Transferring Naive Bayes Classifiers for Text Classification

Wenyuan Dai[†] Gui-Rong Xue[†] Qiang Yang[‡] Yong Yu[†]
[†]Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{dwyak, grxue, yyu}@apex.sjtu.edu.cn
[‡]Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
qyang@cs.ust.hk

Abstract

A basic assumption in traditional machine learning is that the training and test data distributions should be identical. This assumption may not hold in many situations in practice, but we may be forced to rely on a different-distribution data to learn a prediction model. For example, this may be the case when it is expensive to label the data in a domain of interest, although in a related but different domain there may be plenty of labeled data available. In this paper, we propose a novel transfer-learning algorithm for text classification based on an EM-based Naive Bayes classifiers. Our solution is to first estimate the initial probabilities under a distribution $\mathfrak{D}_\ell$ of one labeled data set, and then use an EM algorithm to revise the model for a different distribution $\mathfrak{D}_u$ of the test data which are unlabeled. We show that our algorithm is very effective in several different pairs of domains, where the distances between the different distributions are measured using the Kullback-Leibler (KL) divergence. Moreover, KL-divergence is used to decide the trade-off parameters in our algorithm. In the experiment, our algorithm outperforms the traditional supervised and semi-supervised learning algorithms when the distributions of the training and test sets are increasingly different.

Introduction

Traditional machine learning assumes the training and test data distributions be identical. Unfortunately, in practice, this assumption is not always satisfied. The training data are often either too few, expensive to label, or easy to be outdated. However, there may be sufficient labeled data that are available under a different distribution in a similar domain. For example, there may often be very few labeled blog documents that are categorized into blog types, but there may be plenty newsgroup ones that are labeled by numerous information sources. The newsgroup and the blog documents are under different distributions, but the features are in the same space (that is, words). Since the word usages in the newsgroup and the blogs are different, if we use the newsgroup documents as training data to train a traditional classifier for predicting the class labels of the blog documents, the performance could be very poor. Thus, it is a critical *transfer-learning problem* how to classify the test data that are under a different distribution from the training data.

In this paper, we focus on the problem of classifying text documents across different distributions. We have a labeled set of documents $\mathcal{D}_\ell$ under a probabilistic distribution $\mathfrak{D}_\ell$, and an unlabeled document set $\mathcal{D}_u$ under another distribution $\mathfrak{D}_u$, probably because the unlabeled data are collected from a different domain from the labeled data. Here, we do not assume that $\mathfrak{D}_\ell$ are $\mathfrak{D}_u$ are Gaussian distributions; there has been work that addressed transfer learning using the Gaussian distributions (DauméIII & Marcu 2006). Intuition tells us that when $\mathfrak{D}_\ell$ and $\mathfrak{D}_u$ are sufficiently similar, transfer learning becomes feasible. In this paper we have two related goals. First, we develop a novel Naive Bayes based algorithm that can be trained on $\mathcal{D}_\ell$, and then *tweaked* and applied to $\mathcal{D}_u$ to ensure good performance. Second, we will empirically measure the difference between the two distributions and relate this difference to the performance.

In this work, we relax the identical-distribution assumption on training and test data, and propose a novel transfer-learning version of the Naive Bayes classifiers, allowing the training and test data distributions to be different. We call this new algorithm Naive Bayes Transfer Classifier (or `NBTC` for short). More specifically, we consider both the labeled training documents $\mathcal{D}_\ell$ under one distribution $\mathfrak{D}_\ell$ and the unlabeled test documents $\mathcal{D}_u$ under another distribution $\mathfrak{D}_u$. `NBTC` first estimates an initial model under the $\mathfrak{D}_\ell$ distribution based on the labeled data. Then, an *Expectation-Maximization* (EM) algorithm is applied to fit the model for $\mathfrak{D}_u$. As a result, the EM algorithm gradually finds a local optimal model under the distribution $\mathfrak{D}_u$, which means the Naive Bayes prediction model transfers from $\mathfrak{D}_\ell$ to $\mathfrak{D}_u$.

To measure the difference between two distributions, we use the Kullback-Leibler (KL) divergence (Kullback & Leibler 1951) to measure the distance between the training and test data. We use this distance to estimate the trade-off parameters in our algorithm, and then parameters can be automatically set as a result. Our experimental results show that `NBTC` gives the best performance on all the transfer learning data sets when compared with several state-of-the-art supervised and semi-supervised algorithms.

Related Work

Transfer Learning

Recently, transfer learning has been recognized as an important topic in machine learning research. Early works include (Thrun & Mitchell 1995; Schmidhuber 1994; Caruana

1997), and Ben-David & Schuller (2003) provided a theoretical justification of transfer learning through multi-task learning.

