Cost-sensitive Structured SVM for Multi-category Domain Adaptation

Jiaolong Xu[1,2], Sebastian Ramos[1,2], David Vázquez[2], Antonio M. López[1,2]
[1]Computer Vision Center
[2]C. Sc. Dpt. Univ. Autònoma de Barcelona, Spain
{jiaolong, sramosp, dvazquez, antonio}@cvc.uab.es

***Abstract*—Domain adaptation addresses the problem of accuracy drop that a classifier may suffer when the training data (source domain) and the testing data (target domain) are drawn from different distributions. In this work, we focus on domain adaptation for structured SVM (SSVM). We propose a cost-sensitive domain adaptation method for SSVM, namely COSS-SSVM. In particular, during the re-training of an adapted classifier based on target and source data, the idea that we explore consists in introducing a non-zero cost even for correctly classified source domain samples. Eventually, we aim to learn a more target-oriented classifier by not rewarding (zero loss) properly classified source-domain training samples. We assess the effectiveness of COSS-SSVM on multi-category object recognition.**

I. Introduction

Large amounts of visual data are being generated daily and made available in many domains, for example, images uploaded by internet users. However, the lack of semantic labels for these images makes difficult their use for computer vision applications that rely on learned classifiers. In these situations, it is reasonable to apply a classifier trained with existing labeled data to newly collected data. However, this procedure commonly results in a significant drop of the classifier's accuracy. This is because the training data (source domain) and the testing data (target domain) are drawn from different probability distributions.

On the one hand, the straightforward solution consists in reproducing for the target domain the effort already done for the source domain, *i.e.*, data acquisition and labeling, and posterior training of the desired classifiers. However, this may turn out in a waste of resources since labeling usually involves tiresome manual work prone to errors, and collecting new images is not always straightforward. On the other hand, some domain adaptation methods can exploit existing source-domain labeled data (*e.g.*, Caltech256 [1], PASCAL2010 [2], ImageNet [3]) to complement relatively few target-domain data for obtaining new target-oriented classifiers. When the domain adaptation requires that all the target-domain data is labeled, it is called *supervised*, if this data is totally unlabeled the domain adaptation is denoted as *unsupervised*, otherwise it is denoted as *semi-supervised*.

The domain adaptation problem has received considerable attention by machine learning researchers [4], [5], [6] and most recently by computer vision ones [7], [8], [9] and several domain adaptation techniques have been developed within both communities. In this work, we follow the supervised domain adaptation setting and focus on multi-category object recognition between domains [2], [3].

Training classifiers for multi-category object recognition can be cast as a structured SVM (SSVM) problem. Compared to other approaches such as the one-versus-all strategy, an interesting advantage of SSVM is the possibility of introducing (in a natural way) class-specific penalties during the training of the classifiers. In fact, such classification setup is required in many real-world applications where cost vary for different types of misclassification errors. This type of learning is denoted as cost-sensitive learning [10]. For example, in medical diagnosis, misclassifying a cancer as non-cancer is much more serious than misclassifying a non-cancer as cancer since patients could lose their life because of the delay in the correct diagnosis and treatment. Similarly, misclassifying a spam message as a useful e-mail can be easily and quickly corrected by the human receiver, while misclassifying a useful e-mail as spam can be more risky if it actually contains important information that requires quick attention.

There is an increasing literature on SVM-based domain adaptation approaches [11], [12], [13], [14] that show very good adaptation results for different vision applications. One of the simplest strategies is to retrain a classifier with mixed combination of source and target samples. Another popular approach is the adaptive SVM (ASVM) [11], which learns a perturbation of the source-domain decision classifier by using target-domain training samples. The combination of source and target samples may have limited adaptation ability since it treats all the training samples equally, *i.e.* the misclassification costs on source and target domain samples are regarded as equally important. In this way, the final classifier may operate on both domains but not being sufficiently discriminative in the target domain. The weighted combination method treats the source and target domains differently by setting different hyper-parameters in the SVM loss term, which tends to perform better than the straightforward combination method [15].

Generally, most of the SVM-based domain adaptation methods can be extended to SSVM. However, in this work, we propose a new strategy based on the misclassification cost. Similarly to the weighted combination method, our adaptation process treats the source-domain samples differently than the target-domain ones. In particular, during the re-training (adaptation) of a classifier based on target and source data, we explore the idea of introducing a non-zero cost even for correctly classified source domain samples. If the classifier makes a correct prediction to the label of a source-domain sample, we regard it as risky, thus the loss in the objective function is increased. However, this situation should be less risky than making an incorrect prediction about the label of either a source-domain sample or a target-domain one. In other words, we aim to learn a more target-oriented classifier by not rewarding (zero loss) properly classified source-domain training samples.

