Fast Generalized Distillation for Semi-supervised Domain Adaptation

No Author Given

No Institute Given

Abstract. Semi-supervised domain adaptation (SDA) is a typical setting when we face the problem of domain adaptation in the real applications. How to effectively utilize the unlabeled examples is an important issue in SDA. In this paper, we propose a new paradigm, called *Generalized Distillation Semi-supervised Domain Adaptation* (GDSDA) to solve the SDA problem. We first demonstrate that GDSDA can effectively utilize the unlabeled data to transfer the knowledge from the source domain in the SDA problem. Meanwhile, we illustrate that the imitation parameter of GDSDA can greatly affect the performance of the target model. Then, we propose GDSDA-SVM which uses SVM as the base classifier and can effectively estimate the imitation parameter. Specifically, the imitation parameter is estimated by minimizing the Leave-one-out loss on the target data. Experiment results show that GDSDA-SVM can effectively utilize the unlabeled data to transfer the knowledge between different domains under the SDA setting.

1 Introduction

Domain adaptation can be used in many real applications, which addresses the problem of learning a target domain with the help of a different but related source domain. Previous methods show that carefully modeling the source data to compensate for the domain shift between different domains can significantly improve the performance on the target domain [4]. In real applications, it can be very expensive to obtain sufficient labeled examples while there are abundant unlabeled examples in the target domain. *Semi-supervised domain adaptation* (SDA) tries to use some unlabeled examples as well as a few labeled ones from the target domain to compensate for the domain shift[9]. Typically, the labeled examples are too few to construct a good classifier alone. Therefore, how to effectively utilize the unlabeled examples is an important issue in SDA.

In previous work of SDA, many methods have been proposed to leverage the source knowledge with the unlabeled data. Duan et al.[6] proposed a method to measure the domain shift with Maximum Mean Discrepancy of the labeled and unlabeled data from source and target domains. Daumé et al[3] utilized unlabeled data as a co-regularizer and forced the hypotheses learned from different domains to agree on the unlabeled data. Yao et al.[21] used the unlabeled target examples to discover the underlying intrinsic information in the target domain. Donahue et al.[4] show that using smoothness constraints on the classifier scores over the unlabeled data can lead to improved adaptation results. Most previous work in SDA requires to access the individual example in the source domain to measure the data distribution mismatch between the source and target domain. However, in some situation, we may not be able to access each of the source examples for many reasons. For example, when we use a large dataset as our source domain, it is tedious to access each of the source examples to estimate the data distribution mismatch.

Recently, a framework called *Generalized Distillation* (**GD**) [11] was proposed, which allows the knowledge to be transferred between different models effectively. GD includes two different models, the teacher and student models. The student model tries to distill the knowledge from the teacher model by mimicking the outputs of the teacher model on the training data. Remarkably, GD can be applied in many learning scenarios such as unsupervised, semi-supervised and multitask learning[11]. Given that GD has such ability in various learning scenarios, it is natural to ask the following two questions: (1) Can GD be applied to solve the SDA problem? (2) Is there any obstacle when we apply GD to real SDA applications?

To answer these two questions, in this paper, we first propose a new paradigm, called *Generalized Distillation Semi-supervised Domain Adaptation* (**GDSDA**), and illustrate how it can solve the SDA problem. Specifically, we show that the knowledge transfer process of GDSDA is so effective that the target model can outperform the source model it learned from even without using any ground truth label. Different from many other paradigms which requires to access every single example of the source domain, Moreover, GDSDA only requires the predicted class probabilities of the target domain examples from the source model. Therefore, GDSDA is more efficient especially when the source domain is relatively large and there is a well-trained source model.

