Efficient Learning of Domain-invariant Image Representations

Judy Hoffman
UCB EECS & ICSI
jhoffman@eecs.berkeley.edu

Erik Rodner
UCB EECS & ICSI
erik.rodner@gmail.com

Jeff Donahue
UCB EECS & ICSI
jdonahue@eecs.berkeley.edu

Trevor Darrell
UCB EECS & ICSI
trevor@eecs.berkeley.edu

Kate Saenko
University of Massachusetts, Lowell
saenko@cs.uml.edu

Abstract

We present an algorithm that learns representations which explicitly compensate for domain mismatch and which can be efficiently realized as linear classifiers. Specifically, we form a linear transformation that maps features from the target (test) domain to the source (training) domain as part of training the classifier. We optimize both the transformation and classifier parameters jointly, and introduce an efficient cost function based on misclassification loss. Our method combines several features previously unavailable in a single algorithm: multi-class adaptation through representation learning, ability to map across heterogeneous feature spaces, and scalability to large datasets. We present experiments on several image datasets that demonstrate improved accuracy and computational advantages compared to previous approaches.

1 Introduction

We address the problem of learning domain-invariant image representations for multi-class classifiers. The ideal image representation often depends not just on the task but also on the domain. Recent studies have demonstrated a significant degradation in the performance of state-of-the-art image classifiers when input feature distributions change due to different image sensors and noise conditions [1], pose changes [2], a shift from commercial to consumer video [3, 4], and, more generally, training datasets biased by the way in which they were collected [5]. Learning adaptive representations for linear classifiers is particularly interesting as they are efficient and prevalent in vision applications, with fast linear SVMs forming the core of some of the most popular object detection methods [6, 7].

Previous work proposed to adapt linear SVMs [8, 9, 10], learning a perturbation of the source hyperplane by minimizing the classification error on labeled target examples for each binary task. These perturbations can be thought of as new feature representations that correct for the domain change. The recent HFA method [11] learns both the perturbed classifier and a latent domain-invariant feature representation, allowing domains to have heterogeneous features with different dimensionalities. However, existing SVM-based methods are limited to learning a separate representation for each binary problem and cannot transfer a common, class-independent component of the shift (such as global lighting change) to unlabeled categories, as illustrated in Figure 1. Additionally, the HFA algorithm cannot be solved in linear space and therefore scales poorly to large datasets.
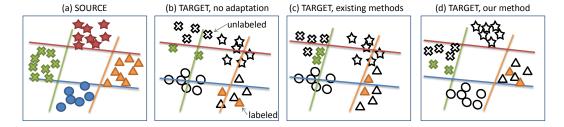
Figure 1: (a) Linear classifiers (shown as decision boundaries) learned for a four-class problem on a fully labeled source domain. (b) Problem: classifiers learned on the source domain do not fit the target domain points shown here due to a change in feature distribution. (c) Existing SVM-based methods only adapt the features of classes with labels (crosses and triangles). (d) Our method adapts all points, including those from classes without labels, by transforming all target features to a new domain-invariant representation.

Recently proposed feature adaptation methods [1, 2, 12, 13, 14, 15] offer a solution by learning a category-independent feature transform that maps target features into the source, pooling all training labels across categories. This enables multi-class adaptation, i.e. transferring the category-independent component of the domain-invariant representation to unlabeled categories. For example, a map learned on the labeled "triangle" class in Figure 1 can also be used to map the unlabeled "star" class to the source domain. An additional advantage of the asymmetric transform method ARC-t [12] over metric learning [1] or the recently proposed Geodesic Flow Kernel (GFK) [15], is that, like HFA [11], ARC-t can map between heterogeneous feature spaces. However, ARC-t has two major limitations: First, the feature learning does not optimize the objective function of a strong, discriminative classifier directly; rather, it maximizes some notion of similarity between the transformed target points and points in the source. Second, it does not scale well to domains with large numbers of points due to the high number of constraints, which is proportional to the product of the number of labeled data points in the source and target.

In this paper, we present a novel technique that combines the desirable aspects of recent methods in a single algorithm, which we call Max-Margin Domain Transforms, or MMDT for short. MMDT uses an asymmetric (non-square) transform W to map target features x to a new representation Wx maximally aligned with the source, learning the transform jointly on all categories for which target labels are available (Figure 1(d)). MMDT provides a way to adapt max-margin classifiers in a multi-class manner, by learning a shared component of the domain shift as captured by the feature transformation W. Additionally, MMDT can be optimized quickly in linear space, making it a feasible solution for problem settings with a large amount of training data.

