Instance Selection and Instance Weighting for Cross-Domain Sentiment Classification via PU Learning

Rui Xia[†], Xuelei Hu[†], Jianfeng Lu[†], Jian Yang[†] and Chengqing Zong[‡]
[†]Department of Computer Science, Nanjing University of Science and Technology, China
[‡]National Laboratory of Pattern Recognition, Institute of Automation, CAS
[†]{rxia, xlhu, lujf, csyang}@njust.edu.cn, [‡]cqzong@nlpr.ia.ac.cn

Abstract

Due to the explosive growth of the Internet online reviews, we can easily collect a large amount of labeled reviews from different domains. But only some of them are beneficial for training a desired target-domain sentiment classifier. Therefore, it is important for us to identify those samples that are the most relevant to the target domain and use them as training data. To address this problem, a novel approach, based on instance selection and instance weighting via PU learning, is proposed. PU learning is used at first to learn an in-target-domain selector, which assigns an in-target-domain probability to each sample in the training set. For instance selection, the samples with higher in-target-domain probability are used as training data; For instance weighting, the calibrated in-target-domain probabilities are used as sampling weights for training an instance-weighted naive Bayes model, based on the principle of maximum weighted likelihood estimation. The experimental results prove the necessity and effectiveness of the approach, especially when the size of training data is large. It is also proved that the larger the Kullback-Leibler divergence between the training and test data is, the more effective the proposed approach will be.

1 Introduction

In the field of natural language processing (NLP), methods for domain adaptation can be classified into two main categorizations: 1) labeling adaptation, which aims to learn a new feature representation or a new labeling function for the target domain; 2) instance adaptation, which learns the importance of labeled data in the source domain based on instance weighting.

Instance adaptation is particularly important for sentiment classification, since we can easily obtain a large collection of labeled data in the vast amount of Internet reviews. But maybe only some of them are beneficial for training a desired target-domain sentiment classifier. Therefore, it is important for us to identify the samples that are the most relevant to the target domain. Under this circumstance, instance adaptation is very necessary for training an effective classifier.

To our knowledge, however, most of the existing domain adaptation studies in sentiment classification are based on labeling adaptation, while the work of instance adaptation are very scarce. Indeed, there are some instance adaptation algorithms proposed by the machine learning community. But most of them are hard to be applied directly to NLP applications with discrete distributions and high dimensional feature space (such as sentiment classification).

In this work, we propose a novel approach, based on instance selection and instance weighting via PU learning, in the context that the labeled training data are domain-mixed[1] and large-size. PU learning is a collection of semi-supervised techniques for training a binary classifier on the positive set $\mathcal{P}$ and the unlabeled set $\mathcal{U}$ only. PU learning is used, in our approach, to learn a binary in-target-domain selector, which assigns an in-target-domain probability to each sample in the training set. Based on the in-target-domain probabilities, two models are proposed: 1) **Instance Selection (PUIS)**, where the instances with higher in-target-domain probability are selected as training data; 2) **Instance Weighting (PUIW)**, where we first calibrate the in-target-domain probability to an appropriate degree, and then use the calibrated probabilities as sampling weights for training an instance-weighted naïve Bayes model, based on the principle of maximum weighted likelihood estimation.

Let us use an artificial example to illustrate our approach. In Figure 1, the red dots denote the target-domain unlabeled data, which are drawn from an unknown distribution. The blue crosses denote the training data, which are drawn from some other distributions. Here, "domain-mixed" means the red dots and blue crosses are partially overlapped, and "large-size" means the number of the blue training data is large. In domain adaptation, our aim is to estimate the distribution of the red-dot target domain, based on the blue-cross training data.

[1] We refer to "domain-mixed" by assuming that the training data may contain some target-domain or target-domain-relevant samples, but we do no know where they are in the mixed training data.

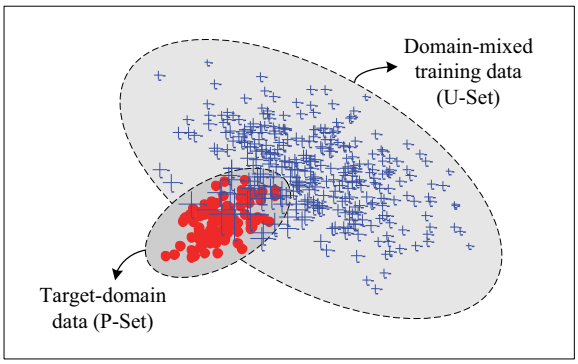

Figure 1: An artificial example of our approach.

