Visual Domain Adaptation Using Regularized Cross-Domain Transforms

Kate Saenko
Brian Kulis
Mario Fritz
Trevor Darrell

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-106
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-106.html

July 1, 2010

Visual Domain Adaptation Using Regularized Cross-Domain Transforms

Kate Saenko Brian Kulis Mario Fritz
Trevor Darrell
UC Berkeley EECS and ICSI, Berkeley, CA
saenko,kulis,mfritz,trevor@eecs.berkeley.edu

July 1, 2010

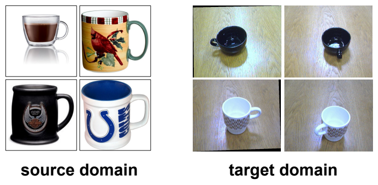
Abstract

We introduce a method that adapts object models acquired in a particular *visual domain* to new imaging conditions by learning a transformation which minimizes the effect of domain-induced changes in the feature distribution. The transformation is learned in a supervised manner, and can be applied to categories unseen at training time. We prove that the resulting model may be kernelized to learn non-linear transformations under a variety of regularizers. In addition to being one of the first studies of domain adaptation for object recognition, this work develops a general theoretical framework for adaptation that could be applied to non-image data. We present a new image database for studying the effects of visual domain shift on object recognition, and demonstrate the ability of our method to improve recognition on categories with few or no target domain labels, moderate to large changes in the imaging conditions, and even changes in the feature representation.

1 Introduction

Supervised classification methods such as kernel-based and nearest-neighbor classifiers have been shown to perform very well on standard object recognition tasks (e.g. [1], [2], [3]). However, many such methods expect the test images to come from the same distribution as the training images, and often fail when presented with a novel *visual domain*. While the problem of *domain adaptation* has received significant recent attention in the natural language processing community, it has been largely overlooked in the object recognition field. In this paper, we present a novel adaptation method, and apply it to the problem of domain shift in the context of object recognition.

Often, we wish to perform recognition in a *target* visual domain where we have very few labeled examples and/or only have labels for a subset of categories, but have access to a *source* domain with plenty of labeled examples in many



train	test	SVM-bow	NBNN [3]
source	source	54 ± 2	61 ± 1
source	target	20 ± 1	19 ± 1

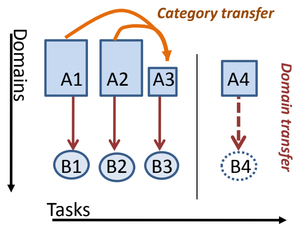


Figure 1: (Left) Example of extreme visual domain shift: The table shows how the classification accuracies (averaged over 31 categories) of two methods (an SVM over a bag-of-words representation (SVM-bow) and the Naive Bayes nearest neighbor (NBNN) classifier of [3]) degrade when they are trained on the source domain (online merchant images) and tested on the target domain (images from office robot). See Sec. 4 for detailed dataset descriptions. (Right) Unlike category transfer methods, our method does not transfer structure between related tasks, but rather transfers the *domain shift* to new tasks (which may or may not be related), such as from tasks 1-3 to task 4, as shown in the figure.

categories. As Figure 1 shows, it is insufficient to directly use object classifiers trained on the source domain, as their performance can degrade significantly on the target domain. Even when the same features are extracted in both domains, and the necessary normalization is performed on the image and the feature vectors, the underlying cause of the domain shift can strongly affect the feature distribution and thus violate the assumptions of the classifier. Typical causes of visual domain shift include changes in the camera, image resolution, lighting, background, viewpoint, and post-processing. In the extreme case, all of these changes take place, such as when shifting from typical object category datasets mined from internet search engines to images captured in real-world surroundings, e.g. by a mobile robot (see Figure 1). Furthermore, the feature vectors could be extracted using different pipelines and could have different dimensionalities.

Recently, domain adaptation methods that attempt to transfer classifiers learned on a source domain to new domains have been proposed for natural language tasks. For example, Blitzer et al. adapt sentiment classifiers learned on book reviews to electronics and kitchen appliances [4]. We argue that addressing the problem of domain adaptation for object recognition is essential for two reasons: 1) while labeled datasets are becoming larger and more available, they still differ significantly from many interesting domains, and 2) it is unrealistic to expect the user to collect many labels in each new domain, especially when one considers the large number of possible object categories.