In this paper, we focus on the topic of learning across different distributions of text data. Among the research works that address cross-distribution learning in text, DauméIII & Marcu (2006) investigated learning from the *in-domain* and *out-of-domain* labeled data to train a statistical classifier (i.e. *Mention Type Classification* and *Tagging*), which is applied for predicting the in-domain unlabeled data. They used *Conditional Expectation Maximization* (CEM) to tackle this problem. Raina, Ng, & Koller (2006) constructed informative priors by discovering highly correlated words, and then used these priors to enhance the classification with limited labeled data. Wu & Dietterich (2004) proposed a classification algorithm using both inadequate training data and large amount of low quality auxiliary data. The auxiliary data could be considered as the labeled data under a different distribution from the test data. They demonstrated some improvement by using the auxiliary data. However, all the above approaches still need some labeled data under the same distribution as the test data in order to work well. In contrast, we assume that the test data are completely unlabeled.

Learning under *sample selection bias* (Zadrozny 2004) or *covariate shift* (Shimodaira 2000), which deals with the case when training and test data are selected from different distributions, could be a similar problem to ours. Sample selection bias has been received plenty of attention in econometrics, e.g. the Novel-prize winning work (Heckman 1979). However, in the sample selection bias problem, the training and test data are selected from the same data source. The only problem they focus is the selection bias, and hence their objective is to correct the bias. Therefore, we consider that correcting sample selection bias is not *real* transfer learning.

Bennett, Dumais, & Horvitz (2003) proposed an ensemble-learning algorithm to tackle the transfer classification problem for text data. They trained a number of basic classifiers based on the training data from different tasks and calculated a *reliability score* for each classifier. An advantage is that they did not need any labeled data under the same distribution as the test set. However, their ensemble based classifiers were selecting among the *predefined* set of basic classifiers, which made their adaptability somewhat limited.

Naive Bayes Classifiers

The Naive Bayes classifiers (Lewis 1992) are known as a simple Bayesian classification algorithm. It has been proven very effective for text categorization. Regarding the text categorization problem, a document $d \in \mathcal{D}$ corresponds to a data instance, where $\mathcal{D}$ denotes the training document set. The document d can be represented as a bag of words. Each word $w \in d$ comes from a set $\mathcal{W}$ of all feature words. Each document d is associated with a class label $c \in \mathcal{C}$, where $\mathcal{C}$ denotes the class label set. The Naive Bayes classifiers estimate the conditional probability $P(c|d)$ which represents the probability that a document d belongs to a class c. Using the Bayes rule, we have

$$P(c|d) \propto P(c) \cdot P(d|c). \tag{1}$$

The key assumption of Naive Bayes classifiers is that the words in the documents are conditionally independent given the class value, so that

$$P(c|d) \propto P(c) \prod_{w \in d} P(w|c). \tag{2}$$

A popular way to estimate $P(w|c)$ is through Laplacian smoothing:

$$P(w|c) = \frac{1 + n(w,c)}{|\mathcal{W}| + n(c)}, \tag{3}$$

where $n(w,c)$ is the number of the word positions that are occupied by w in all training examples whose class value is c. $n(c)$ is the number of word positions whose class value is c. Finally, $|\mathcal{W}|$ is the total number of distinct words in the training set.

Several extensions to the Naive Bayes classifiers have been proposed. Nigam *et al.* (2000) combined the Expectation-Maximization (EM) (Dempster, Laird, & Rubin 1977) and Naive Bayes classifiers for learning from both labeled and unlabeled documents in a semi-supervised algorithm. The EM algorithm is used to maximize the likelihood on both labeled and unlabeled data. The algorithm in (Nigam *et al.* 2000) was applied by (Rigutini, Maggini, & Liu 2005) to the cross-lingual text categorization problem. Liu *et al.* (2002) proposed a heuristic approach Spy-EM that can learn how to handle training and test data with non-overlapping class labels. However, they all assume that the training and test data are under the same distributions, and hence cannot cope well with the transfer-learning problem.

The Naive Bayes Transfer Classification Algorithm

In this section, we present our *Naive Bayes Transfer Classification* algorithm (NBTC). Our main idea is to use the EM algorithm (Dempster, Laird, & Rubin 1977) to find a locally optimal *a Posteriori* hypothesis under the target distribution. We first estimate an initial model based the training data under its distribution $\mathfrak{D}_\ell$. The initial model is treated as an albeit poor estimation of the distribution $\mathfrak{D}_u$ for the test data. The EM algorithm is applied to find a local optimal in the hypothesis space over $\mathfrak{D}_u$, where the estimation should gradually approach the target distribution $\mathfrak{D}_u$. Our algorithm could be considered as an extension of the traditional EM-based Naive Bayes classifiers (Nigam *et al.* 2000) for transfer learning.