Our method can be interpreted as follows: First, we use PU learning (where $\mathcal{P}$ is the set of red dots, and $\mathcal{U}$ is the set of blue crosses) to learn an in-target-domain selector and assign an in-target-domain probability (the value is denoted by the size of the cross) to each blue cross. The blue cross that is closer to the target domain will have a larger size. Second, the in-target-domain probabilities are calibrated to control the rate of size-increase of the blue crosses as the distance becomes small. Finally, the calibrated probabilities are used as sampling weights for training an instance-weighted model, based on the principle of maximum weighted likelihood estimation. Instances that have a larger size will influence the model more than those with a smaller one. In this manner, the gravity-center of the blue crosses will move toward the red dots, and the distribution of the training data will become more approximate to the target-domain distribution.

The approach is evaluated by a range of experiments. The experimental results prove the necessity and effectiveness of the approach, especially when the size of training data is large. Compared to PUIS, PUIW is more effective and stable. The results also indicate that our approach has a good property: the larger the Kullback-Leibler divergence (KLD) between the training and test data, the more effective our approach will be.

2 Related Work

2.1 Domain Adaptation

A survey of domain adaptation and transfer learning can be found in [Pan and Yang, 2010]. There are two main kinds of methods for domain adaptation [2] [Jiang and Zhai, 2007]: labeling adaptation and instance adaptation. We review them respectively.

1) **Labeling adaptation.** Daume III [2007] proposed an easy domain adaptation (EDA) method by duplicating the feature space of both the source and target data. The structural correspondence learning (SCL) model proposed by Blitzer *et al.* [2007] was the representative work of labeling adaptation. Pan proposed two domain adaptation approaches via transfer component analysis (TCA) [Pan *et al.*, 2009] and spectral feature alignment (SFA) [Pan *et al.*, 2010], respectively. [Xia and Zong, 2011] and [Samdani and Yih, 2011] at the same time proposed feature re-weighting based on ensemble learning, but the feature subsets are divided in different ways. Glorot *et al*. [2011] proposed a deep learning approach to address domain adaptation for large-scale sentiment classification. Duan *et al*. [2012] proposed a method based on augmented feature representations for heterogeneous domain adaptation.

2) **Instance adaptation**. In the field of machine learning, this problem is also termed "covariate shift" or "instance selection bias" [Zadrozny, 2004], where the key problem is "density ratio estimation (DRE)". Shimodaira [2000] used kernel density estimation, Dudik *et al.* [2005] used maximum entropy density estimation, and Huang *et al.*[2007] used kernel mean matching, to estimate the density ratio. The KLIEP algorithm proposed by Sugiyama *et al.* [2007] was one of the representative work, where the density ratio was estimated by minimizing the K-L divergence between the true and approximated distributions based on a linear model. Bickel *et al.* [2010] proposed an algorithm based on joint discriminative learning for modeling the covariate shift.

To our knowledge, most of the existing domain adaptation methods in NLP are based on labeling adaptation, while the work of instance adaptation that are particularly designed for NLP applications are very scarce. Recently, Axelrod *et al*. [2011] proposed a method based on a language model, to select parallel training data for machine translation.

In this work, we develop an instance adaptation method for cross-domain sentiment classification, based on in-target-domain selection via PU learning. The process of in-target-domain selection is similar to the work by Bickel *et al.* [2010], where the sampling weights are trained by a joint discriminative learning. Different from that, the sampling weights in our approach are provided by PU learning, rather than a complex joint-training. It leads to computational efficiency and can be easily applied to NLP even when the training size is large.

2.2 PU Learning

Liu *et al*. [2002] summarized several PU learning algorithms and performed a comprehensive evaluation. PU learning has the potential for many NLP applications, especially for the tasks related to distribution similarity. For example, Li *et al.* [2007] use the PU learning framework to identify unexpected instances in the test set which are different in distribution to the training set. Li *et al.* [2010] used PU learning for the problem of entity set expansion, where PU learning was compared with traditional distributional similarity and showed significant superiority.