In this paper, we introduce a novel domain adaptation technique based on cross-domain transformations. The key idea is to learn a regularized non-linear transformation that maps points from one domain to another domain (see Figure 2), using supervised data from both domains. The input consists of pairs of inter-domain examples that are known to be similar (or dissimilar). The output is the learned transformation, which can be applied to previously unseen test data points. We present a general model for learning linear cross-domain transformations, and then prove a novel result showing how to learn non-linear transformations by kernelizing the formulation for a particular class of regularizers. The ability to learn asymmetric and non-linear transformations is key, as it allows us to handle more general types of domain shift and changes in feature type and dimension. Encoding the domain invariance into the feature representation allows our method to benefit a broad range of classification methods, from k-NN to SVM, as well as clustering methods.

Importantly, our approach can be applied to the scenario where some of categories do not have any labels in the target domain. This essentially means transferring the learned "domain shift" to new categories encountered in the target domain. Thus, our approach can be thought of as a form of knowledge transfer from the source to the target domain. However, in contrast to many existing transfer learning paradigms (e.g. [5], [6]), we do not presume any degree of relatedness between the categories that are used to learn the transferred structure and the categories to which the structure is transferred (see Figure 1). The key point is that we are transferring the structure of the domain shift, not transferring structures common to related categories.

In the next section, we relate our approach to existing work on domain adaptation. Section 3 describes the theoretical framework behind our approach, including novel results on the possibility of kernelization of the asymmetric transform, and presents the main algorithm. We evaluate our method on a new dataset designed to study the problem of visual domain shift, which is described in Section 4, and show results of object classifier adaptation on several types of visual domain shift in Section 5.

2 Related Work

The domain adaptation problem has recently started to gain attention in the natural language community. Daume III [7] proposed a domain adaptation approach that works by transforming the features into an augmented space, where the input features from each domain are copied twice, once to a domain-independent portion of the feature vector, and once to the portion specific to that domain. The portion specific to all other domains is set to zeros. While "frustratingly" easy to implement, this approach only works for classifiers that learn a function over the features. With normalized features (as in our experimental results), the nearest neighbor classifier results are unchanged after adaptation. Structural correspondence learning is another method proposed for NLP tasks such as sentiment classification [4]. However, it is targeted towards language domains, and relies heavily on the selection of *pivot* features, which are words that frequently occur in both domains (e.g. "wonderful", "awful") and are correlated with domain-specific words.

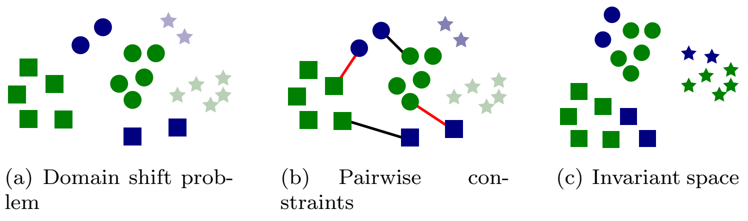

(a) Domain shift problem (b) Pairwise constraints (c) Invariant space

Figure 2: The key idea of our approach to domain adaptation is to learn a transformation that compensates for the domain-induced changes. By leveraging (dis)similarity constraints (b) we aim to reunite samples from two different domains (blue and green) in a common invariant space (c) in order to learn and classify new samples more effectively across domains. The transformation can also be applied to new categories (lightly-shaded stars). This figure is best viewed in color.

Recently, several adaptation methods for the support vector machine (SVM) classifier have been proposed in the video retrieval literature. Yang et al. [8] proposed an Adaptive SVM (A-SVM) which adjusts the existing classifier $f^s(x)$ trained on the source domain to obtain a new SVM classifier $f^t(x)$. Cross-domain SVM (CD-SVM) proposed by Jiang et al. [9] defines a weight for each source training sample based on distance to the target domain, and re-trains the SVM classifier was with re-weighted patterns. The domain transfer SVM (DT-SVM) proposed by Duan et al. [10] used multiple-kernel learning to minimize the difference between the means of the source and target feature distributions. These methods are specific to the SVM classifier, and they require target-domain labels for all categories. The advantage of our method is that it can perform transfer of domain-invariant representations to *novel* categories, with no target-domain labels.

Finally, metric and similarity learning has been successfully applied to a variety of problems in vision and other domains (see [11, 12, 13, 14] for some vision examples) but to our knowledge has not been applied to domain adaptation.