The key idea behind our approach is to simultaneously learn both the projection of the target features into the source domain and the classifier parameters themselves, using the same classification loss to jointly optimize both.

Thus our method learns a feature representation that combines the strengths of max-margin learning with the flexibility of the feature transform. Because it operates over the input features, it can generalize the learned shift in a way that parameter-based methods cannot. On the other hand, it overcomes the two flaws of the ARC-t method: by optimizing the classification loss directly in the transform learning framework, it can achieve higher accuracy; furthermore, replacing similarity constraints with more efficient hyperplane constraints significantly reduces the training time of the algorithm and learning a transformation directly from target to source allows optimization in linear space.

The main contributions of our paper can be summarized as follows (also see Table 1):

- Experiments show that MMDT in linear feature space outperforms competing methods in terms of multi-class accuracy even compared to previous kernelized methods.

- MMDT learns a representation via an asymmetric category independent transform. Therefore, it can adapt features even when the target domain does not have any labeled examples for some categories and when the target and source features are not equivalent.

	ARC-t [12]	HFA [11]	GFK [15]	MMDT (ours)
multi-class	**yes**	no	**yes**	**yes**
large datasets	no	no	**yes**	**yes**
heterogeneous features	**yes**	**yes**	no	**yes**
optimize max-margin objective	no	**yes**	no	**yes**

Table 1: Unlike previous methods, our approach is able to simultaneously learn muti-class representations that can transfer to novel classes, scale to large training datasets, and handle different feature dimensionalities.

- The optimization of MMDT is scalable to large datasets because the number of constraints to optimize is linear in the number of training data points and because it can be optimized in linear feature space.
- Our final iterative solution can be solved using standard QP packages, making MMDT easy to implement.

2   Related Work

Domain adaptation, or covariate shift, is a fundamental problem in machine learning, and has attracted a lot of attention in the machine learning and natural language community, e.g. [16, 17, 18, 19] (see [20] for a comprehensive overview). It is related to multi-task learning but differs from it in the following way: in domain adaptation problems, the distribution over the features $p(X)$ varies across domains while the output labels Y remain the same; in multi-task learning or knowledge transfer, $p(X)$ stays the same (single domain) while the output labels vary (see [20] for more details). In this paper, we perform multi-task learning *across domains*, i.e. both $p(X)$ and the output labels Y can change between domains.

Domain adaptation has been gaining considerable attention in the vision community. Several SVM-based approaches have been proposed for image domain adaptation, including: weighted combination of source and target SVMs and transductive SVMs applied to adaptation in [21]; the feature replication method of [17]; Adaptive SVM [8, 9], where the source model parameters are adapted by adding a perturbation function, and its successor PMT-SVM [10]; Domain Transfer SVM [3], which learns a target decision function while reducing the mismatch in the domain distributions; and a related method [4] based on multiple kernel learning. In the linear case, feature replication [17] can be shown to decompose the learned parameter into $\theta = \hat{\theta} + \theta'$, where $\hat{\theta}$ is shared by all domains [22], in a similar fashion to adaptive SVMs.

Several authors considered learning feature representations for unsupervised and transfer learning [23], and for domain adaptation [18, 24]. For visual domain adaptation, transform-based adaptation methods [1, 12, 13, 2, 14, 11] have recently been proposed. These methods attempt to learn a perturbation over the feature space rather than a class-specific perturbation over the model parameters, typically in the form of a transformation matrix/kernel. The most closely related are the ARC-t method [12], which learns a transformation that maximizes similarity constraints between points in the source and those projected from the target domain, and the recent HFA method [11], which learns a transformation both from the source and target into a common latent space, as well as the classifier parameters. Another related method is the recently proposed GFK [15], which computes a symmetric kernel between source and target points based on geodesic flow along a latent manifold. We will present a detailed comparison to these three methods in the next section.