The usage of PU learning in this work is similar to [Li *et al.*, 2007], but in a reverse direction. They use PU learning to identify unexpected test instances, while we use it to identify the most useful training instances. To our knowledge, it is the

[2] There are still some other types of domain adaptation methods, such as parameter-based methods and semi-supervised-based methods. Due to space limitation, they are not reviewed.

first time that PU learning is used for instance selection and instance weighting in domain adaptation.

3 The Proposed Approach

3.1 PU Learning for In-target-domain Selection

PU learning is stated as follows [Liu *et al*. 2002]: Given a set $\mathcal{P}$ of positive samples of a particular class, and a set $\mathcal{U}$ of unlabeled samples that contains hidden positive and negative samples, a classifier is built using $\mathcal{P}$ and $\mathcal{U}$ for classifying the data in binary decisions.

In our approach, PU learning is employed to build a binary in-target-domain selector. The class labels are positive (in-target-domain) and negative (not-in-target-domain).[3] The target-domain test set $\mathcal{D}_t$ is referred to as the positive set $\mathcal{P}$, and the domain-mixed training set $\mathcal{D}_s$ is referred to as the unlabeled set $\mathcal{U}$. Our aim is to learn a binary classifier based on $\mathcal{P}$ and $\mathcal{U}$, and identify the most positive (i.e., target-domain-likely) samples from the domain-mixed training set (i.e., $\mathcal{U}$ set). There are several PU learning algorithms. In this work, the S-EM algorithm proposed in [Liu *et al.*, 2003] is used to learn the in-target-domain selector. The algorithm is presented in Figure 2.

Algorithm S-EM($\mathcal{P}$, $\mathcal{U}$, a, b)
$\mathcal{N}_r = \varnothing$; % Step 1
$\tilde{\mathcal{P}}$ = Sample($\mathcal{P}$, a%);
Assign each instance in $\mathcal{P} - \tilde{\mathcal{P}}$ the class label $d = 1$;
Assign each instance in $\mathcal{U} \cup \tilde{\mathcal{P}}$ the class label $d = 0$;
Build a NB classifier g using $\mathcal{P} - \tilde{\mathcal{P}}$ and $\mathcal{U} \cup \tilde{\mathcal{P}}$;
Classify each $\mathbf{x} \in \mathcal{U} \cup \tilde{\mathcal{P}}$ using g;
for each $u \in \mathcal{U}$
  if $p(+|u) < b$
    $\mathcal{N}_r \leftarrow \mathcal{N}_r \cup \{u\}$;
$\mathcal{U}_r = \mathcal{U} - \mathcal{N}_r$;
Assign each instance in $\mathcal{P}$ the class label $d = 1$; % Step-2
Assign each instance in $\mathcal{N}_r$ the class label $d = 0$;
Learn an EM classifier f iteratively on $\mathcal{P}$, $\mathcal{N}_r$ and $\mathcal{U}_r$;
for each $\mathbf{x}_n \in \mathcal{U}$ % Final prediction
  Predict $p(d|\mathbf{x}_n)$ using f;
  if $p(d = 1|\mathbf{x}_n) > 0.5$
    Output $\mathbf{x}_n$ as a positive (in-target-domain) sample;
  else
    Output $\mathbf{x}_n$ as a negative (not-in-target-domain) sample;

Figure 2: The S-EM algorithm for In-target-domain Selection.

S-EM contains two main steps. In Step 1, a small set of samples $\tilde{\mathcal{P}}$ is randomly sampled from the positive set $\mathcal{P}$. The positive samples in $\tilde{\mathcal{P}}$ are called "spies". The dataset of $\mathcal{P} - \tilde{\mathcal{P}}$ and $\mathcal{U} \cup \tilde{\mathcal{P}}$ are labeled as positive and negative, respectively. A naive Bayes (NB) classifier is then applied to predict each instance $\mathbf{x} \in \mathcal{U} \cup \tilde{\mathcal{P}}$ and identify a reliable negative sample set $\mathcal{N}_r$ based on a threshold b. In Step 2, given the positive set $\mathcal{P}$, the reliable negative set $\mathcal{N}_r$ and the remaining unlabeled set $\mathcal{U}_r = \mathcal{U} - \mathcal{N}_r$, an Expectation Maximization (EM) algorithm is used for semi-supervised learning, to build a in-target-domain selector that iteratively predict the samples in $\mathcal{U}_r$. Finally, the positive-labeled subset $\mathcal{U}_p \subset \mathcal{U}_r$ is identified as the in-target-domain part. Each training sample $\mathbf{x}_n$ will receive an in-target-domain probability $p(d = 1|\mathbf{x}_n)$.