In order to emphasize that our domain adaptation proposal for SSVM is cost-sensitive, we call it COSS-SSVM. We assess the effectiveness of COSS-SSVM by relying on publicly available object recognition benchmarks. Our experiments show that COSS-SSVM achieves a recognition accuracy comparable to the state-of-the-art methods.

The rest of the paper is organized as follows. In Sect. II we confront COSS-SSVM to related previous works. In Sect. III, COSS-SSVM is described in detail. Sect. IV shows the obtained results in the context of visual object recognition. Finally, Sect. V summarizes the main conclusion of our work.

II. Related Work

The *dataset bias* [16] that refers to the differences between data distributions of train and test sets can hurt a variety of vision tasks. Considering the different datasets as different domains we can find strong connections between this phenomenon and the domain adaptation problem. To address the latter problem, there have been proposed a significant amount of methods within the computer vision community, which commonly are based on feature or model transformation techniques.

Regarding feature-transformation approaches, recently Hoffman *et al.*[17] proposed a method to learn a feature transformation and a classifier in a jointly fashion. Respect to model transformation approaches during the last five years several methods that adapt the parameters of discriminative classifiers, commonly SVM, have been proposed. The Adaptive SVM (ASVM) proposed by Yang *et al.*[11] is a remarkable work that has been the motivation of consecutive successful approaches like the Projective Model Transfer SVM (PMT-SVM) [18], which adapts the SVM parameters between different domains by adding a constrain in the regularization term that enforces the target model to be close to the source one.

In contrast to the model parameter adaptation methods which mainly relies on regularization term, other SVM-based domain adaptation strategies make use of the loss term, *e.g.* weighted combination of both domain samples (WDSVM) [15], [13]. The WDSVM method weights the loss function values differently for the source and target examples by using two distinct SVM hyperparameters C_S and C_T. Cost-sensitive learning as a type of learning that takes into account different misclassification costs has been used in many real-world applications [10], [19], [20], while rarely explored for domain adaptation. These techniques set different classification costs to different samples during the learning. The final goal is to minimize the total cost. By setting a higher cost to the samples of some classes the learned classifier can improve the performance for these classes. A typical application of this learning is to solve the class imbalance problem [10]. Also, it can be applied in multiclass problems [21]. An interesting application for face recognition in control access is proposed in [19]. In this case the cost of confusing an allowed user with other is lower than confusing it with a non-allowed one. These methods motivated us to set different costs to the misclassification error for the source and target samples as a way of improving the accuracy of the adapted classifier in the target domain.

III. Proposed Method

In this work, we focus on SSVM based methods. Such methods consist of a loss term $\mathcal{L}(\mathbf{w};\mathcal{D})$ that captures the error with respect to the training data $\mathcal{D}$ and a regularization term $\mathcal{R}(\mathbf{w})$ that penalizes model complexity. We use SSVM for cross-domain multi-category object recognition. For domain adaptation, first we provide a straightforward extension of the SVM-based domain adaptation methods to SSVM, including the mixed (or weighted) combination of source and target samples strategy and adaptive SVM. Similar to the weighted combination method, we propose a cost-sensitive method, COSS-SSVM, to take into account the differences of the source and target domain samples.

In this work, we denote as $\mathcal{D}_l^S$ the labeled source domain and as $\mathcal{D}_l^T$ the labeled target domain. The vector concatenation is represented as $\mathbf{a} = [\mathbf{b}', \mathbf{c}']'$, where $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ are column vectors, and the zero vector is denoted by $\mathbf{0}$.

A. Structured SVM for multi-category classification

The simplest way to train a multi-category classifier is the one-versus-all strategy, for instance, using SVM as base classifier. However, recent works on object recognition also show promising results when relying on SSVM [22], with the advantage of admitting class-dependent misclassification penalties during the training process. Accordingly, we use SSVM as our base classifier. We study domain adaptation approaches for this classifier, which are rarely explored compared to the SVM-based method. We start by introducing the formulation of the SSVM for multi-category classification.

