Towards Adapting ImageNet to Reality: Scalable Domain Adaptation with Implicit Low-rank Transformations

Erik Rodner
Judith Hoffman
Jeffrey Donahue
Trevor Darrell
Kate Saenko

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-154
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-154.html

August 20, 2013

Towards Adapting ImageNet to Reality: Scalable Domain Adaptation with Implicit Low-rank Transformations

Erik Rodner[1,2,*] Judy Hoffman[1,*] Jeff Donahue[1]
Trevor Darrell[1] Kate Saenko[3]
[1]ICSI & EECS UC Berkeley, [2]University of Jena, [3]UMass Lowell

Abstract

Images seen during test time are often not from the same distribution as images used for learning. This problem, known as domain shift, occurs when training classifiers from object-centric internet image databases and trying to apply them directly to scene understanding tasks. The consequence is often severe performance degradation and is one of the major barriers for the application of classifiers in real-world systems. In this paper, we show how to learn transform-based domain adaptation classifiers in a scalable manner. The key idea is to exploit an implicit rank constraint, originated from a max-margin domain adaptation formulation, to make optimization tractable. Experiments show that the transformation between domains can be very efficiently learned from data and easily applied to new categories. This begins to bridge the gap between large-scale internet image collections and object images captured in everyday life environments.

1. Introduction

Learning from huge datasets comprised of millions of images is one of the most promising directions towards closing the gap between human and machine visual recognition abilities. There has been tremendous success in the area of large-scale visual recognition [4] allowing for learning of tens of thousands of visual categories. However, in parallel, researchers have discovered the bias induced by current image databases and that performing visual recognition tasks across domains cripples performance [18]. Although this is especially common for smaller datasets, like Caltech-101 or the PASCAL VOC datasets [18], the way large image databases are collected (typically using internet search engines) also introduces an inherent bias. This can be seen for example when comparing object images of the ImageNet [4] and SUN2012 database [20] in Figure 1, where the "object-centric" data of ImageNet is of high resolution with centered objects as well as sometimes artificial backgrounds, and the SUN2012 objects are part of scene images leading to blurred appearances with a large degree of occlusion and truncation.

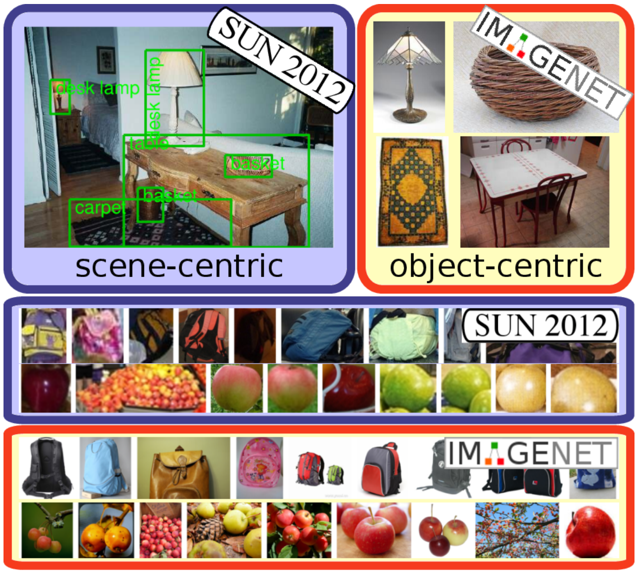

Figure 1. Dataset bias of ImageNet and the SUN2012 database shown for an indoor scene and for the categories *backpack* and *apple* on a bounding box level.

Transform-based domain adaptation overcomes the bias by learning a transformation between datasets. In contrast to classifier adaptation [1, 22, 3, 11], learning a transformation between feature spaces directly allows us to perform adaptation even for (new) categories that are not present in both datasets. Especially for large-scale recognition with a large number of categories, this is a crucial benefit, because we can learn category models for all the categories in a given source domain also in the target domain. Transformations can be learned in an unsupervised manner [12] or by using the labels present in both domains to maximize the margin of the classifier on the source and transformed target data [9, 5].

[*] both authors contributed equally

In this paper, we introduce a novel optimization method that enables transform-learning and associated domain adaptation methods to scale to "big data". We do this by a novel re-formulation of the optimization in [9] as direct dual coordinate descent and by exploiting an implicit rank constraint. Although we learn a linear transformation between domains, which has a quadratic size in the number of features used, our algorithm needs only a linear number of operations in each iteration in both feature dimensions (source and target domain) as well as the number of training examples. This is an important benefit compared to other methods that need to run in kernel space [12, 5] to overcome the high dimensionality of the transformation, a strategy impossible to apply for large-scale settings. The obtained scalability of our method is crucial as it allows the use of transform-based domain adaptation for datasets with a large number of categories and examples, settings in which previous techniques [12, 5, 9] were unable to run in reasonable time. Our experiments on different datasets show the various advantages of transform-based methods, such as generalization to new categories or even handling domains with different feature types.

