Ensemble of Anchor Adapters for Transfer Learning

Fuzhen Zhuang[1], Ping Luo[1], Sinno Jialin Pan[2], Hui Xiong[3] and Qing He[1]

[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, zhuangfz@ics.ict.ac.cn, luop@ict.ac.cn,heq@ics.ict.ac.cn
[2]Nanyang Technological University, Singapore, sinnopan@ntu.edu.sg
[3]MSIS Department, Rutgers University, hxiong@rutgers.edu

ABSTRACT

In the past decade, there have been a large number of transfer learning algorithms proposed for various real-world applications. However, most of them are vulnerable to *negative transfer*[1] since their performance is even worse than traditional supervised models. Aiming at more robust transfer learning models, we propose an ENsemble framework of *anCHOR adapters* (ENCHOR for short), in which an *anchor adapter* adapts the features of instances based on their similarities to a specific *anchor* (i.e., a selected instance). Specifically, the more similar to the anchor instance, the higher degree of the original feature of an instance remains unchanged in the adapted representation, and vice versa. This adapted representation for the data actually expresses the local structure around the corresponding anchor, and then any transfer learning method can be applied to this adapted representation for a prediction model, which focuses more on the neighborhood of the anchor. Next, based on multiple anchors, multiple anchor adapters can be built and combined into an ensemble for final output. Additionally, we develop an effective measure to select the anchors for ensemble building to achieve further performance improvement. Extensive experiments on hundreds of text classification tasks are conducted to demonstrate the effectiveness of ENCHOR. The results show that: when traditional supervised models perform poorly, ENCHOR (based on only 8 selected anchors) achieves $6\% - 13\%$ increase in terms of average accuracy compared with the state-of-the-art methods, and it greatly alleviates negative transfer.

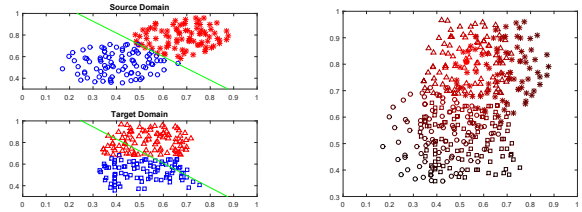
Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning–Machine Learning

1. INTRODUCTION

Traditional classification algorithms often fail to obtain satisfactory performances, since the assumption that the training data from source domain and test data from target domain are drawn from the same distribution does not always hold in real-world applications. Transfer learning focuses on adapting the common knowledge from related source domain to improve the learning performance in target domain with distribution mismatch, and has attracted vast amount of research studies in the past decade [1].

Generally, the proposed transfer learning algorithms can be grouped into two types, i.e., single model [2, 3, 4] learning and ensemble learning [5, 6, 7, 8]. For the first type, Liao et al. [2] first estimated the degree of mismatch of each instance in the source domain with the whole target domain, and then presented an active learning approach for selecting the labeled examples in target domain. Finally they incorporated this information into logistic regression for transfer learning. Chen et al. [4] tried to learn robust data representations by reconstruction, recovering original features from data that are artificially corrupted with noise for transfer learning. All the above methods are only based on single model learning, which may not achieve stable and robust results. In this paper, we try to ensemble the outputs on multi-model level, which considers the local structure of source and target domains simultaneously for each local model. For the second type, there are several transfer learning methods, which focus on learning from multiple source domains [5, 6] and assigning different weights to models [7, 8]. Gao et al. [7] proposed a locally weighted ensemble framework to combine multiple models for transfer learning, where the weights are dynamically assigned according to a model's predictive power on each test example. However, most of these methods do not explicitly exploit the local structures of source and target domain simultaneously.



(a) Source Domain and Target Domain (b) The Selected Anchor and Its Similarity to Other Samples

Figure 1: The Intuitive Example of Why Selecting Anchor

It is often observed that the source and target domains may not be closely related over all the data. However, it is possible to make safe transfer if the training performs only in the same local neighborhood areas of the source and target domains. The intuitive example is shown in Figure 1. In this example, we show a toy data set with two dimensions, and the source domain and target domain data are plotted in Figure 1(a), where red "$*$" and "$\triangle$" denote positive instances and blue "$\circ$" and "$\square$" denote negative ones. Obviously, we can find that the source domain and target domain have different distributions, and the classifier trained from the source domain may not give satisfying predictions on target domain. If we can select the data points from source and target domains that located