Then we argue that the imitation parameter of GDSDA which controls the amount of knowledge transferred from the source can greatly affect the performance of the target model in prediction. However, according to previous works [11, 17], the imitation parameter can only be determined by either brute force search or background knowledge. Therefore, we propose a novel imitation parameter estimation method for GDSDA, called GDSDA-SVM that uses SVM as the base classifier and can determine the imitation parameter automatically. In particular, inspired by [2], we use mean square loss for GDSDA-SVM and show that the Leave-one-out cross validation (LOOCV) loss can be calculated in a closed form. By minimizing the LOOCV loss on the target training data, we can find the optimal imitation parameter for the target model. In our experiments, we show that GDSDA-SVM can effectively find the optimal imitation parameter and achieve competitive performance compared to methods using brutal force search.

To summarize, the main contributions of this paper include: (1) We propose the framework GDSDA for domain adaptation and show that GDSDA can be used in many real SDA problems. (2) We propose the GDSDA-SVM that can effectively find the optimal imitation parameter for GDSDA.

2 Generalized Distillation for Semi-supervised Domain Adaptation

As we mentioned, GDSDA is a paradigm using generalized distillation for semi-supervised domain adaptation. In this section, we first give a brief review of generalized distillation. Then we show the process of GDSDA and demonstrate the reason why GDSDA can work for the SDA problem. Finally, we show the importance of the imitation parameter.

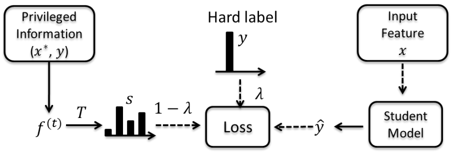


Fig. 1: Illustration of Generalized Distillation training process.

2.1 An overview of Generalized Distillation and GDSDA

Distillation [8] and *Learning Using Privileged Information* (LUPI) [19] are two paradigms that enable machines to learn from other machines. Both methods address the problem of how to build a student model that can learn from the advanced teacher models. Recently, Lopez et al. [11] proposed a framework called *generalized distillation* that unifies both methods and show that it can be applied in many scenarios.

In GD, the training data can be represented as a collection of the triples: $\{(x_1, x_1^*, y_1), (x_2, x_2^*, y_2) \dots (x_n, x_n^*, y_n)\}$. x^* is the privileged information for data x, which is only available in the training set and y is the corresponding label. The process of generalized distillation is as follows: in step 1, a teacher model $f^{(t)}$ is trained using the input-output pairs $\{x_i^*, y_i\}_{i=1}^n$. In step 2, use $f^{(t)}$ to generate the soft label s_i for each training example x_i using the softmax function σ:

$$s_i = \sigma(f^{(t)}(x_i)/T) \tag{1}$$

where T is a parameter called temperature to control the smoothness of the soft label. In step 3, learn the student $f^{(s)}$ from the pairs $\{(x_i, y_i), (x_i, s_i)\}_{i=1}^n$ using:

$$f^{(s)} = \underset{f^{(s)} \in \mathcal{F}^{(s)}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left[\lambda \ell \left(y_i, \sigma(f^{(s)}(x_i)) \right) + (1-\lambda) \ell \left(s_i, \sigma(f^{(s)}(x_i)) \right) \right] \tag{2}$$

Here, $\ell(\cdot, \cdot)$ is the loss function and λ is the imitation parameter to balance the importance between the hard label y_i and the soft label s_i.

As generalized distillation only requires the training inputs $\{x_i, y_i\}_{i=1}^n$ and the output s_i from the teacher function $f^{(t)}$, it can be naturally applied to

SDA. This leads to *Generalized Distillation Semi-supervised Domain Adaptation* (**GDSDA**), where the source model can be used as the teacher to output the soft labels and the student model is the target model. To be consistent with other works in domain adaptation, we use source model and target model to denote the teacher model and the student model in the rest of our paper.

