Transform-based Domain Adaptation for Big Data

Erik Rodner
University of Jena

Judy Hoffman
ICSI/EECS UC Berkeley

Jeff Donahue
ICSI/EECS UC Berkeley

Trevor Darrell
ICSI/EECS UC Berkeley

Kate Saenko
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[1].

1 Introduction

There has been tremendous success in the area of large-scale visual recognition [3] 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 [12]. Although this is especially common for smaller datasets, like Caltech-101 or the PASCAL VOC datasets [12], the way large image databases are collected also introduces an inherent bias.

Transform-based domain adaptation overcomes the bias by learning a transformation between datasets. In contrast to classifier adaptation [1, 14, 2, 8], learning a transformation between feature spaces directly allows us to perform adaptation even for new categories. Especially for large-scale recognition with a large number of categories, this is a crucial benefit, because we can learn category models for all categories in a given source domain also in the target domain.

In our work, 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 [6] as an SVM learning problem 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 kernel methods [9, 4] that overcome the high dimensionality of the transformation by dualization, 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 [9, 4, 6] were unable to run in reasonable time. Our experiments show the advantages of

[1] Source code can be found at: `https://github.com/erodner/liblinear-mmdt`. A version of the paper was also presented in the ICCV 2013 VisDA workshop.

transform-based methods, such as generalization to new categories or even handling domains with different feature types [10].

2 Scalable Transformation Learning

Our new scalable method can be applied to supervised domain adaptation, where we are given source training examples $\mathcal{D} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{n}$ 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 [11, 9, 6]. We denote linear constraints in the source domain using m hyperplanes $\boldsymbol{v}_i \in \mathbb{R}^D$ for $1 \leq i \leq m$. For the approach of [6] these hyperplanes correspond to a hyperplanes of a one-vs-all classifier, but we prefer a more general formulation which also includes the metric learning method of [9] Let us denote with $\tilde{y}_{ij}$ a scalar which controls the sign of the corresponding linear constraint. In the case of [6], this is the binary category label and in the case of [9], $\tilde{y}_{ij}$ is related to the intended similarity between $\boldsymbol{v}_i$ and $\tilde{\boldsymbol{x}}_j$. With this general notation, we can express the standard transformation learning problem with slack variables as follows:

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

Note that this directly corresponds to the transformation learning problem proposed in [6]. Previous transformation learning techniques [11, 9, 6] used a Bregman divergence optimization technique [9], which scales quadratically in the number of target training examples (kernelized version) or the number of feature dimensions (linear version).

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 $\boldsymbol{w} = \text{vec}\,(\mathbf{W})$ 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 $\|\mathbf{W}\|_F^2 = \|\boldsymbol{w}\|_2^2$ and $\boldsymbol{v}_i^T \mathbf{W}\tilde{\boldsymbol{x}}_j = \boldsymbol{w}^T \text{vec}\left(\boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T\right)$. Let $\ell = m(j-1) + i$ be the index ranging over the target examples as well as the m hyperplanes in the source domain. We now define a new set of "augmented" features as follows $\boldsymbol{d}_\ell = \text{vec}\left(\boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T\right)$ and $t_\ell = \tilde{y}_{ij}$. With these definitions, Eq. (1) is equivalent to a soft-margin SVM problem with training set $(\boldsymbol{d}_\ell, t_\ell)_{\ell=1}^{\tilde{n}\cdot m}$. We exploit this result of our analysis by using and modifying the efficient coordinate descent solver proposed in [7]. The key idea is to maintain and update $\boldsymbol{w} = \sum_{\ell=1}^{m\cdot\tilde{n}} \alpha_\ell\ t_\ell\ \boldsymbol{d}_\ell$ explicitly, which leads to a linear time complexity for a single coordinate descent step. Whereas, for standard learning problems an iteration with only a linear 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 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.

Implicit low-rank structure of the transform To derive a low-rank structure of the transformation matrix, let us recall the representation of $\boldsymbol{w}$ as a weighted sum of training examples $\boldsymbol{d}_\ell$ in matrix notation:

$$\mathbf{W} = \sum\nolimits_{i,j=1}^{m,\tilde{n}} \alpha_\ell\ \boldsymbol{v}_i \cdot \tilde{\boldsymbol{x}}_j^T = \sum\nolimits_{i=1}^{m} \boldsymbol{v}_i \Big(\sum\nolimits_{j=1}^{\tilde{n}} \alpha_\ell\, \tilde{\boldsymbol{x}}_j^T\Big)\ .$$

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 [6], 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 $\boldsymbol{\beta}_i = \sum_{j=1}^{\tilde{n}} \alpha_\ell\, \tilde{\boldsymbol{x}}_j^T$. 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

	α_ℓ update	Indirect $\boldsymbol{W}$ update
Our approach	$\mathcal{O}(\tilde{D})$	$\mathcal{O}(\tilde{D})$
Direct representation of $\mathbf{W}$	$\mathcal{O}(D \cdot \tilde{D})$	$\mathcal{O}(D \cdot \tilde{D})$
Bregman optimization (kernel) [9]	-	$\mathcal{O}(n \cdot \tilde{n})$
Bregman optimization (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 $\tilde{D}$.