[1]Here we say the "negative transfer" occurs when the accuracy from transfer learning algorithm is lower than the one of supervised learning algorithm, i.e., Logistic Regression (LR) in this paper.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983690

in the same local neighborhood area, then the classif cation performance would be better. In Figure 1(b), for given anchor red solid "○", we calculate the similarities between the anchor and the rest data points, then we plot all data points with different colors (i.e., indicating different degree of similarities with the anchor.). Note that, cosine distance is used to compute the similarity of two data points. In this f gure, red data points mean the similar ones with the anchor, while the black ones are dissimilar with the anchor. We can f nd that the data points located in the same local neighborhood area can be selected, and based on them the classif cation performance may be much better. Motivated by this observations, we propose an ENsemble framework of *anCHOR adapters* (ENCHOR for short) for transfer learning, in which each *anchor adapter* adapts the features of instances (from both source and target domains) based on their similarities to a specif c *anchor* (an instance selected from source or target domains). Specif cally, the more similar to the anchor instance, the higher degree of the original feature of an instance remains unchanged in the adapted representation, and vice versa. In other words, the instances from both source and target domains are more likely to be unchanged if they are more similar to the selected anchor. Thus, the data distributions after adaption for both source and target domains are more likely to be similar, and the models learned over the adapted data may achieve better performance (at least in the corresponding local area). Then, based on multiple anchors, multiple anchor adapters can be built and combined into an ensemble for f nal output. Additionally, we develop an effective measure to select the anchors for ensemble building to achieve further performance improvement. Extensive experiments on hundreds of text classif cation tasks are conducted to demonstrate the effectiveness of ENCHOR. The results show that: when traditional supervised models perform poorly, ENCHOR (based on only 8 selected anchors) achieves $6\% - 13\%$ increase in terms of average accuracy compared with the state-of-the-art methods, and it greatly alleviates negative transfer.

The remainder of this paper is organized as follows. We introduce the proposed framework in Section 2, and then propose the anchor selection strategy in Section 3. In Section 4, we conduct extensive experiments on text classif cation problems to demonstrate the effectiveness of the proposed algorithm. Related work is summarized in Section 5. Finally, Section 6 concludes the paper.

2. ENSEMBLE OF ANCHOR ADAPTERS

Next, we propose an ensemble of anchor adapters for transfer learning. Note that, bold letters, such as $\boldsymbol{u}$ and $\boldsymbol{v}$, are used to represent vectors. Data matrices are written in bold upper case, such as $\boldsymbol{X}$ and $\boldsymbol{Y}$. Also, $\boldsymbol{X}_{(i,j)}$ indicates the i-th row and j-th column element of matrix $\boldsymbol{X}$. Calligraphic letters, such as $\mathcal{A}$ and $\mathcal{D}$, are used to represent sets. Finally, we use $\mathbb{R}$ and $\mathbb{R}_+$ to denote the sets of real numbers and nonnegative real numbers. Without specif cal illustration, all the vectors are column vectors.

2.1 Anchor-based Adapter

Given the source domain with labeled data $\mathcal{D}_s = \{\boldsymbol{x}_i^{(s)}, y_i^{(s)}\}|_{i=1}^{n_s}$ and target domain with unlabeled data $\mathcal{D}_t = \{\boldsymbol{x}_i^{(t)}\}|_{i=1}^{n_t}$, and their corresponding word-document co-occurrence matrices are $\boldsymbol{X}_s \in \mathbb{R}_+^{m\times n_s}$ and $\boldsymbol{X}_t \in \mathbb{R}_+^{m\times n_t}$, where m is the number of features and n_s, n_t are respectively the numbers of instances in source domain and target domain. First, we randomly select q anchors $(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_q)$ (i.e., an anchor is an instance) from source or target domain, then compute the similarities between each anchor and the source (target) domain data. To compute the similarity, we can use the Gaussian distance function or cosine distance function. In this paper, we adopt the cosine function $\cos(\boldsymbol{a}, \boldsymbol{x}) = \frac{\boldsymbol{a}^\top \boldsymbol{x}}{\sqrt{\boldsymbol{a}^\top \boldsymbol{a}}\cdot\sqrt{\boldsymbol{x}^\top \boldsymbol{x}}}$. Then for the τ-th anchor, we can adapt the data matrices of $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$ from source and target domains to new data matrices $\hat{\boldsymbol{X}}_s^{(\tau)}$ and $\hat{\boldsymbol{X}}_t^{(\tau)}$, called anchor-adapted matrices in this paper. The computations of $\hat{\boldsymbol{X}}_s^{(\tau)}$ and $\hat{\boldsymbol{X}}_t^{(\tau)}$ are as follows,

$$\begin{aligned}\hat{\boldsymbol{X}}_s^{(\tau)} &= (\cos(\boldsymbol{a}_\tau, \boldsymbol{x}_1^{(s)})\cdot\boldsymbol{x}_1^{(s)}, \cdots, \cos(\boldsymbol{a}_\tau, \boldsymbol{x}_{n_s}^{(s)})\cdot\boldsymbol{x}_{n_s}^{(s)}),\\ \hat{\boldsymbol{X}}_t^{(\tau)} &= (\cos(\boldsymbol{a}_\tau, \boldsymbol{x}_1^{(t)})\cdot\boldsymbol{x}_1^{(t)}, \cdots, \cos(\boldsymbol{a}_\tau, \boldsymbol{x}_{n_t}^{(t)})\cdot\boldsymbol{x}_{n_t}^{(t)}).\end{aligned} \tag{1}$$

By the anchor-based adapter, the high similar instances with anchor from source and target domains are retained, while the importance of the instances with low similarities to anchor are degraded.