Assume we are given an image set of K object categories, $\mathcal{D} = \{(\mathbf{x}_i, y_i) | y_i \in \{1, \ldots, K\}\}_{i=1}^N$. A multi-class SVM solves the optimization problem:

$$\begin{array}{l} \min \dfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^N \xi_i \\ \text{s.t.,} \quad \forall i, k, \quad \xi_i \geq 0 \\ \mathbf{w}'_k\phi(\mathbf{x}_i) - \mathbf{w}'_{y_i}\phi(\mathbf{x}_i) \geq \Delta(y_i, k) - \xi_i, \end{array} \tag{1}$$

where $\phi(\mathbf{x}_i)$ is a feature vector of dimension d, each $\mathbf{w}_k$ is the weight vector of linear classifier k, $\mathbf{w} = [\mathbf{w}'_1, \ldots, \mathbf{w}'_K]' \in \mathbb{R}^{Kd}$ concatenates the weights for all classes. $\Delta(y_i, k)$ is the loss function which measures a confusion cost associated with predicting class k when the true label of $\mathbf{x}_i$ is y_i. In this work, we consider 0-1 loss, *i.e.* $\Delta(y_i, k) = 0$, if $y_i = k$, otherwise 1. The multi-class classification problem is solvable by using a SSVM. For that, $\phi(\mathbf{x})$ in Eq.(1) is replaced by $\Phi(\mathbf{x}, y)$, which is a concatenation of the features of each class:

$$\begin{array}{l} \Phi(\mathbf{x}, y) = [\psi_1(\mathbf{x}, y), \ldots, \psi_K(\mathbf{x}, y)]', \\ \psi_k(\mathbf{x}, y) = \begin{cases} \phi(\mathbf{x}), & \text{if } y = k \\ \mathbf{0}, & otherwise. \end{cases} \end{array} \tag{2}$$

B. Baseline 1: Mixture of source and target domain samples (MIX-SSVM and DW-SSVM)

One of the most simplest ways for domain adaptation is to train a classifier on the union of the source and target domain samples, we denote this procedure by MIX-SSVM. Another extension is to consider different weights for the source- and target-domain samples, *i.e.* domain weighted SVM (DWSVM) [15]. For SSVM, we can apply the domain weight strategy and we denote it by DW-SSVM. Note that the MIX-SSVM is the special case of DW-SSVM, where the source and target domain weights are the same. We write the DW-SSVM objective function as follows:

$$\begin{aligned}
&\min \frac{1}{2}\|\mathbf{w}\|^2 + C_T \textstyle\sum_{i=1}^{N^T} \xi_i^T + C_S \sum_{j=1}^{N^S} \xi_j^S \\
&\text{s.t.,} \quad \forall i, j, y, \quad \xi_i^S, \xi_i^T \geq 0 \\
&\mathbf{w}'\Phi(\mathbf{x}_i, y_i) - \mathbf{w}'\Phi(\mathbf{x}_i, y) \geq \Delta(y_i, y) - \xi_i^T, \mathbf{x}_i \in \mathcal{D}_l^T \\
&\mathbf{w}'\Phi(\mathbf{x}_j, y_j) - \mathbf{w}'\Phi(\mathbf{x}_j, y) \geq \Delta(y_j, y) - \xi_j^S, \mathbf{x}_j \in \mathcal{D}_l^S.
\end{aligned} \tag{3}$$

MIX-SSVM corresponds to the case of equal hyperparameters C_S and C_T. Following [15], [13], the values of these hyperparameters are selected by minimizing the cross validation error on the target training set.

C. Baseline 2: Adaptive Structured SVM (ASSVM)

1) Adaptive SVM: ASVM is a model-transform-based method, which adapts the model parameters from the source $\mathcal{D}_l^S$ to the target domain $\mathcal{D}_l^T$ (l indicates labeled samples) by minimizing the following objective function:

$$\min_{\mathbf{w}^T} \frac{1}{2}\|\mathbf{w}^T - \mathbf{w}^S\|^2 + C\mathcal{L}(\mathbf{w}^T; \mathcal{D}_l^T), \tag{4}$$

where the regularization term $\|\mathbf{w}^T - \mathbf{w}^S\|^2$ constrains the target model $\mathbf{w}^T$ to be close to the source one $\mathbf{w}^S$.

2) Adaptive Structured SVM (ASSVM): The extension of ASVM to SSVM is straightforward. The loss term $\mathcal{L}(\mathbf{w}^T; \mathcal{D}_l^T)$ can be written by using the feature representation of Eq.(2) in the structured SVM loss function. The objective function of ASSVM is written as follows:

$$\begin{aligned}
&\min \frac{1}{2}\|\mathbf{w}^T - \mathbf{w}^S\|^2 + C \textstyle\sum_{i=1}^{N^T} \xi_i \\
&\text{s.t.,} \quad \forall i, y, \quad \mathbf{x}_i \in \mathcal{D}_l^T, \quad \xi_i \geq 0 \\
&\mathbf{w}^{T'}\Phi(\mathbf{x}_i, y_i) - \mathbf{w}^{T'}\Phi(\mathbf{x}_i, y) \geq \Delta(y_i, y) - \xi_i,
\end{aligned} \tag{5}$$