2. Related Work

For the task of domain adaptation, two different sets of data are typically considered, the source and the target domain, which are drawn from similar but distinct distributions $p(\boldsymbol{x})$ and $p(\tilde{\boldsymbol{x}})$. The goal is to transfer knowledge from the source domain to the target domain. In the following, we briefly review related work done in the areas of domain adaptation as well as transfer learning. Although transfer learning [13] considers a change of the conditional distribution $p(y \mid \boldsymbol{x})$ rather than a change of the data distribution $p(\boldsymbol{x})$ as in domain adaptation, the methods in both areas often use similar principles and ideas.

Domain adaptation can be applied at different levels of the machine learning pipeline. For example, the adaptive SVM method [22] combines a target classifier and an existing source classifier by linear combination of their continuous outputs. This is related to adding a new regularization term to the SVM objective that forces the target SVM hyperplane parameter to be close to the source hyperplane [21]. Aytar and Zisserman [1] showed the importance of using a scale-invariant similarity measure for this regularization term. Furthermore, the authors of [3] proposed a combination of target, source and transductive SVM. More recently, Khosla *et al*. [11] introduced a method to jointly learn a "visual world model" common across all domains in combination with an additive bias term for each individual domain.

In general, classifier adaptation methods are often limited to cases where labeled training data is given for every class in the source as well as in the target domain. However, we often have a source domain with not only more training examples but also more labeled categories available. Exploiting all the information and learning visual classifiers for new categories in the target domain is possible with metric or transformation-based methods.

Another line of work was started by Gopalan *et al*. [8], who introduced domains as points on a manifold of subspaces. To perform domain adaptation, features are mapped to the subspaces induced by the geodesic from the source to the target domain. This yields several intermediate representations of the input data that can be used for learning a classifier. Gong *et al*. [7] showed how to circumvent sampling only a finite number of subspaces by expressing the representation as a kernel. In contrast, Tommasi *et al*. [17] tackled the domain adaptation problem by learning a shared subspace capturing domain-invariant properties of the categories. Learning for a new dataset is then done by learning an additional domain-specific transformation of the data.

The work of Saenko *et al*. [14] was one of the earliest papers to investigate domain adaptation challenges in visual recognition. The key idea of their work is to apply metric learning techniques that allow for estimating a category-independent metric which related target and source examples, and can be used in a nearest neighbor classifier. Kulis *et al*. [12] extended their work to asymmetric transformations and metrics using a Frobenius norm regularizer. A major bottleneck of their approach is the number of instance (linear) constraints, one for each pair of source and target examples, that need to be considered during optimization and the fact that transforms are learned independently of loss. Therefore, Hoffman *et al*. [9] recently showed how to jointly learn a transformation together with SVM parameters in a max-margin framework, which reduces the number of constraints to the number of categories. The linear transformation was quadratic in the feature dimensionality, and the kernelization as used by [9, 12] was quadratic in the number of training examples. This scales poorly with very large data, and as we show in the experiments section is intractable for even modestly large-scale data.

3. Scalable Transformation Learning

We introduce a method for learning a transformation, which is easy to apply, implement, and can be combined with other large-scale architectures. Our new scalable method can be applied to supervised domain adaptation, where we are given source training examples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ and target examples $\tilde{\mathcal{D}} = \{(\tilde{\boldsymbol{x}}_j, \tilde{y}_j)\}_{j=1}^{\tilde{n}}$.

Our goal is to learn a linear transformation $\mathbf{W}\tilde{\boldsymbol{x}}$ mapping a target training data point $\tilde{\boldsymbol{x}}$ to the source domain. The transformation is learned through an optimization framework which introduces linear constraints between transformed target training points and information from the source and thus generalizes the methods of [14, 12, 9]. To demonstrate the generality of our approach, we denote linear constraints in the source domain using hyperplanes $\boldsymbol{v}_i \in \mathbb{R}^D$ for $1 \leq i \leq m$. Let us denote with $\tilde{y}_{ij}$ a scalar which represents some measure of intended similarity between $\boldsymbol{v}_i$ and the target training data point $\tilde{\boldsymbol{x}}_j$. With this general notation, we can express the standard transformation learning problem with slack variables as follows:

$$\begin{aligned} &\min_{\mathbf{W},\{\boldsymbol{\eta}\}} && \frac{1}{2}\|\mathbf{W}\|_F^2 + \tilde{C}\sum_{i=1,j=1}^{m,\tilde{n}} (\eta_{ij})^p \\ &\text{s.t.} && \tilde{y}_{ij}\left(\boldsymbol{v}_i^T \mathbf{W} \tilde{\boldsymbol{x}}_j\right) \geq 1 - \eta_{ij},\ \eta_{ij} \geq 0 \quad \forall i,j \ . \end{aligned} \tag{1}$$

Note that this directly corresponds to the transformation learning problem proposed in [9]. Previous transformation learning techniques [14, 12, 9] used a Bregman divergence optimization technique [12], which scales quadratically in the number of target training examples (kernelized version) or the number of feature dimensions (linear version). For the large-scale scenario considered in this paper, this is impractical due to the large number of target training examples and categories given, as well as the high dimensionality of the features. Therefore, we show in a new analysis both how to use dual coordinate descent for the optimization of $\mathbf{W}$ and that $\mathbf{W}$ has a low-rank structure, which can be exploited to allow for efficient optimization as verified in our experimental evaluation.

