Similarity-based Weighting Approach for Transfer Learning

Yuan Shi  
Advisor: Jiahai Wang  
Technical Report  
School of Software, Sun Yat-sen University

Abstract - Transfer learning addresses learning problems in which the training and test data are across different domains, i.e. under different distributions or feature spaces. The objective is to transfer useful knowledge in the abundant source-domain data to benefit learning in the target domain. In this report, we present a simple approach to enable instance-based transfer learning. The key idea is to find some source-domain instances similar to target-domain instances using a classifier learnt on target-domain training data. Those similar instances are then given positive weights and integrated into the target-domain training set to build a better classifier for the target domain. The main advantage of our approach over existing weighting approaches lies that the instance-weight is efficiently determined in a heuristic way. Also, experimental results reveal that the approach is highly effective and robust.

1. Introduction
Transfer learning is a recent machine learning framework that aims at reusing knowledge learnt in a source domain to solve problems in a related target domain. Under this framework, the training data in the source domain and the test data in the target domain have different distributions or even different feature sets [1], which violates the fundamental assumption in traditional machine learning research – the training and test data are under the same distribution. When the distribution changes, most machine learning approaches require users to re-collect sufficient training data in the target domain [2], yet transfer learning can reuse the old training data, thus significantly save the expense in labeling new samples.

Transfer learning has been studied for more than one decade and various approaches have been developed. Most of these approaches can be summarized into two contexts: instance-transfer and feature-representation-transfer [2]. In instance-transfer, training samples/instances in the source domain are re-weighted according to their impact on learning in the target domain [3, 4, 5, 6, 7, 8]. Different from instance-transfer, feature-representation-transfer tries to learn a common feature representation across domains to reduce the domain divergence as well as the training error, therefore knowledge in different domains can be transferred easily [9, 10, 11, 12].

In this report, we focus on instance-transfer and adopt the problem setting used in [5], which assumes that a large amount of old training data in the source domain (denoted by diff-distribution data) and a small amount of new training data in the target domain (denoted by same-distribution data) are available. Both the diff-distribution and same-distribution training data are labeled. Note that although a few same-distribution data are provided, they are not adequate to train a good classifier for the target domain. Therefore we need to utilize the same-distribution to find out some useful diff-distribution instances and reuse them to improve the learning in the target domain.

In this work, we aim to develop a novel instance-transfer approach to deal with the above problem.
Our idea is based on the intuition that the useful diff-distribution instances are similar to the same-distribution instances. Here, “similar” means that these diff-distribution instances look like matching the distribution in the target domain. Thus, the key step of our approach is to measure the similarity between the data in the source and target domains. In this report, we present a simple yet effective way to measure the similarity. We first train a classifier using the same-distribution data, and then employ the classifier to predict the labels of diff-distribution instances. For each diff-distribution instance, if the predicted label is consistent with its true label, the instance is considered as similar to the same-distribution data and its weight is set as the labeling confidence of the classifier on this instance. Otherwise, its weight is set to zero. After that, diff-distribution instances with positive weights are added to the same-distribution training data. Finally, the classifier for the target domain is trained on the combined training set.

The proposed approach can be referred to as similarity-based weighting approach, which assigns weights to diff-distribution data in a heuristic way, i.e. according to the measured similarity. Our experimental results show that this approach can build a high-performance classifier for the target domain effectively and efficiently.

The rest of the report is organized as follows: Section 2 briefly reviews related work in instance-transfer. Section 3 gives the formal definitions for the problem. We present the proposed approach in Section 4 and report the experimental results in Section 5. Section 6 concludes the report and discusses our future work.

2. Related Work

In this section, we briefly discuss some related researches with similar problem setting as ours.

Wu and Dietterich [3] presented a way of integrating auxiliary (source-domain) training data into $k$-nearest neighbor and support vector machine algorithms. The methodology is to minimize a weighted sum of two loss functions, one for original training data and the other for auxiliary data. Their experiments demonstrated that using auxiliary data can improve the classification performance when the original training data is inadequate. Note that in their work, all auxiliary training data are given the same weight, but our approach sets the weights for each source-domain instance.