3 Domain Adaptation Using Regularized Cross-Domain Transforms

We begin by describing our general domain adaptation model in the linear setting, focusing on the particular cases that we employ in our experiments, then, in Section 3.1, we prove a novel result showing that kernelization of the model can be achieved under a wide class of regularizers.

In the following, we assume that there are two domains $\mathcal{A}$ and $\mathcal{B}$ (e.g., source and target). Given vectors $\boldsymbol{x} \in \mathcal{A}$ and $\boldsymbol{y} \in \mathcal{B}$, we propose to learn a linear transformation W from $\mathcal{B}$ to $\mathcal{A}$ (or equivalently, a transformation W^T to transform from $\mathcal{A}$ to $\mathcal{B}$). If the dimensionality of the vectors $\boldsymbol{x} \in \mathcal{A}$ is d_A and the dimensionality of the vectors $\boldsymbol{y} \in \mathcal{B}$ is d_B, then the size of the matrix W is

$d_A \times d_B$. We denote the resulting inner product similarity function between $\boldsymbol{x}$ and the transformed $\boldsymbol{y}$ as

$$\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T W \boldsymbol{y}.$$

The goal is to learn the linear transformation given some form of supervision, and then to utilize the learned similarity function in a classification or clustering algorithm. To avoid overfitting, we choose a regularization function for W, which we will denote as $r(W)$ (choices of the regularizer are discussed below). Denote $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_A}]$ as the matrix of n_A training data points (of dimensionality d_A) from $\mathcal{A}$ and $Y = [\boldsymbol{y}_1, ..., \boldsymbol{y}_{n_B}]$ as the matrix of n_B training data points (of dimensionality d_B) from $\mathcal{B}$. We will discuss the exact form of supervision we propose for domain adaptation problems in Section 3.2, but for now assume that it is a function of the learned similarity values $\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y})$ (i.e., a function of the matrix $X^T W Y$), so a general optimization problem would seek to minimize the regularizer subject to supervision constraints given by functions c_i:

$$\begin{array}{ll} \min_W & r(W) \\ \text{s.t.} & c_i(X^T W Y) \geq 0, \quad 1 \leq i \leq c. \end{array} \tag{1}$$

Due to the potential of infeasibility, we can introduce slack variables into the above formulation, or write the problem as an unconstrained problem:

$$\min_W \; r(W) + \lambda \sum_i c_i(X^T W Y).$$

In this paper, we focus on two particular special cases of this general transformation learning problem. The first employs the regularizer $r(W) = \frac{1}{2}\|W\|_F^2$ and constraints of the form $\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y}) \leq b$ for dissimilar pairs or $\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y}) \geq b$ for similar pairs.[1] We call this probelm the *Frobenius-regularized asymmetric transformation problem* with similarity and dissimilarity constraints, or *asymm* for short in the rest of the paper.

Second, we consider the regularizer $r(W) = \mathrm{tr}(W) - \log\det(W)$, and constraints that are a function of the learned *distances* $d_W(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T W (\boldsymbol{x} - \boldsymbol{y})$ (as with similarity constraints, the learned distances are linear functions of $X^T W Y$). Unlike the *asymm* formulation, this regularizer can only be applied when the dimensionalities of the two domains are equal ($d_A = d_B$). This choice of regularizer and constraints has previously been studied as a Mahalanobis metric learning method, and is called *information-theoretic metric learning* (ITML) [13]; we stress, however, that the use of such a regularizer for domain adaptation is novel, as is our method for constructing cross-domain constraints, which we discuss in Section 3.2. We call this approach *symm* for short, since the learned transformation W is always symmetric positive definite.

[1] $\mathrm{sim}_W(\boldsymbol{x}_i, \boldsymbol{y}_j)$ for some $\boldsymbol{x}_i \in \mathcal{A}$ and $\boldsymbol{y}_j \in \mathcal{B}$ can be written as a linear function of $X^T W Y$ via the expression $\boldsymbol{e}_i^T X^T W Y \boldsymbol{e}_j$, where $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are the i-th and j-th standard basis vectors, respectively.

3.1 Kernelization

There are two main limitations to the transformation learning problem (1) presented above. First, it is limited to linear transformations W, which may not be sufficient for some adaptation tasks. Second, the size of W grows as $d_A \cdot d_B$, which may be prohibitively large for some problems. In this section, we prove that (1) may be solved in kernel space, resulting in a non-linear transformation whose complexity is independent of the dimensionalities of the points in either domain. This kernelization result is the first general kernelization result for asymmetric transformation learning, and is critical to obtaining good performance for several domain adaptation tasks.

