Fast Generalized Distillation for Semi-supervised Domain Adaptation

Abstract

Semi-supervised domain adaptation (SDA) can be applied in many real applications. In this paper, we propose our paradigm, called GDSDA, that uses the recently proposed framework *Generalized Distillation* (GD) (Lopez-Paz et al. 2016) to solve the SDA problem. We demonstrate the reason why GDSDA can work for SDA problem and show that it can work in the different setting such as unsupervised and semi-supervised learning scenarios. Then to make GDSDA more practical for real applications, we propose a novel parameter estimation method, called GDSDA-SVM which uses SVM as the base learner for GDSDA and can effectively estimate the key parameter of GDSDA, i.e. the imitation parameter. Specifically, we use ℓ_2-loss in GDSDA-SVM and show that we can effectively estimate the imitation parameter by minimizing the Leave-one-out loss of the target model on the training data. Experiment results show that our method can estimate the imitation parameter effectively in domain adaptation.

Introduction

Domain adaptation can be used in many real applications, which assumes that the data distribution in the target domain differs from the data distribution in the source domain. Previous methods show that carefully modeling the source data to compensate the domain shift between different domains can significantly improve the performance on the target domain (Donahue et al. 2013). In real applications, it is often very expensive to obtain sufficient labeled examples while there are abundant unlabeled examples in the target domain. *Semi-supervised domain adaptation* (SDA) tries to utilize some unlabeled data from the target domain to compensate the domain shift as well as a few labeled data (Karl, Bidigare, and Letelier 2001). Typically, the labeled data are too few to construct a good classifier alone. How to effectively utilize the unlabeled data is an important issue in SSDA.

Recently, a framework called *Generalized Distillation* (**GD**) (Lopez-Paz et al. 2016) was proposed, which allows the knowledge to be transferred between the teacher and student models effectively. GD can be considered as a hybrid framework of two popular paradigms, *Distillation* (Hinton, Vinyals, and Dean 2014) and *privileged information* (Vapnik and Izmailov 2015). In GD, the student learner tries to distill the knowledge of teacher model trained with the privileged information, and mimick 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 (Lopez-Paz et al. 2016). Given that GD has such ability in various learning scenarios, people would ask the following two questions: (1) Can GD be applied to solve the SDA problem? (2) Is there any obstacle when we apply GDSDA for real 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. Secondly, we propose a novel algorithm GDSDA-SVM, that makes GDSDA more effective for practical applications. Specifically, to answer the first question, we show that the machine learner trained with our GDSDA framework can effectively exploit the knowledge from the source domain and outperform the source model it learned from when labeled data is sparse. Different from many other paradigms where the source knowledge is required in both training and testing process, such as (Kuzborskij and Orabona 2013), the source knowledge is only required in the training process of GDSDA and the machine learner can work on itself along when testing. More interestingly, we demonstrate that GDSDA can utilize the knowledge from most of the existing classifier.

Then we argue that the imitation parameter of GDSDA controls the amount of knowledge transferred from the source which can greatly affect the performance of the machine learner in real applications (see Section). However, according to previous work (Lopez-Paz et al. 2016; Tzeng et al. 2015), the imitation parameter can only be determined by either brute force search or background knowledge. It is ideal to find a method that can determine the imitation parameter automatically, especially when there are more than one imitation parameter to be determined.

Therefore, we propose a novel imitation parameter estimation method for GDSDA, called GDSDA-SVM that uses SVM as the base learner and can determine the imitation parameter automatically. In particular, inspired by (Cawley 2006), we use ℓ_2-loss for GDSDA-SVM and show that the Leave-one-out cross validation (LOOCV) loss can be cal-

culated 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. In addition, 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 domain adaptation problems. (2) We propose the GDSDA-SVM that can effectively find the optimal imitation parameter for GDSDA.

The rest of this paper is organized as follow: In Section , we describe the related work on privileged information and distillation in domain adaptation. Section we propose our framework of GDSDA and provide some statistic analysis. Based on that, we propose our GDSDA-SVM in Section . Experimental results are shown in Section . Some discussion and conclusion are provided in Section .

Related Work

As we use GD to solve SDA problem, we will introduce the related work on both areas.

