for domain-invariant vehicle detection in traffic surveillance videos. Although we have demonstrated the effectiveness of our approach on the task of vehicle detection, it can be potentially applied to other object detection problems. Future work includes extending this model to multiple source domains, multiple object categories, and using class labels from the source or target domains when they are available.

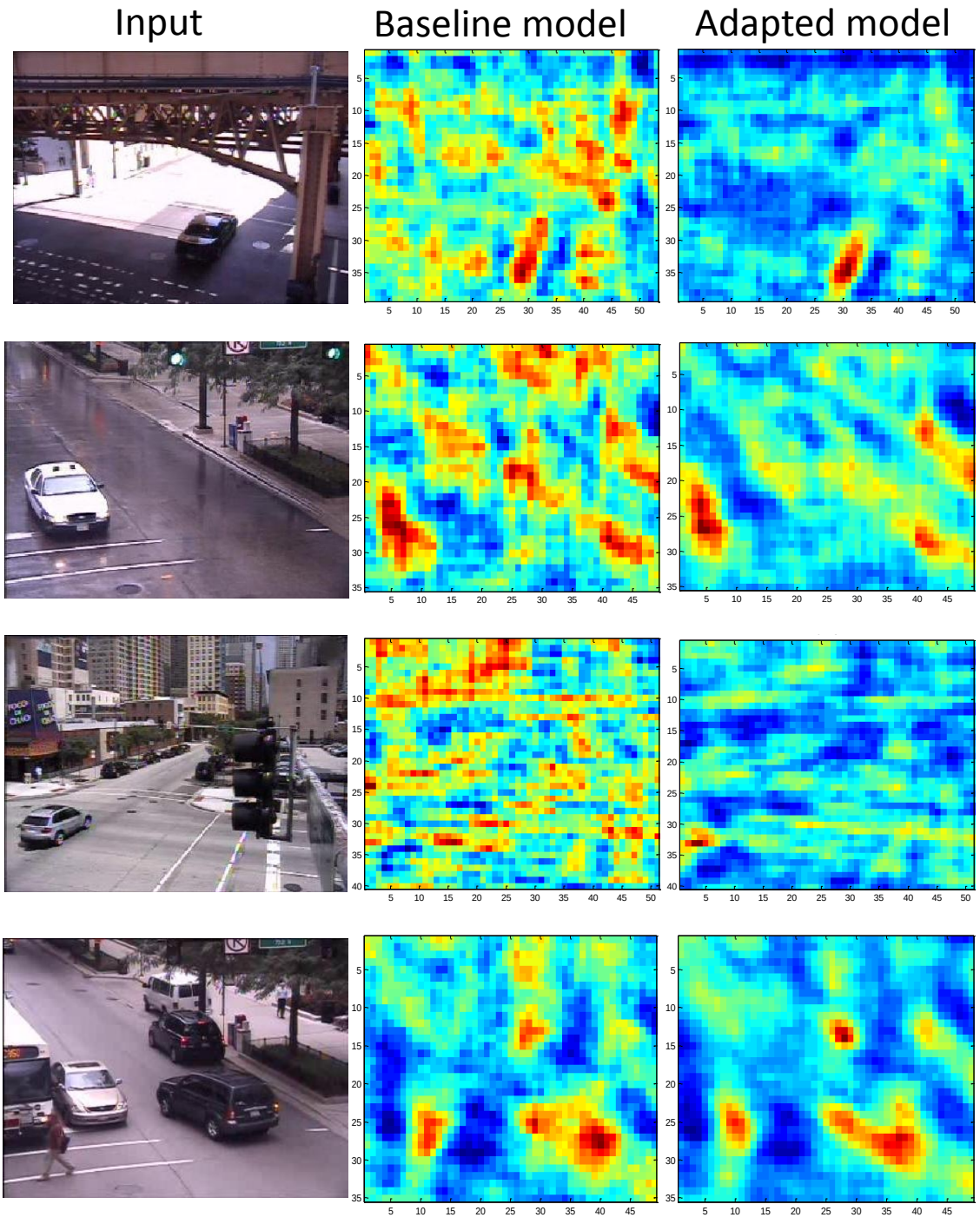


Figure 2.8: Qualitative results demonstrating the improvements obtained by domain adaptation.

Chapter 3

Sampling For Fully Unsupervised Domain Adaptive Object Detection

3.1 Introduction

Discriminative learning algorithms for classification perform well when training and test data are drawn from the same distribution. Often, however, we have sufficient labeled training data from single or multiple source domains but wish to learn a classifier which performs well on a target domain with a different distribution and no labeled training data. In object detection, for example, where the goal is to determine the position and size of all of the objects within one category appearing in a given image, it may be infeasible to collect training data to model the enormous variety of possible combinations of pose, background, resolution, and lighting conditions affecting object appearance. Thus, in realistic applications, we expect to encounter domains at test time for which we have seen little or no training data.

For this reason, domain adaptation techniques have gained considerable attention in computer vision applications with some promising results [6–9, 25]. Previous works have addressed the case in which a few positive and negative examples, with or without their labels, are available from the target domain. Even in the case where labels are not provided for target samples [9], some weak information about class labels is still used in adapting to the new domain simply because the number of positive and negative samples are roughly of the same order, i.e, the two classes are

balanced. Here, in contrast, we focus on the extreme case where no samples from the new domain are given and so they have to be obtained automatically for a *fully unsupervised* adaptation approach.[1]

Unfortunately, fully unsupervised domain adaptation for object detection is a chicken and egg problem: in order to best adapt to the target domain, class labels are needed to balance the set of positive and negative samples; but, class labels can only be obtained automatically with a model that works sufficiently well on the target domain. Here, the main challenge arises from the fact that there are only a limited number of object instances but almost an infinite number of samples of the background class, since any portion of an image that does not contain the object of interest is considered an example of the background class. If training samples were obtained by randomly sampling the image, the positive and negative samples would be highly unbalanced and the model would only adapt to the appearance of the more pervasive background class.

To address this problem, Mirrashed et al. [25] proposed an approach for bootstrapping the target domain with the source trained detector, assuming that the source and target domain are sufficiently close to obtain a more balanced training set from the target domain. However their work is limited to adaptation from only a single source domain. In real world applications, especially with the ever-increasing numbers of public data sets, it would be desirable to make use of classifiers pre-trained on multiple independent datasets (regarding each dataset as a

[1] An adaptive approach where the samples from the target domain are obtained automatically and used without their class labels.