The main idea behind the following results is to show that i) the learned similarity function resulting from solving (1) can be computed only using inner products between data points in $\mathcal{A}$ and inner products between data points in $\mathcal{B}$, and ii) (1) can be reformulated as an optimization problem involving such inner products and whose size is independent of the dimensionalities d_A and d_B. Then we can replace standard inner products with arbitrary kernel functions, resulting in non-linear learned transformations in the input space. In the following analysis, the input kernel matrices over within-domain points are given as $K_A = X^T X$ and $K_B = Y^T Y$. We begin with the first result (proof in Appendix A).

Lemma 3.1. *Assume that the regularizer r is convex, $r(W) = \sum_j r_j(\sigma_j)$, where $\sigma_1, ..., \sigma_d$ are the singular values of W, and that $\min_x r_j(x) = 0$. Then the optimal solution W^* to* (1) *(or the corresponding unconstrained version) is of the form $W^* = X K_A^{-1/2} L K_B^{-1/2} Y^T$, where L is some $n_A \times n_B$ matrix.*

Note that the assumption that $\min_x r_j(x) = 0$ can be eliminated, and this leads to a W^* of the form $\alpha I + X K_A^{-1/2} L K_B^{-1/2} Y^T$. One important consequence of the above lemma is that, given arbitrary points $\boldsymbol{x}$ and $\boldsymbol{y}$, the function $\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y})$ can be computed in kernel space—by replacing W with $X K_A^{-1/2} L K_B^{-1/2} Y^T$, the expression $\boldsymbol{x}^T W \boldsymbol{y}$ can be written purely in terms of inner products.

The above result demonstrates the existence of such a matrix L, but does not show how to compute it. Using the above lemma, we now show how to rewrite the optimization (1) in terms of the kernel matrices to solve for L (proof in Appendix A):

Theorem 3.2. *Assume the conditions of Lemma 3.1 hold. If W^* is the optimal solution to* (1) *and L^* is the optimal solution to the following problem:*

$$\begin{array}{ll} \min_L & r(L) \\ s.t. & c_i(K_A^{1/2} L K_B^{1/2}) \geq 0, \quad 1 \leq i \leq c, \end{array} \tag{2}$$

then $W^* = X K_A^{-1/2} L^* K_B^{-1/2} Y^T$. In the unconstrained case, the corresponding optimization is $\min_L \; r(L) + \lambda \sum_i c_i(K_A^{1/2} L K_B^{1/2})$.

In summary, we have proven that the general asymmetric transformation learning problem may be applied in kernel space under a class of convex regularizers of the form $r(W) = \sum_j r_j(\sigma_j)$. Though our focus on this paper is on two particular regularizers—the squared Frobenius norm and a LogDet-type regularizer (for the symmetric case)—one can imagine applying our analysis to other regularizers. For example, the trace norm $\mathrm{tr}(W)$ falls under our framework; because the trace-norm as a regularizer is known to produce low-rank matrices W, it would be desirable in dimensionality-reduction settings.

3.2 Algorithm

Both regularizers considered in our paper ($r(W) = \frac{1}{2}\|W\|_F^2$ and $r(W) = \mathrm{tr}(W) - \log\det(W)$) are strictly convex, and the constraints we consider are linear. In either case, one can use a variety of possible optimization techniques. We opted for an alternating projection method using Bregman's algorithm; this method can be easily implemented to scale to large problems and has fast convergence in practice. See Censor and Zenios for details on Bregman's algorithm [15].

Generating Cross-Domain Constraints: Assume that there are k categories, with data from each category i denoted as D_i, consisting of $(\boldsymbol{x}, y)$ pairs of input data and category labels. There are two cases that we consider. In the first case, we have many labeled examples for each of the n categories in the source domain data, $D^s = \{D_1^s, ..., D_n^s\}$, and a few labeled examples for each category in the target domain data, $D^t = \{D_1^t, ..., D_n^t\}$. In the second case, we have the same D^s but only have labels for a subset of the categories in the target domain, $D^t = \{D_1^t, ..., D_m^t\}$, where $m < k$. Here, our goal is to adapt the classifier trained on the tasks $m+1, ..., k$, which only have source domain labels, to obtain a new classifier, which reduces the predictive error on the target domain by accounting for the domain shift. We do this by applying the transformation learned on the m categories to the features in the source domain training set of the new categories, and re-training the classifier.