There are several advantages can be benef ted from the proposed framework, 1) For each anchor, only the similar instances from source and target domains are selected according to the similarities, then the new constructed source domain and target domain become more similar and their distribution difference is decreased. 2) Since the distributions of the new constructed data matrices are similar, we believe the better prediction results can be achieved. 3) The proposed framework ENCHOR is very easy to parallelize, since the anchors are independent of each other.

2.2 Ensemble of Anchor Adapters

Given q randomly selected anchors $(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_q)$ and according to the anchor-based adapter, we can obtain q pairs of new source and target domains $((\hat{\boldsymbol{X}}_s^{(1)}, \hat{\boldsymbol{X}}_t^{(1)}), (\hat{\boldsymbol{X}}_s^{(2)}, \hat{\boldsymbol{X}}_t^{(2)}), \cdots, (\hat{\boldsymbol{X}}_s^{(q)}, \hat{\boldsymbol{X}}_t^{(q)}))$. We can perform any transfer learning algorithm on each pair of new source domain and target domain, and the output predictions on target domain data can be obtained. For the selected q anchors, we can f nally get q predictions $(\hat{\boldsymbol{G}}_t^{(1)}, \hat{\boldsymbol{G}}_t^{(2)}, \cdots, \hat{\boldsymbol{G}}_t^{(q)})$ for target domain, where $\hat{\boldsymbol{G}}_t^{(\tau)} \in \mathbb{R}_+^{n_t\times c}$ ($\tau \in \{1, \cdots, q\}$, c is the number of instance classes.). Each row of $\hat{\boldsymbol{G}}_t$ denotes the prediction vector of an instance yielding to $\sum_j^c \hat{\boldsymbol{G}}_{t(i,j)} = 1$, and $\hat{\boldsymbol{G}}_{t(i,j)}$ is the prediction probability of instance $\boldsymbol{x}_i^{(t)}$ belonging to class j.

In the proposed framework ENCHOR, all the output predictions can be combined in two ways, one is in weighted manner and the other with simply average predictions,

$$\bar{\boldsymbol{G}}_{t(i,\cdot)}^{w} = \frac{\sum_{\tau=1}^q \cos(\boldsymbol{a}_\tau, \boldsymbol{x}_i^{(t)})\cdot\hat{\boldsymbol{G}}_{t(i,\cdot)}^{(\tau)}}{\sum_{\tau'=1}^q \cos(\boldsymbol{a}_{\tau'}, \boldsymbol{x}_i^{(t)})}, \bar{\boldsymbol{G}}_{t(i,\cdot)}^{a} = \frac{\sum_{\tau=1}^q \hat{\boldsymbol{G}}_{t(i,\cdot)}^{(\tau)}}{q}. \tag{2}$$

Actually, these two ways achieve very similar results in our experiments, so we will only list the weighted version in the experimental section. Our framework ENCHOR is a general framework, which can be adapted to any transfer learning with probabilistic outputs. In this paper, we implement ENCHOR based on non-negative matrix tri-factorization for transfer learning [9].

3. SELECTION STRATEGY FOR ANCHORS

As you have noted in Section 2, the anchors are randomly selected. Actually, the randomly selected anchors may lead to poor performance, since they may be outliers or located in low density area. How to select the effective anchors is very important and challenging to obtain outstanding transfer learning performance. To the best of our knowledge, there has not yet previous work devoting to this task. Next, we will propose our strategy and some baseline strategies for selecting anchors.

3.1 The Proposed Strategy

Given the q anchors $(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_q)$ and the corresponding output predictions $(\hat{\boldsymbol{G}}_t^{(1)}, \hat{\boldsymbol{G}}_t^{(2)}, \cdots, \hat{\boldsymbol{G}}_t^{(q)})$ by the iterating algorithm, we propose an effective measure to select the high-quality

anchors. We require the measure have the following properties, 1) the value range of the measure to be [0,1]; 2) larger value of the measure indicates the better of the anchor. Therefore, we will only select the top ℓ anchors with the largest values of the proposed measure for final ensemble of anchor adapters.