In SDA, many works have been proposed to utilize the unlabeled data. Yao et al. (Yao et al. 2015) proposed a framework, named Semi-supervised Domain Adaptation with Subspace Learning (SDASL) to correct data distribution mismatch and leverages unlabeled data. Donahue et al.(Donahue et al. 2013) proposed a framework for adapting classifiers by "borrowing" the source data to the target domain using a combination of available labeled and unlabeled examples. Daume et al. (Daumé III, Kumar, and Saha 2010) proposed a method by augmenting the feature space to compensate the domain shift. Duan et al. (Duan et al. 2012) proposed a method using the unlabeled data to measure the mismatch between the domains based on the maximum mean discrepancy.

There are also many works related to GD in computer vision community. Sharmanska et al. (Sharmanska, Quadrianto, and Lampert 2013) proposed a Rank Transfer method that uses attributes, annotator rationales, object bounding boxes, and textual descriptions as the privileged information for object recognition. Motiian et al. (Motiian et al. 2016) proposed the information bottleneck method with privileged information (IBPI) that leverage the auxiliary information such as like supplemental visual features, bounding box annotations and 3D skeleton tracking data to improve visual recognition performance. Tzeng et al. (Tzeng et al. 2015) proposed a CNN architecture for domain adaptation to leverage the knowledge from limited or no labeled data using the soft label. Urban et al. (Urban et al. 2016) use a small shallow net to mimick the output of a large deep net while using layer-wised distillation with ℓ_2 loss of the outputs of student and teacher net. Similarly, Luo et al. (Luo et al. 2016) use ℓ_2 loss to train a compressed student model from the teacher model for face recognition. Gupta et al. (Gupta, Hoffman, and Malik 2016) use supervision transfer to distill the knowledge from a trained CNN with unlabeled data or just a few labeled data.

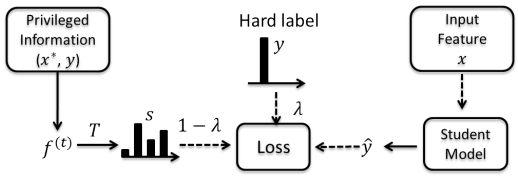


Figure 1: Illustration of Generalized Distillation training process.

Generalized Distillation for Semi-supervised Domain Adaptation

As we mentioned, GDSDA is a paradigm using generalized distillation for 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 with the SDA problem. Finally, we show that to achieve an optimal model, we should carefully set the value of imitation parameter to minimize the training error.

An overview of Generalized Distillation and GDSDA

Distillation (Hinton, Vinyals, and Dean 2014) and *Learning Using Privileged Information* (LUPI) (Vapnik and Izmailov 2015) are two paradigms that enable machines to learn from other machines. Both methods address the problem how to build a student model that can learn from the advanced teacher models. Recently, Lopez et al. (Lopez-Paz et al. 2016) proposed a framework called *generalized distillation* that unifies both techniques and show that it can be applied in many scenarios.

In GD, the training data can be represented as a collection of the triples:

$$\{(x_1, x_1^*, y_1), (x_2, x_2^*, y_2) \dots (x_n, x_n^*, y_n)\}$$

where x^* is the privileged information associate to the label y which is not accessible in the test process. Therefore, the goal of GD is to train a model, called student model with the guidance of the the privileged information.

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 with 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 $f^{(s)}$ using the pair $\{(x_i, y_i), (x_i, s_i)\}_{i=1}^n$ using:

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

Here, λ is the imitation parameter to balance the importance between the hard label y_i and the soft label s_i.

GD can be used in many scenarios such as multi-task learning, semi-supervised learning, and reinforcement learning. As generalized distillation only required for the training inputs $\{x_i, y_i\}_{i=1}^n$ and the output s from the teacher function f_t, it can be naturally applied in domain adaptation, called *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 respectively.

An important issue the extend GD to SDA is that, in Eq. (2), each example is assigned a hard label (ground truth) and a soft label. However, in SDA, we are not able to obtain the hard label for unlabeled data. Here similar to GD (Lopez-Paz et al. 2016), we use the "fake label" strategy to the unlabeled data: for the labeled examples, we use *one-hot* strategy to encode their labels while use gray code (all 0s) to the unlabeled examples as their label. Thus, each example in the target domain has its own label now. It is reasonable to argue that the "fake label" strategy would introduce extra noise and degrade the performance. However, we will show in our experiment that when the labeled examples are rare, we can still achieve improved performance by setting the proper value to the imitation parameter (See the single source experiment in Section).