To generate similarity constraints $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}$ and dissimilarity constraints $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}$ necessary to learn the domain-invariant transformation, we use the following procedure. We sample a random pair consisting of a labeled source domain sample $(\boldsymbol{x_i^s}, y_i^s)$ and a labeled target domain sample $(\boldsymbol{x_j^t}, y_j^t)$, and create a constraint

$$\begin{aligned} \mathrm{sim}_W(\boldsymbol{x}_i, \boldsymbol{x}_j) &\geq u \quad \text{if} \quad y_i = y_j, \\ \mathrm{sim}_W(\boldsymbol{x}_i, \boldsymbol{x}_j) &\leq \ell \quad \text{if} \quad y_i \neq y_j. \end{aligned} \tag{3}$$

when running *asymm* and

$$\begin{aligned} d_W(\boldsymbol{x}_i, \boldsymbol{x}_j) &\leq u \quad \text{if} \quad y_i = y_j, \\ d_W(\boldsymbol{x}_i, \boldsymbol{x}_j) &\geq \ell \quad \text{if} \quad y_i \neq y_j. \end{aligned} \tag{4}$$

when running *symm*. Alternatively, we can generate constraints based not on class labels, but on information of the form: target sample $\boldsymbol{x_i}$ is similar to source sample $\boldsymbol{x_j}$. This is particularly useful when the source and target data include

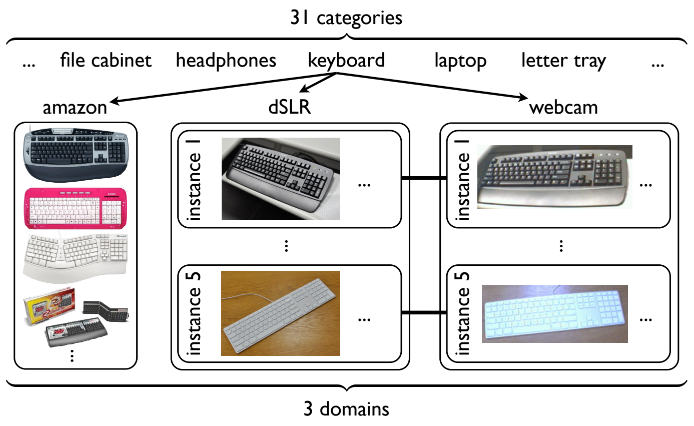


Figure 3: New dataset for investigating domain shifts in visual category recognition tasks.

images of the same object, as it allows us to best recover the structure of the domain shift, without learning anything about particular categories. We refer to these as *correspondence* constraints. It is important to generate constraints between samples of different domains, as including same-domain constraints can make it difficult for the algorithm to learn a domain-invariant metric. In fact, we show experimentally that generating constraints based on class labels without regard for domain boundaries, in the style of metric learning, does considerably worse than our method.

4 A Database for Studying Effects of Domain Shift in Object Recognition

As detailed earlier, effects of domain shift have been largely overlooked in previous object recognition studies. Therefore, one of the contributions of this paper is a database that allows researchers to study, evaluate and compare solutions to the domain shift problem by establishing a multiple-domain labeled dataset and benchmark. The database, benchmark code, and code for our method will be made available to the community upon publication. In addition to the domain shift aspects, this database also proposes a challenging office environment category learning task which reflects the difficulty of real-world indoor robotic object recognition, and may serve as a useful testbed for such tasks. It contains a total of 4652 images originating from the following three domains:

Images from the web: The first domain consists of images from the web downloaded from online merchants (www.amazon.com). This has become a very popular way to acquire data, as it allows for easy access to large amounts of data that lends itself to learning category models. These images are of products shot at medium resolution typically taken in an environment with studio lighting conditions. We collected two datasets: *amazon* contains 31 categories[2] with an average of 90 images each. The images capture the large intra-class variation of these categories, but typically show the instances only from a canonical viewpoint. *amazonINS* contains 17 object *instances* (e.g. can of *Taster's Choice* instant coffee) with an average of two images each.

Images from a digital SLR camera: The second domain consists of images that are captured with a digital SLR camera in realistic environments with natural lighting conditions. The images have high resolution (4288x2848) and low noise. We have recorded two datasets: *dslr* has images of the 31 object categories, with 5 different objects for each, in an office environment. Each object was captured with on average 3 images taken from different viewpoints, for a total of 423 images. *dslrINS* contains 534 images of the 17 object instances, with an average of 30 images per instance, taken in a home environment.