An important issue of applying GD to SDA is that, in Eq. (2), each example is assigned with a hard label y (true label) and a soft label s (class probabilities from the teacher). However, in SDA, we are not able to obtain the hard labels of the unlabeled data. Here we follow the GD work[11] and use the "fake label" strategy to label the unlabeled data: for the labeled examples, we use *one-hot* strategy to encode their labels while using the *gray code* (all 0s) as the label of the unlabeled examples. Thus, each example in the target domain is assigned with a label. It is arguable that the "fake label" strategy would introduce extra noise and degrade the performance. However, we will show in our experiment that this noise can be well controlled by setting a proper value to the imitation parameter and we can still achieve improved performance (See the single source experiment). The process of GDSDA is shown in Figure 2. Suppose we have

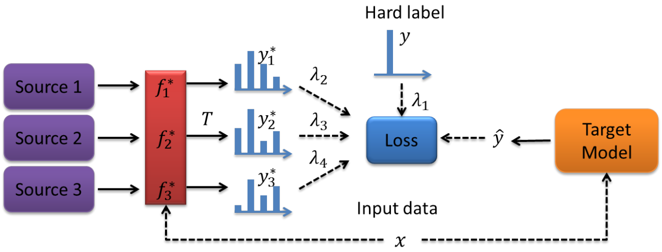


Fig. 2: Illustration of GDSDA training process.

$M-1$ source domains denoted as $D_s^{(j)} = \{X^{(j)}, Y^{(j)}\}_{j=1}^{M-1}$ and the target domain $D_t = \{X, Y\}$ encoded with the "fake label" strategy. Similar to GD, the process of GDSDA is as follows:

1. Train the source models f_j^* for each of the $M-1$ domain with $\{X^{(j)}, Y^{(j)}\}$.
2. For each of the training example x_i in the target domain, computer the corresponding soft label y_{ij}^* with each of the source model f_j^* and the temperature $T > 0$.
3. Learn the target model f_t using the $(M+1)$-tuples $\{x_i, y_i, y_{i1}^*, \dots, y_{i(M-1)}^*\}_{i=1}^L$ with the imitation parameters $\{\lambda_i\}_{i=1}^M$ using (3):

$$\begin{aligned} f_t(\lambda) = \arg\min_{f_t \in \mathcal{F}} \frac{1}{L} \sum_{i=1}^{L} \left[\lambda_1 \ell\left(y_i, f_t(x_i)\right) + \sum_{j=1}^{M-1} \lambda_{j+1} \ell\left(y_{ij}^*, f_t(x_i)\right) \right] \\ \text{s.t.} \qquad \sum_i \lambda_i = 1 \end{aligned} \tag{3}$$

Compared to other works of SDA which require to use each example of the source domain, by either re-weighting [4, 6] or augmentation [3], GDSDA only requires the trained model from the source domain to generate the soft labels. Moreover, GDSDA is able to handle the multi-class scenario while some previous works, such as SHFA[5] can only solve the binary classification problem in SSDA. Moreover, GDSDA is compatible with any type of source model that is able to output the soft label (i.e. class probabilities).

2.2 Why does GDSDA work

In this part, we provide demonstrate the scenarios where GDSDA would work. Before we provide our analysis, we first introduce the two basic assumptions for GDSDA to work well in domain adaptation: the *assumption of distillation and the assumption of the source model.*

Assumption of Distillation: The capacity (or VC dimension) of the target model f_t is smaller than the capacity of source model f^*. This assumption is inherited from distillation. **Assumption of the source model:** The source model f^* should work better than a target model f'_t trained only with the hard labels. For example, when we only have a single labeled example for each class in the target training set, it is reasonable to assume that the source model trained from another domain could perform better than any model trained only with the target training data on the target task. Based on this two assumptions, we will show that GDSDA can effectively leverage the source model and transfer the knowledge between different domains under the SDA setting.

According to ERM principle[20], the simple model has better generalization ability than the complex one if they both have the same training error. By mimicking the output of the source model f^* on the target domain, the target model f_t can achieve similar training error to the training error of the source model f^* on the target domain and we can expect that the target model has better generalization ability.

It is worthy to notice that in this process, the target model only has to mimick the output of the source model (soft label) without considering the hard labels of the examples. In another word, GDSDA provide an effective way to utilize the unlabeled data.