D. Cost-sensitive Structured SVM for Domain Adaptation (COSS-SSVM)

Our proposed domain adaptation method is based on MIX-SSVM. As it is shown in Eq.(3), the MIX-SSVM method simply combines source and target samples but does not take into the misclassification cost on different domains. Since the labeled target training samples are generally much less than the source samples, the final classifier can still be source-domain oriented. We propose a cost-sensitive method which considers different miss-classification cost for different domain samples. Concretely, when the classifier correctly predicts the class label for a target domain sample, the cost is 0. However, if the sample comes from the source domain, we still increase a soft loss in the objective function. Since our goal is to minimize the training error in the target domain, correctly predicting a source domain sample may confuse the target domain classifier. Using this cost-sensitive approach, the final classifier is expected to obtain higher accuracy in the target domain. Our cost-sensitive objective function is defined as follows:

$$\begin{aligned}
&\min \frac{1}{2}\|\mathbf{w}\|^2 + C \textstyle\sum_{i=1}^{N^T} \xi_i^T + C \sum_{j=1}^{N^S} \xi_j^S \\
&\text{s.t.,} \quad \forall i, j, y, \quad \xi_i^S, \xi_i^T \geq 0 \\
&\mathbf{w}'\Phi(\mathbf{x}_i, y_i) - \mathbf{w}'\Phi(\mathbf{x}_i, y) \geq \Delta^T(y_i, y) - \xi_i^T, \mathbf{x}_i \in \mathcal{D}_l^T \\
&\mathbf{w}'\Phi(\mathbf{x}_j, y_j) - \mathbf{w}'\Phi(\mathbf{x}_j, y) \geq \Delta^S(y_j, y) - \xi_j^S, \mathbf{x}_j \in \mathcal{D}_l^S.
\end{aligned} \tag{6}$$

For $\Delta^T(y_i, y)$, we use the same 0-1 loss as in Eq.(1), while for $\Delta^S(y_j, y)$, we define the following loss function:

$$\Delta^S(y_j, y) = \begin{cases} \gamma \in [0, 1], & \text{if } y_j = y, \\ 1, & otherwise, \end{cases} \tag{7}$$

where γ is a *soft* loss for the source-domain correct predictions. This means that even when the classifier correctly predicts the label of a source-domain sample, since it may confuse the target-domain classification, we treat it as a $light$ misclassification. Thus, we increase the loss in the objective function by $\gamma \in [0, 1]$. If $\gamma = 0$, we have the standard MIX-SSVM. If $\gamma = 1$, correctly predicting the label of a source-domain sample is treated with the same cost as incorrectly predicting the label of any other sample, *i.e.* source and target domain ones. In Sect. IV, we show the impact of choosing different γ values.

IV. Experimental Results

In the following, we evaluate our approach on standard multi-category object recognition datasets. We first compare the SSVM with the one-versus-all linear SVM on multi-class classification. The classifiers are trained with labeled source or target domain samples without domain adaptation. Then, we study the impact of the soft cost γ in the proposed COSS-SSVM. [1] We test on all the source-target domain splits with different γ values. Finally, following the evaluation protocol in [17], we compare the proposed method to MIX-ASSVM, DW-SSVM, ASSVM as well as other state-of-the-art methods.

A. Dataset and Experiment Setting

Datasets: We use the benchmark domain adaptation datasets called *Office* [7] and *Caltech256* [1]. The *Office* dataset is a collection of images from three different domains: *amazon*, *webcam*, and *dslr*. Each domain contains 31 categories of common office objects. The *amazon* domain is a collection of product images from amazon.com. The *webcam* and *dslr* contain images taken by a webcam and a dslr camera, respectively. The *Caltech256* dataset contains 256 object categories from a single domain. We use the 10 common categories shared by the the *Caltech256* and *Office* for our experiments.

Image representation: We use the image representation provided by [23] (SURF bag of words image features with a dictionary of 800 words). Following [23], we apply PCA to the source and target data and use 20-dimensional features in all the experiments.

[1] We will make the source code available.

Experiment setup: We follow the setup of [7], [23] and [17]. This means that we use 20 training examples per category for *amazon* source and 8 for all other source domains and 3 labeled examples per category for each target domain. We use the same 20 random train/test splits available from [17] for reporting the average performance.

B. Multi-category Classification Accuracy of SSVM classifier

Before performing the domain adaptation, we investigate the performances of our base classifiers:

SRC: trained with only the source domain samples.

TAR: trained with only target domain samples.

We compare our SSVM multi-class classifier to the *classic* one-versus-all classifier. Table I shows the detailed comparison on each source-target domain pair. For the one-versus-all linear SVM, we tested three different solvers: Liblinear [24], Vlfeat Pegaso [25] and Mosek QP solver [26]. Our SSVM solver is implemented based on the LBFGS optimization toolbox [27]. The hyperparameter C of each SVM is first tuned by cross-validation according to the mean accuracy on all target dataset, then we fix the value for all the experiments. The results show that our SSVM classifier constantly outperforms one-versus-all methods. Accordingly, based on this SSVM base classifier, we implement our domain adaptation algorithms.