Based on the in-target-domain probabilities, we propose two models for domain adaptation:

1) Instance Selection (PUIS). The instances with higher in-target-domain probability are selected as training data;

2) Instance Weighting (PUIW). In PUIW, we assume that the target-domain test data are generated by the following In-target-domain Selection process: a target-domain instance is drawn by first sampling $\mathbf{x}$ from the distribution of the training set $p_s(\mathbf{x})$, and then a in-target-domain indicator variable d decides whether $\mathbf{x}$ is moved into the target domain ($d = 1$) or not ($d = 0$). On this basis, the target-domain distribution $p_t(\mathbf{x})$ could be estimated as:

$$p_t(\mathbf{x}) \propto p(d = 1|\mathbf{x}) p_s(\mathbf{x}), \quad (1)$$

where the in-target-domain probabilities $p(d = 1|\mathbf{x}_n)$ will be used as weights for instance weighting.

In the following sections, we introduce PUIW in detail. We first calibrate the in-target-domain probability to an appropriate degree (Section 3.2), and then use the calibrated probabilities as sampling weights for training an instance-weighted naïve Bayes model (Section 3.4), according to the principle of maximum weighted likelihood estimation (Section 3.3).

3.2 In-target-domain Probability Calibration

In the S-EM algorithm, each training sample is assigned with an in-target-domain probability $p(d = 1|\mathbf{x}_n)$. However, S-EM tends to generate a probability either arbitrarily close to 0 or arbitrarily close to 1, due to the naive Bayes assumption. It is important to obtain well-calibrated in-target-domain probabilities for an accurate instance weighting.

It is known that the posterior probability output can also be written in the form of a sigmoid function of the log-odds output $p(d = 1|\mathbf{x}) = \frac{1}{1+e^{-f(\mathbf{x})}}$, where $f(\mathbf{x}) = \log \frac{p(d=1|\mathbf{x})}{p(d=0|\mathbf{x})}$ is the log-odds. We calibrate the in-target-domain probability by multiplying a calibration parameter α ($0 < \alpha < 1$) to the log-odds:

$$q(d = 1|\mathbf{x}) = \frac{1}{1 + e^{-\alpha f(\mathbf{x})}}, \quad (2)$$

where α is used to prevent the in-target-domain probabilities from being too arbitrary and make them more smooth.

[3] The positive and negative labels in PU learning should be distinguished from the sentiment positive and negative labels. In PU learning, positive means in-target-domain; negative means not-in-target-domain.

Replacing the in-target-domain probability in Equation (1) with the calibrated one, we obtain the approximated target-domain distribution:

$$q_t(\mathbf{x}) \propto \frac{1}{1+e^{-\alpha f(\mathbf{x})}} p_s(\mathbf{x}). \quad (3)$$

Note that Equation (1) is a special case of Equation (3) when $\alpha = 1$.

3.3 PUIW for Domain Adaptation

In this section, we use the maximum likelihood estimation (MLE) principle to derive the PUIW framework for domain adaptation.

Assume that $\mathcal{X}$ and $\mathcal{Y}$ are the feature space and label space respectively, and $p(\mathbf{x}, y)$ is the joint probability of $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$. In cross-domain classification, our aim is to find the best parameter θ^* that maximizes the (expected) likelihood of the data drawn from the target domain:

$$\mathcal{L}(\theta) = \int_{\mathbf{x}\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_t(\mathbf{x}, y) \log p(\mathbf{x}, y|\theta) d\mathbf{x}.$$

We use $q_t(\mathbf{x})$ (Equation 3) as an estimation to the real target-domain distribution, and assume $p_s(y|\mathbf{x}) \approx p_t(y|\mathbf{x})$. Then, the MLE problem could be written as:

$$\mathcal{L}(\theta) = \int_{x\in\mathcal{X}} q(d=1|\mathbf{x}) p_s(\mathbf{x}) \sum_{y\in\mathcal{Y}} p_s(y|\mathbf{x}) \log p(\mathbf{x}, y|\theta) d\mathbf{x}.$$

Since the target-domain labeled data are not available, we maximize the (empirical) likelihood of the source-domain training data instead. It turns out to be a weighted version of maximum likelihood estimation (MWLE):

$$\theta^* = \arg\max_\theta \frac{1}{N_s} \sum_{n=1}^{N_s} q(d=1|\mathbf{x}_n) \log p(\mathbf{x}_n, y_n|\theta), \quad (4)$$

where $q(d=1|\mathbf{x}_n)$ is the sampling weight (calibrated in-target-domain probability) of the training instance $\mathbf{x}_n$, and N_s is the size of training data.