Suppose we have $M-1$ 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 follow:

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 some temperature $T > 0$.
3. Learn the target model f_t using the $(M+1)$-tuple $\{x_i, y_i, y_{i1}^*, \ldots, y_{M-1}^*\}_{i=1}^L$ with some imitation parameters $\{\lambda_i\}_{i=1}^M$ using (3):

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

Compared to other works of SDA which require to utilize each example of the source domain, by either re-weighting (Donahue et al. 2013; Duan et al. 2012) or augmentation (Daumé III, Kumar, and Saha 2010), GDSDA only requires the trained model from the source domain to generate the soft label. Considering the fact that it is more convenient to manipulate the source model than each of the examples in the source domain, GDSDA can be more effective than those previous method. For example, if we want to use ImageNet (Deng et al. 2009) as the source domain, it is almost impossible to access each of the millions of the examples while there are many well trained models publicly available online to be leveraged. Also, GDSDA is able to handle the multi-class scenario while some previous work, such as SHFA(Duan, Xu, and Tsang 2012) can only solve the problem of binary class in SSDA. Moreover,it is clear that GDSDA is compatible with any type of source model that can output the soft label.

Statistic analysis of GDSDA

In this part, we provide statistic analysis for the scenarios where GDSDA would work. Before we provide the analysis, there are two basic assumptions for GDSDA to work well in domain adaptation: the *assumption of distillation and the assumption of transfer*.

Assumption of Distillation: The capacity (i.e. VC dimension) of target model f_t is smaller than the capacity of source model f^*. This assumption is inherited from distillation. **Assumption of Transfer:** 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 some source models trained from other domain could perform better than any model trained only with the target training data on the target task.

Let $\hat{R}(f)$ be the training error of the model f with finite VC dimension h on a training data with size L. According to ERM principle, the generalization error bound $R(f)$ of the model is:

$$R(f) \leq \hat{R}(f) + O\left(\sqrt{\frac{h}{L}}\right) \tag{4}$$

As long as the target model can achieve similar training error on the target training data to the training error of the source model, i.e. $\hat{R}(f_t) \approx \hat{R}(f^*)$, it could have better generalization error than the source model, i.e. $R(f_t) \leq R(f^*)$ (Hinton, Vinyals, and Dean 2014). It is worthy to notice that in this process, we don't require any labeled examples from the training set. This means GDSDA can effectively utilize the unlabeled data in SDA problem.

Arguably, the source model is biased on the target task due to the domain shift. More interestingly, as it is suggested in (Hinton, Vinyals, and Dean 2014), we can use some labeled data from the target domain to adjust the bias and achieve a better performance on the target task with Eq. (2). This indicates that the target model trained with GDSDA can be further improved with the help of a few labeled data. Here, we use the imitation parameter λ to control the importance of the soft label from the source model and the true label, i.e. the similarity between the source and target tasks. In addition, GDSDA can effectively utilize the unlabeled data to transfer the knowledge from source domain while using a few labeled data to compensate the domain shift (see the experiment results in Section).

Key parameter: the imitation parameter

From above we can see that GDSDA can effectively transfer the knowledge between source and target tasks. In this part, we theoretically demonstrate that the imitation parameter can greatly affect the performance of the target model.

In GDSDA, we have to decide the values of 2 parameters, the temperature T and imitation parameter λ. The temperature T control the smoothness of the soft label and the imitation parameter λ controls how similar the target model is to the source model. Many previous works have addressed the importance of the similarity control when transferring the knowledge between different domains (Duan, Xu, and Tsang 2012; Duan et al. 2012). Without carefully controlling the amount of the knowledge transferred from the source domain, it is easy for the target model to get degraded performance or even suffer from negative transfer (Pan and Yang 2010). Here we show that the value of imitation parameter can greatly affect the performance of the target model. Specifically, we show that we should choose the imitation parameter that minimizes the empirical risk on the training data for distillation.

Let $f(x, \lambda)$ be a function with a finite VC dimension h that minimizes the number of errors on the training data $\{x_i, y_i\}_{i=1}^{L}$. Let $v(\lambda)$ be the training error. Then according to the VC theory (Vapnik 1999), for an arbitrary loss function $\mathcal{L}(\cdot)$ we have the following bound:

$$P\left(\mathcal{L}\left(y, f(x, \lambda)\right) \geq 0\right) < v(\lambda) + O\left(\sqrt{\frac{h}{L}}\right) \tag{5}$$

In other word, the optimal imitation parameter should be the one that can minimize the training error.