Images from a webcam: The third domain consists of images of the 31 categories recorded with a simple webcam. The images are of low resolution (640x480) and show significant noise and color as well as white balance artifacts. Many current imagers on robotic platforms share a similarly-sized sensor, and therefore also possess these sensing characteristics. The resulting *webcam* dataset contains same 5 objects as for in *dSLR*, for a total of 795 images.

The database represents several interesting visual domain shifts. First of all, it allows us to investigate the adaptation of category models learned on the web to dSLR and webcam images, which can be thought of as in situ observations on a robotic platform in a realistic office or home environment. Second, domain transfer between the high-quality dSLR images to low-resolution webcam images allows for a very controlled investigation of category model adaptation, as the same objects were recorded in both domains. Finally, the amazonINS and dslrINS datasets allow us to evaluate adaptation of product instance models from web data to a user environment, in a setting where images of the same products are available in both domains.

5 Experiments

In this section, we evaluate our domain adaptation approach by applying it to k-nearest neighbor classification of object categories and instances. We use the database described in the previous section to study different types of domain shifts and compare our new approach to several baseline methods. First, we will detail our processing pipeline before we describe the different setting and elaborate on our empirical findings.

Image Processing: All images were resized to the same width and converted to grayscale. Local scale-invariant interest points were detected using the SURF [16] detector to describe the image. SURF features have been shown

[2]The 31 categories in the database are: backpack, bike, bike helmet, bookcase, bottle, calculator, desk chair, desk lamp, computer, file cabinet, headphones, keyboard, laptop, letter tray, mobile phone, monitor, mouse, mug, notebook, pen, phone, printer, projector, puncher, ring binder, ruler, scissors, speaker, stapler, tape, and trash can.

		No shift	Baseline Methods				Our Method	
domain A	domain B	knnAA	knnAB	knnBB	ITML(A+B)	ITML(B)	asymm	symm
webcam	dslr	0.34	0.14	0.20	0.18	0.23	0.25	**0.27**
dslr	webcam	0.31	0.25	0.23	0.23	0.28	0.30	**0.31**
amazon	webcam	0.33	0.03	0.43	0.41	0.43	**0.48**	0.44

Table 1: Domain adaptation results for categories seen during training in the target domain.

		Baseline Methods		Our Method	
domain A	domain B	knnAB	ITML(A+B)	asymm	symm
webcam	dslr	0.37	0.38	**0.53**	0.49
webcam-800	dslr-600	0.31	n/a	**0.43**	n/a
amazonINS	dslrINS	0.23	0.25	**0.30**	0.25

Table 2: Domain adaptation results for categories not seen during training in the target domain.

to be highly repeatable and robust to noise, displacement, geometric and photometric transformations. The blob response threshold was set to 1000, and the other parameters to default values. A 64-dimensional non-rotationally invariant SURF descriptor was used to describe the patch surrounding each detected interest point. After extracting a set of SURF descriptors for each image, vector quantization into visual words was performed to generate the final feature vector. A codebook of size 800 was constructed by k-means clustering on a randomly chosen subset of the amazon database. All images were converted to histograms over the resulting visual words. No spatial or color information was included in the image representation for these experiments.

In the following, we compare k-NN classifiers that use the proposed cross-domain transformation to the following baselines: 1) k-NN classifiers that operate in the original feature space using a Euclidean distance, and 2) k-NN classifiers that use traditional supervised metric learning, implemented using the ITML [17] method, trained using all available labels in both domains. We kernelize the metric using an RBF kernel with width $\sigma = 1.0$, and set $\lambda = 10^2$. As a performance measure, we use accuracy (number of correctly classified test samples divided by the total number of test samples) averaged over 10 randomly selected train/test sets.

Same-category setting: In this setting, all categories have (a small number of) labels in the target domain (3 in our experiments.) We generate constraints between all cross-domain image pairs in the training set based on their class labels, as described in Section 3.2. Table 1 shows the results. In the first result column, to illustrate the level of performance without the domain shift, we plot the accuracy of the Euclidean k-NN classifier trained on the source domain $\mathcal{A}$ and tested on images from the same domain (*knn_AA*). The next column shows the same classifier, but trained on $\mathcal{A}$ and tested on $\mathcal{B}$ (*knn_AB*). Here,