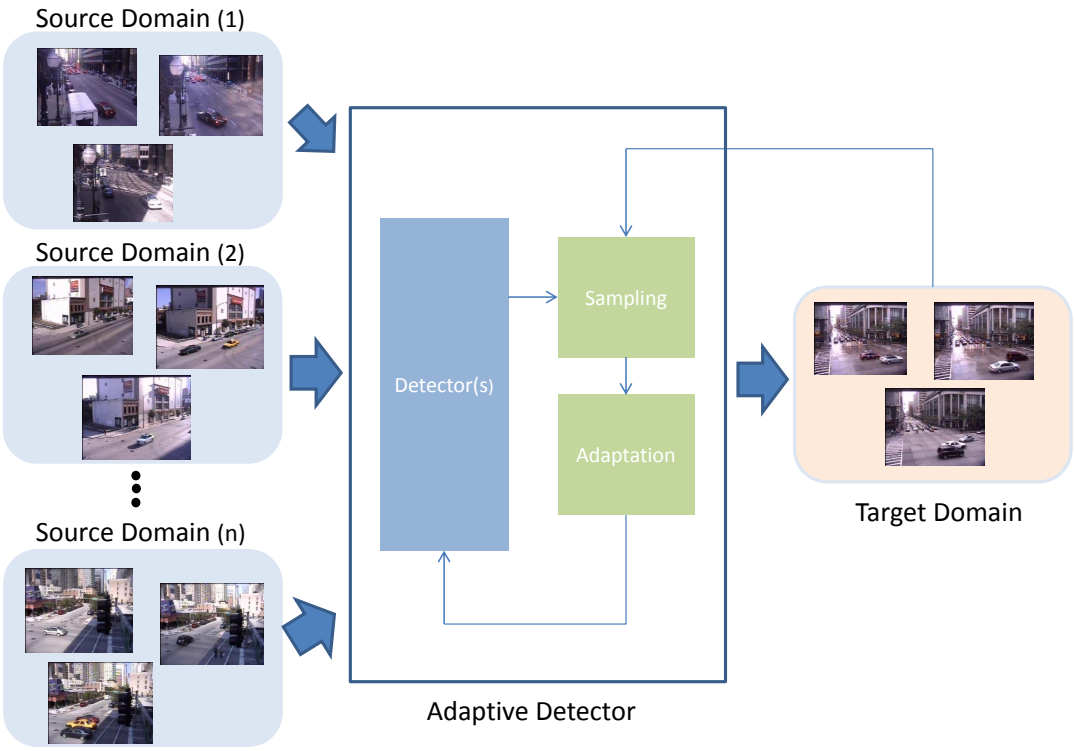


Figure 3.1: Our framework for fully unsupervised adaptive object detection. With detectors trained on training data from multiple source domains, we bootstrap the target domain for positive and negative samples. We then retrain the detectors with training data adapted to samples from the target domain. And the whole process is reiterated.

source domain) when adapting to a new domain. Our target application is the scenario of vehicle detection in urban surveillance videos, where videos are collected from cameras in multiple locations and each camera is treated as a separate domain, representing varying viewpoint, illumination conditions (e.g. sunlight, shadows, reflections, rain, snow) and traffic patterns. Our goal, then, is to leverage all available

domains (camera locations) for which we have labeled data to adapt to a new domain without labeled data.

Traditionally, semi-supervised learning methods are employed to address problems where both labeled and unlabeled data are used to yield improved classification [26]. The authors in [27, 28] proposed a co-training strategy, which unlike the original work of Blum et. al in [29], does not assume independence and redundancy in the feature space. Instead, an ensemble of learners with different inductive biases (e.g. decision trees, naive Bayes, SVMs, etc.) are trained separately on the same labeled data set. They then make predictions on the unlabeled data. If a majority of learners confidently agree on the class of an unlabeled sample, that sample with its predicted label is added to the training data. All learners are retrained on the updated training set. The final prediction is made with a variant of a weighted majority vote among all the learners.

While the intuition behind this procedure is similar to our problem of sampling for adaptive object detection, there are three main complications in our setting compared to that of [27, 28] and semi-supervised approaches in general. First, it is assumed that both labeled and unlabeled data are sampled from an identical underlying distribution which does not hold for our problem setting; second, we have different sets of labeled data, from multiple source domains; and third, we do not necessarily have different types of classifiers with different inductive biases. To address the first problem we use a domain adaption technique, such as TCA [10], to project all the training data into a common space before re-training the classifiers. To address the second and third situations, we show that training the same learning

algorithm, in particular linear SVM's which are the most common classifiers used in computer vision, over different domains or data sets with generally different biases [30, 31] will result in the different inductive biases needed for successful co-training. So in fact, in our case, having labeled data from different source domains with generally different biases substitutes for having different classifiers with different inductive biases.

Another important difference between our adaptive co-learning method and the original version [29] or other variations of co-learning algorithm [27, 28] is that we do not use the new labeled data in retraining the classifiers. Instead, we ignore their labels and use the new samples from the target domain as a representative of the data distribution in the feature space for that domain. We then iteratively learn a common latent subspace (using TCA) underlying source and target domains in which we train and run our classifiers. Therefore, since we do not explicitly require the labels of samples from the target domain, the labeling noise in the machine-labeled predictions will not be detrimental as it would be to noise sensitive supervised learning algorithms.

We compare our approach to two baseline methods: (1) an adaptive approach based on a recently proposed discriminative framework [32] that explicitly estimates a bias for each source domain and approximates an unbiased classifier for an unseen target domain (referred to as *visual world*); and, (2) the adaptive approach in [25] where a single classifier trained on all the labeled data from multiple source domains is used to bootstrap the detection in target domain for adaptation.

The remainder of the paper is organized as follows. We describe the two

	D_{t1}	D_{t2}	D_{t3}	D_{t4}	D_{t5}	D_{t6}	D_{t7}	D_{t8}	D_{t9}	D_{t10}	Average
Single source (no adaptation)	.48	.27	.21	.28	.35	.22	.50	.21	.40	.06	.30
Multi source (no adaptation)	.65	.48	**.31**	.27	.51	.15	.67	.35	.36	.01	.37
Unbiased learning (W_{vw})	.57	.38	.20	.17	.50	.13	.57	.26	.31	.01	.31
Unbiased learning ($W_{vw} + \Delta_{tar}$)	.68	.51	.15	.20	.47	.19	.64	.70	.47	**.22**	.42
Adaptive self-learning	.74	.58	.22	.37	.64	**.24**	**.76**	.70	.54	.19	.50
Adaptive co-learning	**.76**	**.69**	.20	**.46**	**.65**	.22	**.76**	**.77**	**.55**	.21	**.53**

Table 3.1: Vehicle detection results. The detector was trained on labeled data from one of the 4 groups of 3 source domains and tested on one of the 10 target domains. The numbers are average precision. The performance for each target domain is averaged over 4 possible scenarios resulting from the 4 different multi-source domain groups

alternative methods in sections 3.2.1 and 3.2.2 and explain our proposed approach in section 3.3. A detailed description of the experiments and results are given in section3.4, followed by a conclusion in section 4.5.

3.2 Baseline Approaches

The following sections describe the two baseline methods that we employed to address the problem of automatic sampling from the target domain in a fully unsupervised domain adaptive object detection framework. We analyze a setting in which we have plentiful labeled training data drawn either from multiple source domains with uncontrolled various distributions or data sets with naturally different

biases [30, 31] but no labeled training data is available from the target domain of interest.

3.2.1 Unbiased Learning

Similar to our proposed method in 3.3, this approach relies on the assumption that the bias between datasets (or domains) can be identified in the feature space, i.e. the features used to describe the images are rich enough to capture the bias in the data distribution of a domain. With that assumption, Khosla et al in [32] present an algorithm, which is largely based on max-margin learning (SVM), to explicitly model the bias vector in the feature space for each training dataset. Based on the observation that different image datasets are biased samples of a more general dataset (the visual world), they model the weight vector (W_i) learned for a specific dataset (d_i) as a linear additive function of the corresponding bias term (Δ_i) and the weight vector for the visual world (W_{wv}).

For our adaptive detection framework, we employ this method in an iterative mode. In the first iteration, using labeled data from multiple source domain, we learn W_{wv} and bootstrap the (unseen) target domain to obtain high confidence positive and negative samples. Then considering those samples along with their predicted labels as a new "source" domain, we learn the new bias vector for that domain, Δ_{tar}, in the second and following iterations.

3.2.2 Adaptive Self-learning

The most common method of semi-supervised learning is self-learning (also known as self-training or bootstrapping) [26], in which a given model predicts the classes for the unlabeled portion of the data. The automatically labeled examples are then added to the training set, the model is retrained, and the whole process is iterated.

However, in our setting the labeled and unlabeled data are not drawn from the same probability distribution. So similar to [25], we use TCA [10], as a domain adaptation technique to adapt the learned model on training data to the target distribution of interest. In other words, unlike semi-supervised learning algorithm, we do not use the new self-predicted labels in retraining the classifier. We instead use these new predictions as a sample of the data distribution in the target domain to learn a common latent subspace for both the source and target domains. Since TCA does not use class labels for training, the labeling noise of self-learning will not be detrimental (note that most machine learning approaches perform poorly in the presence of labeling noise). Also to prevent a classification mistake from reinforcing itself over iterations, only the most and least confident predictions by the baseline detectors are used as positive and negative samples for the target domain.