Along this line, we first introduce two criterions based on Shannon Entropy to measure the confidence and class ratio when performing prediction. For each test instance, we hope the classifiers trained on the anchor-adapted data sets can predict the instance to some class with the highest confidence. Take the binary classification problem as an example, if an instance $\boldsymbol{x}$ belongs to the first class, then we think the prediction vector (1, 0) is better than the one (0.6, 0.4), since the classifier assigns $\boldsymbol{x}$ with the label of first class with 100% confidence in the first vector. Following, we give the definition of confidence criterion on the i-th instance in target domain as

$$E_{c(i)} = \sum_{j=1}^{c} \hat{\boldsymbol{G}}_{t(i,j)} \log_c \hat{\boldsymbol{G}}_{t(i,j)}, \tag{3}$$

where $\log_c$ is a base-c logarithm, which can ensure $E_{c(i)} \in [-1, 0]$, and c is the number of instance classes. Large value of $E_{c(i)}$ indicates the classifier can predict the instance $\boldsymbol{x}_i$ into some class with high confidence. Since $E_{c(i)} \in [-1, 0]$, so we make a slight amendment to let $\hat{E}_{c(i)} = E_{c(i)} + 1 \in [0, 1]$, then the confidence criterion over all target domain data is defined,

$$E_c = \frac{1}{n_t} \sum_{i=1}^{n_t} \hat{E}_{c(i)},\ E_c \in [0, 1]. \tag{4}$$

Obviously, when the classifier predicts all the instances from target domain with 100% confidence, E_c gets the largest value 1.

For the class ratio criterion, we assume that it would be better if the classifier can give the predictions with the true class ratio of target domain. Assuming that the true class distribution of target domain is $\boldsymbol{p} = (p_1, p_2, \cdots, p_c)$, $\sum_i^c p_i = 1$, and the predicted class distribution is $\hat{\boldsymbol{p}} = (\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_c)$, $\sum_i^c \hat{p}_i = 1$, where $n_t^{(j)}$ be the number of instances belonging to class j and the label of $\boldsymbol{x}_i^{(t)}$ is computed as $\max_j \hat{\boldsymbol{G}}_{t(i,j)}$, then the class ratio criterion is defined as

$$E_r = \sum_{j=1}^{c} \frac{\hat{p}_i}{p_i \cdot N} \log_c \frac{p_i \cdot N}{\hat{p}_i},\ E_r \in [0, 1], \tag{5}$$

where $N = \sum_{j=1}^{c} \frac{\hat{p}_i}{p_i}$ is the normalization factor. When the predicted class distribution is the same as the true class distribution, i.e., $\hat{p}_c = p_c$, E_r gets the largest value 1.

Based on the above two criterions, we finally come to the proposed measure for selecting high-quality anchors,

$$E_{cr} = E_c \times E_r,\ E_{cr} \in [0, 1]. \tag{6}$$

In this measure, we simultaneously consider the confidence criterion and class ratio criterion. In other word, we hope the classifier not only can predict the instances with high confidence, but also the predicted class distribution can be the same as the true one. However, in real-world applications, we always do not know the true class distribution of target domain. In this case, we simple set $\boldsymbol{p} = (\frac{1}{c}, \frac{1}{c}, \cdots, \frac{1}{c})$ when computing our measure. Fortunately, the experimental results in Section 4 show that the proposed measure can also work well even when the true class distribution is unknown.

After sorting the q values of E_{cr} (each anchor corresponding to a value of E_{cr}), we select the top ℓ anchors with the largest values of E_{cr}. Then, the final output is

$$\bar{\boldsymbol{G}}^{*w}_{t\ (i,\cdot)} = \frac{\sum_{\tau=1}^{\ell} \boldsymbol{w}_{t(i)}^{(\tau)} \cdot \hat{\boldsymbol{G}}_{t(i,\cdot)}^{(\tau)}}{\sum_{\tau'=1}^{\ell} \boldsymbol{w}_{t(i)}^{(\tau')}}. \tag{7}$$

The proposed framework with and without selection strategy are denoted as ENCHOR* and ENCHOR, respectively.

3.2 Some Baseline Strategies

To validate the effectiveness of our strategy, we also introduce some baseline strategies to select anchors. Kullback-Leibler (KL) divergence [10] and Maximum Mean Discrepancy (MMD) [11] are used to measure the distribution difference between different domains, and some works [12, 13] have adopted them for transfer learning. Smaller values of KL divergence and MMD can usually lead to better transfer learning performance, so we also adapt them to anchor selection.

4. EXPERIMENTAL EVALUATION

In this section, we construct hundreds of classification problems to validate the effectiveness of the proposed framework ENCHOR. Note that we only focus on binary classification problems, while obviously our algorithm can handle multi-class classification problems.