C. The Impact of γ in the COSS-SSVM Domain Adaptation

In this section, we evaluate the domain adaptation of the COSS-SSVM under different γ values. Fig.1 shows the classification accuracy on each source-target domain pair. We tested γ from 0 to 1 with a step of 0.1. When $\gamma = 0$, which is equal to the MIX-SSVM training, the domain adaptation has the lowest accuracy on all testing pairs. As γ increases, the accuracy of the adapted classifier constantly improves. However, when γ reaches 1, most of the adapted classifiers become worse, except the ones on $A \rightarrow W$, $W \rightarrow A$, $C \rightarrow W$ and $C \rightarrow D$. The figure on the right hand side draws the overall accuracy on all domain splits. It shows clearly that the domain adaptation reaches maximum accuracy at $\gamma = 0.9$. This experiment proves our assumption that correctly predicting the label of source domain samples should be taken into account as a missclassification cost. Accordingly, we fix $\gamma = 0.9$ in the following experiments.

D. Comparison with more Baselines

In this section, we compare COSS-SSVM with multiple baselines. As we follow the same setting of Hoffman:2013, we make use of the available evaluation protocol as well as some available baseline results.

ASVM: Adaptive SVM [11].

PMT-SVM: Projective model transfer SVM [18].

ASVM and PMT-SVM are trained with only the target domain samples and a pre-trained source model. We use the implementation from [18] and use the MOSEK optimization toolkit [26] in our experiments. The multi-category classifiers are learned in a one-vs-all manner.

ARCT: A general feature transform method proposed in [28], using both domain data. We compare with the results available from [17].

HFK: A feature transform based method that learns a latent common space between source and target as well as a common space classifier [8]. We compare with the results available from [17].

GFK: The geodesic flow kernel method [23], using all source and target data (including testing data). We apply 1-nearest neighbour classifier with the kernel as in [17].

MMDT: Max-margin domain transfer method of [17], which learns a mapping from target domain to source domain as well as a discriminative classifier trained with the mapped target features and the source domain ones.

ASSVM, *MIX-SSVM* and *DW-SSVM*, see Sect. III.

Results The multi-category accuracy for each domain split is shown in Table II. Table III shows the average accuracy of each algorithm on all domain splits. The recently proposed method [17] reported state-of-the-art results and our COSS-SSVM achieves comparable accuracy on all domain splits, being slightly better on the average accuracy. Our *ASSVM* implementation also reaches the same performances as *MMDT*, showing promising performance of SSVM-based domain adaptation methods. Compared to MIX-SSVM, our COSS-SSVM shows a significant improvement for all domain splits. In fact, the MIX-SSVM is the special case of COSS-SSVM characterized by $\gamma = 0$.

E. Discussion

The proposed COSS-SSVM method is a simple while effective strategy of domain adaptation. In this work we simply fix γ for all the source domain samples, but considering instance level cost-sensitive learning and using a dynamic cost may improve the accuracy resulting from domain adaptation. Although our cost-sensitive approach is actually built on a very simple mixed combination of different domain samples, it could be complementary to other SVM-based domain adaptation methods.

V. Conclusions

In this paper, we present a simple but effective approach to perform the task of adapting classifiers between different domains. Our method is based on the assumption that correctly predicting the label of a source domain sample could be considered as a missclassification cost for a target domain classifier. Thus we propose a cost-sensitive training strategy to handle missclassification costs of source and target domain samples differently, what we denoted as cost-sensitive structured SVM (COSS-SSVM). We apply this method to the problem of multi-category object recognition between different domains. Experiments on standard benchmarks demonstrate the effectiveness of our approach by a significant improvement to the cost-insensitive method *i.e.* MIX-ASSVM and comparable accuracy to other state-of-the-art methods. Finally, note that our method can be also regarded as a complement to other SVM-based domain adaptation methods. For instance we could extend our method for unsupervised domain adaptation exploiting the intrinsic compact structures of categories across

	A → W	A → D	A → C	W → A	W → D	W → C
SRC (Liblinear)	39.7 ± 1.4	37.7 ± 1.0	38.4 ± 0.5	32.6 ± 1.0	64.2 ± 0.9	26.8 ± 0.6
SRC (Pegasos)	37.1 ± 1.3	34.5 ± 1.1	36.6 ± 0.3	35.9 ± 0.8	63.8 ± 1.2	28.4 ± 0.6
SRC (QP-Mosek)	36.9 ± 1.3	36.7 ± 1.1	37.9 ± 0.4	32.3 ± 1.0	62.8 ± 1.0	27.1 ± 0.5
SRC-SSVM	**42.3** ± 0.8	**38.4** ± 0.6	**39.8** ± 0.3	**39.2** ± 0.4	**65.3** ± 0.7	**33.7** ± 0.5