3   Max-Margin Domain Transforms

We propose a novel method for multi-task domain adaptation of linear SVMs by learning a target feature representation. Denote the normal to the affine hyperplane associated with the k'th binary SVM as θ_k, $k = 1, ..., K$, and the offset of that hyperplane from the origin as b_k. Intuitively, we would like to learn a new target feature representation that is shared across multiple categories. We propose to do so by estimating a transformation W of the input features, or, equivalently, a transformation W^T of the source hyperplane parameters θ_k. Let $x_1^s, \ldots, x_{n_S}^s$ denote the training points in the source domain ($\mathcal{D}_S$), with labels $y_1^s, \ldots, y_{n_S}^s$. Let $x_1^t, \ldots, x_{n_T}^t$ denote the labeled

points in the target domain ($\mathcal{D}_T$), with labels $y_1^t, \ldots, y_{n_T}^t$. Thus our goal is to jointly learn 1) affine hyperplanes that separate the classes in the common domain consisting of the source domain and target points projected to the source and 2) the new feature representation of the target domain determined by the transformation W mapping points from the target domain into the source domain. The transformation should have the property that it projects the target points onto the correct side of each source hyperplane.

For simplicity of presentation, we first show the optimization problem for a binary problem (dropping k) with no slack variables. Our objective is as follows:

$$\min_{W,\theta,b} \quad \frac{1}{2}||W||_F^2 + \frac{1}{2}||\theta||_2^2 \tag{1}$$

$$\text{s.t.} \quad y_i^s\left(\begin{bmatrix} x_i^s \\ 1 \end{bmatrix}^T \begin{bmatrix} \theta \\ b \end{bmatrix}\right) \geq 1 \qquad \forall i \in \mathcal{D}_S \tag{2}$$

$$y_i^t\left(\begin{bmatrix} x_i^t \\ 1 \end{bmatrix}^T W^T \begin{bmatrix} \theta \\ b \end{bmatrix}\right) \geq 1 \quad \forall i \in \mathcal{D}_T \tag{3}$$

Note that this can be easily extended to the multi-class case by simply adding a sum over the regularizers on all θ_k parameters and pooling the constraints for all categories. The objective function, written as in Equations (1)-(3), is not a convex problem and so is both hard to optimize and is not guaranteed to have a global solution. Therefore, a standard way to solve this problem is to do alternating minimization on the parameters, in our case W and (θ, b). We can effectively do this because when each parameter vector is fixed, the resulting optimization problem is convex.

We begin by re-writing Equations (1)-(3) for the more general problem with soft constraints and K categories. Let us denote the hinge loss as: $\mathcal{L}(y, x, \theta) = \max\{0, 1 - \delta(y, k) \cdot x^T\theta\}$. We define a cost function

$$
\begin{aligned}
J(W, \theta_k, b_k) \quad = \quad & \frac{1}{2}||W||_F^2 + \sum_{k=1}^{K}\left[\frac{1}{2}||\theta_k||_2^2\right. \\
& \left. + C_S\sum_{i=1}^{n_S}\mathcal{L}\left(y_i^s, \begin{bmatrix} x_i^s \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}\right) + C_T\sum_{i=1}^{n_T}\mathcal{L}\left(y_i^t, W \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}\right)\right]
\end{aligned}
\tag{4}
$$

where the constant C_S penalizes the source classification error and C_T penalizes the target adaptation error. Finally, we define our objective function with soft constraints as follows:

$$\min_{W,\theta_k,b_k} J(W, \theta_k, b_k) \tag{5}$$

To solve the above optimization problem we perform coordinate descent on W and (θ, b).

1. Set iteration $j = 0$, $W^{(j)} = 0$.

2. Solve the sub-problem $(\theta_k^{(j+1)}, b_k^{(j+1)}) = \arg\min_{\theta_k, b_k} J(W^{(j)}, \theta_k, b_k)$ by solving:

$$\min_{\theta,b} \sum_{k=1}^{K}\left[\frac{1}{2}||\theta_k||_2^2 + C_S\sum_{i=1}^{n_S}\mathcal{L}\left(y_i^s, \begin{bmatrix} x_i^s \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k; \\ b_k \end{bmatrix}\right) + C_T\sum_{i=1}^{n_T}\mathcal{L}\left(y_i^t, W^{(j)} \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}\right)\right]$$

Notice, this corresponds to the standard SVM objective function, except that the target points are first projected into the source using $W^{(j)}$. Therefore, we can solve this intermediate problem using a standard SVM solver package.