4.1 Data Preparation

We adopt the widely use data set *20Newsgroups*[2], which is one of the benchmark data sets for evaluating transfer learning algorithms [1, 7, 14, 3]. This corpus has approximately 20,000 newsgroup documents, which are evenly divided into 20 subcategories. Some similar subcategories are grouped into a top category, e.g., the four subcategories *sci.crypt*, *sci.electronics*, *sci.med* and *sci.space* belong to the top category *sci*. We select three top categories, i.e., *rec*, *sci* and *talk*, to construct the classification problems. The top categories are used for classification, e.g., in the combination *rec vs. sci*, the data from *rec* are positive instances, while the data from *sci* are negative ones.

To construct the transfer learning tasks, we follow the approach in [9]. For the combination *rec vs. sci*, we randomly select a subcategory from *rec* as positive class and a subcategory from *sci* as negative class to produce the source domain. The target domain is similarly constructed, thus in totally 144 ($P_4^2 \cdot P_4^2$) classification tasks are generated for this combination. We have three combinations for the three top categories *rec*, *sci* and *talk*, i.e., *rec vs. sci*, *rec vs. talk* and *sci vs. talk*, and in totally 432 (144×3) classification problems are constructed.

For the above constructed classification tasks, they almost have balanced class distribution. To validate our algorithm can also perform well when the classes are unbalanced, we change the ratios of positive instances and negative instances in target domain on 144 problems of the combination *rec vs. sci*. Specifically, we only randomly sample 50%, 40%, 30%, 20% and 10% negative instances for each task, then the corresponding ratios are 2:1, 2.5:1, 3.3:1, 5:1 and 10:1. The average results are reported on three independent trials.

4.2 Experimental Settings

4.2.1 Baselines

we compare our algorithm ENCHOR with the following state-of-the-art baselines, including

- The supervised algorithm: Logistic Regression (LR) [15];
- The transfer learning methods:
  – Transfer learning based on non-negative matrix tri-factorization

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

(MTrick) [9], which does not consider the local transfer and anchor selection strategy;
− Locally weighted Ensemble (LWE) [7], in which the classif cation models are assigned with different weights according to the local structure of each test instance;
− Marginalized Stacked Denoising Autoencoders (mSDA) [4], which learns robust features for transfer learning;

- We also compare our anchor selection strategy E_{cr} with the following four strategies, conf dence Criterion (E_c) in Eq. (4), class ratio criterion (E_r) in Eq. (5), Kullback-Leibler (KL) divergence [10] and Maximum Mean Discrepancy (MMD) [11].

4.2.2 Parameter Settings

In ENCHOR, the parameters of the number of word clusters k, trade-off factor β, error threshold ε and the maximal number of iterations T are set following the suggestion of MTrick [9], $k = 50$, $\beta = 1$, $\varepsilon = 10^{-11}$, $T = 100$. After some preliminary test, the parameters of q and ℓ are set as 50 and 8 for all selection strategies. The parameters of the baselines are set according to the suggestions of their original papers.

The variables $\hat{\boldsymbol{F}}_s$, $\hat{\boldsymbol{F}}_t$ and $\hat{\boldsymbol{G}}_t$ are initialized as the feature clustering results by PLSA [16] on the whole data set of the source and target domain. We adopt the Matlab implementation of PLSA[3] in the experiments. $\hat{\boldsymbol{G}}_s$ is initialized as the true class information in the source-domain, and $\hat{\boldsymbol{G}}_t$ is initialized as the predicted results of any supervised classif er, which is trained based on the source domain data. In this experiment Logistic Regression is adopted to give these initial results.

4.3 Results on Classif cation Problems with Balanced Classes

4.3.1 Comparison among ENCHOR, MTrick, LWE, mSDA and LR

Indeed, the anchors can be randomly selected from source domain or target domain, and there are different explanations for these two ways. If we select the anchors from source domain, then each time we choose the similar instances from target domain to predict; while if the anchors selected from target domain, each time the similar instances from source domain are chosen for training. In these two ways, the propose algorithms are denoted as ENCHOR$_s$, ENCHOR$_s^*$ for selecting anchors from source domain, and ENCHOR$_t$, ENCHOR$_t^*$ for selecting anchors from target domain. ENCHOR$_s^*$ and ENCHOR$_t^*$ are output by the proposed anchor selection strategy. The detailed results on *rec vs. sci* of all compared algorithms are shown in Figure 2 (The results on the other two combinations are very similar with *rec vs. sci*), and the average accuracies are reported in Table 1. In Figure 2, the classif cation problems are sorted with the increasing order by the accuracies of LR. Thus, the x-axes in these f gures can also indicate the degree of diff culty in knowledge transferring. To clearly validate the superiority of our algorithms, the 144 problems are divided into two parts in Table 1. The f rst part includes the problems with accuracies from MTrick lower than 90%, on which we think MTrick could not perform well. The other part contains the ones whose accuracies are greater than or equivalent to 90%, on which MTrick can achieve satisfying performance. In Table 1, the third column indicates the number of problems whose accuracies from MTrick $< 90\%$ or $\geq 90\%$. The number in the parentheses denotes how many problems suffering from negative transfer for the compared algorithms.