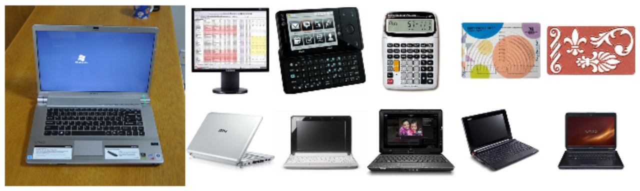


Figure 4: Examples of the 5 nearest neighbors retrieved for a *webcam* query image (right image) from the *amazon* dataset, using the *knn_AB* baseline in Table 1 (top row of smaller images) and the learned cross-domain *symm* kernel (bottom row).

the effect of the domain shift is evident, as the performance drops for all domain pairs, dramatically so in the case of the *amazon* to *webcam* shift. We can also train k-NN using the few available $\mathcal{B}$ labels (*knn_BB*, third column). The fourth and the fifth columns show the metric learning baseline, trained either on all pooled training data from both domains ($ITML(A+B)$, or only on $\mathcal{B}$ labels ($ITML(B)$). The last two columns show the symmetric and asymmetric variants of our domain adaptation method. Note that *knn_BB* does not perform as well because of the limited amount of labeled examples we have available in $\mathcal{B}$. Even the more powerful metric-learning based classifier fails to perform as well as the k-NN classifier using our domain-invariant transform, given the small amount of labeled target data.

The shift between dslr and webcam domains represents a moderate amount of change, mostly due to the differences in the cameras, as the same objects were used to collect both datasets. Since webcam actually has more training images, the reverse webcam-to-dslr shift is probably better suited to adaptation. In both these cases, *symm* outperforms *asym*, possibly due to the more symmetric nature of the shift and/or lack of training data to learn a more general tranformation. The shift between the amazon and the dslr/webcam domains is the most drastic (bottom row of Table 1.) Even for this challenging problem, the adapted k-NN classifier outperforms the non-adapted baselines, with *asymm* doing better than *symm*. Figure 4 show example images retrieved by our method from *amazon* for a query from *webcam*.

New-category setting: In this setting, the test data belong to categories (or instances) for which we only have labels in the source domain. We use the first half of the categories to learn the transformation, forming correspondence constraints between images of the same object instances in roughly the same pose. We test the metric on the remaining categories. The results of adapting *webcam* to *dslr* are shown the first row of Table 2. Our approach clearly learns something about the domain shift, significantly improving the performance over the baselines, with *asymm* beating *symm*. Note that the overall accuracies are higher as this is a 16-way classification task. The second row in Table 2 shows results of adapting between heterogeneous feature sets: *webcam*, where features are computed using an 800-codeword vocabulary, and *dslr-600*, where a different vocabulary is used, with 600 codewords computed on dSLR images. The baseline k-NN is achieved by mapping each $\mathcal{B}$ codeword to it's nearest neighbor in $\mathcal{A}$. This clearly illustrates the advantage of our asymmetric method, as it is able to handle such drastic transformations of the input features. The last row shows results on an instance classification task, tackling the shift from Amazon to user environment images.

6 Conclusion

We presented a detailed study of domain shift in the context of object recognition, and introduced a novel adaptation technique that projects the features into a domain-invariant space via a transformation learned from labeled source and target domain examples. Our approach can be applied to adapt a wide range of visual models which operate over similarities between samples, and works both on cases where we need to classify novel test samples from categories seen at training time, and on cases where the test samples come from new categories which were not seen at training time. This is especially useful for object recognition, as large multi-category object databases can be adapted to new domains without requiring labels for all of the possibly huge number of categories. Our results show the effectiveness of our technique for adapting k-NN classifiers to a range of domain shifts.

References

[1] Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR. (2007)

[2] Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV. (2007)

[3] Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2008)

[4] Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. ACL (2007)

[5] Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: ICCV. (2009)

[6] Fink, M.: Object classification from a single example utilizing class relevance metrics. In: Proc. NIPS. (2004)

[7] Daume III, H.: Frustratingly easy domain adaptation. In: ACL. (2007)

[8] Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. ACM Multimedia (2007)

[9] Jiang, W., Zavesky, E., Chang, S., Loui, A.: Cross-domain learning methods for high-level visual concept classification. In: ICIP. (2008)

[10] Duan, L., Tsang, I.W., Xu, D., Maybank, S.J.: Domain transfer svm for video concept detection. In: CVPR. (2009)

[11] Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proc. CVPR. (2005)

[12] Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: CVPR. (2004)