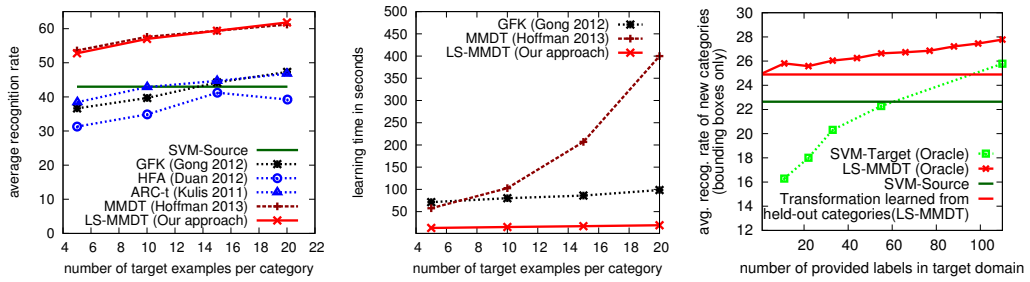

Figure 1: (Left) recognition rates and (Center) learning times when using the first 20 categories of the Bing/Caltech256 (source/target) dataset. Times of ARC-t [9] and HFA [4] are off-scale (12min and 55min for 10 target points per category). (Right) New category scenario: our approach is used to learn 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.

$D \times \tilde{D}$ for $\mathbf{W}$. It also allows for very efficient updates with a computation time even independent of the number of categories. Details about the algorithm and further speed-ups achieved with caching are given in [10]. The asymptotic time needed in each iteration of the solver is shown in Table 1.

3 Experiments

In our experiments, we give empirical validation that our optimization algorithm allows for significantly faster learning than the one used by [6] without loss in recognition performance and that we can learn a transformation between large-scale datasets that can be used for transferring new category models without any target training examples. Further experimental results are given in the corresponding technical report [10].

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*). Furthermore, we evaluate the performance of the geodesic flow kernel (*GFK*) presented by [5] and integrated in a nearest neighbor approach. The metric learning approach of [9] (*ARC-t*) and the shared latent space method of [4] (*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. The parameter $\tilde{C}$ is tuned using cross-validation on a smaller dataset and as features we used pre-computed features provided with the datasets [6].

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 [6] and allows us to compare our new optimization technique with the one used by [6] and also with other state-of-the-art domain adaptation methods [9, 4, 5]. We use the data splits provided by [2] and the Bing dataset is used as source domain with 50 source examples per category. Figure 1 contains a plot for the recognition results (left) and the training time (center plot) with respect to the number of target training examples per category in the Caltech dataset. As Figure 1 shows, our solver is significantly faster than the one used in [6] and achieves the same recognition accuracy.

Figure 2: 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.

Furthermore, it outperforms other state-of-the-art methods, like ARC-t [9], HFA [4], and GFK [5], in both learning time and recognition accuracy.

In-scene classification ImageNet/SUN2012 Figure 2 shows some of the results we obtained for in-scene classification with the ImageNet/SUN2012 datasets (Target domain: SUN2012) 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 [13], we are not given the exact number of objects for each category in the image, making our problem setting more difficult and realistic.

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

As we can see in Figure 1, this transfer method ("Transf. learned from held-out 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 [12].

4 Conclusions

We briefly outlined 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 models learned from biased internet sources to real-world scene understanding datasets.

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

References

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

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

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009.

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

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

[6] Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. In *Proc. ICLR*, 2013.

[7] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proc. ICML*, 2008.

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

[9] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. CVPR*, 2011.

[10] Erik Rodner, Judith Hoffman, Jeffrey Donahue, Trevor Darrell, and Kate Saenko. Towards adapting imagenet to reality: Scalable domain adaptation with implicit low-rank transformations. Technical Report UCB/EECS-2013-154, EECS Department, University of California, Berkeley, Aug 2013.

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

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

[13] Gang Wang, David Forsyth, and Derek Hoiem. Comparative object similarity for improved recognition with few or no examples. In *Proc. CVPR*, pages 3525–3532, 2010.

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