3. Solve the subproblem $W^{(j+1)} = \arg\min_W J(W, \theta^{(j+1)}, b^{(j+1)})$ by solving

$$\min_{W} \quad \frac{1}{2}||W||_F^2 + C_T\sum_{k=1}^{K}\sum_{i=1}^{n_T}\mathcal{L}\left(y_i^t, W \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k^{(j+1)} \\ b_k^{(j+1)} \end{bmatrix}\right)$$

and increment j. This optimization sub-problem is convex and is in a form that a standard QP optimization package can solve.

4. Iterate steps 2 & 3 until convergence.

It is straightforward to show that both stages (2) and (3) cannot increase the global cost function $J(W, \theta, b)$. Therefore, this algorithm is guaranteed to converge to a local optimum. A proof is included in the supplemental material.

It is important to note that since both steps of our iterative algorithm can be solved using standard QP solvers, the algorithm can be easily implemented. Additionally, since the constraints in our algorithm grow linearly with the number of training points and it can be solved in linear feature space, the optimization can be solved efficiently even as the number of training points grows.

Relation to existing work: We now analyze the proposed algorithm in the context of the previous feature transform methods ARC-t [12], HFA [11] and GFK [15]. ARC-t introduced similarity-based constraints to learn a mapping similar to that in step 3 in our algorithm. This approach creates a constraint for each labeled point x_i^s in the source and labeled point x_i^t in the target, and then learns a transformation W that satisfies constraints of the form $(x_i^s)^T W x_i^t > u$ if the labels of x_i^s and x_i^t are the same, and $(x_i^s)^T W x_i^t < l$ if the labels are different, for some constants u, l.

The ARC-t formulation has two distinct limitations that our method overcomes. First, it must solve $n_S \cdot n_T$ constraints, whereas our formulation only needs to solve $K \cdot n_T$ constraints, for a K category problem. In general, our method scales to much larger source domains than with ARC-t. The second benefit of our max-margin transformation learning approach is that the transformation learned using the max-margin constraints is learned jointly with the classifier, and explicitly seeks to optimize the final SVM classifier objective. While ARC-t's similarity-based constraints seek to map points of the same category arbitrarily close to one another, followed by a separate classifier learning step, we seek simply to project the target points onto the correct side of the learned hyperplane, leading to better classification performance.

The HFA formulation also takes advantage of the max-margin framework to directly optimize the classification objective while learning transformations. HFA learns the classifier and transformations to a common latent feature representation between the source and target. However, HFA is formulated to solve a binary problem so a new feature transformation must be learned for each category. Therefore, unlike MMDT, HFA cannot learn a representation that generalizes to novel target categories. Additionally, due to the difficulty of defining the dimension of the latent feature representation directly, the authors optimize with respect to a larger combined transformation matrix and a relaxed constraint. This transformation matrix becomes too large when the feature dimensions in source and target are large so the HFA must usually be solved in kernel space. This can make the method slow and cause it to scale poorly with the number of training examples. In contrast, our method can be efficiently solved in linear feature space which makes it fast and potentially more scalable.

Finally, GFK [15] formulates a kernelized representation of the data that is equivalent to computing the dot product in infinitely many subspaces along the geodesic flow between the source and target domain subspaces. The kernel is defined by the authors to be symmetric and so can not handle source and target domains of different initial dimension. Additionally, GFK does not directly optimize a classification objective. In contrast, our method, MMDT, can handle source and target domains of different feature dimensions via an asymmetric W, as well as directly optimizes the classification objective.

4   Experiments on Image Datasets

We now present experiments using the *Office* [1], *Caltech256* [25] and *Bing* [21] datasets to evaluate our algorithm according to the following four criteria. 1) Using a subset of the *Office* and *Caltech256* datasets we evaluate multi-class accuracy performance in a standard supervised domain adaptation setting, where all categories have a small number of labeled examples in the target. 2) Using the full *Office* dataset we evaluate multi-class accuracy for the supervised domain adaptation setting where the source and target have different feature dimensions. 3) Using the full *Office* dataset we evaluate multi-class accuracy in the multi-task domain adaptation setting with novel target categories at test time. 4) Using the *Bing* dataset we assess the ability to scale to larger datasets by analyzing timing performance.