Arguably, because of the domain shift, the source model is biased towards the source domain when applying on the target task. However, as it is suggested in [8], we can use a few labeled data from the target domain to compensate for the domain shift and achieve a better performance on the target task with Eq. (3). Specifically, we use the imitation parameter λ to control the relative importance of the soft label from the source model and the hard label, which in turn reflects the similarity between the source and target tasks. For example, in Figure 2, when we set $\lambda_2 = 0$, we actually ignore the knowledge from source domain 1. As a result, with proper imitation parameter, GDSDA can compensate for the domain shift under the setting of SDA (for more details, please see the experiment section).

As we have mentioned above, the imitation parameter controls the importance of the soft label, i.e. the knowledge transferred from source domain. Many previous works have addressed the importance of knowledge control in domain adaptation [5, 6]. Without carefully controlling the amount of knowledge transferred from the source domain, it is easy for the target model to get degraded performance or even suffer from negative transfer [13]. How to choose the imitation parameter is essential for GDSDA. In the previous works, the imitation parameter can only be determined by either brute force search [11] or background knowledge [17]. On the other hand, in real applications, it is common that there could be multiple source domains to be exploited. As it is suggested in [16], learning from multiple related sources simultaneously can significantly improve the performance of the target model. However, these previous works become more difficult to apply when there are multiple sources and imitation parameters to be determined. For these reasons, we proposed a method, GDSDA-SVM that can estimate the transfer parameter automatically.

3 GDSDA-SVM

As we mentioned, it is important to find a method that can determine the imitation parameter effectively. In this section, we propose our method GDSDA-SVM that uses SVM as the base classifier and can effectively estimate the imitation parameter by minimizing the training error on the target domain.

3.1 Distillation with multiple sources

As it is suggested in [19], the optimal imitation parameter should be the one that can minimize the training error on the target domain. Based on that, we propose our method GDSDA-SVM that can estimate the imitation parameter effectively.

In our GDSDA-SVM, instead of using hinge loss, we use Mean Squared Error (MSE) as our loss function for the following two reasons: (1) Several recently works [1, 12, 14, 18] show that MSE is also an efficient measurement for the target model to mimick the behavior of the source model. (2) MSE can provide a closed form cross-validation error estimation. Thus we can effectively estimate the imitation parameter.

Suppose we have L examples $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{L}$ from and N classes in the target domain where $X \in R^{L\times d}, Y \in R^{L\times N}$. Meanwhile, there are $M-1$ the source (teacher) models providing the soft labels $Y^* = \{\mathbf{y}^*_{ij} | j = 1, ..., L; i = 1, ..., M-1\}$ for each of the L examples. For simplicity, we combine the hard label Y and soft label Y^* and use new label matrix: $S \in R^{M\times L\times N}$ to denote them. To solve this N-class classification problem, we adopt the One-vs-All strategy to build N binary SVMs. To obtain the nth binary SVM, we have to solve the following optimization problem:

$$\min \quad \frac{1}{2}||\mathbf{w}_n||^2 + C\sum_{i,j}\lambda_i e^2_{ijn} \quad s.t. \quad e_{ijn} = s_{ijn} - \mathbf{w}_n\mathbf{x}_j; \sum_i \lambda_i = 1; \tag{4}$$

To find the saddle point,

$$\frac{\partial L}{\partial \mathbf{w}_n} = 0 \rightarrow \mathbf{w}_n = \sum_j \alpha_{ij}^{(n)} \mathbf{x}_j; \quad \frac{\partial L}{\partial e_{ijn}} = 0 \rightarrow \alpha_{ij}^{(n)} = 2C\lambda_i e_{ijn} \tag{5}$$