	D → A	D → W	D → C	C → A	C → W	C → D
SRC (Liblinear)	32.2 ± 0.7	70.8 ± 1.1	25.2 ± 0.4	37.0 ± 0.8	30.8 ± 1.5	32.0 ± 1.0
SRC (Pegasos)	34.3 ± 0.6	71.3 ± 0.7	27.9 ± 0.3	38.9 ± 1.0	29.9 ± 1.3	31.7 ± 1.3
SRC (QP-Mosek)	31.2 ± 0.9	69.4 ± 1.1	25.6 ± 0.5	38.3 ± 0.9	30.2 ± 1.3	30.1 ± 1.1
SRC-SSVM	**35.4** ± 0.5	**71.5** ± 0.7	**29.8** ± 0.2	**42.2** ± 0.7	**37.0** ± 1.2	**39.5** ± 1.1

	A → W	A → D	A → C	W → A	W → D	W → C
TAR (Liblinear)	57.1 ± 1.0	44.2 ± 0.9	26.5 ± 0.6	45.1 ± 1.2	47.9 ± 1.7	25.5 ± 0.8
TAR (Pegasos)	60.0 ± 0.8	47.0 ± 0.9	28.9 ± 0.8	46.3 ± 1.0	52.9 ± 1.6	27.0 ± 0.7
TAR (QP-Mosek)	**64.7** ± 1.1	51.4 ± 1.1	30.8 ± 0.6	48.5 ± 1.1	54.0 ± 1.4	29.8 ± 1.0
TAR-SSVM	**64.5** ± 0.9	**52.1** ± 1.1	**33.5** ± 0.8	**50.5** ± 0.7	**56.0** ± 1.0	**31.3** ± 0.8

	D → A	D → W	D → C	C → A	C → W	C → D
TAR (Liblinear)	43.0 ± 1.0	56.4 ± 0.9	26.6 ± 0.7	44.4 ± 1.1	56.6 ± 1.0	44.5 ± 1.6
TAR (Pegasos)	44.9 ± 0.9	60.9 ± 1.0	28.4 ± 0.8	45.8 ± 1.1	58.2 ± 1.0	50.0 ± 1.4
TAR (QP-Mosek)	47.8 ± 1.0	63.1 ± 1.1	29.8 ± 0.8	**49.5** ± 1.0	**63.1** ± 1.2	52.6 ± 1.3
TAR-SSVM	**49.4** ± 0.8	**66.2** ± 0.9	**33.2** ± 0.8	**50.2** ± 0.9	**63.4** ± 1.1	**53.8** ± 1.2

TABLE I. MULTI-CATEGORY RECOGNITION ACCURACY ON TARGET DOMAINS. THE DOMAIN NAMES ARE IN ABBREVIATIONS: A: *amazon*, W: *webcam*, D: *dslr*, C: *Caltech256*

.

	A → W	A → D	A → C	W → A	W → D	W → C
SRC-SSVM	42.3 ± 0.8	38.4 ± 0.6	39.8 ± 0.3	39.2 ± 0.4	65.3 ± 0.7	33.7 ± 0.5
TAR-SSVM	64.5 ± 0.9	52.1 ± 1.1	33.5 ± 0.8	**50.5** ± 0.7	56.0 ± 1.0	31.3 ± 0.8
ARCT	55.7 ± 0.9	50.2 ± 0.7	37.0 ± 0.4	43.4 ± 0.5	71.3 ± 0.8	31.9 ± 0.5
HFA	61.8 ± 1.1	*52.7* ± 0.9	31.1 ± 0.6	45.9 ± 0.7	57.1 ± 1.0	29.4 ± 0.6
GFK	56.5 ± 0.8	45.3 ± 0.9	38.6 ± 0.4	45.8 ± 0.6	**73.8** ± 0.7	32.6 ± 0.6
MMDT	*65.1* ± 1.2	**54.5** ± 1.0	39.7 ± 0.5	**50.6** ± 0.8	62.5 ± 1.0	34.8 ± 0.8
ASVM	*65.0* ± 1.0	51.6 ± 1.1	30.9 ± 0.6	48.6 ± 1.1	54.4 ± 1.5	29.8 ± 1.0
PMT-SVM	**65.9** ± 1.0	*52.6* ± 1.1	32.3 ± 0.6	49.0 ± 1.1	57.9 ± 1.6	30.4 ± 0.9
ASSVM	60.0 ± 0.9	49.7 ± 0.8	42.6 ± 0.5	*49.5* ± 0.5	67.4 ± 0.7	*37.3* ± 0.5
MIX-SSVM	51.9 ± 1.1	45.5 ± 0.5	41.9 ± 0.4	46.4 ± 0.4	68.2 ± 0.8	*37.1* ± 0.4
DW-SSVM	55.9 ± 1.1	48.7 ± 0.6	**44.5** ± 0.4	49.2 ± 0.4	*72.0* ± 1.0	**38.0** ± 0.5
COSS-SSVM	57.2 ± 1.1	48.8 ± 0.6	*44.4* ± 0.4	*50.3* ± 0.4	*72.4* ± 1.1	**37.8** ± 0.6

