Scalable Transform-based Domain Adaptation

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

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. Source code can be found at: https://github.com/erodner/liblinear-mmdt.

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, 13, 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 reformulation 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 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 $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ 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 [11, 9, 6]. 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:

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

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 reformulate 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

*both authors contributed equally

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 $(\boldsymbol{d}_\ell, t_\ell)_{\ell=1}^{\tilde{n}\cdot K}$. 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:

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

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 $D \times \tilde{D}$ 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]. A summary of the asymptotic time needed in each iteration of the solver is shown in Table 1.

3. Experiments

We only give a brief summary of our experimental results here and refer the reader to [10] for details:

1. An evaluation on the Bing/Caltech dataset of [2] shows that our algorithm is several orders of magnitude faster than [9, 5, 4, 6]. Furthermore, it achieves the same performance as [6] and therefore outperforms [9, 5, 4] in terms of accuracy (Figure 1).

	α_ℓ update	Indirect $\boldsymbol{W}$ 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) [9]	-	$\mathcal{O}(n \cdot \tilde{n})$
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 $\tilde{D}$.

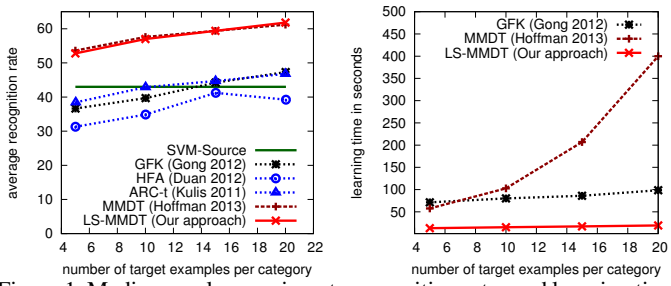

Figure 1. Medium-scale experiment: recognition rates and learning times when using the first 20 categories of the Bing/Caltech256 (source/target) dataset. Times of ARC-t [9] and HFA [4] are off-scale (12min and 55min for 10 target points per category).

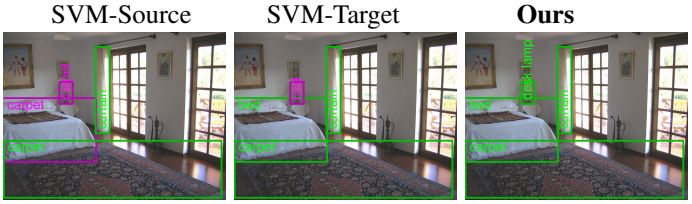

Figure 2. Results for object classification with given bounding boxes and scene prior knowledge: columns show the results of (1) SVM-Source, (2) SVM-Target, and (3) transform-based domain adaptation using our method. Correct classifications are highlighted with green borders. The figure is best viewed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

2. Experiments on a new ImageNet/SUN domain adaptation challenge show significant performance gains of our adaptation algorithm, especially when transferring new category models to the target domain.
3. The approach is even able to handle different feature dimensions in source and target domain.

Figure 2 shows some results from a bounding box recognition task, where category models are adaptated from ImageNet. Note that running previous methods [9, 6] is impossible in this scenario, because of the large number of categories and examples.

4. Conclusions

We briefly showed how to extend transform-based domain adaptation towards large-scale scenarios with a large number of examples and feature dimensionality. The method is easy to implement and to apply and achieves significant performance gains in several different adaptation tasks.

References

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

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

[3] 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

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

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

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

[7] 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. 2

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

[9] 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

[10] E. Rodner, J. Hoffman, J. Donahue, T. Darrell, and K. 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. 1, 2

[11] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, pages 213–226, 2010. 1

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

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