3.1. Learning W with dual coordinate descent

We now re-formulate Eq. (1) as a vectorized optimization problem suitable for dual coordinate descent that allows us to use efficient optimization techniques. We use $\boldsymbol{w} = \text{vec}\left(\mathbf{W}\right)$ to denote the vectorized version of a matrix $\mathbf{W}$ obtained by concatenating the rows of the matrix into a single column vector. With this definition, we can write:

$$\|\mathbf{W}\|_F^2 = \|\text{vec}\left(\mathbf{W}\right)\|_2^2 = \|\boldsymbol{w}\|_2^2 \tag{2}$$

$$\boldsymbol{v}_i^T \mathbf{W} \tilde{\boldsymbol{x}}_j = \boldsymbol{w}^T \text{vec}\left(\boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T\right) \quad . \tag{3}$$

Let $\ell = m(j-1)+i$ be the index ranging over the target examples as well as the m hyperplanes in the source domain, which we also denote as $\ell = (i,j)$ for convenience. We now define a new set of "augmented" features as follows:

$$\boldsymbol{d}_\ell = \text{vec}\left(\boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T\right) \in \mathbb{R}^{D \times \tilde{D}} \quad , \tag{4}$$

$$t_\ell = \tilde{y}_{ij} \quad . \tag{5}$$

With these definitions, Eq. (1) is equivalent to a soft-margin SVM problem with training set $(\boldsymbol{d}_\ell, t_\ell)_{\ell=1}^{\tilde{n}\cdot K}$. We exploit this result of our analysis by using and modifying the efficient coordinate descent solver proposed in [10], which solves the SVM optimization problem in its dual form with respect to the dual variables α_ℓ:

$$\min_{\boldsymbol{\alpha} \geq 0} g(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \bar{\mathbf{Q}} \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha} \quad . \tag{6}$$

We have considered the L_2-SVM formulation ($p = 2$ in Eq. (1)), although our techniques presented in this paper also hold for the standard L_1-SVM case. The matrix $\mathbf{Q}$ is a regularized kernel matrix incorporating the labels, *i.e.* $\bar{Q}_{\ell,\ell'} = t_\ell t_{\ell'}\, \boldsymbol{d}_\ell^T \boldsymbol{d}_{\ell'} + \lambda\,\delta\left[i = j\right]$ with $\lambda = \frac{1}{2\tilde{C}}$. The key idea is to maintain and update $\boldsymbol{w}$ explicitly:

$$\boldsymbol{w} = \sum_{\ell=1}^{m\cdot\tilde{n}} \alpha_\ell\, t_\ell\, \boldsymbol{d}_\ell \quad . \tag{7}$$

This dramatically reduces the computational complexity of the gradient computation in α_ℓ compared to classical dual solvers commonly used for kernel SVM:

$$\nabla_\ell\, g(\boldsymbol{\alpha}) = y_i \cdot \boldsymbol{w}^T \boldsymbol{d}_\ell + \lambda\,\alpha_\ell - 1 \quad , \tag{8}$$

which requires a number of operations linear in the dimensionality of the given (augmented) feature vectors $\boldsymbol{d}_\ell$. A single coordinate descent step can then be done by:

$$\alpha_\ell \leftarrow \max\left(0, \alpha_\ell - \frac{\nabla_\ell\, g(\boldsymbol{\alpha})}{\|\boldsymbol{d}_\ell\| + \lambda}\right) \tag{9}$$

in the same asymptotic time. Note that explicitly maintaining $\boldsymbol{w}$ is essential for easily computable coordinate descent steps; therefore, given the change $\triangle\alpha_\ell$ of the step, we have to update $\boldsymbol{w}$ so that Eq. (7) is again fulfilled:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \triangle\alpha_\ell\, t_\ell\, \boldsymbol{d}_\ell \quad . \tag{10}$$

Whereas, for standard learning problems an iteration with only a linaer number of operations in the feature dimensionality already provides a sufficient speed-up, this is not the case when learning domain transformations $\mathbf{W}$. When the dimension of the source and target feature space is D and $\tilde{D}$, respectively, the features $\boldsymbol{d}_\ell$ of the augmented training set have a dimensionality of $D \cdot \tilde{D}$, which is impractical for vision tasks with high-dimensional input features. For this reason, we show in the following how we can efficiently exploit an implicit low-rank structure of $\mathbf{W}$ for a small number of hyperplanes inducing the constraints.