Liao et al. [4] introduced an auxiliary variable for each auxiliary sample to reflect the distribution mismatch and the auxiliary variables are estimated as a byproduct, along with the classifier. They also proposed an active learning method to select unlabeled target-domain samples to be labeled. Our approach differs from their method in that the instance weights are directly estimated by the learnt classifier, without introducing additional variables into the learning process.

Dai et al. [5] proposed a boosting algorithm, namely $TrAdaBoost$, as an extension of the $AdaBoost$ algorithm. The idea is to use a small number of target-domain data to find the useful source-domain training data by iteratively adjusting their weights. In each iteration, a base classifier is trained on the weighted training data and used to predict the labels of target-domain data. The weights are then updated based on the prediction error. Yet iteratively re-weighting can be computationally expensive given that it may need to re-train a base learner many times. Our method, by contrast, determines the weights more efficiently.

Xing et al. [6] provided a perspective different from the above work. They attempted to refine the classification labels instead of the classification model. The proposed algorithm utilizes the mixture distribution of the training and test data as a bridge to enable better transfer learning.

3. Problem Statement

We use the same problem setting in [5], where the source and target domain data are assumed to adopt
the same feature set $X$ and the label set $Y$ (in this report, $Y = \{0, 1\}$), but their distributions are different. The training set (denoted by $T$) is composed of two data sets: same-distribution training set $T_s$ and the diff-distribution training set $T_d$. $T_s$ represents the labeled data in the target domain: $T_s = \{(x'_i, y'_i), (x''_i, y''_i), \ldots, (x'_n, y'_n)\}$, where $x'_i, y'_i \in X(i = 1, 2, \ldots, n)$ is an input feature vector and $y'_i \in Y$ is its corresponding label. $n$ is the number of instances in $T_s$. $T_d$ represents the labeled data in the source domain: $T_d = \{(x''_i, y''_i), (x''_i, y''_i), \ldots, (x''_m, y''_m)\}$, where $x''_i, y''_i \in X(i = 1, 2, \ldots, m)$ and $y''_i \in Y$ are the input feature vector and its label, respectively. $m$ is the number of instances in $T_d$. In general, $n \ll m$, meaning that there are a large number of diff-distribution training data but a small number of same-distribution training data. In addition, the test data set is denoted by $S$, which contains some unlabeled data in the target domain: $S = \{x'_r, x''_r, \ldots, x'_r\}$, where $x'_r \in X(i = 1, 2, \ldots, r)$ are the feature vector and $r$ denotes the number of test instances. Note that $S$ and $T_s$ are under the same distribution. The problem of transfer learning is to learn a function $f: X \rightarrow Y$ that minimizes the prediction error on the unlabeled test set $S$.

4. Similarity-based Weighting Approach

The proposed approach aims at using some diff-distribution instances similar to the same-distribution data to train a better classifier. There are three main steps in the approach:

Step 1) train a classifier $\hat{\mathbf{c}}$ using $T_s$.

Step 2) use $\hat{\mathbf{c}}$ to predict the label of each instance in $T_d$. If the prediction is correct, the weight of the diff-distribution instance is set as the labeling confidence of the classifier. Otherwise, its weight is set to zero.

Step 3) Construct a new training set $T'$ by adding the diff-distribution instances with positive weights to $T$, then train a new classifier $\hat{\mathbf{c}}'$ on $T'$.

To illustrate the approach in detail, we use the weighted $k$-nearest neighbor ($k$-NN) [13] to implement the classifier and set the weights of all same-distribution data as one. Since $k$-NN is a lazy learner i.e. no computation is performed in the training process, we only focus on Step 2), which is conducted as follows. For each diff-distribution instance $(x''_i, y''_i)$ in $T_d$, let $\Omega_i$ denote the set of its $k$-nearest neighboring instances in $T_s$, if the majority of instances in $\Omega_i$ have the same label as $(x''_i, y''_i)$, then its weight is set to the proportion of the same-label instances, i.e. the labeling confidence of $k$-NN. Mathematically, the weight is calculated by

$$w''_i = 1 - \frac{1}{k} \sum_{x'_j \in \Omega_i} |y'_j - y''_i|$$

Otherwise, we set its weight as zero.