3.4 Instance-weighted Naïve Bayes Model

Note that PUIW is a framework that could be integrated with different classification algorithms. In this work, we employ the naive Bayes algorithm, and propose an instance-weighted naïve Bayes (WNB) model.[4]

Parameters of traditional NB are estimated by the MLE principle, which turns out to be the ratio of empirical counts. In the context of WNB, the parameters should be estimated by the MWLE principle (Equation 4), which results in a weighted version of the ratio of empirical counts:

$$p(c_j|\theta^*) = \frac{\sum_{n=1}^{N_s} I(y_n = c_j) q_n}{\sum_{n=1}^{N_s} q_n},$$

$$p(w_i|c_j;\theta^*) = \frac{\sum_{n=1}^{N_s} I(y_n = c_j) q_n N(w_i, \mathbf{x}_n)}{\sum_{n=1}^{N_s} I(y_n = c_j) q_n \sum_{k=1}^{M} N(w_k, \mathbf{x}_n)},$$

where $I(\cdot)$ is the identification function, q_n is short for the calibrated in-target-domain probability, $N(w_k, \mathbf{x}_n)$ is the count of word w_k appears in document $\mathbf{x}_n$, and M is the size of vocabulary. It is reasonable that the counts of features in target-domain-relevant samples will contribute more to the model than those irrelevant samples.

4 Experimental Study

4.1 Experimental Settings and Datasets

In order to evaluate our approach extensively, we conduct the experiments under two different settings: 1) Experiments with a normal-size training set; 2) Experiments with a large-size training set. The results are reported in Section 4.2 and 4.3, respectively.

In Setting 1, the Movie review dataset[5] is used as the source domain data, and each domain of the Multi-domain sentiment datasets[6] (Book, DVD, Electronics, and Kitchen) is used as the target domain. We artificially construct a domain-mixed training data by randomly choosing 200 labeled data from the target domain, and mixing them with 2000 source-domain labeled data. The remaining data in the target domain are used as test set. Unigrams and Bigrams with term frequency no less than 4 are used as features. We repeat the experiments for 10 times, and report the average results.

In Setting 2, we extract one large domain (Video) from the unprocessed part of the Multi-domain sentiment datasets. Reviews of 1 or 2 stars are marked as negative, and reviews of 4 or 5 stars are marked as positive. It finally contains 10,000 class-balanced training samples. Reviews from each of the other 12 domains are used as the target-domain test data. Unigrams with term frequency on less than 5 are used as features. The experiments are also repeated for 10 times and the average results are reported.

4.2 Experiments with A Normal-size Training Set

We first report the results using a normal-size training set. In Figure 3, we draw the accuracy curve of three different methods: 1) For **PUIS**, the bottom x-axis (which denotes the number of selected samples) is used. All training samples are sorted in a descending order according to the in-target-domain probabilities, and we report the system performance trained with an increasing number of selected samples; 2) For **Random Selection** (denoted by "Random"), we report the performance of the same number of randomly selected samples; 3) For **PUIW**, the top x-axis (which

[4] Here, NB is used for sentiment classification. It should be distinguished from the usage of NB in PU learning (S-EM).

[5] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

denotes the value of calibration parameter α) is used. In Table 1, we explicitly compare the best results obtained by different methods, where "All" denotes the results using all training samples. The K-L divergence (KLD) between the training and test data is also reported for each task. The results of PUIW are in terms of that obtained by WNB, and the results of PUIS are in terms of that obtained by NB trained with selected samples.

It should be noticed that we have not compared our model with traditional labeling adaptation approaches (such as SCL) due to different task settings. The focus of this paper is instance adaptation for sentiment classification. Our method do not rely on any additional target domain data, while the labeling adaptation methods need either a small amount of labeled data or a large amount of unlabeled data in the target domain for help.