For each example $\mathbf{x}_j$ and its constraint of label s_{ijn}, we have $e_{ijn} + \mathbf{w}_n \mathbf{x}_j = s_{ijn}$. Replacing $\mathbf{w}_n$ and e_{ijn}, we have:

$$\lambda_i \mathbf{x}_j \sum_k \alpha_{ik}^{(n)} \mathbf{x}_k + \frac{\alpha_{ij}^{(n)}}{2C} = \lambda_i s_{ijn} \tag{6}$$

Summing over each constraint of example x_j, we have:

$$\underbrace{\sum_i \lambda_i}_{=1} \mathbf{x}_j \sum_k \alpha_{ik}^{(n)} \mathbf{x}_k + \sum_i \frac{\alpha_{ij}^{(n)}}{2C} = \sum_i \lambda_i s_{ijn} \tag{7}$$

Let $\eta_{jn} = \sum_i \alpha_{ij}^{(n)}$, we have:

$$\sum_j \eta_{jn} \mathbf{x}_j x_i + \frac{\eta_{in}}{2C} = \sum_i \lambda_i s_{ijn} \tag{8}$$

This implies that solving the optimization problem (4) is equivalent to solve a standard LS-SVM [15] whose the target is the weighted sum of each label $\sum_i \lambda_i s_{ijn}$.

Here we use $\varOmega$ to denote the matrix $\varOmega = [K + \frac{\mathbf{I}}{2C}]$ where K is the kernel matrix $K = \{\mathbf{x}_i \mathbf{x}_j | i, j \in 1 \dots L\}$. To simplify our notation, let $\eta'_n = M^{-1} S_n$ where S_n is the matrix $S_n = \{s_{ijn} | i \in M; j \in L\}$ and $\varOmega^{-1}$ is the inverse of matrix $\varOmega$.

Let $\eta_{jn} = \sum_i \lambda_i \eta'_{ijn}$. According to [2], the Leave-one-out estimation of the example $\mathbf{x}_j$ for the nth binary SVM can be written as:

$$\hat{y}_{jn} = \sum_i \lambda_i \left(s_{ijn} - \frac{\eta'_{ijn}}{\varOmega_{jj}^{-1}} \right) \tag{9}$$

where $\varOmega_{jj}^{-1}$ is the jth diagonal element of $\varOmega^{-1}$.

3.2 Cross-entropy loss for imitation parameter estimation

From the previous part, we have already found a effective way to estimate the output of the SVM. The optimal imitation parameters, can be found by solving the following optimization problem:

$$\min \quad L_c(\lambda) = \frac{1}{2} \sum_i^M ||\lambda_i||^2 + \frac{1}{L} \sum_{j,n} \ell\left(y_{in}, \hat{y}_{jn}(\lambda)\right) \quad s.t. \quad \sum \lambda_i = 1 \tag{10}$$

Algorithm 1 GDSDA-SVM

```
Input: Input examples $X = \{\mathbf{x}_1, ..., \mathbf{x}_L\}$, number of classes $N$, number of sources $M$,
  3D label matrix, $S = [Y_1, Y_2, ..., Y_M]$ with size $L \times M \times N$, temperature $T$
Output: Target model $f_t = Wx$
  Compute $\Omega = [K + \frac{\mathbf{I}}{2C}]$
  Compute imitation parameter $\lambda$ with Algorithm 2
  Compute new label $Y_{new} = \sum_i \lambda_i Y_i$
  Compute $\eta = \Omega^{-1} Y_{new}$
  Compute $w_n = \sum_j \eta_{jn} x_j$
```

Here we use the ℓ-2 regularization term to control the complexity of λs so that the target model can achieve better generalization performance. For the loss function $\ell(\cdot, \cdot)$, We use the cross-entropy loss function.

$$\ell\left(y_{in}, \hat{y}_{jn}\left(\lambda\right)\right) = y_{in}\log(P_{jn}) \quad for \quad P_{jn} = \frac{e^{\hat{y}_{jn}}}{\sum_h e^{\hat{y}_{jh}}} \tag{11}$$