In the previous work, the imitation parameter can only be determined by either brute force search (Lopez-Paz et al. 2016) or background knowledge (Tzeng et al. 2015) which greatly reduces its effectiveness. In domain adaptation, it is common that there could be multiple source models to be exploited. It is ideal to find a method that can determine the imitation parameter automatically.

GDSDA-SVM

In step 3 of GDSDA, we have to decide the value for the imitation parameter. According to (5), to achieve the best performance of the target model, the imitation parameter should be set to minimize the training error. In this section, we illustrate our method GDSDA-SVM that uses SVM as the base learner and can effectively estimate the imitation parameter.

Distillation with multiple sources

As we discussed Section , we should find a method that can estimate the optimal imitation parameter effectively for real applications. To solve this problem, we propose a novel method that can determine the imitation parameter λ autonomously, called GDSDA-SVM. In our GDSDA-SVM, instead of using cross-entropy loss, we use Mean Squared Error (MSE) as our loss function for the following two reasons: (1) Some recently work (Ba and Caruana 2014; Luo et al. 2016; Romero et al. 2015; Urban et al. 2016) show that MSE is also an efficient measurement for the targets model to mimic the behavior of the source model. (2) MSE can provide a closed form cross-validation error estimation and help us to choose the proper imitation parameter effectively.

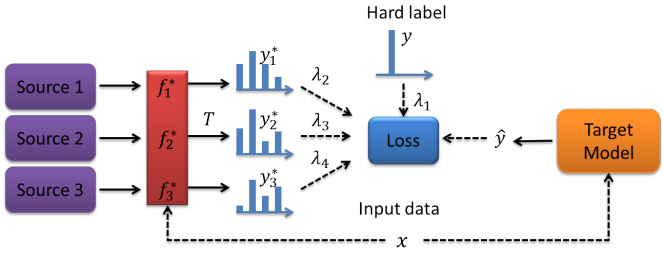


Figure 2: Illustration of GDSDA training process.

Suppose we have L examples $\{x_i, y_i\}_{i=1}^{L}$ 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 $\{Y_i^*\}_{i=1}^{M-1}$ for each of the L examples. For simplicity, we combining the hard label Y and soft label Y^* and use new label matrix: $S = R^{M\times L\times N}$ 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:

$$\begin{aligned} \min \quad & \frac{1}{2}||w_n|\,|^2 + C\sum_{i,j}\lambda_i e_{ijn}^2 \\ s.t. \quad & e_{ijn} = s_{ijn} - w_n x_j \\ & \sum_i \lambda_i = 1 \\ & \lambda_i \in [0,1]; i \in M; j \in L \end{aligned} \tag{6}$$

To solve this optimization problem, we use KKT theorem (Cristianini and Shawe-Taylor 2000) and add the dual sets of variables to the Lagrangian of the optimization problem:

$$\begin{aligned} \mathcal{L} =& \frac{1}{2}|w_n|^2 + C\sum_{i,j}\lambda_i e_{ijn}^2 + \sum_{i,j}\alpha_{ij}^{(n)}\left(s_{ij} - wx_j - e_{ij}\right) \\ & + \beta^{(n)}\left(\sum_i \lambda_i - 1\right) \end{aligned} \tag{7}$$

To find the saddle point,

$$\begin{aligned} \frac{\partial L}{\partial w_n} &= w_n - \sum_j \alpha_{ij}^{(n)} x_j = 0 \rightarrow w_n = \sum_j \alpha_{ij}^{(n)} x_j \\ \frac{\partial L}{\partial e_{ijn}} &= 2C\lambda_i e_{ijn} - \alpha_{ij}^{(n)} = 0 \rightarrow \alpha_{ij}^{(n)} = 2C\lambda_i e_{ijn} \end{aligned} \tag{8}$$

For each example x_j and its constraint of label s_{ijn}, we have $e_{ijn} + w_n x_j = s_{ijn}$. Replacing w_n and e_{ijn}, we have:

$$\lambda_i x_j \sum_k \alpha_{ik}^{(n)} x_k + \frac{\alpha_{ij}^{(n)}}{2C} = \lambda_i s_{ijn} \tag{9}$$