	svm$_s$	**svm**$_t$	**arct** [12]	**hfa** [11]	**gfk**[15]	**mmdt** (ours)
a → w	33.9 ± 0.7	62.4 ± 0.9	55.7 ± 0.9	61.8 ± 1.1	58.6 ± 1.0	64.6 ± 1.2
a → d	35.0 ± 0.8	55.9 ± 0.8	50.2 ± 0.7	52.7 ± 0.9	50.7 ± 0.8	56.7 ± 1.3
w → a	35.7 ± 0.4	45.6 ± 0.7	43.4 ± 0.5	45.9 ± 0.7	44.1 ± 0.4	47.7 ± 0.9
w → d	66.6 ± 0.7	55.1 ± 0.8	71.3 ± 0.8	51.7 ± 1.0	70.5 ± 0.7	67.0 ± 1.1
d → a	34.0 ± 0.3	45.7 ± 0.9	42.5 ± 0.5	45.8 ± 0.9	45.7 ± 0.6	46.9 ± 1.0
d → w	74.3 ± 0.5	62.1 ± 0.8	78.3 ± 0.5	62.1 ± 0.7	76.5 ± 0.5	74.1 ± 0.8
a → c	35.1 ± 0.3	32.0 ± 0.8	37.0 ± 0.4	31.1 ± 0.6	36.0 ± 0.5	36.4 ± 0.8
w → c	31.3 ± 0.4	30.4 ± 0.7	31.9 ± 0.5	29.4 ± 0.6	31.1 ± 0.6	32.2 ± 0.8
d → c	31.4 ± 0.3	31.7 ± 0.6	33.5 ± 0.4	31.0 ± 0.5	32.9 ± 0.5	34.1 ± 0.8
c → a	35.9 ± 0.4	45.3 ± 0.9	44.1 ± 0.6	45.5 ± 0.9	44.7 ± 0.8	49.4 ± 0.8
c → w	30.8 ± 1.1	60.3 ± 1.0	55.9 ± 1.0	60.5 ± 0.9	63.7 ± 0.8	63.8 ± 1.1
c → d	35.6 ± 0.7	55.8 ± 0.9	50.6 ± 0.8	51.9 ± 1.1	57.7 ± 1.1	56.5 ± 0.9
mean	40.0 ± 0.6	48.5 ± 0.8	49.5 ± 0.6	47.4 ± 0.8	51.0 ± 0.7	52.5 ± 1.0

Table 2: Multi-class accuracy for the standard supervised domain adaptation setting: All results are from our implementation. When averaged across all domain shifts the reported average value for **gfk** was 51.65 while our implementation had an average of 51.0 ± 0.7. Therefore, the result difference is within the standard deviation over data splits. Red indicates the best result for each domain split. Blue indicates the group of results that are close to the best performing result. The domain names are shortened for space: a: `amazon`, w: `webcam`, d: `dslr`, c: *Caltech256*

***Office* Dataset**   The *Office* dataset is a collection of images that provides three distinct domains: `amazon`, `webcam`, and `dslr`. The dataset has 31 categories consisting of common office objects such as chairs, backpacks and keyboards. The `amazon` domain contains product images (from amazon.com) containing a single object, centered, and usually on a white background. The `webcam` and `dslr` domains contain images taken in"the wild" using a webcam or a dslr camera, respectively. They are taken in an office setting and so have different lighting variation and background changes (see Figure 1 for some examples.) We use the SURF-BoW image features provided by the authors [1]. More details on how these features were computed can be found in [1]. The available features are vector quantized to 800 dimensions for all domains and additionally for the `dslr` domain there are 600 dimensional features available (we denote this as `dslr-600`).

***Office + Caltech256* Dataset**   This dataset consists of the 10 common categories shared by the *Office* and *Caltech256* datasets. To better compare to previously reported performance, we use the features provided by [15], which are also SURF-BoW 800 dimensional features.

***Bing* Dataset**   To demonstrate the effect that constraint set size has on run-time performance, we use the *Bing* dataset from [21], which has a larger number of images in each domain than *Office*. The source domain has images from the Bing search engine and the target domain is from the *Caltech256* benchmark. We run experiments using the first 20 categories and set the number of source examples per category to be 50. We use the train/test split from [21] and then vary the number of labeled target examples available from 5 to 25.

Baselines   We use the following baselines as a comparison in the experiments where applicable.[1]

- **svm**$_s$: A support vector machine using source training data.
- **svm**$_t$: A support vector machine using target training data.
- **arc-t**: A category general feature transform method proposed by [12]. We implement the transform learning and then apply both a KNN classifier (as originally proposed) and an SVM classifier.
- **hfa**: A max-margin transform approach that learns a latent common space between source and target as well as a classifier that can be applied to points in that common space [11].
- **gfk**: The geodesic flow kernel [15] applied to all source and target data (including test data). Following [15], we use a 1-nearest neighbor classifier with the kernel.