Cross-entropy pays less attention to a single incorrect prediction which reduces the affect of the outliers in the training data. Moreover, cross-entropy works better for the unlabeled data with our "fake label" strategy. As we mentioned in our "fake label" strategy, we use one-hot strategy to encode the hard labels of the labeled examples while encoding the unlabeled examples with gray code. When we use cross-entropy, it can automatically ignore penalties of the unlabeled examples and reduce the noise introduced by our "fake label" strategy. The derivative can be written as:

$$\frac{\partial \ell(\lambda)}{\partial \lambda_i} = \sum_n \left(s_{ijn} - \frac{\eta'_{ijn}}{\Omega^{-1}_{jj}}\right)(P_{jn} - y_{jn}) \tag{12}$$

To summarize, we describe GDSDA-SVM in Algorithm 1. As the optimization problem (10) is strongly convex, it is easy to prove that Algorithm 2 can converge to the optimal λ with the rate of $O(\log(t)/t)$ where t is the optimization iteration (We are not able to show our proof here due to the space limit).

4 Experiments

In this section, we demonstrate the empirical performance of our algorithm GDSDA-SVM on the benchmark dataset Office. Specifically, we provide two different settings: single source and multi-source transfer scenarios for GDSDA-SVM.

Dataset: There are 3 subsets in Office datasets, Webcam (795 examples), Amazon (2817 examples) and DSLR (498 examples), sharing 31 classes. In our experiments, we use DSLR and Webcam as the source domain and Amazon as the target domain. We use the features extracted from Alexnet [10] FC7 as the input features for both source and target domain. The source models are trained with multi-layer perception (MLP) on the whole source dataset.

Algorithm 2 λ Optimization

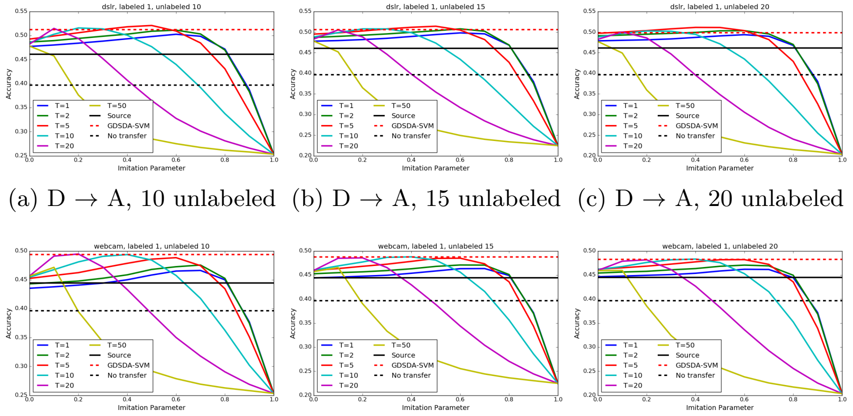
```
Input: Input examples $X$, number of classes $N$, size of sources $M$, 3D label matrix $S$,
  temperature $T$, optimization iteration $iter$, Kernel matrix $\varOmega$
Output: Imitation parameter $\lambda$
  Initialize $\lambda = \frac{1}{M}$,
  Let $S_n$ be the label matrix of $S$ for class $n$
  for Each label $S_n$ do
    Compute $\eta'_n = \varOmega^{-1} S_n$
  end for
  for $it \in \{1, ..., iter\}$ do
    Compute $\hat{y}_{jn}$ and $P_{jn}$ with (9) and (11)
    for each $\mathbf{x}_j$ in $X$ do
      $\varDelta_\lambda = \varDelta_\lambda + \sum_n \left( s_{ijn} - \frac{\eta'_{ijn}}{\varOmega^{-1}_{jj}} \right) (P_{jn} - y_{jn})$
    end for
    $\varDelta_\lambda = \varDelta_\lambda / L$, $\lambda = \lambda - \frac{1}{it}(\varDelta_\lambda + \lambda)$ , $\lambda = \lambda / \sum \lambda_i$
  end for
```

4.1 Single Source for Office datasets

In this experiment, we compare our algorithm under the setting where there is just one source model. Specifically, we perform two groups of experiments using Amazon dataset as the target domain and DSLR and Webcam datasets as the source domains respectively. As we mentioned, there are significantly fewer labeled examples than unlabeled ones in real SDA applications. Therefore, in each group of experiment, we just use 1 labeled example per class with 3 different sizes of unlabeled example (10, 15 and 20 per class).