3.3 Proposed Method: Adaptive Co-learning

In this approach, instead of training a single classifier on all the labeled data from all the source domains, we train a classifier separately on the training dataset

from each of the source domains. Then, each of the classifiers make predictions separately on the data from the target domain. The final prediction is made with a variant of a weighted majority vote among all the classifiers. Using a vote weighted by a measure of confidence eliminates the possibility that a majority of learners make the same wrong predictions each with very low confidence.

The rest of the process, including adaptation and re-learning of the classifiers, is the same as in 3.2.2. Once again, the most and least confident predictions by these baseline detectors are used correspondingly as positive and negative samples of the target domain for TCA training. Then all the baseline classifier are re-trained and tested on the target domain in the common latent subspace learned by TCA and the whole process is iterated. To compare the confidence values of predictions between these different classifiers, we use the score calibration method described in [33].

The idea behind this strategy is that since multiple classifiers are trained on different training datasets with different biases, they learn diverse models with different inductive biases that can complement each other. The hope is that using the consensus between these hypotheses with different biases would result in more accurate predictions on the target domain with the unknown and possibly different bias.

3.4 Experiments and Results

We evaluated our proposed methods and baseline approaches for vehicle detection on the dataset used in [25]. This dataset consists of videos from 50 different traffic surveillance cameras, located in a large North American city. From each camera viewpoint, frames were collected at different times of the day and contain large variations in illumination due to the changes in the direction of sunlight and the resulting reflections and shadows from buildings. Apart from the viewpoint, which changes significantly across the cameras, the amount and type of traffic also varies. On average each test image contains one to three vehicles. We chose a subset of 22 domains, 12 as source and 10 as target domains, so that there is no overlap between the location of the cameras or the intersection that are looked at between the source and target domains. There are at least 100 annotated frames within each target domain. Dividing the 12 source domains into 4 groups of 3 multi-source domains, we perform experiments on 4×10 possible testing scenarios.

We used HOG features (as implemented by [12]) with a dimension of 55,648 to represent detection windows. We used a sliding-window detector where an SVM with linear kernel (as implemented by *LibLinear* [34]) is applied as the binary classifier to each window location at multiple scales. Classifiers were trained on a a fixed number of 300 positive and 300 negative samples from each of the source domains. Samples were drawn randomly from all source domains and the performance was averaged over 10 iterations. The number of positive samples automatically sampled from target domains for adaptation was set to 50 for all the methods. Following the

result of pilot experiments in [25], the dimension of the subspace in TCA was set to 15 for all the experiments. Performance is measured by average precision, the area under the precision-recall curve.

Table 3.1 summarizes the results of our experiments for each target domain and for each of the baseline and proposed methods. The reported average precision per each target domain is averaged over the 4 testing scenarios with 4 sets of multi-source domains. The last column in the table shows the overall performance averaged over all the target domains.

The first and second rows in table 3.1 show the results of baseline detectors with no adaptation, where the training labeled data comes from either only one out of three source domains (single source) or is accumulated from all three source domains (multi-source). While on average more training data can result in slightly better classification, in 4 out of 10 cases (Dt4, Dt6, Dt9 , and Dt10) that assumption does not hold, likely due to the degree of difference between the training sample distributions.

The last two rows show the results for adaptive self-learning and adaptive co-learning detection, where the performance is increased respectively 35% and 43% over the multi-source baseline method with no adaptation. While our proposed method of adaptive co-learning outperforms adaptive self-learning in the majority of cases, the simplicity and lower computational cost of self-learning can still make it an attractive and competitive choice for time-sensitive applications.

The results of unbiased learning are reflected in rows 3 and 4. Row 3 shows the scenario where an unbiased weight vector (W_{wv}) learned in the first iteration is

used for detection on the unseen target domain,as suggested by [32]. However in our case, it significantly underperforms the multi-source baseline detector (row 2) over all of the 10 test domains. Row 4 shows the results of our extension to the algorithm where a bias term specific to samples obtained from the target domain in the first iteration is learned and used for classification at the second iteration ($W_{vw} + \Delta_{tar}$). This time, while the performance increases with respect to the baseline detectors (first iteration), it still falls short compared to the other two methods of adaptive self-learning and co-learning.

The sampling and re-training iteration was repeated five times for each algorithm. On average the performances changed only by .9% across all methods from iteration 1 to iteration 5. Consequently, the reported results in table 3.1 indicate those from the first iteration for all methods.

3.5 Conclusion

We have presented and evaluated an approach for fully unsupervised domain adaptive vehicle detection from multiple source domains in traffic surveillance videos and showed its superior performance compared to some alternative methods. Although we have demonstrated the effectiveness of our approach on the task of vehicle detection, it can be potentially applied to other classes of object. The generality of our proposed adaptive object detection framework also extends to employment of any domain adaptation technique or supervised learning algorithm.

Chapter 4

Domain Adaptive Classification: A Novel Approach

4.1 Introduction

Discriminative learning algorithms rely on the assumption that models are trained and tested on the data drawn from the same marginal probability distribution. In real world applications, however, this assumption is often violated and results in a significant performance drop. For example, in visual recognition systems, training images are obtained under one set of lighting, background, view point and resolution conditions while the recognizer could be applied to images captured under another set of conditions. In speech recognition, acoustic models trained by one speaker need to be used by another. In natural language processing, part-of-speech taggers, parsers, and document classifiers are trained on carefully annotated training sets, but applied to texts from different genres or styles where there is mismatched distributions of words and their usages.

For these reasons domain adaptation techniques have received considerable attention in machine learning applications. Some previous efforts [6, 23, 35, 36] consider *semi-supervised* domain adaptation where some labeled data from the target domain is available. We focus on the *unsupervised* scenarios when there is no labeled data from the target domain available. Some earlier work in unsupervised domain adaptation assumes that there are discriminative "pivot" features that are common

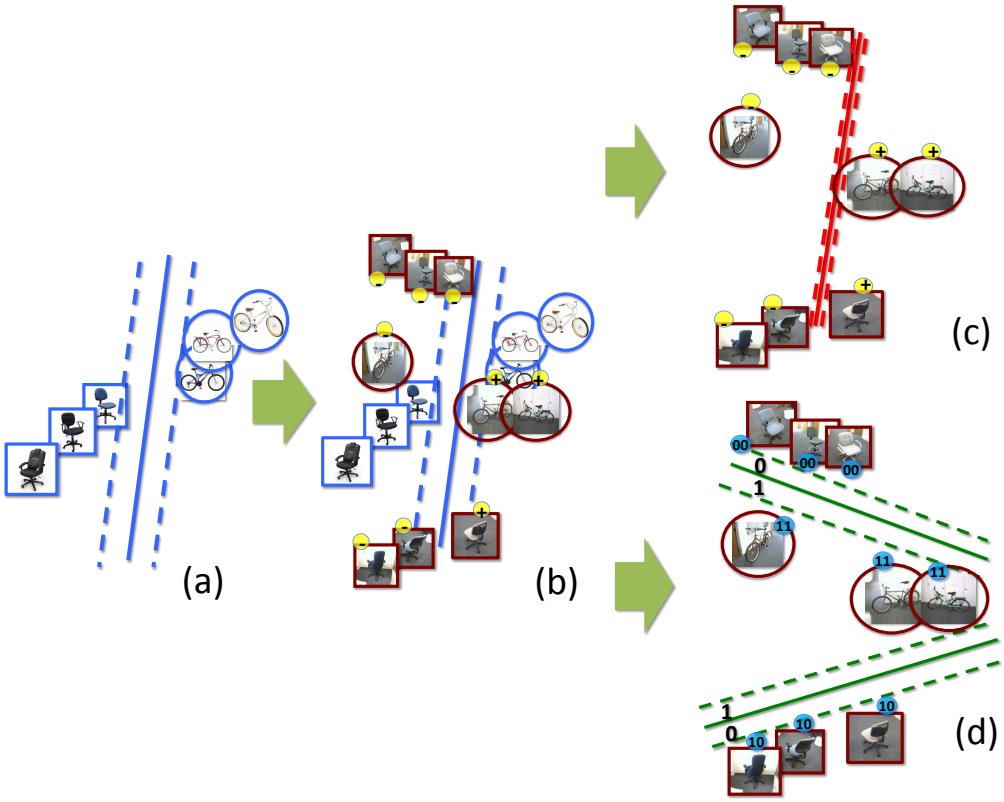


Figure 4.1: This figure summarizes the overall idea of our method. (a) shows a classifier that is trained on data for two categories from the **source** domain (internet images). In (b) we classify the data from the **target** domain (webcam images) using the classifier trained in (a). In (c) and (d) we want to use roughly predicted labels in the target domain to find hyperplanes that are discriminative across categories and also have large margins from samples. (c) illustrates a hyperplane that perfectly separates positive and negative samples but has a small margin. (d) shows two hyperplanes that are not perfectly discriminative but they are binarizing data in the target domain with a large margin. The binarized samples by these two hyperplanes are linearly separable.

to both domains [20,21]. While such methods might work well in language domains, in visual world typical histogram-based image descriptors (visual words) can change significantly across domains. A recent work [37] considers the labeled source data at the instance level to detect a subset of them (landmarks) that could model the distribution of the data in the target domain well. A drawback of such methods is that they do not use the information from all the samples in the source domain available for training the classifier, as they use only landmark points and prune the rest.