---

[1]We used the LIBSVM package [26] for kernelized methods and Liblinear [27] package for linear methods.

source	target	$\mathbf{svm}_t$	arc-t	hfa	mmdt
amazon	dslr-600	52.9 ± 0.7	58.2 ± 0.6	57.8 ± 0.6	62.3 ± 0.8
webcam	dslr-600	51.8 ± 0.6	58.2 ± 0.7	60.0 ± 0.6	63.3 ± 0.5

Table 3: Multiclass accuracy results on the standard supervised domain adaptation task with different feature dimensions in the source and target. The target domain is `dslr` for both cases.

source	$\mathbf{svm}_s$	arc-t	gfk	mmdt
amazon	10.3 ± 0.6	41.4 ± 0.3	38.9 ± 0.4	44.6 ± 0.3
webcam	51.6 ± 0.5	59.4 ± 0.4	62.9 ± 0.5	58.3 ± 0.5

Table 4: Multiclass accuracy results on the *Office* dataset for the domain shift of `webcam→dslr` for target test categories not seen in at training time. Following the experimental setup of [12]. We compare against pmt-svm [10] and ARC-t [12] using both knn and svm classification.

Standard Domain Adaptation Experiment For our first experiment, we use the *Office+Caltech256* domain adaptation benchmark dataset to evaluate multi-class accuracy in the standard domain adaptation setting where a few labeled examples are available for all categories in the target domain. We follow the setup of [1] and [15]: 20 training examples for `amazon` source (8 for all other domains as source) and 3 labeled examples per category for the target domain. We created 20 random train/test splits and averaged results across them.

The multi-class accuracy for each domain pair is shown in Table 2. Our method produced the highest multi-class accuracy for 9 out of 12 of the domain shifts and competitively on the other 3 shifts. This experiment demonstrates that our method achieves a high recognition performance and is able to outperform the most recent domain adaptation algorithms. Our method especially stands out in the settings where the domains are initially very different. The most similar domains in this dataset are `webcam` and `dslr` and we see that our algorithm does not perform as well on those two shifts as **gfk**. This fits with our intuition since **gfk** is a 1-nearest neighbor approach and so is more suitable when the domains are initially similar.

Additionally, an important observation is that our linear method on average outperforms all the baselines, even though they each learn a non-linear transformation.

Asymmetric Transform Experiment Next, we analyze the effectiveness of our asymmetric transform learning by experimenting with the source and target having different feature dimensions. We use the same experimental setup as previously, but use the *Office* dataset and the alternate representation for the `dslr` domain that is 600-dimensional (denoted as `dslr-600`). We compare against **svm**$_t$, **arc-t** and **hfa**, the baselines that can handle this scenario. The results are shown in Table 3. Again, we find that our method can effectively learn a feature representation for the target domain that optimizes the final classification objective.

Generalizing to Novel Categories Experiment We next consider the setting of practical importance where labeled target examples are not available for all objects. Recall that this is a setting that many category specific adaptation methods cannot generalize to, including **hfa** [11]. Therefore, we compare our results for this setting to the **arc-t** [12] method which learns a category independent feature transform and the **gfk** [15] method which learns a category independent kernel to compare the domains. Following the experimental setup of [12], we use the full *Office* dataset and allow 20 labeled examples per category in the source for `amazon` and 10 labeled examples for the first 15 object categories in the target (`dslr`). For the `webcam→dslr` shift, we use 8 labeled examples per category in the source for `webcam` and 4 labeled examples for the first 15 object categories in the target `dslr`.

The experimental results for the domain shift of `webcam→dslr` are evaluated and shown in Table 4; MMDT outperforms the baselines for the `amazon→dslr` shift and offers adaptive benefit over **svm**$_s$ for the shift from `webcam` to `dslr`. As in the first experiment, both **arc-t** and **gfk** use nearest neighbor classifiers on a learned kernel are more suitable to the shift between `webcam` and `dslr`, which are initially very similar.