Applying the EM Algorithm

We first consider the following problem: given the training set $\mathcal{D}_\ell$ under a distribution $\mathfrak{D}_\ell$, the test set $\mathcal{D}_u$ under a different distribution $\mathfrak{D}_u$, the objective is to find a local optimum of the *Maximum a Posteriori* hypothesis

$$h_{\mathrm{MAP}} = \arg\max_h P_{\mathfrak{D}_u}(h) \cdot P_{\mathfrak{D}_u}(\mathcal{D}_\ell, \mathcal{D}_u|h). \tag{4}$$

In this equation, $P_{\mathfrak{D}_u}(\cdot)$ means the probability that is estimated under the distribution $\mathfrak{D}_u$. Note that, we have to estimate the optimal *a Posteriori* likelihood by considering $\mathcal{D}_\ell$

together with $\mathcal{D}_u$, since $\mathcal{D}_u$ is unlabeled. As all the probabilities are estimated under the test distribution $\mathfrak{D}_u$, we will not be concerned much with the fact that $\mathcal{D}_\ell$ is under a different distribution $\mathfrak{D}_\ell$ from the target distribution $\mathfrak{D}_u$. Estimating the probabilities under $\mathfrak{D}_u$ is a constraint that we use to ensure that the model is designed for $\mathfrak{D}_u$. As we will see later, this constraint is useful for distinguishing the prior probabilities of data under these two distributions.

Instead of maximizing $P_{\mathfrak{D}_u}(h) \cdot P_{\mathfrak{D}_u}(\mathcal{D}_\ell, \mathcal{D}_u|h)$ in Equation (4), we can work with the log-likelihood $\ell(h|\mathcal{D}_\ell, \mathcal{D}_u) = \log P_{\mathfrak{D}_u}(h|\mathcal{D}_\ell, \mathcal{D}_u)$ so that

$$\begin{aligned}\ell(h|\mathcal{D}_\ell, \mathcal{D}_u) \propto \log P_{\mathfrak{D}_u}(h) \\ + \sum_{d\in\mathcal{D}_\ell} \log \sum_{c\in\mathcal{C}} P_{\mathfrak{D}_u}(d|c,h) \cdot P_{\mathfrak{D}_u}(c|h) \\ + \sum_{d\in\mathcal{D}_u} \log \sum_{c\in\mathcal{C}} P_{\mathfrak{D}_u}(d|c,h) \cdot P_{\mathfrak{D}_u}(c|h). \quad (5)\end{aligned}$$

In Equation (5), we may also weaken the influence of the unlabeled data $\mathcal{D}_u$ with a parameter λ ($0 < \lambda < 1$), like the EM-λ algorithm in (Nigam *et al.* 2000), when the size of the labeled data set $\mathcal{D}_\ell$ is small. But, in this paper, we assume the labeled data are sufficient.

The EM algorithm is able to find a local maximum estimation of $\ell(h|\mathcal{D}_\ell, \mathcal{D}_u)$. Similar to the last section, we consider each data instance as a document d, represented by a bag of words $\{w|w \in d \wedge w \in \mathcal{W}\}$ as its features. In the training data $\mathcal{D}_\ell$ under the distribution $\mathfrak{D}_\ell$, each d is associated with a class label $c \in \mathcal{C}$. In the test data $\mathcal{D}_u$ under the distribution $\mathfrak{D}_u$, the class labels are all unknown to us. The EM algorithm looks for a local maximum of $\ell(h|\mathcal{D}_\ell, \mathcal{D}_u)$ by iterating through the following two steps:

- **E-Step:**

$$P_{\mathfrak{D}_u}(c|d) \propto P_{\mathfrak{D}_u}(c) \prod_{w\in d} P_{\mathfrak{D}_u}(w|c). \quad (6)$$

- **M-Step:**

$$P_{\mathfrak{D}_u}(c) \propto \sum_{i\in\{\ell,u\}} P_{\mathfrak{D}_u}(\mathcal{D}_i) \cdot P_{\mathfrak{D}_u}(c|\mathcal{D}_i) \quad (7)$$

$$P_{\mathfrak{D}_u}(w|c) \propto \sum_{i\in\{\ell,u\}} P_{\mathfrak{D}_u}(\mathcal{D}_i) \cdot P_{\mathfrak{D}_u}(c|\mathcal{D}_i) \cdot P_{\mathfrak{D}_u}(w|c,\mathcal{D}_i) \quad (8)$$