Another research theme in domain adaptation is to assume there is an underlying common subspace [9, 10, 38] where the source and target domains have the same (or similar) marginal distributions, and the posterior distributions of the labels are also the same across domains. Hence, in this subspace a classifier trained on the labeled data from the source domain would likely perform well on the target domain. However, transforming data only with the goal of modeling the target domain distribution does not necessarily result in accurate classification. Our goal is to identify a transformation that not only models the distribution of a target domain, but also is discriminative across categories.

We propose a simple yet effective adaptation approach that directly learns a new feature space from the unlabeled target data. This feature space is optimized for classification in the target domain. Motivated by [39], our new feature space, composed of binary attributes, is spanned by max-margin non-orthogonal hyperplanes learned directly on the target domain. Our new binary feature sets are discriminative and at the same time are robust against the change of distributions

of data points in the original feature space between the source and target domains. We refer to this property as *predictability*. The notion of predictability is based on the idea that subtle variations of the data point positions in the original space should not result in different binary codes. In other words, a particular bit in the binary code should be identical (predictable) for all the data samples that are close to each other in the feature space. Figure 4.1 illustrates the essential idea behind our approach.

Our experimental evaluations show that our method significantly outperforms state-of-the-art results on several benchmark datasets which are extensively studied for domain adaptation. In fact in many cases we even reach the upperbound accuracy that is obtained when the classifier is trained and tested on the target domain itself. We also investigate the dataset bias problem, recently studied in [30, 32]. We show that our adaptive classification technique can successfully overcome the bias differences between the datasets in cross-dataset classification tasks. The joint optimization criteria of our model can be solved efficiently and is very easy to implement. Our MATLAB code and data is available at [*removed due to anonymity*].

4.2 Related Work

While it is still not clear how exactly to quantify a domain shift between the train (source) and test (target) data sets, several methods have been devised that show improved performance for cross-domain classification.

In language processing, Daume et al [3] model the data distribution corre-

sponding to source and target domains as a common shared component and a component that is specific to the individual domains. Blitzer et al [20, 21] proposed a structural correspondence learning approach that detects some pivot features that occur frequently and behave similarly in both domains. They used these pivot features to learn an adapted discriminative classifier for the target domain. In visual object recognition, Saenko et al [6] proposed a metric learning approach that uses labeled data in the source and target domains for all or some of the corresponding categories to learn a regularized transformation for mapping between the two domains.

In unsupervised settings where there is no label information available from the target domain, several methods have been recently proposed. Pan et al [10] devise a dimensionality reduction technique that learns an underlying subspace where the difference between the data distributions of the two domains is reduced. However they obtain this subspace by aligning distribution properties that are not class-aware; therefore it does not guarantee that the same class from separate domains will project onto the same coordinates in the shared subspace. Gopalan et al [9] take an incremental learning approach, following a geodesic path between the two domains modeled as points on a Grassmann manifold. Gong et al [38] advance this idea by considering a kernel-based approach; i.e. they integrate an infinite number of subspaces on that geodesic path rather than sampling a finite number of them. In [37], Gong et al, however, consider only a subset of training data in the source domain for their geodesic flow kernel approach; the ones that are distributed similarly to the target domain, .

In [30, 32], the varying data distribution between the train and test sets have been studied under the "dataset bias" They point out how existence of various types of bias, such as capture and negative set bias, between datasets can hurt visual object categorization. This is a similar problem to domain adaptation where each dataset can be considered as a domain.

Another set of related methods are those that use binary code descriptors for recognition. Recent method shows that even with a few bits of binary descriptor one can reach state-of-the-art performance in object recognition. Gong et al [40] optimized to find a rotation of data that minimizes binary quantization error. They used CCA in order to leverage labels' information. In [39] they proposed a technique to map the data into a hamming space where each bit is predictable from neighboring visual data. At the same time the binary code of an image needs to be discriminative across the categories. Our method is motivated by their approach. We are also looking for a set of discriminative binary codes but in our problem data comes from different domains with mismatched distributions in the feature space. In section 4.3 we explain how our method solves this problem by a joint optimization over solving a linear SVM and finding a binary projection matrix.

4.3 Proposed Method

Our goal is to identify useful information for classification in the target domain. We represent this information by a number of hyperplanes in the feature space created using data from the target domain. We call each of these hyperplanes, an

attribute. These attributes must be discriminative across categories and predictable across domains. We explain our notion of predictability in section 4.3.2. We use these attributes as feature descriptors and train a classifier on the labeled data in the source domain. When we apply this classifier to the target domain, we achieve a much higher accuracy rate than the baseline classifier for the target data. The baseline is simply a classifier trained on the source data in the original feature space.

Each attribute is a hyperplane in feature space; it divides the space into two subspaces. We assign a binary value to each instance by its "sidedness" with respect to the hyperplane. We construct a K-bit binary code for each image using K hyperplanes. To produce consistent binary codes across domains, each binary value needs to be predictable from instances across domains. Predictability is the key to the performance of our method. We also want the attributes to be discriminative across categories. i.e. the K-bit attribute descriptors of the samples from same category should be similar to each other and different from the other categories.

4.3.1 Problem Description

First we explain the notations that we use throughout this section. Superscripts $\mathcal{S}$ and $\mathcal{T}$ indicates source and target domains respectively and superscript T indicates matrix transpose. x_i is a d-dimensional column vector that represents the i^{th} instance feature and X is a matrix created by concatenation of all x_i's. l_i is the category label of the i^{th} instance. Without loss of generality, we assume that $l_i \in \{1, -1\}$. A is a $d \times K$ matrix whose k^{th} column, a_k, is the normal vector of a

hyperplane (attribute) in the original feature space. w is the K-dimensional normal vector of a classifier that classifies one category from the others in the binary attribute space. sgn(.) is the sign function

We want to directly optimize for better classification in the target domain. Therefore, we need to find K hyperplanes, a_k, in the target domain such that when we use $\text{sgn}(A^T x_i)$ as a new feature space, and learn a classifier on source data projected onto this space, we can predict the class labels of the data in the target domain. Of course we do not have the class labels for the data in the target domain $l_i^{\mathcal{T}}$. In order to train the classifier and attributes (hyperplanes) in target domain, we add a constraint to our optimization to force the $l_i^{\mathcal{T}}$ to be predictable from the source domain's classifier. More specifically, our optimization is a combination of two max-margin SVM-like classifiers that are interconnected via the attribute mapping matrix A.

$$\min_{A,w^{\mathcal{S}},w^{\mathcal{T}},l^{\mathcal{T}},\xi^{\mathcal{S}},\xi^{\mathcal{T}}} \|w^{\mathcal{S}}\| + \|w^{\mathcal{T}}\| + C_1 \sum \xi^{\mathcal{S}} + C_2 \sum \xi^{\mathcal{T}}$$

$$s.t.$$

$$\begin{aligned} & l_i^{\mathcal{S}}({w^{\mathcal{S}}}^T \text{sgn}(A^T x_i^{\mathcal{S}})) > 1 - \xi_i^{\mathcal{S}}, \\ & l_j^{\mathcal{T}}({w^{\mathcal{T}}}^T \text{sgn}(A^T x_j^{\mathcal{T}})) > 1 - \xi_j^{\mathcal{T}}, \\ & l_j^{\mathcal{T}} = \text{sgn}({w^{\mathcal{S}}}^T \text{sgn}(A^T x_j^{\mathcal{T}})), \end{aligned} \tag{4.1}$$

It is not straightforward to solve the optimization in Eq 4.1 because matrix A in the constraints requires a combinatorial search for the optimal solution. But if we constrain the possible solutions for A, then we can solve it efficiently. As we will explain in section 4.3.2, we do this by forcing predictability constraints on all the

a_k vectors.