To show the effectiveness of GDSDA-SVM, we show the performance of GDSDA using brute force to search the imitation parameter λ in the range $[0, 0.1, ..., 1]$ with different temperature T as the baselines. Meanwhile, we show the performance of the source model on the target task, denoted as "Source" and the performance of a target model (using LIBLINEAR[7]) trained with only labeled examples in the target domain denoted as "No transfer". To avoid the randomness, we perform each experiment 10 times and report the average result. For GDSDA-SVM, we use temperature $T = 20$ for all experiments in this part. The experimental results are shown in Figure 3.

From the results of brutal force search, it is clear that the value of imitation parameter can greatly affect the performance of the target model. Also, we can see that, when we only use the unlabeled data for distillation, i.e. $\lambda = 0$, as we expected, GDSDA can still slightly outperform the source model. This means GDSDA can effectively transfer the knowledge between different domains even merely with the unlabeled data. As we increase the value of imitation parameter, i.e. introducing the hard labels from the target domain, the performance of GDSDA can be further improved. As we mentioned before, even though our "fake label" strategy would introduce extra noise, the noise can be limited by setting the proper value to imitation parameter and the target model can still



(a) D → A, 10 unlabeled (b) D → A, 15 unlabeled (c) D → A, 20 unlabeled

(d) W → A, 10 unlabeled (e) W → A, 15 unlabeled (f) W → A, 20 unlabeled

Fig. 3: Experiment results on DSLR→Amazon and Webcam→Amazon when there are just a few labeled examples. The experiments use only 1 labeled example per class. The results of DSLR→Amazon and Webcam→Amazon are shown in figure (a)-(c) and (d)-(e) respectively. GDSDA-SVM is trained with temperature $T = 20$. λ on the X-axis denotes the imitation parameter for the hard label and the corresponding imitation parameter for the soft label is set to $1 - \lambda$.

get improved performance compared to the baselines. More importantly, we can see that GDSDA-SVM can achieve the competitive results compared to baselines using brutal force search in D→A experiments. In W→A experiments, it achieves the best performance among all methods. This indicates that we can effectively (about 6 times faster than brutal force search) obtain a good target model with our imitation parameter estimation method.

4.2 Multi-Source for Office datasets

In this experiment, we train the target model for the Amazon dataset and adapt the knowledge from the rest of two source domains, Webcam and DSLR. We use the similar settings as our single source experiment and perform 2 groups of experiments using 1 labeled and 2 labeled examples per class respectively. We use temperature $T = 5$ and the results of multi-source GDSDA-SVM are denoted as SVM_Multi. Here we use two single source GDSDA-SVMs (SVM_w and SVM_d trained with Webcam and DSLR respectively) as the baselines. We also show the best performance of the brutal force search model (SVM_BF). We search temperature in range $T = [1, 2, 5, 10, 20, 50]$ and each imitation parameter in range $[0, 0.1, ..., 1]$, making sure that their sum equals to 1. The experiment results are shown in Figure 4.