3.2. Implicit low-rank structure of the transform

To derive a low-rank structure of the transformation matrix, let us recall Eq. (7) in matrix notation:

$$\mathbf{W} = \sum_{i=1,j=1}^{m,\tilde{n}} \alpha_\ell\, \boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T = \sum_{i=1}^{m} \boldsymbol{v}_i \left(\sum_{j=1}^{\tilde{n}} \alpha_\ell\, \tilde{\boldsymbol{x}}_j^T\right) \quad . \tag{11}$$

Thus, $\mathbf{W}$ is a sum of m dyadic products and therefore a matrix of at most rank m, with m being the number of hyperplanes in the source used to generate constraints. Note that for our experiments, we use the MMDT method [9], for which the number of hyperplanes equals the number of object categories we seek to classify. We can exploit the low rank structure by representing $\mathbf{W}$ indirectly using:

$$\boldsymbol{\beta}_i = \sum_{j=1}^{\tilde{n}} \alpha_\ell \, \tilde{\boldsymbol{x}}_j^T \quad . \tag{12}$$

This is especially useful when the number of categories is small compared to the dimension of the source domain, because $[\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m]$ only has a size of $m \times \tilde{D}$ instead of $D \times \tilde{D}$ for $\mathbf{W}$. It also allows for very efficient updates with a computation time even independent of the number of categories.

First, with the given low-rank representation and the $\boldsymbol{\beta}_i$, we can easily speed up the scalar product in Eq. (8):

$$\boldsymbol{w}^T \boldsymbol{d}_\ell = \boldsymbol{v}_i^T \mathbf{W} \tilde{\boldsymbol{x}}_j = \sum_{i'=1}^{m} \boldsymbol{v}_i^T \boldsymbol{v}_{i'} \, \boldsymbol{\beta}_{i'}^T \tilde{\boldsymbol{x}}_j = \sum_{i'=1}^{m} \rho_{i,i'} \boldsymbol{\beta}_{i'}^T \tilde{\boldsymbol{x}}_j \ ,$$

where the matrix $\mathbf{R} = (\rho_{i,i'}) \in \mathbb{R}^{m \times m}$ can be calculated in advance. Furthermore, we can cache $\boldsymbol{\beta}_{i'}^T \tilde{\boldsymbol{x}}_j$, leading to a only a cost of $\mathcal{O}(\tilde{D})$ (Details in Sect. 3.3).

The matrix $\mathbf{R}$ contains the correlations between hyperplanes and also shows the multi-task fashion of the approach: the $\boldsymbol{\beta}_i$ vectors can be seen as linear classifiers in the target domain and the matrix $\mathbf{R}$ combines all of them taking the dependencies between classes into account. This is an interesting and important aspect of our method in scenarios with a large number of categories. A linear classifier $\boldsymbol{v}_i$ is mapped to the target domain by:

$$\tilde{\boldsymbol{v}}_i = \mathbf{W}^T \boldsymbol{v} = \sum_{i'=1}^{m} \boldsymbol{\beta}_i \, \rho_{i,i'} \tag{13}$$

and therefore uses correlations to other categories, which is similar to transfer learning approaches [16]. To allow for efficient α-updates in Eq. (9), we further need to consider an efficient calculation of the feature vector norm $\|\boldsymbol{d}_\ell\|^2$:

$$\|\boldsymbol{d}_\ell\|^2 = \|\boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T\|^2 = \|\boldsymbol{v}_i\|^2 \cdot \|\tilde{\boldsymbol{x}}_j\|^2 \quad . \tag{14}$$

Finally the update formula in Eq. (10) can be translated into updating $\boldsymbol{\beta}_i$ in only $\mathcal{O}(\tilde{D})$ operations:

$$\boldsymbol{\beta}_i \leftarrow \boldsymbol{\beta}_i + \triangle\alpha_\ell \, t_\ell \, \tilde{\boldsymbol{x}}_j \quad . \tag{15}$$

3.3. Algorithmic details and complexity

In this section, we briefly discuss some implementation details of the solver used in our experiments (Sect. 5). Code for our efficient dual coordinate descent transform solver, adapted from `liblinear` [6], will be made publicly available online. The shrinking heuristics presented in [10] that maintain a set $\mathcal{S}$ of dual variables that have been set to zero during optimization and that are likely not to change in the future are also implemented in our approach. An algorithmic outline of our approach is given in Figure 2.

	α_ℓ update	$\boldsymbol{W}$ update
Our approach	$\mathcal{O}(\tilde{D})$	$\mathcal{O}(\tilde{D})$
Direct rep. of $\mathbf{W}$	$\mathcal{O}(D \cdot \tilde{D})$	$\mathcal{O}(D \cdot \tilde{D})$
Bregman opt. (kernel) [12]	-	$\mathcal{O}(n \cdot \tilde{n})$
Bregman opt. (linear)	-	$\mathcal{O}(D \cdot \tilde{D})$