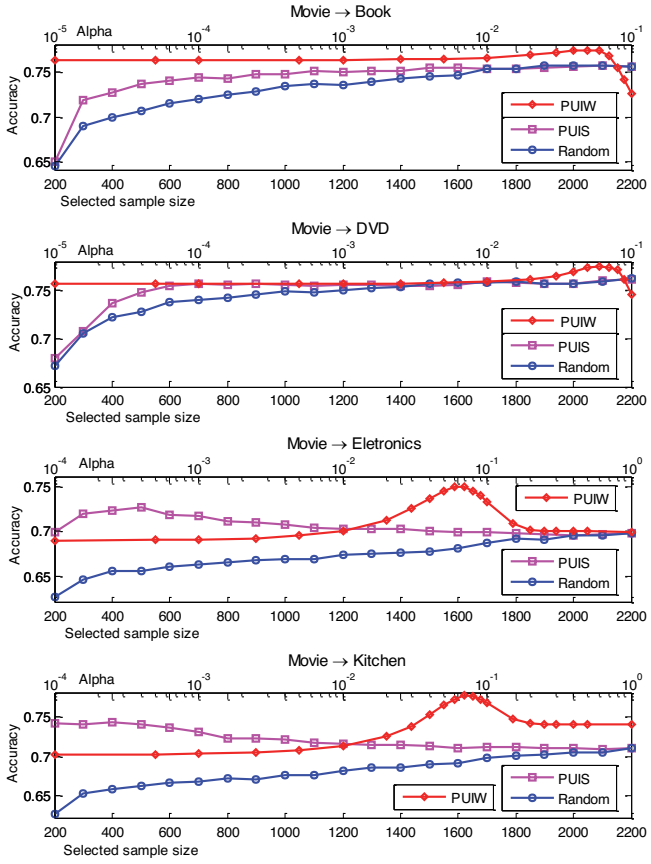

Figure 3: Accuracy curve using a normal-size training set. For Random and PUIS, the bottom x-axis (number of selected samples) is used. For PUIW, the top x-axis is used (the value of α).

Task	KLD	ALL	PUIS	PUIW
Movie → Book	43.81	0.7568	0.7572	**0.7747**
Movie → DVD	33.54	0.7622	0.7622	**0.7818**
Movie → Electronics	104.52	0.6975	0.7265	**0.7500**
Movie → Kitchen	119.70	0.7097	0.7435	**0.7775**
Average	—	0.7316	0.7474	**0.7710**

Table 1: Accuracy comparison using a normal-size training set.

We observe the results in the following aspects:

1) Random Selection. In all tasks, with the increase of training samples, the performance of Random Selection is gradually improved. This agrees with our general knowledge that more training data will improve the machine learning performance.

2) PUIS. Using the same number of training data, PUIS is significantly better than Random Selection. It indicates that PU learning is effective at selecting the most useful training samples. The accuracy increase is significant, especially in high-KLD tasks. For example, in Movie → {Electronics and Kitchen}, the improvements are 2.90 and 3.38 percentages, respectively. When the KLD is relatively low, however, the effect of PUIS are limited. The average increase is 1.58 percentages.

3) PUIW. Compared to PUIS, the effect of PUIW is more significant and robust. Across four tasks, the improvements are 1.79, 1.96, 5.25 and 6.78 (in average 3.94) percentages, respectively. It is reasonable since the calibrated in-target-domain probabilities used as sampling weight are more appropriate than the usage of 0/1 weight in instance selection.

4.3 Experiments with A Large-size Training Set

In this section, we report the experimental results using a large-size training set, which contains 10,000 labeled Video reviews. The reviews from each of the other 12 domains are used as test data respectively. In Figure 4, we draw the accuracy curve of Random Selection, PUIS and PUIW. Due to the space limitation, we only present four of them. In Table 2, we report the best results obtained by different methods as well as the KLD across all 12 tasks.

We observe the results the same way as in Section 4.2. But this time we lay the emphasis on the results that are special for large-size training data

1) The effect of adding more training samples. It can be observed in Random Selection that, when the size of training samples is relatively small (<2000), the accuracy increases significantly as the training size increases. But when the size becomes larger (>3000), the improvement by adding more training data is very slight. It indicates that when the size of training data is already large, adding more training samples will not cause significant improvements.