4.3.2 Predictability

In different domains data appears with different distributions. Consider a picture of a car taken by a mobile phone's camera and the same picture taken from a professional high quality camera. Due to differences in the two photo capturing systems such as resolution, the two images will be mapped to two different points in visual feature space despite being the same object from the same category. For better classification, however, ideally we would like to create a feature space that would map these two images onto the same or nearby points. In other words, we would like to have a class-compact and domain-invariant feature space for these images. For a sample, an attribute is a binary value derived from a hyperplane in the raw feature space. If this hyperplane produces different binary values for samples that are nearby to each other, then we say that the values coming from this hyperplane are not predictable. Therefore, this attribute would not be robust against the variations of samples from different domains in the raw feature space.

Predictability is the ability to predict the value of a given bit of a sample by looking at the corresponding bit of the nearest neighbors of that sample. For example, if the k^{th} bit in most of the nearest neighbors of a sample is **1** then we can infer that the k^{th} bit of that sample would also be **1**.

Consider the situation where a hyperplane crosses a dense area of samples. There would be many samples in proximity to each other that are assigned dif-

ferent binary values. The binary values obtained by this hyperplane are thus not *predictable*. The binary values obtained by a hyperplane are *predictable* when the hyperplane has large margin from samples. There are several methods that try to model the transfer of distribution between domains [9, 10, 38]. All of these methods rely on discovering some orthogonal basis of the feature space such as principle components. However these orthogonal basis are not appropriate as hyperplanes for attributes. Figure 4.2 illustrates a demonstration of the hyperplanes defined by orthogonal basis (PCA) in green lines. Note that PCA hyperplanes cross dense areas of samples. If we binarize the samples by the PCA hyperplanes, then samples in the red circle will have different binary codes even though they are nearby each other and strongly clustered. The hyperplanes that are shown in orange are our predictable attributes, which enforce the large margins from samples.

To enforce the predictability constraint on binary values of attributes, we regulate our optimization by adding a max-margin constraint on A as follows:

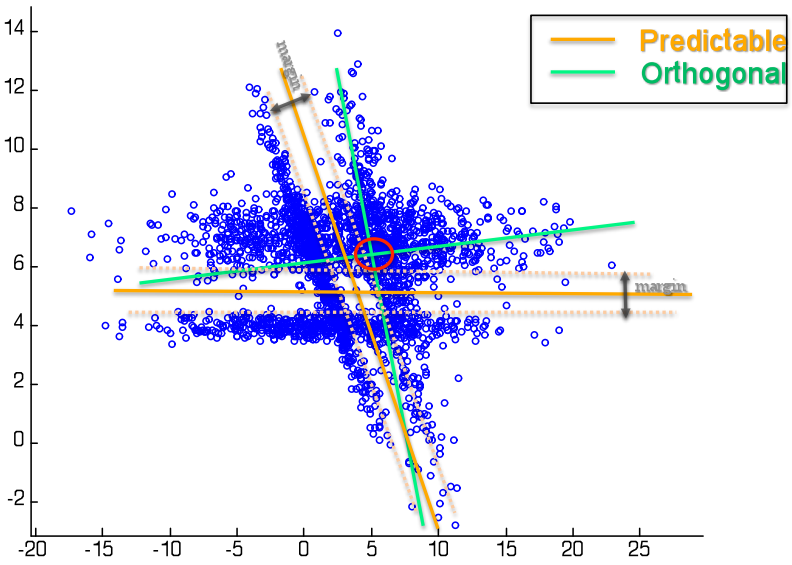


Figure 4.2: Comparison of predictable hyperplanes and orthogonal hyperplanes. Note that the hyperplanes learned by large margin divide the space, avoiding the fragmentation of sample distributions by the help of *predictability* constraints implemented by max-margin regularization.

$$\min_{A,w^{\mathcal{S}},w^{\mathcal{T}},l^{\mathcal{T}},\xi^{\mathcal{S}},\xi^{\mathcal{T}},\xi^{A}} \|w^{\mathcal{S}}\| + \|w^{\mathcal{T}}\| + \|A\|_F + C_1 \sum \xi^{\mathcal{S}} + C_2 \sum \xi^{\mathcal{T}} + C_3 \sum \xi^{A}$$

$$s.t.$$

$$\begin{aligned} & l_i^{\mathcal{S}}(w^{\mathcal{S}T}\mathrm{sgn}(A^T x_i^{\mathcal{S}})) > 1 - \xi_i^{\mathcal{S}}, \\ & l_j^{\mathcal{T}}(w^{\mathcal{T}T}\mathrm{sgn}(A^T x_j^{\mathcal{T}})) > 1 - \xi_j^{\mathcal{T}}, \\ & l_j^{\mathcal{T}} = \mathrm{sgn}(w^{\mathcal{S}T}\mathrm{sgn}(A^T x_j^{\mathcal{T}})), \\ & b_{kj} = \mathrm{sgn}(a_k^T x_j^{\mathcal{T}}), \\ & b_{kj}(a_k^T x_j^{\mathcal{T}}) > 1 - \xi_{kj}^{A}, \end{aligned} \tag{4.2}$$

Where b_{kj} is the binary value of the k^{th} bit (attribute) of the j^{th} sample in the

target domain. In fact, each attribute is a max-margin classifier in feature space and b_{jk} is the label of the j^{th} sample when classified by the k^{th} attribute classifier. This optimization can be easily conducted using block coordinate descent. If we fix $w^{\mathcal{T}}$ and A, then solving the optimization for $w^{\mathcal{S}}$ is a simple linear SVM in the attribute space. Accordingly, once we determine $w^{\mathcal{S}}$, we can compute $l^{\mathcal{T}}$. Then solving for $w^{\mathcal{T}}$ and A is a standard attribute discovery problem in the target domain and can be solved using the method (DBC) in [39]. We iterate over these two steps: finding $w^{\mathcal{S}}$, and then solving for $w^{\mathcal{T}}$ and A. We don't know how to obtain a good initialization for $w^{\mathcal{T}}$ and A, but luckily we don't necessarily need them. We only need to have an initialization for $l^{\mathcal{T}}$ so that we can solve the attribute discovery problem for A and $w^{\mathcal{T}}$. An intuitive way to initialize $l^{\mathcal{T}}$ is to learn a classifier on the labeled data in the source domain, $x^{\mathcal{S}}$ and $l^{\mathcal{S}}$, and then apply it on $x^{\mathcal{T}}$, the data in the target domain. Algorithm 1 summarizes our method.

Algorithm 1 Adaptive Classification

Input: $X^{\mathcal{S}}$, $l^{\mathcal{S}}$, $X^{\mathcal{T}}$, K.

Output: $l^{\mathcal{T}}$, A, $w^{\mathcal{S}}$, $w^{\mathcal{T}}$.