In Equation (7), $P_{\mathfrak{D}_u}(c|\mathcal{D}_i)$ can be calculated by

$$P_{\mathfrak{D}_u}(c|\mathcal{D}_i) = \sum_{d\in\mathcal{D}_i} P_{\mathfrak{D}_u}(c|d) \cdot P_{\mathfrak{D}_u}(d|\mathcal{D}_i). \quad (9)$$

In Equation (8), $P_{\mathfrak{D}_u}(w|c,\mathcal{D}_i)$ can be estimated by the Laplacian smoothing. Thus,

$$P_{\mathfrak{D}_u}(w|c,\mathcal{D}_i) = \frac{1 + n_{\mathfrak{D}_u}(w,c,\mathcal{D}_i)}{|\mathcal{W}| + n_{\mathfrak{D}_u}(c,\mathcal{D}_i)}, \quad (10)$$

where

$$n_{\mathfrak{D}_u}(w,c,\mathcal{D}_i) = \sum_{d\in\mathcal{D}_i} |d| \cdot P_{\mathfrak{D}_u}(w|d) \cdot P_{\mathfrak{D}_u}(c|d), \quad (11)$$

$$n_{\mathfrak{D}_u}(c,\mathcal{D}_i) = \sum_{d\in\mathcal{D}_i} |d| \cdot P_{\mathfrak{D}_u}(c|d). \quad (12)$$

In the Equations (7) and (8), $P_{\mathfrak{D}_u}(\mathcal{D}_i)$ is the trade-off parameter for the data set $\mathcal{D}_i$. From the form of $P_{\mathfrak{D}_u}(\mathcal{D}_i)$, it could be understood as a kind of relevance metric between $\mathcal{D}_i$ and $\mathfrak{D}_u$, for $i \in \{\ell, u\}$. Note that the probabilities are measured under the distribution $\mathfrak{D}_u$. Usually, $P_{\mathfrak{D}_u}(\mathcal{D}_u)$ should be greater than $P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$, since $\mathcal{D}_u$ is drawn under $\mathfrak{D}_u$, while $\mathcal{D}_\ell$ is not under the same distribution. Moreover, $P_{\mathfrak{D}_u}(\mathcal{D}_u) > P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$ also ensures the probabilities are estimated towards $\mathfrak{D}_u$.

The probabilities $P(w|d)$ and $P(d|\mathcal{D}_i)$ are independent from $\mathfrak{D}_u$. Hence, we simply use $P(w|d)$ and $P(d|\mathcal{D}_i)$ to approximate $P_{\mathfrak{D}_u}(w|d)$ in Equation (11) and $P_{\mathfrak{D}_u}(d|\mathcal{D}_i)$ in Equation (9), respectively.

Algorithm 1 The Naive Bayes Transfer Classification (NBTC) Algorithm.

Input: A labeled training set $\mathcal{D}_\ell$ under the distribution $\mathfrak{D}_\ell$; An unlabeled test set $\mathcal{D}_u$ under the distribution $\mathfrak{D}_u$; A set $\mathcal{C}$ of all the class labels; A set $\mathcal{W}$ of all the word features; and the maximum number of iterations T.
Output: $P^{(T)}_{\mathfrak{D}_u}(c|d)$

```
1: Initialize $P^{(0)}_{\mathfrak{D}_u}(w|c)$, $P^{(0)}_{\mathfrak{D}_u}(c)$, and $P^{(0)}_{\mathfrak{D}_u}(d)$ via traditional Naive Bayes classifier algorithm.
2: for $t \leftarrow 1$ to $T$ do
3:     for each $c \in \mathcal{C}$ and $d \in \mathcal{D}_u$ do
4:         Calculate $P^{(t)}_{\mathfrak{D}_u}(c|d)$ based on $P^{(t-1)}_{\mathfrak{D}_u}(c)$, $P^{(t-1)}_{\mathfrak{D}_u}(w|c)$ and the Equation (6).
5:     end for
6:     for each $c \in \mathcal{C}$ do
7:         Calculate $P^{(t)}_{\mathfrak{D}_u}(c)$ based on $P^{(t)}_{\mathfrak{D}_u}(c|d)$ and the Equation (7).
8:         for each $w \in \mathcal{W}$ do
9:             Calculate $P^{(t)}_{\mathfrak{D}_u}(w|c)$ based on $P^{(t)}_{\mathfrak{D}_u}(c|d)$ and the Equation (8).
10:        end for
11:    end for
12: end for
```

A description of our NBTC algorithm is shown in Algorithm 1. We first estimate an initial value of $P_{\mathfrak{D}_u}(\cdot)$ by $P_{\mathfrak{D}_\ell}(\cdot)$ using a traditional Naive Bayes classifier. Since the EM algorithm is defined under the distribution $\mathfrak{D}_u$, which is ensured by $P_{\mathfrak{D}_u}(\mathcal{D}_u) > P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$ as discussed before, $P^{(t)}_{\mathfrak{D}_u}(\cdot)$ should gradually converge to a local optimal value of $P_{\mathfrak{D}_u}(\cdot)$ as a result of the iterations. In other words, NBTC transfers the learned model from $P_{\mathfrak{D}_\ell}(\cdot)$ to $P_{\mathfrak{D}_u}(\cdot)$ by using EM algorithm, which shows why NBTC intuitive makes sense.