2) The necessity of PUIS/PUIW. Compared to the results in Section 4.2, the effects of PUIS and PUIW in this case are more significant. The average of increase of PUIS and PUIW are 3.32 and 4.48 percentages, respectively. It is worth noting that in many tasks, only about 10% of selected training samples could result in better performance than that trained with all samples. It indicates that instance adaptation is very necessary when the size of training data is large.

3) The stability of PUIW. In addition to its remarkable performance, it also can be observed that the accuracy curve of PUIW is unimodal and quite stable. While it is not easy to determine the best number of selected instances in PUIS, most of the best accuracy in PUIW are obtained when α locates at the area around 0.1. It suggests another advantage of PUIW: the stability in model selection.

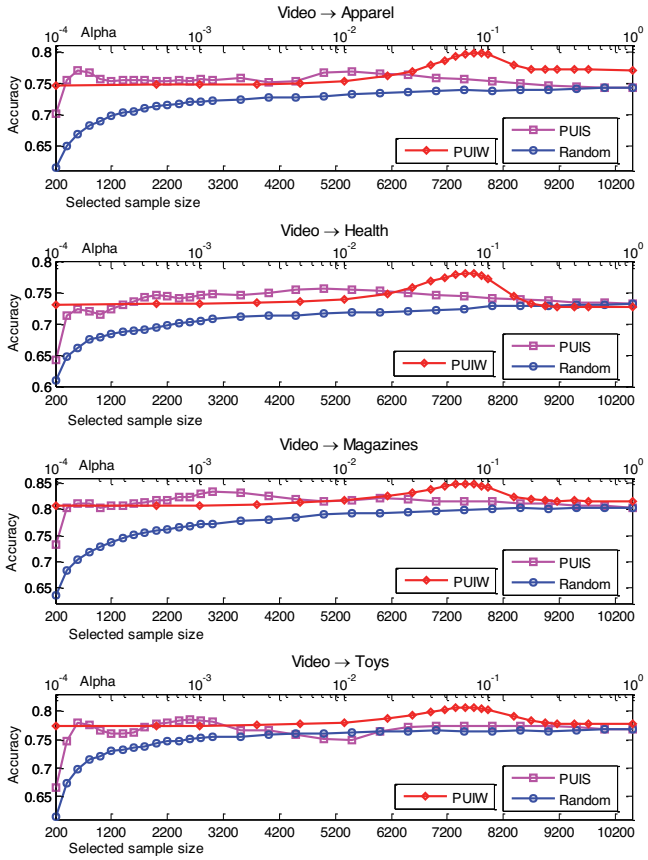

Figure 4: Accuracy curve using a large-size training set. For Random and PUIS, the bottom x-axis (number of selected samples) is used. For PUIW, the top x-axis is used (the value of α).

Task	KLD	ALL	PUIS	PUIW
Video → Apparel	166.69	0.7440	0.7704	**0.7995**
Video → Baby	160.50	0.7494	0.7759	**0.7932**
Video → Books	85.61	0.7328	**0.7952**	0.7893
Video → Camera	146.61	0.7747	0.8164	**0.8278**
Video → DVD	66.71	0.7877	0.8169	**0.8180**
Video → Electronics	143.87	0.7213	0.7603	**0.7712**
Video → Health	159.73	0.7331	0.7576	**0.7826**
Video → Kitchen	155.72	0.7424	0.7736	**0.7980**
Video → Magazines	122.53	0.8030	0.8344	**0.8484**
Video → Music	99.49	0.7562	0.7581	**0.7734**
Video → Software	136.48	0.7411	**0.8078**	0.7830
Video → Toys	134.84	0.7679	0.7858	**0.8066**
Average	—	0.7545	0.7877	**0.7993**

Table 2: Accuracy comparison using a large-size traing set.

4.3 Further Discussion

We finally investigate the relation between K-L divergence (KLD) and the accuracy improvements of our approach. It is known that KLD measures the difference of two distributions. In our tasks, KLD represents the distributional change from the training set to test set. Hence, when KLD is small, the space of improvements in domain adaptation is limited; when KLD increases, the space of improvements also becomes larger.

In Figure 5, we draw the relation of KLD and the accuracy increase gained by PUIW. It can be observed that, KLD and accuracy increase are in a linear relation generally, except for few aberrant points. It indicates that our approach has a good property: the larger the KLD of the training and test data is, the more effective our approach will be.