Table 1. Asymptotic times for one iteration of the optimization, where a single constraint is taken into account. There are n source training points of dimension D and $\tilde{n}$ target training points of dimension $\tilde{D}$.

```
Optimization of $\mathbf{W}$ in our method
1. For $1 \leq i, i' \leq K$: $\rho_{i,i'} = \boldsymbol{v}_i^T \boldsymbol{v}_{i'}$
2. For $1 \leq j \leq \tilde{n}$: $q_j = \|\tilde{\boldsymbol{x}}_j\|^2$
3. Repeat until convergence of $\boldsymbol{\alpha}$
    (a) Loop through the active set $\ell = (i, j) \in \mathcal{S}$
        i. $s = \sum_{i'=1}^{m} \rho_{i,i'} \, \boldsymbol{\beta}_{i'}^T \tilde{\boldsymbol{x}}_j$ using cached $\boldsymbol{\beta}_{i'}^T \tilde{\boldsymbol{x}}_j$
        ii. $G = \delta\,[\tilde{y}_j = i] \cdot s + \lambda \alpha_\ell - 1$
        iii. $PG = \begin{cases} G & \alpha_\ell > 0 \\ \min(G, 0) & \alpha_\ell = 0 \end{cases}$
        iv. if $PG \neq 0$
            A. $\alpha_\ell \leftarrow \max(\alpha_\ell - G/(q_j \cdot \rho_{i,i} + \lambda), 0)$
            B. $\boldsymbol{\beta}_i \leftarrow \boldsymbol{\beta}_i + \triangle\alpha_\ell \, \delta\,[\tilde{y}_j = i] \, \tilde{\boldsymbol{x}}_j$
```

Figure 2. Pseudo code for $\mathbf{W}$ optimization without shrinking heuristics and caching details.

Computational complexity The asymptotic times are summarized in Table 1. While the asymptotic time for the kernel Bregman optimization used in [12, 9] depends on the number of source examples, the time we need to iteratively take one constraint into account is independent of the number of examples in either the source or target domain. One pass over all constraints takes time $\mathcal{O}(\tilde{n} \cdot m)$, which finally leads to a linear asymptotic time in the product of the number of target points and the target dimension, independent of the size of the source training set. Therefore, our method allows for using transform-based adaptation in large-scale settings, where previous approaches [12, 14] were unable to run at all.

Identity regularizer As described in previous sections, the transformation $\mathbf{W}$ has a low-rank structure when using the original MMDT formulation. In situations with only a small number of categories, this can be too restrictive for the class of transformations. However, when using the identity regularizer $\|\mathbf{W} - \mathbf{I}\|_F^2$, we obtain $\mathbf{W} = \mathbf{I} + \sum_i \boldsymbol{v}_i \boldsymbol{\beta}_i^T$, which allows to estimate full rank matrices. The efficient updates in each coordinate descent iteration do not change significantly and are omitted here due to the lack of space.

Caching techniques As mentioned earlier, we cache the scalar products $\boldsymbol{\beta}_i^T \tilde{\boldsymbol{x}}_j$ to allow for fast computation. Each time the vector $\boldsymbol{\beta}_i$ is updated, all $\tilde{n}$ cached values $\boldsymbol{\beta}_i^T \tilde{\boldsymbol{x}}_j$ are invalid and have to be updated in one of the next steps where $\tilde{\boldsymbol{x}}_j$ is taken into account. When using a fully randomized order of the dual variables α_ℓ as suggested by [10], this invalidation happens on average every Kth step leading to a low probability that the cached value can be used in between. For this reason, we only consider a random order of j and iterate normally through all the K categories. Therefore, we can use the cached values in each of the K blocks.

Convergence properties Our solver maintains all the convergence properties of dual coordinate descent solvers. In particular, we have at least a linear convergence rate [10, Theorem 1] and an ϵ-accurate solution can be obtained in $\mathcal{O}(-\log(\epsilon))$ iterations.

4. Domain adaptation datasets

In the following, we briefly describe the datasets used in our experiments for the source as well as the target domain.

ImageNet ILSVRC2010 to SUN2012 Whereas ImageNet images were obtained using object category names and therefore contain a large portion of advertisement images, the creation of the SUN database was done by searching for scene categories and labeling objects in the images afterwards. Therefore, there is a significant domain shift between the two datasets (Figure 1). In fact, Torralba and Efros's experiments in [18] consistently showed that the domain shift between ImageNet and SUN is one of the most severe among all pairs of benchmark datasets they surveyed.

For this reason, we assembled a new challenge for domain adaptation methods by matching a subset of the object categories from the SUN2012 dataset [20] (target domain) with the ones present in the hierarchy of the ImageNet 2010 challenge [2] (source domain). The matching of the category names in both datasets is done by using the manually maintained WordNet matchings of the SUN2012 dataset [20]. Using the WordNet descriptions, a large set of SUN2012 descriptions can be mapped to nodes of the WordNet subgraph related to the ILSVRC2010 challenge; *i.e.*, to sets of ILSVRC2010 categories (leaf nodes). Finally, we consider pairs of SUN2012 labels and ILSVRC2010 category sets that lead to more than 20 examples. This leads to a total of 84 categories[1]. The final set of examples consists of cropped bounding boxes not labeled as difficult or truncated. Classification with these examples without context knowledge can be considered as very challenging.

To allow for easy reproducibility of the results, we use the bag of visual words (BoW) features provided for the ImageNet challenge. Furthermore, features in the SUN database are extracted by computing bag of visual words features inside of the given bounding boxes. This is also done with the feature extraction code provided for the ImageNet challenge.