Summing over each constraint of example x_j, we have:

$$\underbrace{\sum_i \lambda_i}_{=1} x_j \sum_k \alpha_{ik}^{(n)} x_k + \sum_i \frac{\alpha_{ij}^{(n)}}{2C} = \sum_i \lambda_i s_{ijn} \tag{10}$$

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

$$\sum_j \eta_{jn} x_j x_i + \frac{\eta_{in}}{2C} = \sum_i \lambda_i s_{ijn} \tag{11}$$

This implies that solving the optimization problem (6) is equivalent to solve a standard LS-SVM whose the target is encoded as $\sum_i \lambda_i s_{ijn}$, i.e. the weighted label matrix S.

Here we use Ω to denote the matrix $\Omega = [K + \frac{\mathbf{I}}{2C}]$ where K is the kernel matrix $K = \{x_i x_j | i, j \in 1 \dots L\}$. 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 Ω^{-1} is the inverse of matrix Ω.

Let $\eta_{jn} = \sum_i \lambda_i \eta'_{ijn}$. The Leave-one-out estimation of the example x_j for the nth binary SVM can be written as (Cawley 2006):

$$\sum_i \lambda_i s_{ijn} - \hat{y}_{jn} = \frac{\eta_{jn}}{\Omega_{jj}^{-1}} = \frac{\sum_i \lambda_i \eta'_{ijn}}{\Omega_{jj}^{-1}}$$

$$\hat{y}_{jn} = \sum_i \lambda_i \left(s_{ijn} - \frac{\eta'_{ijn}}{\Omega_{jj}^{-1}} \right) \tag{12}$$

where Ω_{jj}^{-1} is the jth diagonal element of Ω^{-1}. Now, we have found an efficient way to estimate the output of each binary student model for example x_j. In the following part, we will introduce how to find the optimal λ_i for each of the source models.

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:

$$\min \quad L_c(\lambda) = \frac{1}{2} \sum_i^M ||\lambda_i||^2 + \frac{1}{L} \sum_{j,n} \ell\left(y_{in}, \hat{y}_{jn}(\lambda)\right) \tag{13}$$

Here we use the L2 regularization term to control the complexity of λ so that the estimated λ can perform well on the testing data. For the loss $\ell(\cdot)$, We choose the cross-entropy (CE) as the loss function. Compared to MSE, cross-entropy pay less attention to a single incorrect predicted example which reduced the affect of the outliers in the training data. Moreover, cross-entropy works better with unlabeled data. 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 ignore penalties of the unlabeled example and limit the noise introduced by our "fake label" strategy. Therefore, Let:

$$\mu_{ijn} = s_{ijn} - \frac{\eta'_{ijn}}{\Omega_{jj}^{-1}} \qquad P_{jn} = \frac{e^{\hat{y}_{jn}}}{\sum_h e^{\hat{y}_{jh}}} \tag{14}$$

The derivative can be written as:

$$\frac{\partial \ell(\lambda)}{\partial \lambda_i} = \sum_n \mu_{ijn} \left(P_{jn} - y_{jn}\right) \tag{15}$$

Algorithm 1 GDSDA-SVM

Input: Input examples $X = \{x_1, ..., x_L\}$, number of classes N, size of sources M, 3D label matrix, $S = [Y_1, Y_2, ..., Y_M]$ with size $L \times M \times N$, temperature T, optimization iteration $iter$
Output: Target model $f_t = Wx$
Compute $\Omega = [K + \frac{\mathbf{I}}{2C}]$
Compute imitation parameter λ with Algorithm 2
Compute new label $Y_{new} = \sum_i \lambda_i Y_i$
Compute $\eta = \Omega^{-1} Y_{new}$
Compute $w_n = \sum_j \eta_{jn} x_j$

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 Ω
Output: Imitation parameter λ
Initialize $\lambda = \frac{1}{M}$,
Let S_n be the label matrix of S for class n
for Each label S_n **do**
  Compute $\eta'_n = \Omega^{-1} S_n$
end for
Compute μ using (14)
for $it \in iter$ **do**
  Compute $\hat{y}_{jn}$ and P_{jn} with (12) and (15)
  $\Delta_\lambda \leftarrow 0$
  for each x_j in X **do**
    $\Delta_\lambda = \Delta_\lambda + \sum_n \mu_{ijn} \left(P_{jn} - y_{jn}\right)$
  end for
  $\Delta_\lambda = \Delta_\lambda / L$, $\lambda = \lambda - \frac{1}{itr}(\Delta_\lambda + \lambda)$
end for

In addition, we describe GDSDA-SVM in Algorithm 1. As the optimization problem (13) is strongly convex, it is easy to proof that Algorithm 2 can converge to the optimal λ with the rate of $O(\log(t)/t)$ where t is the optimization iteration[1].