	D → A	D → W	D → C	C → A	C → W	C → D
SRC-SSVM	35.4 ± 0.5	71.5 ± 0.7	29.8 ± 0.2	42.2 ± 0.7	37.0 ± 1.2	39.5 ± 1.1
TAR-SSVM	49.4 ± 0.8	66.2 ± 0.9	33.2 ± 0.8	50.2 ± 0.9	*63.4* ± 1.1	**53.8** ± 1.2
ARCT	42.5 ± 0.5	78.3 ± 0.5	33.5 ± 0.4	44.1 ± 0.6	55.9 ± 1.0	50.6 ± 0.8
HFA	45.8 ± 0.9	62.1 ± 0.7	31.0 ± 0.5	45.5 ± 0.9	60.5 ± 0.9	51.9 ± 1.1
GFK	45.8 ± 0.4	*80.3* ± 0.7	33.3 ± 0.5	46.4 ± 0.7	61.0 ± 1.4	*52.7* ± 1.2
MMDT	*50.4* ± 0.7	74.2 ± 0.7	*35.7* ± 0.7	51.1 ± 0.7	*62.9* ± 1.1	*53.0* ± 1.0
ASVM	48.0 ± 1.1	63.5 ± 1.1	29.9 ± 0.8	49.5 ± 1.0	*63.2* ± 1.2	*52.7* ± 1.3
PMT-SVM	48.6 ± 1.1	66.5 ± 1.2	30.9 ± 0.8	50.0 ± 1.0	**64.3** ± 1.2	52.2 ± 1.3
ASSVM	48.6 ± 0.5	74.6 ± 0.6	*35.5* ± 0.5	*53.4* ± 0.7	**63.6** ± 1.2	*52.7* ± 1.0
MIX-SSVM	45.8 ± 0.4	74.9 ± 0.5	33.8 ± 0.4	50.7 ± 0.6	55.3 ± 1.0	49.4 ± 0.9
DW-SSVM	49.2 ± 0.4	*80.1* ± 0.5	*36.4* ± 0.4	*53.3* ± 0.5	59.5 ± 1.1	51.4 ± 0.8
COSS-SSVM	**51.0** ± 0.4	**80.8** ± 0.5	**36.9** ± 0.6	**53.5** ± 0.5	60.4 ± 1.1	51.9 ± 0.1

TABLE II. MULTI-CATEGORY RECOGNITION ACCURACY ON TARGET DOMAINS ($\gamma = 0.9$). BOLD INDICATES THE BEST RESULT FOR EACH DOMAIN SPLIT. UNDERLINE INDICATES THE SECOND BEST RESULT. THE DOMAIN NAMES ARE IN ABBREVIATIONS: A: *amazon*, W: *webcam*, D: *dslr*, C: *Caltech256*

.

ARCT [28]	HFA [8]	GFK [23]	MMDT [17]	ASVM [11]	PMT-SVM [18]	ASSVM	MIX-SSVM	DW-SSVM	COSS-SSVM
49.5 ± 0.6	47.4 ± 0.8	51.0 ± 0.7	52.9 ± 0.9	48.9 ± 1.1	48.9 ± 1.1	52.9 ± 0.7	50.1 ± 0.6	53.2 ± 0.6	**53.8** ± 0.9

TABLE III. THE AVERAGE MULTI-CATEGORY RECOGNITION ACCURACY ON ALL DOMAIN SPLITS ($\gamma = 0.9$).