From the results in Figure 2 and Table 1, we have the following insightful observations:

- In Figure 2(a) and Table 1, we f nd that 1) both ENCHOR$_t^*$ and ENCHOR$_s^*$ are signif cantly better than ENCHOR$_t$ and ENCHOR$_s$, which indicates the effectiveness of the proposed selection strategy E_{cr}, and the randomly selected anchors maybe outliers or located in low dense area; 2) from a general view, the results of selecting anchors from target domain are very similar with the ones of selecting anchors from source domain. Therefore, in Figure 2(b) and the following sections, we only list the results of ENCHOR$_t^*$.
- Overall, all the transfer learning algorithms are better than LR, which implies that the traditional learning algorithm may be not suitable for transfer learning tasks.
- ENCHOR$_t^*$ and MTrick dramatically outperform LWE and mSDA. MTrick achieves unstable performance in Figure 2(b), especially when the accuracy of LR is low, which indicates that MTrick can not perform well on diff cult transfer learning problems. Overall, our algorithm ENCHOR$_t^*$ obtains the best results.
- To clear validate the robustness of ENCHOR$_t^*$ and ENCHOR$_s^*$, it can be observed from Table 1 that ENCHOR$_t^*$ and ENCHOR$_s^*$ perform signif cantly better than all the baselines when the accuracy of MTrick is lower than 90%.
- Except ENCHOR$_t^*$ and ENCHOR$_s^*$, all the compared algorithms suffer from negative transfer learning, which again shows the robustness and effectiveness of the proposed anchor selection strategy.

In a word, all the results demonstrate the superiority of the proposed algorithms ENCHOR$_t^*$ and ENCHOR$_s^*$.

4.3.2 Comparison of Anchor Selection Strategies

To investigate the effectiveness of the proposed anchor selection strategy E_{cr}, we compare it with the other four strategies, i.e., E_c, E_r, KL and MMD. T average accuracies over 144 problems are list in Table 2. From these results, we can f nd that the selection strategies E_{cr} and E_r are signif cantly better than the other three strategies MMD, KL and E_c, and E_c obtains the worst results. It is also observed that, all the selection strategies outperform LR with respect to the average accuracies in Table 2. Though E_r performs very similar with E_{cr}, these results are based on the conditions that the problems are with balanced class distributions. In Section 4.4, we will show that the results of E_{cr} is better than E_r when the tasks are with unbalanced classes.

Table 2: Average results (%) on Three Combinations for Selection Strategy Comparison

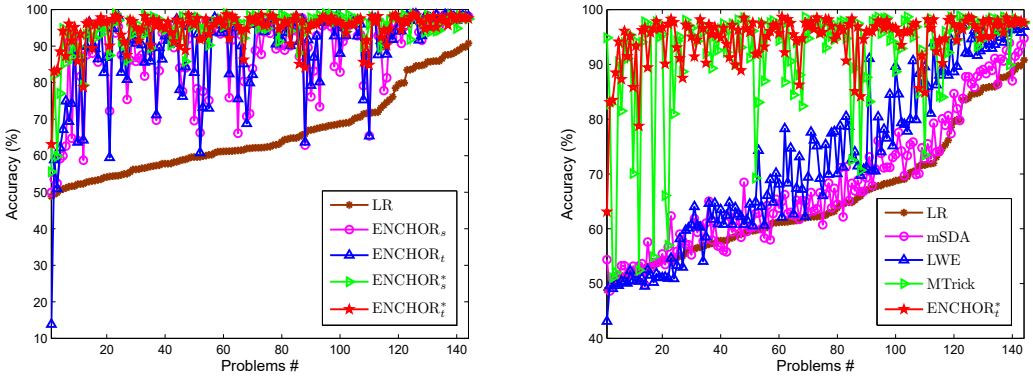
Combination	E_{cr}	E_r	MMD	KL	E_c	LR
rec vs. sci	94.77	95.10	90.69	90.69	78.93	65.57
rec vs. talk	94.43	95.47	92.58	91.81	86.35	72.46
sci vs. talk	91.88	91.85	88.65	86.63	76.54	70.65

Note that, the result of E_{cr} is the same as the one of ANCHOR$_t^*$.