Experiments

We verify our algorithm GDSDA-SVM on the benchmark dataset Office. Moreover, we provide two different settings: single source and multi-source scenarios for GDSDA-SVM.

Dataset

There are 3 subsets in offices 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 (Krizhevsky, Sutskever, and Hinton 2012) FC7 as the input features for both source and target domain. The source model is trained with multi-layer perception (MLP) on the whole source dataset.

[1]The proof is similar to the proof of Lemma 1 in (Shalev-Shwartz et al. 2011).

Single Source for Office datasets

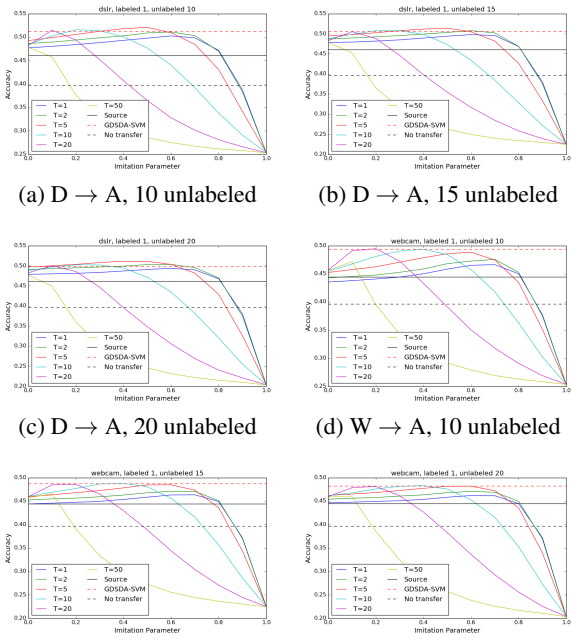
As we mentioned, there are significantly fewer labeled examples than unlabeled ones in real SDA applications. In this experiment, we compare our algorithm with the baselines in this situation where there is just one source model that generates the soft label. Specifically, we perform two groups of experiment using Amazon dataset as the target domain and DSLR and Webcam datasets as the source domain respectively. In each group of experiment, we show 3 results with different settings. We set the size of the labeled example to be 1 per class, and the size of the unlabeled example to be 10, 15 and 20 per class respectively.

To show the effectiveness of GDSDA-SVM, we also use brute force to search the imitation parameter λ in the range $[0, 0.1, ..., 1]$ for comparison with different temperature T. We also show the performance of the source model on the target task, denoted as "Source" and the performance of a target model 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 results. In GDSDA-SVM, we use temperature $T = 20$. The experimental results are shown in figure 3. The imitation parameter of the figures denotes the imitation parameter λ for the hard label and the corresponding imitation parameter for the soft label is set to $1 - \lambda$.

From the results of brutal force search, it is clear that the value of the 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 slightly outperform the source model. This means GDSDA can effectively transfer the knowledge between different domains with just the unlabeled data. As we increase the imitation parameter, i.e. considering the labels from the target domain, the performance of GDSDA can be further improved. As we mentioned in Section , even though our "fake label" strategy would introduce extra noise, it can be limited by setting proper imitation parameter and the target model can still get improved performance compared to the model trained with ground truth labels, i.e. No transfer model in our experiment. More importantly, we can see that the result of GDSDA-SVM can achieve the best performance in most of the situations compared to baselines using brutal force. This indicates that we can effectively (about 6 times times faster) obtain a good target model with our imitation parameter estimation method.

Multi-Source for Office datasets

In this experiment, the target domain, Amazon dataset, adapts the knowledge from the rest of two source domains, Webcam and DSLR. We use the identical settings as the 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 is 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 results (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 1. The experiment results are shown in Figure 4.



(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

Figure 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 $T = 20$.