Estimating the Parameter Setting for EM

Our NBTC algorithm depends on the trade-off parameters $P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$ and $P_{\mathfrak{D}_u}(\mathcal{D}_u)$. Since $P_{\mathfrak{D}_u}(\mathcal{D}_i)$ could be understood as the relevance between $\mathcal{D}_i$ and $\mathfrak{D}_u$, it is natural to use a relevance metric to measure the value of $P_{\mathfrak{D}_u}(\mathcal{D}_i)$.

For this reason, we turn to *relative entropy*, or Kullback-Leibler divergence (Kullback & Leibler 1951) measures, on the feature space $\mathcal{W}$ between the training and test data. This

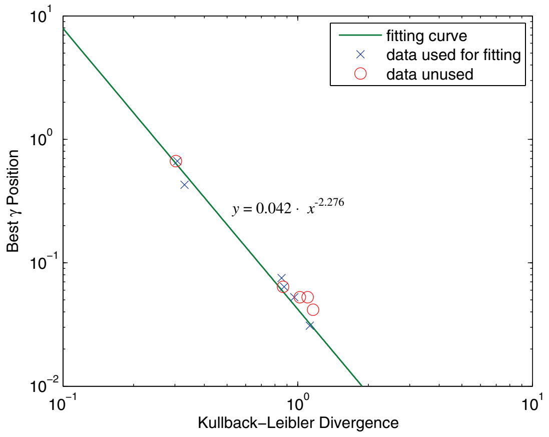

Figure 1: Best γ Position versus KL-divergence

measure provides the distribution distance between the training and test data. More formally, the KL-divergence is defined as,

$$\mathrm{KL}(\mathfrak{D}_\ell||\mathfrak{D}_u) = \sum_{w\in\mathcal{W}} P_{\mathfrak{D}_\ell}(w)\log_2\frac{P_{\mathfrak{D}_\ell}(w)}{P_{\mathfrak{D}_u}(w)}. \tag{13}$$

In the EM algorithm, we estimate $\mathfrak{D}_\ell$ and $\mathfrak{D}_u$ using the feature statistics of $\mathcal{D}_\ell$ and $\mathcal{D}_u$ to obtain the estimations on the respective distributions $\hat{\mathfrak{D}}_\ell$ and $\hat{\mathfrak{D}}_u$. This estimation can be computed as $\mathrm{KL}(\mathfrak{D}_\ell||\mathfrak{D}_u) \approx \mathrm{KL}(\hat{\mathfrak{D}}_\ell||\hat{\mathfrak{D}}_u)$, where

$$\mathrm{KL}(\hat{\mathfrak{D}}_\ell||\hat{\mathfrak{D}}_u) = \sum_{w\in\mathcal{W}} P(w|\mathcal{D}_\ell)\log_2\frac{P(w|\mathcal{D}_\ell)}{P(w|\mathcal{D}_u)}. \tag{14}$$

After the above calculation, we set $\gamma = \frac{P_{\mathfrak{D}_u}(\mathcal{D}_\ell)}{P_{\mathfrak{D}_u}(\mathcal{D}_u)}$ as a *unified parameter* for the EM algorithm. Our objective then becomes how to find the value for γ based on $\mathrm{KL}(\mathfrak{D}_\ell||\mathfrak{D}_u)$.

We propose to empirically find the estimations for the parameters by examining a collection of potential training and test data. As an example, in our experiments to be presented in the next section, we manually tune the γ values on 6 different data sets which are chosen among all the 11 data sets. Each of the data sets consists of a training and a test data sets. For each pair of data sets, we calculate the KL value and plot it against the corresponding γ value. The result is shown in in Figure 1.

From the above figure, we can empirically estimate the best fitting curve as a function:

$$y = 0.042 \cdot x^{-2.276}. \tag{15}$$

As a result, in our NBTC algorithm, we can estimate $P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$ by $\frac{\hat{\gamma}}{1+\hat{\gamma}}$, where $\hat{\gamma} = 0.042 \cdot \mathrm{KL}(\hat{\mathfrak{D}}_\ell||\hat{\mathfrak{D}}_u)^{-2.276}$, and $P_{\mathfrak{D}_u}(\mathcal{D}_u) = 1 - P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$.