From the results, we can see that, when we have 2 source domains, SVM_Multi can still leverage the knowledge effectively and outperform any single source

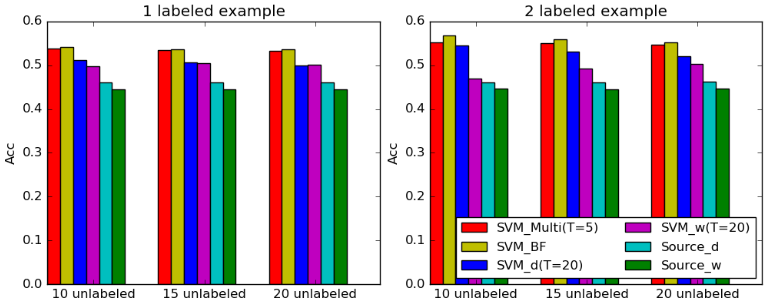


Fig. 4: D+W→A, Multi-source results comparison.

model trained with GDSDA. This shows that the imitation parameter estimated by our method can effectively balance the importance of each source to achieve improved performance. SVM_Multi performs slightly worse than the best result found by brutal force search in some experiments. However, considering their time complexity (GDSDA-SVM is around 30 times faster than brutal force search), SVM_Multi still has its advantage in real applications.

5 Conclusion

In this paper, we propose a framework called *Generalized Distillation Semi-supervised Domain Adaptation* (GDSDA) that can effectively leverage the knowledge from the source domain for SDA problem. To make GDSDA more effective in real applications, we proposed a method called GDSDA-SVM and show that GDSDA-SVM can effectively determine the imitation parameter for GDSDA.

References

1. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Advances in neural information processing systems. pp. 2654–2662 (2014)
2. Cawley, G.C.: Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In: Neural Networks, 2006. IJCNN'06. International Joint Conference on. pp. 1661–1668. IEEE (2006)
3. Daumé III, H., Kumar, A., Saha, A.: Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing. pp. 53–59. Association for Computational Linguistics (2010)
4. Donahue, J., Hoffman, J., Rodner, E., Saenko, K., Darrell, T.: Semi-supervised domain adaptation with instance constraints. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
5. Duan, L., Xu, D., Tsang, I.W.: Learning with augmented features for heterogeneous domain adaptation. In: Proceedings of the International Conference on Machine Learning. pp. 711–718. Omnipress, Edinburgh, Scotland (2012)

6. Duan, L., Xu, D., Tsang, I.W.H., Luo, J.: Visual event recognition in videos by learning from web data. vol. 34, pp. 1667–1680. IEEE (2012)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research 9(Aug), 1871–1874 (2008)
8. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. NIPS Deep Learning and Representation Learning Workshop (2014)
9. Karl, D., Bidigare, R., Letelier, R.: Long-term changes in plankton community structure and productivity in the north pacific subtropical gyre: The domain shift hypothesis. Deep Sea Research Part II: Topical Studies in Oceanography 48(8), 1449–1470 (2001)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1106–1114 (2012)
11. Lopez-Paz, D., Schölkopf, B., Bottou, L., Vapnik, V.: Unifying distillation and privileged information. In: International Conference on Learning Representations (2016)
12. Luo, P., Zhu, Z., Liu, Z., Wang, X., Tang, X.: Face model compression by distilling knowledge from neurons. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
13. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
14. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: In Proceedings of International Conference on Learning Representations (2015)
15. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural processing letters 9(3), 293–300 (1999)
16. Tommasi, T., Orabona, F., Caputo, B.: Learning categories from few examples with multi model knowledge transfer. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36(5), 928–941 (2014)
17. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: The IEEE International Conference on Computer Vision (ICCV) (2015)
18. Urban, G., Geras, K.J., Kahou, S.E., Aslan, O., Wang, S., Caruana, R., rahman Mohamed, A., Philipose, M., Richardson, M.: Do deep convolutional nets really need to be deep (or even convolutional)? In: International Conference on Learning Representations (workshop track) (2016)
19. Vapnik, V., Izmailov, R.: Learning using privileged information: Similarity control and knowledge transfer. Journal of Machine Learning Research 16, 2023–2049 (2015)
20. Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks 10(5), 988–999 (1999)
21. Yao, T., Pan, Y., Ngo, C.W., Li, H., Mei, T.: Semi-supervised domain adaptation with subspace learning for visual recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2142–2150 (2015)