The intuition behind the above rule is that, if a diff-distribution instance is similar to the same-distribution data, then its label must be consistent with those of its same-distribution neighbors (See the instance A in Fig. 1). On the contrary, if a diff-distribution instance is very different to the same-distribution data, its label tends to differ from those of its neighbors (See the instance B in Fig. 1). Empirically, a diff-distribution instance more similar to the same-distribution data should be given a higher weight.
What should be pointed out is that the labeling confidence may not reflect the similarity correctly when the diff-distribution instance is very distant from all the same-distribution data. Fig. 2 illustrates the case. Although the predicted label is correct for the distant instance, we cannot regard the instance as similar to the same-distribution data, because the distances between the diff-distribution instance to its neighboring same-distribution instances are relatively large. In such case, the predication by $k$-NN makes little sense, also do other learners. Luckily, this is not a serious problem for $k$-NN because when testing an unlabeled instance in the target domain, the chance for the “bad” instance, i.e. the distant diff-distribution instance, to be selected as a neighbor is relatively small. Therefore, the “bad” instance has little effects on the final classification performance.

However, for some other learners, like the neural networks [14] and support vector machines [15], adding such a “bad” instance to the new training set may make these learners overfitting in the training process. An easy method to solve the problem is to preprocess the diff-distribution data by filtering out the instances too distant from the same-distribution data. The distance threshold can be set manually by prior knowledge or learnt automatically from the same-distribution data.

Fig. 1 Illustration of the intuition behind our approach

Fig. 2 A case when the labeling confidence does not correctly reflect the similarity

5. Experiments
In this section, we test the performance of the proposed approach using the mushroom data set [16]. To fit the transfer learning setting, the data set is split according to the feature stalk-shape, resulting in an enlarging-mushroom subset (containing 4608 instances) and a tapering-mushroom subset (containing
3516 instances) [5]. The KL-divergence [17] between the two subsets is 1.315, which shows that they are under different distributions. Additionally, we randomly select 1500 tapering-mushroom instances to constitute the diff-distribution training set ($T_j$), 1000 enlarging-mushroom instances to form the test set ($S$) and $n$ enlarging-mushroom instances to form the same-distribution training set ($T_i$). Note, in the experiments, we will change the size $n$ to test the impact of the transfer learning.

Since we intend to study the effectiveness of the similarity-based weighting, we just compare our approach (denoted by Tr-$k$-NN) with $k$-NN algorithm using only the same-distribution data. (simply denoted by $k$-NN). In the experiments, $k$ is set to 5. Fig. 3 pictures the experimental results when the ratio between the same-distribution and diff-distribution data is gradually increased from 0.01 ($n = 10$) to 0.25 ($n = 250$). As can be observed, when the ratio is less than 0.1, Tr-$k$-NN always outperforms $k$-NN in terms of the error rate on the test set; when the ratio is larger than 0.1, Tr-$k$-NN achieves the same performance as $k$-NN. This reveals that when there are relatively small amount of the same-distribution data, using our approach to transfer knowledge from the diff-distribution data is highly effective. Also, during the ratio change, Tr-$k$-NN is always better or at least no worse than $k$-NN, which demonstrates the robustness of the proposed approach. Clearly, to give a comprehensive evaluation of our approach, further experiments are required, such as testing on different data sets and comparing with other approaches.

![Fig. 3 The error rate curves when the ratio changes](image)

6. Conclusions and Future Work
In this report, a novel and simple instance-transfer approach, namely similarity-based weighting, is proposed. This approach efficiently measures the similarity of each source-domain instance to the target-domain instances using a classifier trained only in the target domain. The approach is further implemented by $k$-NN and experimental results demonstrate its effectiveness and robustness. Moreover, as we state, the classifier can be implemented by other learning algorithms, yet a preprocess procedure may need to be incorporated to ensure the better learning. In the future, we will explore the following issues:

- Test the proposed approach extensively in different data sets.
- Theoretical analyze the properties of the approach.
- Incorporate distance metric learning [18] to make our approach more robust and effective.
- Extend our approach to handle noisy source-domain instances.
References