Putting the above together, the NBTC algorithm with automatic parameter setting is presented in Algorithm 2.

Algorithm 2 NBTC Algorithm with Automatic Parameter Setting

```
Input: A labeled training set $\mathcal{D}_\ell$ under the distribution $\mathfrak{D}_\ell$; An unlabeled test set $\mathcal{D}_u$ under the distribution $\mathfrak{D}_u$; A set $\mathcal{C}$ of all the class labels; A set $\mathcal{W}$ of all the word features.
Output: $P_{\mathfrak{D}_u}(c|d)$
1: Calculate $\mathrm{KL}(\mathfrak{D}_\ell||\mathfrak{D}_u)$ based on Equation (14).
2: $\gamma \leftarrow 0.042 \cdot \mathrm{KL}(\mathfrak{D}_\ell||\mathfrak{D}_u)^{-2.276}$.
3: Set $P_{\mathfrak{D}_u}(\mathcal{D}_\ell) \leftarrow \frac{\gamma}{1+\gamma}$ and $P_{\mathfrak{D}_u}(\mathcal{D}_u) \leftarrow 1 - P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$.
4: Call NBTC in Algorithm 1 providing $P_{\mathfrak{D}_u}(\mathcal{D}_\ell)$ and $P_{\mathfrak{D}_u}(\mathcal{D}_u)$, and get back a local maximum a posterior hypothesis $P_{\mathfrak{D}_u}(c|d)$.
```

Experiments

In this section we evaluate our algorithm NBTC empirically. We focus on binary text classification problems in the experiment. As we will show later, NBTC significantly improves over the traditional supervised learning methods as well as semi-supervised learning algorithms when applied to data with different distributions.

Data Sets

In order to evaluate the properties of our framework, we conducted experiments on three data sets, 20 Newsgroups[1], SRAA[2] and Reuters-21578[3]. The three data sets we used are not originally designed for evaluating transfer learning. Thus, we split the original data in a way to make the distributions of the training and test data different, as follows.

First, we observe that all three data sets have hierarchical structures. For example, the 20 Newsgroups corpus contains seven top categories. Under the top categories, there are 20 subcategories. We define the tasks as top-category classification problems. When we split the data to generate training and test sets, the data are split based on subcategories instead of based on random splitting. For example, if $\mathbf{A}$ and $\mathbf{B}$ are two top categories, each has two subcategories. Consider a classification task to distinguish the test instances between $\mathbf{A}$ and $\mathbf{B}$. Under $\mathbf{A}$, there are two subcategories $\mathbf{A_1}$ and $\mathbf{A_2}$, while $\mathbf{B_1}$ and $\mathbf{B_2}$ are two subcategories under $\mathbf{B}$. We may split the data set in such a way that $\mathbf{A_1}$ and $\mathbf{B_1}$ are used as training data, and $\mathbf{A_2}$ and $\mathbf{B_2}$ are used as test data. Then, the training and test sets contain data in different subcategories. Their distributions also differ as a result.

The first three columns of Table 1 shows the statistical properties of the data sets. The first two data sets are from SRAA corpus. The next six are generated using 20 Newsgroups data set. The last three are from Reuters-21578 test collection. KL-divergence values between the training and test sets in each test are presented in the second column in the table, sorted in decreasing order from top down. It can be seen that the KL-divergence values for all the data sets are much larger than the case when we simply split the same data set into test and training data, which has a KL value of nearly zero. The next column titled "Documents" shows the size of the data sets used.

[1]http://people.csail.mit.edu/jrennie/20Newsgroups/

[2]http://www.cs.umass.edu/~mccallum/data/sraa.tar.gz

[3]http://www.daviddlewis.com/resources/testcollections/

Data Set	Kullback-Leibler Divergence	Documents		SVM		NBC		NBTC	Improvement	
		$\lvert\mathcal{D}_\ell\rvert$	$\lvert\mathcal{D}_u\rvert$	$\mathcal{D}_\ell$–$\mathcal{D}_u$	$\mathcal{D}_u$–CV	$\mathcal{D}_\ell$–$\mathcal{D}_u$	$\mathcal{D}_u$–CV	$\mathcal{D}_\ell$–$\mathcal{D}_u$	SVM	NBC
real vs simulated	1.161	8,000	8,000	0.266	0.032	0.254	0.035	0.072	72.9%	71.7%
auto vs aviation	1.126	8,000	8,000	0.228	0.033	0.149	0.031	0.037	83.8%	75.2%
rec vs talk	1.102	3,669	3,561	0.233	0.003	0.262	0.006	0.007	97.0%	97.3%
rec vs sci	1.021	3,961	3,965	0.212	0.007	0.155	0.008	0.015	92.9%	90.3%
comp vs talk	0.967	4,482	3,652	0.103	0.005	0.024	0.004	0.005	95.1%	79.2%
comp vs sci	0.874	3,930	4,900	0.317	0.012	0.207	0.021	0.022	93.1%	89.4%
comp vs rec	0.866	4,904	3,949	0.165	0.008	0.077	0.007	0.009	94.5%	88.3%
sci vs talk	0.854	3,374	3,828	0.226	0.009	0.232	0.011	0.023	89.8%	90.1%
orgs vs places	0.329	1,079	1,080	0.454	0.085	0.379	0.247	0.317	30.2%	16.4%
people vs places	0.307	1,239	1,210	0.266	0.113	0.217	0.117	0.195	26.7%	10.1%
orgs vs people	0.303	1,016	1,046	0.297	0.106	0.291	0.129	0.246	17.2%	15.5%