4.4 Results on Classif cation Problems with Unbalanced Classes

In this section, we want to show that our algorithm can also perform well under the unbalanced class condition. We intentionally change the ratio of positive and negative instances in the target domain of 144 problems on *rec vs. sci*, and the ratios includes 2:1, 2.5:1, 3.3:1, 5:1 and 10:1. The strategy E_r can achieve good results when the classes are balanced in Section 4.3.2, therefore we compare ENCHOR$_t^*$ with MTrick, E_r and LR here. The detailed results are shown in Table 3. Sometimes, the class ratio maybe unknown and can not be estimated, so in this case we simply set $\boldsymbol{p} = (\frac{1}{c}, \cdots, \frac{1}{c})$, i.e., the uniform class distribution. We denote

[3]http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/code/index.html



(a) Comparison among ENCHOR$^*_t$, ENCHOR$^*_s$, ENCHOR$_t$, ENCHOR$_s$ and LR

(b) Comparison among ENCHOR$^*_t$, MTrick, LWE, mSDA and LR

Figure 2: The Performance Comparison among ENCHOR, MTrick, LWE, mSDA and LR on *rec vs. sci*

Table 1: Average Results (%) on Three Combinations for the Comparison of ENCHOR, MTrick, LWE, mSDA and LR

Combination		Problems #	ENCHOR$^*_t$	ENCHOR$^*_s$	ENCHOR$_t$	ENCHOR$_s$	MTrick	LWE	mSDA	LR
rec vs. sci	< 90	31	90.84(0)	91.34(0)	75.16(2)	77.01(2)	77.61(0)	67.16(10)	63.01(5)	61.22
	≥ 90	113	95.85(0)	95.49(0)	91.47(0)	92.24(1)	96.19(0)	73.59(14)	69.22(20)	66.76
	Total		94.77(0)	94.59(0)	87.96(2)	88.96(3)	92.19(0)	72.20(24)	67.88(25)	65.57
rec vs. talk	< 90	13	92.56(0)	95.88(0)	83.09(2)	84.32(2)	86.89(0)	83.86(0)	77.13(0)	70.75
	≥ 90	131	94.61(0)	95.64(0)	92.93(3)	93.66(3)	96.23(0)	78.11(20)	77.86(0)	72.63
	Total		94.43(0)	95.66(0)	92.04(5)	92.82(5)	95.39(0)	78.63(20)	77.79(0)	72.46
sci vs. talk	< 90	27	87.45(0)	84.55(0)	77.17(2)	75.55(4)	80.99(2)	69.42(4)	65.81(2)	62.15
	≥ 90	117	92.90(0)	93.55(0)	89.68(2)	90.38(2)	94.61(0)	82.78(1)	75.75(13)	72.61
	Total		91.88(0)	91.86(0)	87.33(4)	87.60(6)	92.06(2)	80.27(5)	73.89(15)	70.65

Table 3: Average Results (%) on *rec vs. sci* for Unbalanced Class Classif cation

Ratio		Problems #	ENCHOR$^{*(u)}_t$	$E^{(u)}_r$	ENCHOR$^*_t$	E_r	MTrick	LR
2:1	< 90	51	87.13	80.32	89.60	83.20	68.50	57.86
	≥ 90	93	93.72	87.56	94.58	89.34	95.48	66.60
	Total		91.38	85.00	92.81	87.16	85.92	63.51
2.5:1	< 90	71	85.38	77.82	89.30	83.87	68.21	59.59
	≥ 90	73	92.04	85.10	94.11	88.49	95.51	66.09
	Total		88.76	81.51	91.74	86.21	82.05	62.89
3.3:1	< 90	91	77.77	71.59	86.80	81.38	64.10	59.09
	≥ 90	53	88.11	81.04	93.35	88.91	95.19	67.50
	Total		81.58	75.07	89.21	84.15	75.54	62.19
5:1	< 90	114	68.34	66.77	83.92	81.37	61.31	58.13
	≥ 90	30	82.53	75.84	91.25	88.22	93.26	73.88
	Total		71.29	68.66	85.45	82.79	67.97	61.41
10:1	< 90	131	61.29	59.22	78.40	77.38	54.13	58.60
	≥ 90	13	75.64	68.94	88.53	88.98	93.89	79.12
	Total		62.58	60.10	79.32	78.43	57.72	60.45

our algorithms and E_r as ENCHOR$^{*(u)}_t$ and $E^{(u)}_r$ in Table 3 when the true class distribution of target domain is unknown.

From these results, we f nd that 1) ENCHOR$^*_t$ outperforms all the baselines under different ratios for unbalanced class classif cation; 2) with the increasing values of ratio, the accuracies from all algorithms decrease and the number of problems whose accuracies from MTrick lower than 90% increases; 3) ENCHOR$^{*(u)}_t$ also performs better than all baselines, which indicates that all the baselines are signif cantly inf uenced by the class imbalance. In summary, these results again validate the effectiveness of our algorithm.