Bing/Caltech256 dataset We also use the Bing dataset of [3], which contains images for each category of the Caltech256 dataset. In contrast to the ImageNet/SUN2012 scenario, both datasets have been created using internet search images and category keywords. In total, this dataset consists of 256 object categories. Features for this dataset are provided by the authors of [3].

5. Experiments

In our experiments, we give empirical validation for the following claims:

1. Our optimization algorithm allows for significantly faster learning than the one used by [9] without loss in recognition performance (Sect. 5.2).
2. Our transform-based approach can be used for large-scale domain adaptation datasets and achieves state-of-the-art performance, significantly outperforming the geodesic flow kernel method of [7] (Sect. 5.3).
3. We can learn a transformation between large-scale datasets that can be used for transferring new category models without any target training examples (Sect. 5.4) even in the case of different feature dimensions (Sect. 5.5).

5.1. Baseline methods

We compare our approach to the standard domain adaptation baseline, which is a linear SVM trained with only target or only source training examples (*SVM-Target*/*SVM-Source*). Note that for new category experiments, where some classes do not have training examples in the target domain, the *SVM-Target* baseline cannot be used. Furthermore, we evaluate the performance of the geodesic flow kernel (*GFK*) presented by [7] and integrated in a nearest neighbor approach. The metric learning approach of [12] (*ARC-t*) and the shared latent space method of [5] (*HFA*) can only be compared to our approach in a medium-scale experiment which is tractable for kernelized methods. For our experiments, we always use the source code from the authors.

We refer to our method as large-scale max-margin domain transform (*LS-MMDT*) in the following.

[1] tree, chair, cabinet, table, lamp, curtain, box, car, bed, mountain, desk, fence, mirror, skyscraper, bottle, rug, basket, bench, towel, vase, bannister, ball, stove, bookcase, magazine, refrigerator, bucket, clock, glass, hat, oven, boat, fan, shoe, dishwasher, telephone, airplane, loudspeaker, apparel, keyboard, bar, gate, bus, mug, bridge, umbrella, bicycle, backpack, laptop, washer, bathtub, roof, pitcher, fish, tower, flower, apple, file, teapot, minibike, printer, garage, guitar, ashcan, dog, dune, piano, ship, crane, newspaper, mouse, microphone, cliff, bell, elephant, shirt, toaster, orange, remote control, knife, helmet, grape, stick, shop

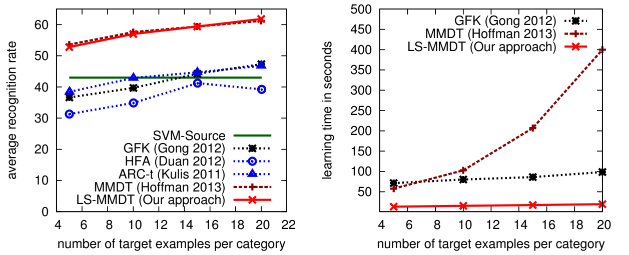

Figure 3. Medium-scale experiment: recognition rates and learning times when using the first 20 categories of the Bing/Caltech256 (source/target) dataset. Times of ARC-t [12] and HFA [5] are off-scale (12min and 55min for 10 target points per category).

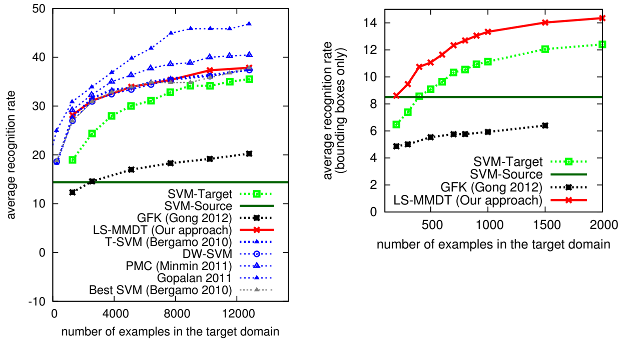

Figure 4. Large-scale experiment with the Bing/Caltech256 domain shift (76K source examples; left) as well as the ImageNet/SUN2012 domain shift (8K source examples; right) and a varying number of target examples.

5.2. Comparison to other adaptation methods

We first evaluate our approach on a medium-scale dataset comprised of the first 20 categories of the Bing/Caltech dataset. This setup is also used in [9] and allows us to compare our new optimization technique with the one used by [9] and also with other state-of-the-art domain adaptation methods [12, 5, 7]. We use the data splits provided by [3] and the Bing dataset is used as source domain with 50 source examples per category. Figure 3 contains a plot for the recognition results (left) and the training time (right plot) with respect to the number of target training examples per category in the Caltech dataset. As Figure 3 shows, our solver is significantly faster than the one used in [9] and achieves the same recognition accuracy. Furthermore, it outperforms other state-of-the-art methods, like ARC-t [12], HFA [5], and GFK [7], in both learning time and recognition accuracy.

5.3. Experiments with a large number of categories

In the next experiment, we use the Bing/Caltech256 dataset [3] with all 256 categories and our Imagenet/SUN2012 subset, settings in which the optimization techniques used in [9] *cannot be applied* due to the large number of target training examples. Furthermore, we test the performance of our method on the new domain dataset presented in Sect. 4 and we restrict the comparison to methods that provide generalization to new categories.