Scaling to Larger Datasets Experiment With our last experiment we show that our method not only offers high accuracy performance it also scales well with an increasing dataset size. Specifi-
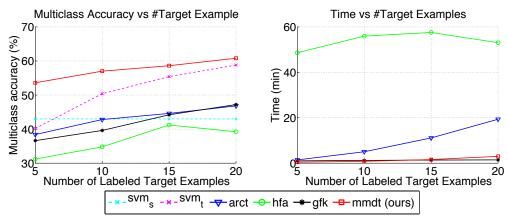
Figure 2: Left: multiclass accuracy on the *Bing* dataset using 50 training examples in the source and varying the number of available labeled examples in the target. Right: training time comparison.

cally, the number of constraints our algorithm optimizes scales linearly with the number of training points. Conversely, the number of constraints that need to be optimized for the **arc-t** baseline is quadratic in the number of training points.

To demonstrate the effect that constraint set size has on run-time performance, we use the *Bing* [21] dataset, which has a larger number of images in each domain than *Office*. The source domain has images from the Bing search engine and the target domain is from the *Caltech256* benchmark. We run experiments using the first 20 categories and set the number of source examples per category to be 50. We use the train/test split from [21] and then vary the number of labeled target examples available from 5 to 20. The left-hand plot in Figure 2 presents multi-class accuracy for this setup. Additionally, the training time of our method (run to convergence) and that of the baselines is shown on the right-hand plot.

Our **mmdt** method provides a considerable improvement over all the baselines in terms of multiclass accuracy. It is also considerably faster than all but the **gfk** method. An important point to note is that both our method and **arc-t** scale approximately linearly with the number of target training points which is empirical verification for our claims. Note that **hfa** and **gfk** do not vary significantly as the number of target training points increases. However, for **hfa** the main bottleneck time is consumed by a distance computation between each pair of training points. Therefore, since there are many more source training points than target, adding a few more target points does not significantly increase the overall time spent for this experiment, but would present a problem as the size of the dataset grew in general.

5 Conclusion

In this paper, we presented a feature learning technique for domain adaptation that combines the ability of feature transform-based methods to perform multi-task adaptation with the performance benefits of directly adapting classifier parameters.

We validated the computational efficiency and effectiveness of our method using two standard benchmarks used for image domain adaptation. Our experiments show that 1) our method is a competitive domain adaptation algorithm able to outperform previous methods, 2) is successfully able to generalize to novel target categories at test time, and 3) can learn asymmetric transformations. In addition, these benefits are offered through a framework that is scalable to larger datasets and achieves higher classification accuracy than previous approaches.

So far we have focused on linear transforms because of its speed and scalability; however, our method can also be kernelized to include nonlinear transforms. In future work, we would like to explore the kernelized version of our algorithm and especially experiment with the geodesic flow kernel as input to our algorithm.

References

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

[2] A. Farhadi and M. K. Tabrizi. Learning to recognize activities from the wrong view point. In *Proc. ECCV*, 2008.

[3] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer svm for video concept detection. In *CVPR*, 2009.

[4] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *Proc. CVPR*, 2010.

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

[6] D. McAllester P. Felzenszwalb, R. Girshick and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010.

[7] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. ICCV*, 2009.

[8] J. Yang, R. Yan, and A. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *ICDM Workshops*, 2007.

[9] X. Li. Regularized adaptation: Theory, algorithms and applications. In *PhD thesis, University of Washington, USA*, 2007.

[10] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Proc. ICCV*, 2011.

[11] Lixin Duan, Dong Xu, and Ivor W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proc. ICML*, 2012.

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

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

[14] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across different feature spaces. In *Proc. NIPS*, 2008.

[15] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, 2012.

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

[17] H. Daume III. Frustratingly easy domain adaptation. In *Proc. ACL*, 2007.

[18] S. Ben-david, J. Blitzer, K. Crammer, and O. Pereira. Analysis of representations for domain adaptation. In *In NIPS*. MIT Press, 2007.

[19] J. Jiang and C. X. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, 2007.

[20] J. Jiang. A literature survey on domain adaptation of statistical classifiers. `http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/`.

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

[22] W. Jiang, E. Zavesky, S. Chang, and A. Loui. Cross-domain learning methods for high-level visual concept classification. In *ICIP*, 2008.

[23] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *Journal of Machine Learning Research - Proceedings Track*, 27:17–36, 2012.

[24] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128, 2006.

[25] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[26] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[27] 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*, 9:1871–1874, 2008.