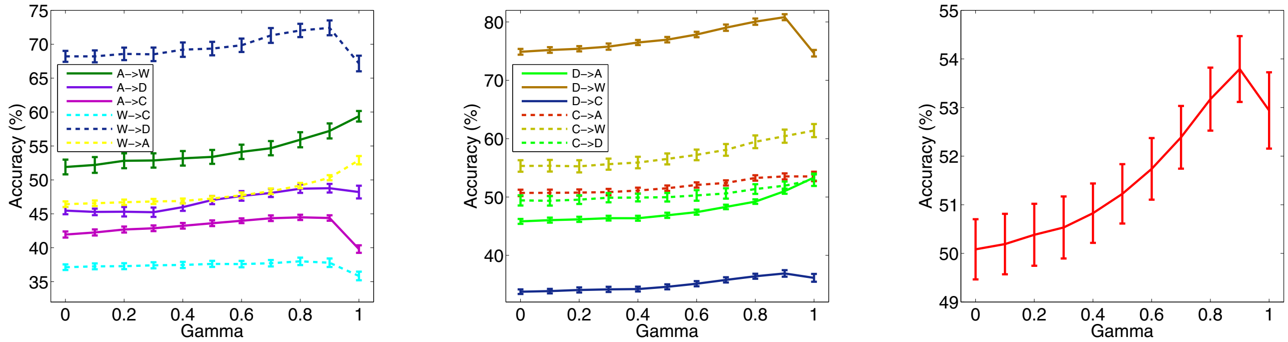

Fig. 1. The multi-category classification accuracy of COSS-SSVM with different γ value. The figure on the right shows the mean accuracy of all domain splits.

different domains as in [29]. Moreover, other recent state-of-the-art methods will be also included in our comparison (*e.g.*, [29], [30]). We let all these as future work.

ACKNOWLEDGMENT

This work is supported by the Spanish MICINN projects TRA2011-29454-C03-01 and TIN2011-29494-C03-02, the Chinese Scholarship Council (CSC) grant No.2011611023 and Sebastian Ramos' FPI Grant BES-2012-058280.

REFERENCES

[1] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep., 2007.

[2] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. Journal on Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, Florida, US, 2009.

[4] H. D. III, "Frustratingly easy domain adaptation," in *Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 2007.

[5] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1, pp. 151–175, 2009.

[6] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[7] K. Saenko, B. Hulis, M. Fritz, and T. Darrel, "Adapting visual category models to new domains," in *European Conf. on Computer Vision*, Hersonissos, Heraklion, Crete, Greece, 2010.

[8] L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Int. Conf. on Machine Learning*, Edinburgh, Scotland, 2012.

[9] D. Vázquez, A. López, J. Marín, D. Ponsa, and D. Gerónimo, "Virtual and real world adaptation for pedestrian detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 797–809, 2014.

[10] C. Ling and V. Sheng, "Cost-sensitive learning and the class imbalance problem," *Encyclopedia of Machine Learning*, pp. 231–5, 2008.

[11] J. Yang, R. Yan, and A. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *ACM Multimedia*, Augsburg, Germany, 2007.

[12] L. Duan, I. Tsang, D. Xu, and S. Maybank, "Domain transfer svm for video concept detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami Beach, FL, USA, 2009.

[13] A. Bergamo and L. Torresani, "Exploring weakly-labeled web images to improve object classification: a domain adaptation approach," in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, 2010.

[14] D. Vázquez, A. López, and D. Ponsa, "Unsupervised domain adaptation of virtual and real worlds for pedestrian detection," in *Int. Conf. in Pattern Recognition*, Tsukuba, Japan, 2012.

[15] G. B. Schweikert, C. Widmer, B. Schlkopf, and G. Rtsch, "An empirical analysis of domain adaptation algorithms for genomic sequence analysis," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2008.

[16] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *European Conf. on Computer Vision*, Florence, Italy, 2012.

[17] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell, "Efficient learning of domain invariant image representations," in *Int. Conf. on Learning Representations*, Arizona, USA, 2013.

[18] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *Int. Conf. on Computer Vision*, 2011.

[19] Y. Zhang and Z. Zhou, "Cost-sensitive face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1758–1769, 2010.

[20] t. Jan, D. Wang, C. Lin, and H. Lin, "A simple methodology for soft cost-sensitive classification," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 2012.

[21] H. Lin, "A simple cost-sensitive multiclass classification algorithm using one-versus-one comparisons," National Taiwan University, Tech. Rep., 2010.

[22] S. Branson, O. Beijbom, and S. Belongie, "Efficient large-scale structured learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013.

[23] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[24] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear lassification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[25] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[26] Mosek, "Optimization toolkit," http://www.mosek.com.

[27] M. Schmidt, "Minconf - projection methods for optimization with simple constraints in matlab," http://www.di.ens.fr/ mschmidt/Software/minConf.html.

[28] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2011.

[29] F. Mirrashed and M. Rastegar, "Domain adaptive classification," in *Int. Conf. on Computer Vision*, Sydney Australia, 2013.

[30] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Int. Conf. on Computer Vision*, Sydney, Australia, 2013.