Table 1: Description of the data sets for transfer text classification, including performances given by SVM, NBC and NBTC. The name of the data set orgs vs people indicates that all the positive instances are from the category orgs, while negative ones from people. "$\mathcal{D}_\ell$–$\mathcal{D}_u$" means training on $\mathcal{D}_\ell$ and testing on $\mathcal{D}_u$; "$\mathcal{D}_u$–CV" means 10-fold cross-validation on $\mathcal{D}_u$.

Experimental Results

In Table 1, in the columns under "SVM", "NBC" and "NBTC", we show two groups of classification results. First, "$\mathcal{D}_\ell$–$\mathcal{D}_u$" denotes the error rate obtained when a classifier is trained based on training set $\mathcal{D}_\ell$ and applied to $\mathcal{D}_u$; this measures the transfer-learning ability of the respective classifier. The column titled "$\mathcal{D}_u$–CV" denotes the best case obtained by the corresponding classifier, where the best case is to conduct a 10-fold cross-validation on test set $\mathcal{D}_u$ using that classifier. Note in obtaining the best case for each classifier, the training part is labeled data from $\mathcal{D}_u$ and the test part is also from $\mathcal{D}_u$, according to different folds. Thus, this is the case of same-distribution classification, which gives the best possibly result for that classifier.

Finally, the last two columns of the table show the improvement by NBTC over SVM and NBC, respectively on transfer learning (that is, over the "$\mathcal{D}_\ell$–$\mathcal{D}_u$" experiments). As explained earlier, NBC (Lewis 1992) is the traditional Naive Bayes classifiers. SVM is the Support Vector Machines (Boser, Guyon, & Vapnik 1992) that is implemented by SVM^{light} with a linear kernel function as in (Joachims 2002). As can be seen from Table 1, our NBTC algorithm dramatically outperforms both traditional SVM and NBC on all data sets. Furthermore, the improvement over these two classifiers are more profound when the KL difference is very large (on top of the table). In addition, when compared with the best-case scenarios, NBTC stays very close to the best cases of both classifiers. NBTC makes the most significant improvement when the KL-divergence is between 0.85 and 1.10. Furthermore, as expected, when the KL-divergence is too large or too small, the improvement will be lessened.

We further compared NBTC with semi-supervised learning. Semi-supervised learning uses the labeled information and statistical distribution information on the unlabeled data to make a joint prediction on all unlabeled data. In our problem, one can consider the training data as labeled, and the test data as unlabeled. By comparing with semi-supervised learning, we show that even a slightly more sophisticated treatment of the data sets will not result in the good performance we see in NBTC, if no distribution difference is taken into account. For semi-supervised learning, we implemented the Transductive Support Vector Machines (TSVM) (Joachims 1999) and traditional EM-based Naive Bayes Classifier (ENBC) (Nigam *et al.* 2000). For TSVM, we used a linear kernel function as in (Joachims 2002), for the semi-supervised transductive learning. The experimental results are shown for all three algorithms on all the data sets in Table 2, where we show the error rates. It can be seen that the performance of NBTC is always better than those of TSVM and ENBC on the data sets.

Data Set	K-L	TSVM	ENBC	NBTC
real vs simulated	1.161	0.130	0.090	0.072
auto vs aviation	1.126	0.102	0.045	0.037
rec vs talk	1.102	0.040	0.008	0.007
rec vs sci	1.021	0.062	0.028	0.015
comp vs talk	0.967	0.097	0.006	0.005
comp vs sci	0.874	0.183	0.042	0.022
comp vs rec	0.866	0.098	0.021	0.009
sci vs talk	0.854	0.108	0.030	0.023
orgs vs places	0.329	0.436	0.323	0.317
people vs places	0.307	0.231	0.197	0.195
orgs vs people	0.303	0.297	0.253	0.246

Table 2: Comparing NBTC with two semi-supervised methods TSVM and ENBC

In Figure 2, we present the convergence curves on all the experimental data sets. It can be seen that NBTC always converges at or close to the best performance points, where the rates of the convergence are very fast. NBTC converges in less than 20 iterations on most data sets, and converges on all the data sets within 30 iterations. Therefore, we believe NBTC is very efficient for these data.