1: $\theta \leftarrow$ Learn a classifier on $X^{\mathcal{S}}$ and $l^{\mathcal{S}}$

2: $l^{\mathcal{T}} \leftarrow$ Test the classifier θ on $X^{\mathcal{T}}$ //Initialization for $l^{\mathcal{T}}$

3: **repeat**

4: $\quad w^{\mathcal{T}}, A \leftarrow \text{DBC}(X^{\mathcal{T}}, l^{\mathcal{T}}, K)$

5: $\quad w^{\mathcal{S}} \leftarrow$ Learn a linear SVM on $\text{sgn}(A^T X^{\mathcal{S}})$ and $l^{\mathcal{S}}$

6: $\quad l^{\mathcal{T}} \leftarrow \text{sgn}({w^{\mathcal{S}}}^T \text{sgn}(A^T X^{\mathcal{T}}))$

7: **until** convergence on $l^{\mathcal{T}}$

4.4 Experiments and Results

We first evaluate our method on two benchmark datasets extensively used for domain adaptation in the contexts of object recognition [6,7,9,37,38] and sentiment analysis [9,21,37]. We compare our method to several previously published domain adaptation methods. Empirical results show that our method not only outperforms all prior techniques in almost all cases, but also in many cases we achieve the same-domain classification, the upper bound, accuracy, i.e. when the classifier is trained and tested on the target domain itself.

Furthermore, we test the performance of our method on an inductive setting of unsupervised domain adaptation. In the inductive setting we test our adapted classifier on a set of unseen and unlabeled instances from target domain- separate from the target domain data used to learn the attribute model. And finally, we investigate the dataset bias problem, recently studied in [30,32], and we show that our adaptive classification technique can successfully overcome the bias differences in both single and multiple source domains scenarios.

4.4.1 Cross-Domain Object Recognition

First, we evaluate our method for cross-domain object recognition. We followed the setup of [37,38] which use the three datasets of object images studied in [6,7,9]: Amazon ($\boldsymbol{A}$) (images downloaded from online merchants), Webcam ($\boldsymbol{W}$) (low-resolution images taken by a web camera), and DSLR ($\boldsymbol{D}$) (high-resolution images taken by a digital SLR camera) plus Caltech-256 ($\boldsymbol{C}$) [41]as a fourth dataset.

Each dataset is regarded as a domain. The domain shift is caused by factors including change in resolution, pose, lighting, background, etc. The experiments are conducted on 10 object classes common to all 4 datasets. There are 2533 images in total and the number of images per class ranges from 15 (in DSLR) to 30 (Webcam), and up to 100 (Caltech and Amazon). We used the publicly available feature sets [1], and the same protocol as in all the previous work were used for representing images: The 64-dimensional SURF features [42] were extracted from the images, and a codebook of size 800 was generated by k-means clustering on a random subset of Amazon database. Then, the images from all domains are represented by an 800-bin normalized histograms corresponding to the codebook.

We report the results of our evaluation on all 12 pairs of source and target domains and compare it with methods as reported in [37] (table 4.4.1). The other methods include transfer component analysis (tca) [10], geodesic flow sampling (gfs) [9], Geodesic Flow Kernel (gfk) [38], structural correspondence learning (scl) [20], kernel mean matching (kmm) [43], and a metric learning method (metric) [6] for semi-supervised domain adaptation, where label information (1 instance per category) from the target domains is used. We also report a baseline results of no adaptation, where we train a kernel SVM on labeled data from the source domain in the original feature space. A linear kernel function is used for the SVM. For each pair of domains the performance is measured by classification accuracy (number of correctly classified instances over total test data from target).

As explained in [37], due to its small number of samples (157 for all 10 cate-

[1] http://www-scf.usc.edu/ boqinggo/da.html

gories), DSLR was not used as a source domain and so the results for other methods have been reported only for 9 out of 12 pairings. Table 4.4.1 shows that our method outperforms all the previous methods in all cases except when DSLR is the target domain. The culprit is the small number of samples in DSLR being insufficient for training the attribute model. In all our experiments in this paper, we used a binary attribute space with 256 dimensions. To learn each attribute hyperplane we used linear SVM coupled with kernel mapping. None of the hyperparameters for SVM classifiers and DBC model were tuned. They were all left at their default values. One might get better results by tuning these parameters.

4.4.2 Cross-Domain Sentiment Analysis

Next, we consider the task of cross-domain sentiment analysis in text [21]. Again we compare the performance of our approach with the same set of domain adaptation methods as reported in [37] and listed in 4.4.1. We used the dataset in [21] which includes product reviews from amazon.com for four different products: books ($\boldsymbol{B}$), DVD ($\boldsymbol{D}$), electronics ($\boldsymbol{E}$), and kitchen appliances ($\boldsymbol{K}$). Each product is considered as a domain. Each review has a rating from 0 to 5, a reviewer name and location, review text, among others. Reviews with rating higher than 3 were classified as positive, and those less than 3 were classified negative. The goal is to determine whether the process of learning positive/ negative reviews from one domain, is applicable to another domain. We used the publicly available feature sets for the collection in which bag-of-words features are used and the dimensionality of

	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$	$W \to A$	$W \to C$	$W \to D$	$D \to W$	$D \to C$	$D \to A$
No Adaptation	41.7	41.4	34.2	51.8	54.1	46.8	31.1	31.5	70.7	38.2	34.6	38.2
TCA [10]	35.0	36.3	27.8	41.4	45.2	32.5	24.2	22.5	80.2	N/A	N/A	N/A
GFS [9]	39.2	36.3	33.6	43.6	40.8	36.3	33.5	30.9	75.7	N/A	N/A	N/A
GFK [38]	42.2	42.7	40.7	44.5	43.3	44.7	31.8	30.8	75.6	N/A	N/A	N/A
SCL [20]	42.3	36.9	34.9	49.3	42.0	39.3	34.7	32.5	83.4	N/A	N/A	N/A
KMM [43]	42.2	42.7	42.4	48.3	53.5	45.8	31.9	29.0	72.0	N/A	N/A	N/A
Metric [6]	42.4	42.9	49.8	46.6	47.6	42.8	38.6	33.0	**87.1**	N/A	N/A	N/A
Landmark [37]	45.5	47.1	46.1	56.7	**57.3**	49.5	40.2	35.4	75.2	N/A	N/A	N/A
Ours	**75.15**	**51.59**	**52.54**	**91.54**	49.68	**60.34**	**74.22**	**53.34**	76.43	**81.02**	**56.03**	**72.03**

Table 4.1: **Cross-domain Object recognition:** accuracies for all 12 pairs of source and target domains are reported (C: Caltech, A: Amazon, W: Webcam, and D: DSLR). Due to its small number of samples, DSLR was not used as a source domain by the other methods and so their results have been reported only for 9 pairings. Our method significantly outperforms all the previous methods except for 2 out of 3 cases when DSLR , whose number of samples are insufficient for training our attribute model, is the target domain.

data is reduced to 400 (the 400 words with the largest mutual information with the labels).

Table 4.2 shows the results; our method outperforms all the previous methods by a relatively large margin (25% average improvement over baseline and 19% over state-of-art).

	$K \to D$	$D \to B$	$B \to E$	$E \to K$
No Adaptation	72.7	73.4	73	81.4
TCA [10]	60.4	61.4	61.3	68.7
GFS [9]	67.9	68.6	66.9	75.1
GFK [38]	69.0	71.3	68.4	78.2
SCL [20]	72.8	76.2	75.0	82.9
KMM [43]	72.2	78.6	76.9	83.5
Metric [6]	70.6	72.0	72.2	77.1
Landmark [37]	75.1	79.0	78.5	83.4
Ours	**92.1**	**93.15**	**94.94**	**95.65**

Table 4.2: **Cross-Domain Sentiment Classification:** accuracies for 4 pairs of source and target domains are reported. K: kitchen, D: dvd, B: books, E: electronics. Our method outperforms all the previous methods.

4.4.3 Comparing to Same-Domain Classification

How accurate are the domain adapted classifiers compared to classifiers trained on labeled data from the target domain? To investigate this, we divide each dataset into two equal parts, one of which is used for training and the other for testing. This balances the number of samples used for within domain training and testing and cross domain adaptive training and testing.