[13] Kulis, B., Jain, P., Grauman, K.: Fast similarity search for learned metrics. IEEE PAMI **39** (2009) 2143–2157

[14] Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. Pattern Recognition and Image Analysis (2009)

[15] Censor, Y., Zenios, S.A.: Parallel Optimization: Theory, Algorithms, and Applications. Oxford University Press (1997)

[16] Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: ECCV. (2006)

[17] Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. ICML (2007)

A Appendix: Proofs

Proof of Lemma 3.1: Let W have singular value decomposition $U\Sigma\tilde{U}^T$. We can therefore write W as $W = \sum_j \sigma_j \boldsymbol{u}_j \tilde{\boldsymbol{u}}_j^T$. For every $\boldsymbol{u}_j$, either it is in the range space of X or the null space of X. If it is in the range space, then $\boldsymbol{u}_j = X\boldsymbol{z}_j$, for some $\boldsymbol{z}_j$; if it is in the null space, then $X^T\boldsymbol{u}_j = 0$. An analogous statement holds for $\tilde{\boldsymbol{u}}_j$.

Consider computation of $c_i(X^TWY)$. Expanding W via the SVD yields

$$X^TWY = X^T\Big(\sum_j \sigma_j \boldsymbol{u}_j \tilde{\boldsymbol{u}}_j^T\Big)Y = \sum_j \sigma_j (X^T\boldsymbol{u}_j\tilde{\boldsymbol{u}}_j^T Y).$$

If either $\boldsymbol{u}_j$ is in the null space of X or $\tilde{\boldsymbol{u}}_j$ is in the null space of Y, then the corresponding terms in the sum will be zero. As a result, σ_j is completely unconstrained, and can be chosen to minimize f_j, which is assumed to be at 0.

Therefore, let us assume that the singular values are ordered such that the first t are such that the corresponding singular vectors $\boldsymbol{u}$ are in the range space of X and $\tilde{\boldsymbol{u}}$ are in the range space of Y. The remainder of the singular values are equal to 0. Then we have

$$\begin{aligned} W &= \sum_{j=1}^{t} \sigma_j \boldsymbol{u}_j \tilde{\boldsymbol{u}}_j^T = \sum_{j=1}^{t} \sigma_j X\boldsymbol{z}_j\tilde{\boldsymbol{z}}_j^T Y^T \\ &= X\Big(\sum_{j=1}^{t} \sigma_j \boldsymbol{z}_j\tilde{\boldsymbol{z}}_j^T\Big)Y^T = X\tilde{L}Y^T. \end{aligned}$$

With the transformation $L = K_A^{1/2}\tilde{L}K_B^{1/2}$, we can equivalently write as $W = XK_A^{-1/2}LK_B^{-1/2}Y^T$.

Proof of Theorem 3.2: Denote $V_A = XK_A^{-1/2}$ and $V_B = YK_B^{-1/2}$. Note that V_A and V_B are orthogonal matrices. From the lemma, $W = V_A L V_B^T$; let $V_A^\perp$ and $V_B^\perp$ be the orthogonal complements to V_A and V_B, and let $\bar{V_A} = [V_A V_V^\perp]$ and $\bar{V_B} = [V_B V_B^\perp]$. Then

$$r\Bigg(\bar{V_A}\begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix}\bar{V_B}^T\Bigg) = r\Bigg(\begin{bmatrix} W & 0 \\ 0 & 0 \end{bmatrix}\Bigg) = r(W) + const.$$

One can easily verify that, given two orthogonal matrices V_1 and V_2 and an arbitrary matrix M, that $r(V_1MV_2) = \sum_j r_j(\sigma_j)$ if σ_j are the singular values of M. So

$$r\Bigg(\bar{V_A}\begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix}\bar{V_B}^T\Bigg) = \sum_j r_j(\bar{\sigma_j}) + const = r(L) + const,$$

where $\bar{\sigma_i}$ are the singular values of L. Thus, $r(W) = r(L) + const$.

Now rewrite the constraints c_i using $W = XK_A^{-1/2}LK_B^{-1/2}Y^T$:

$$c_i(X^TWY) = c_i(K_AK_A^{-1/2}LK_B^{-1/2}K_B) = c_i(K_A^{1/2}LK_B^{1/2}).$$

The theorem follows by rewriting r and the c_i functions using the above derivations. Note that both r and the c_i's can be computed independently of the dimensionality, so simple arguments show that the optimization may be solved in polynomial time independent of the dimensionality when the r_j functions are convex.