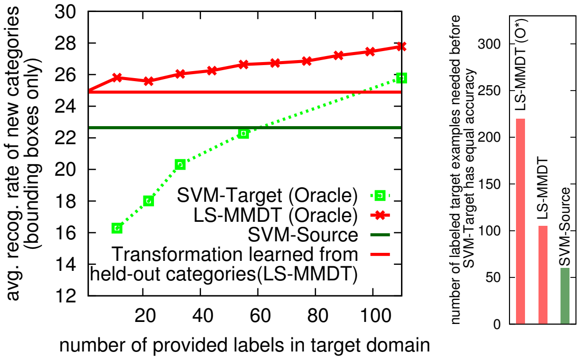

Figure 6. New category scenario: our approach is used to learn a transformation from held-out categories and to transfer new category models directly from the source domain without target examples. The performance is compared to an oracle SVM-Target and MMDT that use target examples from the held-out categories.

The results are given in Figure 4 and we see that we outperform again the geodesic flow method of [7] in both cases. Focusing on the right plot (Imagenet/SUN2012 dataset), notice that our method continues to have a performance benefit over SVM-Target even as the number of labeled target examples increases. This is due to the small number of training examples available for several of the categories, which is typical for real-world datasets [15]. Providing more labeled training data is only possible for some of the categories and without adaptation the recognition rates of less common classes cannot be improved.

Figure 5 shows some of the results we obtained for in-scene classification and 700 provided target training examples, where during test time we are given ground-truth bounding boxes and context knowledge about the set of objects present in the image. The goal of the algorithm is then to assign the weak labels to the given bounding-boxes. With this type of scene knowledge and by only considering images with more than one category, we obtain an accuracy of 59.21% compared to 57.53% for SVM-Target and 53.14% for SVM-Source. In contrast to [19], we are not given the exact number of objects for each category in the image, making our problem setting more difficult and realistic.

5.4. Transferring new category models

A key benefit of our method is the possibility of transferring category models to the target domain even when no target domain examples are available at all. In the following experiment, we selected 11 categories[2] from our ImageNet/SUN2012 dataset and only provided training examples in the source domain for them. The transformation is