Table 4.3 shows the results for all 16 pairs of domains in sentiment dataset and 4 pairs of domains from object recognition datasets. In the latter we could use

	K	E	B	D
K	97.9	97.4	96.6	95.2
E	97.9	97.4	96.5	95.4
B	97.8	97.4	96.6	95.3
D	97.7	97.3	96.6	95.4

	C	A
C	75.6	92.2
A	74.4	92.2

Table 4.3: **Comparing to Same-Domain Classification :** (Left) Accuracies for all 16 pairs of source and target domains in sentiment dataset are reported in the left table. K: kitchen, D: dvd, B: books, E: electronics. (Right) Accuracies for 4 pairs of source and target domains are reported. C: Caltech, A: Amazon. Rows and columns correspond to source and target domains respectively. Our method reaches the upper bound accuracies (diagonal) for cross-domain classification.

only the two domains (Caltech, Amazon) that had sufficient number of samples to be divided into two groups (train/test)

The rows correspond to the source domains and columns to the target domains. We can see how on this data set our adaptive classification method reaches the upper bound performance in all cases.

4.4.4 Transductive vs Inductive Cross-Domain Classification

In the previous experiments, we follow the same protocol as [37, 38] for a fair comparison. So, we had access to all the samples in the target domain at training time and our goal was to predict their labels. This is a transductive learning problem

except that the test data was drawn from a different domain. In an inductive setting we do not have access to the test data at training time. So, to create an inductive setting for the unsupervised domain adaptation problem, we make only a fraction of the data from the target domain accessible at training time for learning our adaptive feature space. The rest, which we refer to as out-of-sample data from the target domain, is set aside for inductive classification tests.

Table 4.4, reports the results for this experiment on the sentiment data set where we have balanced number of samples across domains. Our adaptive classification results on out-of-sample data still outperform the corresponding performance for in-sample data by other methods in 3 out of 4 cases. Nevertheless, it does show a drop in performance compared with our own in-sample results. As we show later, however, this is not necessarily the case. In section 4.4.5 we show how our out-of-sample results reasonably perform compared to the corresponding in-sample ones. (table 4.5)

4.4.5 Dataset Bias

Most of the images in the datasets studied in sections 4.4.1 and 4.4.2 contain the object of interest centered and cropped on a mostly uniform background. To evaluate our method on a wider range of images with unconstrained backgrounds and clutter, as well as to see how it deals with the data set bias problem addressed in [30, 32], we extend our cross-domain object recognition experiments to four widely used computer vision datasets- Pascal2007 [44], SUN09 [45], LabelMe [46], Caltech101

		$K \rightarrow D$	$D \rightarrow B$	$B \rightarrow E$	$E \rightarrow K$
In-samples	No Adaptation	72.7	77.1	75.2	82.8
	Adapted (Ours)	**97.2**	**96.6**	**98.0**	**98.1**
Out-samples	No Adaptation	70.5	75.6	74.4	82.8
	Adapted (Ours)	**77.5**	**76.9**	**80.7**	**84.4**

Table 4.4: **Transductive vs Inductive Cross-domain Classification:** The first two rows show the results in transductive setting where all the data from the target domains are accessible during training. The last two rows show the results in inductive setting where we test our classifier only on a subset of data in the target domain that was not accessible during training time

[41].

We follow the same protocol as [32], where they run experiments on five common object categories- "bird", "car", "chair", "dog", and "person". We used the publicly available feature sets for this data [2]. Using a bag-of-words representation, Grayscale SIFT descriptors [47] at multiple patch sizes of 8, 12, 16, 24 and 30 with a grid spacing of 4 were extracted. Using k-means clustering on randomly sampled descriptors from the training set of all datasets, a codebook of size 256 is constructed. The baseline SVM is implemented using Liblinear [48] coupled with a Gaussian kernel mapping function [49]. The results are evaluated by average precision (AP).

Table4.5 reports the results of our cross-dataset classification in both the in-

[2] http://undoingbias.csail.mit.edu/features.tar

		Caltech	LabelMe	Pascal07	SUN09
In-samples	No Adaptation	78.7	71.6	76.1	70.9
	Adapted (Ours)	**99.4**	**92.7**	**92.6**	**94.9**
Out-samples	No Adaptation	79.1	75.1	75.0	74.2
	Adapted (Ours)	**94.6**	**86.4**	**90.1**	**87.8**

Table 4.5: **Cross-Dataset Object Recognition:** The 4 rightmost columns show the classification results for when we hold out one dataset as the target domain and use the other 3 as source domains, in both the inductive (first two rows) and transductive (last two rows) settings. The reported results are averaged over 5 categories of objects.

ductive (in-sample) and transductive (out-of-sample) settings. Each column of the table correspond to the situation where one dataset is considered as the target domain and all the remaining datasets are considered as the source domain (multi-source domain). These result shows that our approach is robust against varying biases when the training data comes from multiple datasets and the test data comes from another one. The reported results are averaged over all 5 categories. The average performance improvement by our adaptive method over the baseline (no adaptation) is 28% for out-of-sample data and 18% for in-sample data. The only related work that we are aware of that has performed theses cross-dataset classifications experiments with the same settings is [32] where they report an average performance improvement of only 2.5% across all datasets and all categories.

4.4.6 Effectiveness of Predictability

Now, we show the importance of the predictability of attributes by quantitative and qualitative evaluations.

Quantitative evaluation: To see how learning binary attributes by itself is contributing to our performance increase, we ignore the adaptation and use the attribute features learned only from the source domain. In this setting we learn the binary attribute space from the labeled data in the source domain, project the data from both source and target domain onto this space where we train a classifier on the source data and test it on the target data. We then compare the results with corresponding ones by our adapted model. We used the same experiment setup in section 4.4.5 for this evaluation (Figure 4.3).

Qualitative evaluation: Here we show that the discovered attributes are consistent across domains. We pick an attribute classifier learned by our method, then we find images (from both source and target) that are most positively and negatively confident when classified by this attribute classifier. In Figure 4.4 the left two rows use DSLR as source domain and Amazon as target. Similarly, the right two rows use Amazon as source and Webcam as target. The green arrow represent an attribute classifier which is trained on target domain. The dashed part of the arrow illustrates that the same hyperplane which is trained in target domain is applied in the source domain. Images on the right side of the green arrow are the most positive and on the left side are the most negative one. As can be seen in both cases the attribute classifiers are consistent across domains. In the first case, the

attribute consistently separates round shapes from dark-volumed shapes in both domains and in the second case, the attribute consistently discriminates between objects with keypad and objects with dark-volumed shape. This observation is consistent with our intuition of predictability in our optimization.

4.5 Conclusion

We introduce a method for adaptive classification when the train and test data come from different domains. Our method is based on learning a predictable binary code that captures the structural information of the data distribution in the target domain itself. These binary codes prove to be highly effective for classification since they are optimized to be robust against the variations of data distribution in the feature space, while they maintain their discriminative properties. We designed a joint optimization that learns both binary projection matrix and the classifier and is very easy to implement.

Our empirical evaluations demonstrate an impressive and consistent performance gain by our method on standard benchmarks previously studied for domain adaptation problem. In many cases our domain adaptive method could reach the gold standard accuracies; i.e. when the classifier is trained on the labeled from the target domain itself. We also show how our method can successfully generalize over the bias variations among widely-used computer vision datasets.