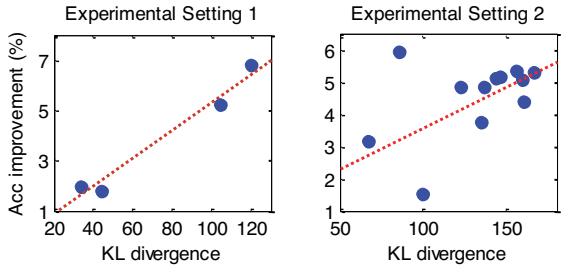

Figure 5: The relation between KLD and Accuracy Increase across all of the tasks (Each point represents one task).

5 Conclusions

In this paper, we propose a novel approach for cross-domain sentiment classification, based on instance selection and instance weighting via PU learning. PU learning is first used to learn an in-target-domain selector, and assign an in-target-domain probability to each sample in the training set. Based on the in-target-domain probabilities, two models namely PUIS and PUIW, are developed. The experimental results prove the necessity and effectiveness of the approach, especially when the size of training data is large. The results also indicate another good property of our approach: the larger the K-L divergence between the training and test data is, the more effective our approach will be.

Shortcomings of this work contain two aspects: 1) Explicit model selection, such as the determination of the number of selected samples and the value of the calibration parameter α, are not involved; 2) It lacks the consideration for labeling adaptation (we simply assume $p_s(y|\mathbf{x}) \approx p_t(y|\mathbf{x})$ in Section 3.3) in instance adaptation. Both of them are very important issues, and we will perform some related investigation in our future work.

Acknowledgments

The research work is supported by the Jiangsu Provincial Natural Science Foundation of China under Grant No. BK2012396, the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20123219120025, the Open Project Program of the National Laboratory of Pattern Recognition (NLPR). The work is also partially supported by the Hi-Tech Research and Development Program (863 Program) of China under Grant No. 2012AA011102 and 2012AA011101, and the National Science Fund for Distinguished Young Scholars under Grant No. 61125305.

References

[Axelrod *et al.*, 2011] A. Axelrod, X. He and J. Gao. Domain Adaptation via Pseudo In-target-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355-362, 2011.

[Bickel *et al.*, 2010] S. Bikckle, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10:2137-2155, 2009.

[Blitzer *et al.*, 2007] J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440-447, 2007.

[Daume III, 2007] H. Daume III. Frustratingly Easy Domain Adaptation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 256-263, 2007.

[Duan *et al.*, 2012] L. Duan, D. Xu and I. W. Tsang. Learning with Augmented Features for Heterogeneous Domain Adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

[Dudik *et al.*, 2005] M. Dudik, R. Schapire, and S. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[Glorot *et al.*, 2011] X. Glorot, A. Bordes and Y. Bengio. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *Proceedings of the International Conference on Machine Learning (ICML)* , pages 513-520, 2011.

[Huang *et al.*, 2007] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[Jiang and Zhai, 2007] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 264-271, 2007.

[Li *et al.*, 2007] X. Li, B. Liu, and S.-K. Ng. Learning to identify unexpected instances in the test set. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, 2007.

[Li *et al.*, 2010] X. Li, L. Zhang, B. Liu, and S.-K. Ng. Distributional Similarity vs. PU Learning for Entity Set Expansion. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2010.

[Liu *et al.*, 2002] B. Liu, W. S. Lee, P. S. Yu, and X. Li. Partically supervised text classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 387-394, 2002.

[Liu *et al.*, 2003] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the IEEE International Conference on Data Mining (ICDM),* pages 179-188, 2003.

[Pan *et al.*, 2009] J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang. Domain Adaptation via Transfer Component Analysis. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, 2009.

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

[Pan *et al.*, 2010] J. Pan, X. Ni, J Sun, Q. Yang and Z Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the International World Wide Web Conference (WWW)* , pages 751-760, 2010.

[Samdani and Yih, 2011] R. Samdani, W. Yih. Domain Adaptation with Ensemble of Feature Groups. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1458-1464, 2011.

[Shimodaira, 2000] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227-244, 2000.

[Sugiyama *et al.*, 2007] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bunau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1433-1440, 2007.

[Xia and Zong, 2011] R. Xia and C. Zong. A POS-based Ensemble Model for Cross-domain Sentiment Classification. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)* , pages 614-622, 2011.

[Zadrozny, 2004] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.