Conclusion and Future Work

In this paper, we addressed the issue of how to classify text documents across different distributions. In our setting, the labeled training data are available but have a different distribution from the unlabeled test data. We have developed a transfer-learning algorithm based on the Naive Bayes classifiers, called NBTC. The NBTC algorithm applies the EM

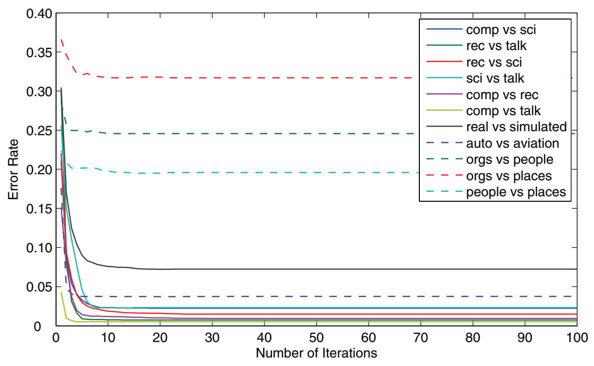

Figure 2: The Error Rate Curves during Iteration on all the Data Sets

algorithm to adapt the NB model learned from the old data for the new. It first estimates the model based under the distribution ($\mathfrak{D}_\ell$) of the training data. Then, an EM algorithm is designed under the distribution ($\mathfrak{D}_u$) of the test data. KL-divergence measures are used to represent the distribution distance between the training and test data. An empirical fitting function based on KL-divergence is used to estimate the trade-off parameters in the EM algorithm. Experiments are conducted to show that `NBTC` can give the best performance among all the learning methods tested on all the data sets. Moreover, `NBTC` also shows excellent convergence property.

There are several areas in which we can improve this work. First, in the future we will try to extend the `NBTC` algorithm for application domains other than text classification. Second, when depicting the relation between best γ and KL values, we empirically adapt the best fitting curve to the distribution distances. However, this curve might not be very precise, and it would be nice to derive some theoretical measures as well. Besides the KL-divergence, there might be some other parameters that can be shown to correlate with the trade-off parameters.

Acknowledgements

Wenyuan Dai, Gui-Rong Xue and Yong Yu are supported by a grant from National Natural Science Foundation of China (NO. 60473122). Qiang Yang is supported by a grant from Hong Kong research grant (HK-RGC 621606). We thank the anonymous reviewers for their great helpful comments.

References

Ben-David, S., and Schuller, R. 2003. Exploiting task relatedness for multiple task learning. In *Proceedings of the Sixteenth Annual Conference on Learning Theory*.

Bennett, P. N.; Dumais, S. T.; and Horvitz, E. 2003. Inductive transfer for text classification using generalized reliability indicators. In *Proceedings of ICML-03 Workshop on The Continuum from Labeled and Unlabeled Data*.

Boser, B. E.; Guyon, I.; and Vapnik, V. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

DauméIII, H., and Marcu, D. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26:101–126.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39:1–38.

Heckman, J. J. 1979. Sample selection bias as a specification error. *Econometrica* 47:153–161.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of Sixteenth International Conference on Machine Learning*.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines*. Ph.D. Dissertation, Kluwer.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *Annals of Mathematical Statistics* 22(1):79–86.

Lewis, D. D. 1992. *Representation and learning in information retrieval*. Ph.D. Dissertation, Amherst, MA, USA.

Liu, B.; Lee, W. S.; Yu, P. S.; and Li, X. 2002. Partially supervised classification of text documents. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 387–394. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using em. *Machine Learning* 39(2-3):103–134.

Raina, R.; Ng, A. Y.; and Koller, D. 2006. Constructing informative priors using transfer learning. In *Proceedings of Twenty-Third International Conference on Machine Learning*.

Rigutini, L.; Maggini, M.; and Liu, B. 2005. An em based training algorithm for cross-language text categorization. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*.

Schmidhuber, J. 1994. On learning how to learn learning strategies. Technical Report FKI-198-94, Fakultat fur Informatik.

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

Thrun, S., and Mitchell, T. M. 1995. Learning one more thing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*.

Wu, P., and Dietterich, T. G. 2004. Improving svm accuracy by training on auxiliary data sources. In *Proceedings of the Twenty-First International Conference on Machine Learning*.

Zadrozny, B. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*.