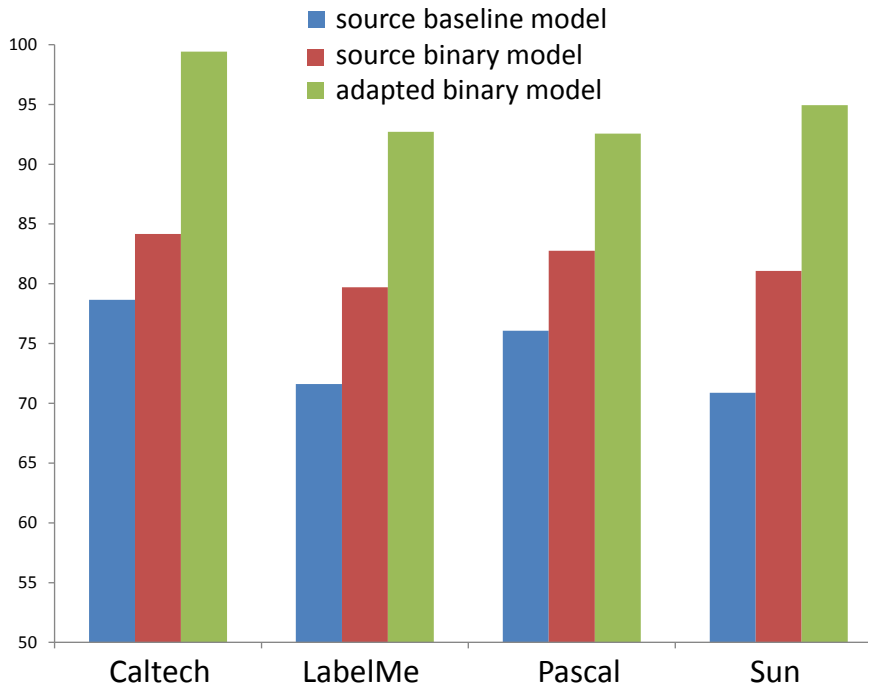


Figure 4.3: **Quantitative Evaluation of Predictability:** The blue bars show the classification accuracies when the classifier is simply trained on the data from the source domain in original feature space (baseline). The red bars show the results when the classifier is trained in a binary attribute space learned from the data in the source domain (source binary). The green bars show the results of our adapted model when the classifier is trained on labeled source data in a binary attribute space learned in the target domain (adapted binary). In average the source binary model is increasing the performance by 10% over the baseline while the adapted binary model does that by 28%

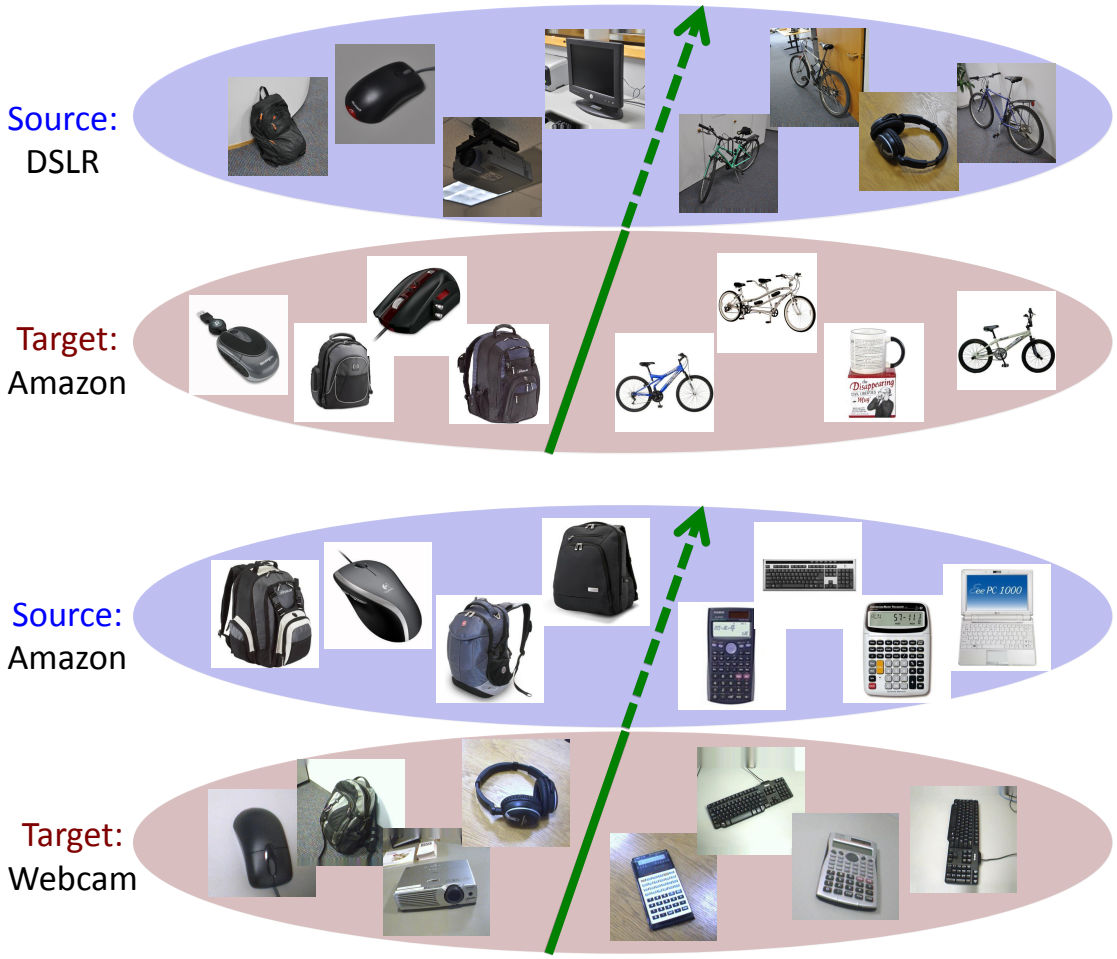


Figure 4.4: **Quantitative Evaluation of Predictability:** This figure illustrates two examples where an attribute hyperplane (green arrow), learned by our joint optimization, discriminates visual properties consistently across two different domains. In the left case, the hyperplane is discriminating between the objects with round shapes vs the ones with more surface area. In the right example, the hyperplane is discriminating the keypad-like objects against the more bulky ones. The dashed part of the arrow indicates that the same hyperplane which is trained in target domain is applied in the source domain.

Bibliography

[1] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.

[3] Hal Daumé, III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 2006.

[4] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. Technical report, 2009.

[5] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jenn Wortman. Learning bounds for domain adaptation. In *NIPS*, 2008.

[6] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[7] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[8] V. Jain and E. Learned-Miller. Online domain-adaptation of a pre-trained cascade of classifiers. In *CVPR*, 2011.

[9] R. Gopalan ad R. Li and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[10] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2), 2011.

[11] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 1979.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010.

[13] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010.

[14] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[15] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[16] Pingkun Yan, Saad M. Khan, and Mubarak Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007.

[17] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.

[18] Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. *AISTATS*, 2010.

[19] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan. A theory of learning from different domains. *Machine learning*, 2010.

[20] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, 2006.

[21] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[22] C.Wang and S. Mahadevan. Manifold alignment without correspondence. In *IJCAI*, 2009.

[23] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach. In *NIPS*, 2010.

[24] K. Lai and D. Fox. Object recognition in 3D point clouds using web data and domain adaptation. *International Journal of Robotics Research*, 2010.

[25] F. Mirrashed, V. Morariu, B. Siddiquie, R. Feris, and L. Davis. Domain adaptive object detection. In *WACV*, 2013.

[26] Xiaojin Zhu. Semi-Supervised Learning Literature Survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.

[27] Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *ICML*, 2000.

[28] Yan Zhou and Sally A. Goldman. Democratic Co-Learning. In *International Conference on Tools with Artificial Intelligence*, 2004.

[29] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[30] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[31] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang, and A. Zisserman. Dataset issues in object recognition. In *Toward Category-Level Object Recognition, volume 4170 of LNCS*, 2006.

[32] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

[33] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[34] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9, 2008.

[35] Hal Daume III, Abhishek Kumar, and Avishek Saha. Co-regularization based semi-supervised domain adaptation. In *NIPS*, 2010.

[36] Hal Daume III. Frustratingly easy domain adaptation. 2007.

[37] Boqing Gong, Yuan Shi, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, 2013.

[38] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[39] Mohammad Rastegari, Ali Farhadi, and David Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*, 2012.

[40] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, 2011.

[41] A. Holub G. Griffin and P. Perona. Caltech-256 object category dataset. In *Technical report*, 2007.

[42] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *In ECCV*, 2006.

[43] Huang J., A.J. Gretton, K.M. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2007.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 2010.

[45] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[46] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 2007.

[47] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2, 2004.

[48] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 2008.

[49] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.