From the results, we can see that, when we have 2 source domains, SVM_Multi can still leverage the knowledge effectively and performs better than any single source model. 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 brutal force result in some experiments. However, considering their time complexity (GDSDA-SVM is around 30 times faster than brutal force search), we still think that we still believe that SVM_Multi is more effectively in real applications.

Conclusion

In this paper, we propose a framework called *Generalized Distillation* that can effectively leverage the knowledge from the source domain in the SDA scenario. We illustrate several advantages of GDSDA such as effective in small data regimes, compatible with various of kinds of classifiers and effective in knowledge transfer. In particular, we demonstrate that the knowledge can be effectively transferred between different domains by distillation with the unlabeled examples in the target domain. Moreover, we have shown

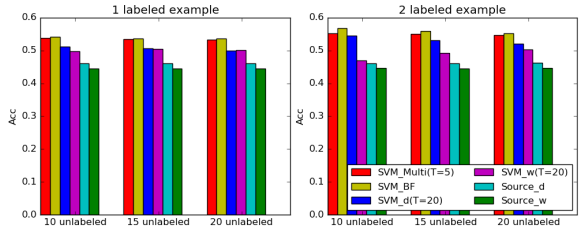


Figure 4: D+W $\rightarrow$ A, Multi-source results comparison.

that the performance of the target model can be further improved with the help of just one labeled example for each class. We also address the importance of the imitation parameter in GDSDA. To make GDSDA more effective in real applications, we proposed a method called GDSDA-SVM which uses SVM as the base learner and can effectively determine the imitation parameter.

References

Ba, J., and Caruana, R. 2014. Do deep nets really need to be deep? In *Advances in neural information processing systems*, 2654–2662.

Cawley, G. C. 2006. Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, 1661–1668. IEEE.

Cristianini, N., and Shawe-Taylor, J. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Daumé III, H.; Kumar, A.; and Saha, A. 2010. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 53–59. Association for Computational Linguistics.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Donahue, J.; Hoffman, J.; Rodner, E.; Saenko, K.; and Darrell, T. 2013. Semi-supervised domain adaptation with instance constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Duan, L.; Xu, D.; Tsang, I. W.-H.; and Luo, J. 2012. Visual event recognition in videos by learning from web data. volume 34, 1667–1680. IEEE.

Duan, L.; Xu, D.; and Tsang, I. W. 2012. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the International Conference on Machine Learning*, 711–718. Edinburgh, Scotland: Omnipress.

Gupta, S.; Hoffman, J.; and Malik, J. 2016. Cross modal distillation for supervision transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*.

Karl, D.; Bidigare, R.; and Letelier, R. 2001. 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.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1106–1114.

Kuzborskij, I., and Orabona, F. 2013. Stability and hypothesis transfer learning. In *Proceedings of the 30th International Conference on Machine Learning*, 942–950.

Lopez-Paz, D.; Schölkopf, B.; Bottou, L.; and Vapnik, V. 2016. Unifying distillation and privileged information. In *International Conference on Learning Representations*.

Luo, P.; Zhu, Z.; Liu, Z.; Wang, X.; and Tang, X. 2016. Face model compression by distilling knowledge from neurons. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Motiian, S.; Piccirilli, M.; Adjeroh, D. A.; and Doretto, G. 2016. Information bottleneck learning using privileged information for visual recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. Fitnets: Hints for thin deep nets. In *In Proceedings of International Conference on Learning Representations*.

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A. 2011. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming* 127(1):3–30.

Sharmanska, V.; Quadrianto, N.; and Lampert, C. H. 2013. Learning to rank using privileged information. In *The IEEE International Conference on Computer Vision (ICCV)*.

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous deep transfer across domains and tasks. In *The IEEE International Conference on Computer Vision (ICCV)*.

Urban, G.; Geras, K. J.; Kahou, S. E.; Aslan, O.; Wang, S.; Caruana, R.; rahman Mohamed, A.; Philipose, M.; and Richardson, M. 2016. Do deep convolutional nets really need to be deep (or even convolutional)? In *International Conference on Learning Representations (workshop track)*.

Vapnik, V., and Izmailov, R. 2015. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research* 16:2023–2049.

Vapnik, V. N. 1999. An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10(5):988–999.

Yao, T.; Pan, Y.; Ngo, C.-W.; Li, H.; and Mei, T. 2015. Semi-supervised domain adaptation with subspace learning for visual recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2142–2150.