[2] laptop,phone,toaster,keyboard,fan,printer,teapot,chair,basket,clock,bottle

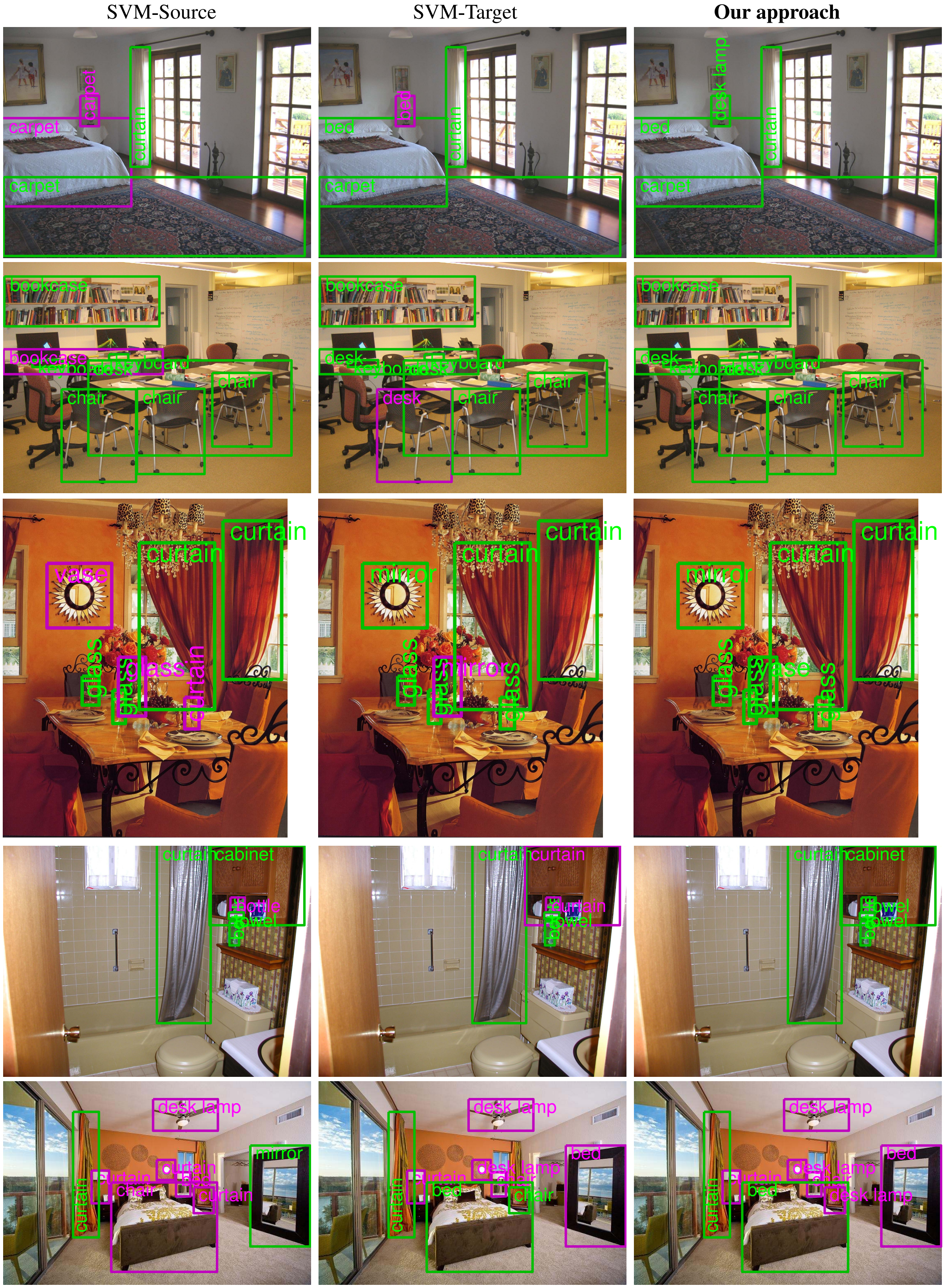

Figure 5. Results for object classification with given bounding boxes and scene prior knowledge: columns show the results of (1) SVM-Source, (2) SVM-Target, and (3) transform-based domain adaptation using our method. Correct classifications are highlighted with green borders. The figure is best viewed in color.

learned from all other categories with both labeled examples in the target and the source domain.

As we can see in Figure 6, this transfer method ("Transf. learned from other categories") even outperforms learning in the target domain (SVM-Target Oracle) with up to 100 labeled training examples. Especially with large-scale datasets, like ImageNet, this ability of our fast transform-based adaptation method provides a huge advantage and allows using all visual categories provided in the source as well as in the target domain. Furthermore, the experiment shows that we indeed learn a category-invariant transformation that can compensate for the observed dataset bias [18].

5.5. Adapting from different feature types

Transform-based domain adaptation can be also applied when source and target domain have different feature dimensionality. To show the applicability of our method in this setting we use the same setup as in the previous experiment, but we computed 1500-dimensional BoW features for objects in the SUN2012 dataset and learned a transformation from the 1000 dimensional features in the ImageNet dataset. Adaptation with our approach achieves a recognition rate of 18.2% compared to 16.9% of SVM-Target using one target training example per category. This can be seen as one of the most difficult adaptation scenarios, where we estimate the domain transformation from different categories and between completely different feature spaces.

6. Conclusions

In this paper, we showed how to extend transform-based domain adaptation towards large-scale scenarios. Our method allows for efficient estimation of a category-invariant domain transformation in the cases of large feature dimensionality and a large number of training examples. This is done by exploiting an implicit low-rank structure of the transformation and by making explicit use of a close connection to standard max-margin problems and efficient optimization techniques for them. Our method is easy to implement and apply, and achieves significant performance gains when adapting visual recognition models learned from biased internet sources to real-world scene understanding datasets.

An important take-home message of this paper is that collecting more and more annotated visual data does not necessarily help for solving scene understanding in general. However, domain adaptation can help to bridge the gap by learning category-invariant transformations without significant additional computational overhead.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Proc. ICCV*, 2011. 1, 2

[2] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge, 2010. http://www.imagenet.org/challenges/LSVRC/2010/. 5

[3] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Proc. NIPS*, 2010. 1, 2, 5, 6

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009. 1

[5] L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proc. ICML*, 2012. 1, 2, 5, 6

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 4

[7] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, 2012. 2, 5, 6

[8] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. ICCV*, 2011. 2

[9] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. In *Proc. ICLR*, 2013. 1, 2, 3, 4, 5, 6

[10] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proc. ICML*, 2008. 3, 4, 5

[11] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *Proc. ECCV*, 2012. 1, 2

[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. 1, 2, 3, 4, 5, 6

[13] S. J. Pan and Q. Yang. A survey on transfer learning. *Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010. 2

[14] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, pages 213–226, 2010. 2, 3, 4

[15] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Proc. CVPR*, pages 1481–1488, 2011. 6

[16] T. Tommasi and B. Caputo. The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In *Proc. BMVC*, 2009. 4

[17] T. Tommasi, N. Quadrianto, B. Caputo, and C. H. Lampert. Beyond dataset bias: Multi-task unaligned shared knowledge transfer. In *Proc. ACCV*, 2012. 2

[18] A. Torralba and A. Efros. Unbiased look at dataset bias. In *Proc. CVPR*, 2011. 1, 5, 8

[19] G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *Proc. CVPR*, pages 3525–3532, 2010. 6

[20] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, pages 3485–3492, 2010. 1, 5

[21] J. Yang, R. Yan, and A. G. Hauptmann. Adapting SVM classifiers to data with shifted distributions. In *Proc. ICDMW*, pages 69–76, 2007. 2

[22] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. *ACM Multimedia*, 2007. 1, 2