5. RELATED WORKS

Transfer learning has provoked suff cient attention in recent years, and there are many algorithms are proposed following different pipelines. Here, we group them into two types, namely learning from single source domain and learning multiple source domains for transfer learning.

For the works of learning from single source domain, Dai et al. [17] proposed a novel transfer-learning algorithm based on an EM-based Naive Bayes classif ers, which f rst estimated the initial probabilities under a distribution of source domain data and then used an EM algorithm to revise the model for the distribution of target domain data. Si et al. [18] proposed a transfer subspace learning framework, which includes two items. The f rst one is the general subspace learning framework, while the second one is to minimize the Bregman divergence between the distribution of source and target domains in the selected subspace. There are also several works based on matrix factorization. Zhuang et al. [9]

f rst argued the associations between word clusters and document classes maybe stable across different domains, and then proposed a new transfer learning algorithm based on non-negative matrix factorization. Pan et al.[19] proposed transfer component analysis to learn some transfer components across domains in a reproducing kernel Hilbert space using maximum mean discrepancy. However, these methods do not make full use of the neighbourhood structures of source and target domains. It is excited that the proposed framework can be easily adapted to these matrix factorization based method to improve performance.

Along the second pipeline, Duan et al. [20] proposed to learn a robust decision function for the target domain data by leveraging a set of auxiliary/source classif ers from multiple source domains. Mansour et al. [21] presented a theoretical analysis of the problem of domain adaptation with multiple sources, and remarked that for any f xed target function, there exists a distribution weighted combining rule that has a loss of at most ϵ with respect to any target mixture of the source distributions. Dredze at al. [8] developed a new multi-domain online learning framework based on parameter combination from multiple classif ers to a new target domain, based on an online learning algorithm for linear classif ers that incorporates conf dence about each parameter into the update. Ge et al. [22] proposed a novel two-phase framework to effectively transfer knowledge from multiple sources. However, all the above algorithms need multiple source domains. Our method focuses on one source domains, and through elaborately selecting the anchors, the f nal prediction is the weighted sum of outputs of anchor adapters.

6. CONCLUSIONS

In this paper, we argue that the transfer learning algorithms performing on the original source and target domains may not obtain stable performance due to the large gap of distribution difference. Then along this line, we propose a transfer learning framework, called ensemble of anchor adapters, and implement it based on non-negative matrix tri-factorization. Specif cally, the anchors are f rst selected from source domain or target domain, and then only the similar instances from both domains are chosen to form anchor-adapted matrices, between which the distribution difference can be reduced. Moreover, we design an entropy based strategy to select high-quality anchors, leading to the outstanding and robust results. Finally, extensive experiments on text classif cation demonstrate the effectiveness and robustness of the proposed method.

7. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 9154610306, 61573335, 61473273), Guangdong provincial science and technology plan projects (No. 2015 B010109005).

8. REFERENCES

[1] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, pages 1345–1359, 2010.

[2] Xuejun Liao, Ya Xue, and Lawrence Carin. Logistic regression with an auxiliary data source. In *Proceedings of the 22nd ICML*, pages 505–512, 2005.

[3] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th ICML*, pages 193–200, 2007.

[4] Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th ICML*, 2012.

[5] Ping Luo, Fuzhen Zhuang, Hui Xiong, Yuhong Xiong, and Qing He. Transfer learning from multiple source domains via consensus regularization. In *Proceedings of the 17th ACM CIKM*, pages 103–112, 2008.

[6] Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *IEEE CVPR*, pages 1338–1345, 2012.

[7] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD*, 2008.

[8] Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by conf dence-weighted parameter combination. *Machine Learning*, pages 123–149, 2010.

[9] Fuzhen Zhuang, Ping Luo, Hui Xiong, Qing He, Yuhong Xiong, and Zhongzhi Shi. Exploiting associations between word clusters and document classes for cross-domain text categorization. *Statistical Analysis and Data Mining*, pages 100–114, 2011.

[10] Solomon Kullback. Letter to the editor: the kullback-leibler distance. *AMERICAN STATISTICIAN*, 1987.

[11] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, pages e49–e57, 2006.

[12] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classif cation for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD*, pages 210–219, 2007.

[13] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, pages 677–682, 2008.

[14] Fuzhen Zhuang, Ping Luo, Hui Xiong, Yuhong Xiong, Qing He, and Zhongzhi Shi. Cross-domain learning from multiple sources: a consensus regularization perspective. *IEEE TKDE*, pages 1664–1678, 2010.

[15] David Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. Wiley, New York, 2000.

[16] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, pages 177–196, 2001.

[17] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classif ers for text classif cation. In *AAAI*, pages 540–545, 2007.

[18] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE TKDE*, pages 929–942, 2010.

[19] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, Qiang Yang, et al. Domain adaptation via transfer component analysis. *IEEE TNN*, pages 199–210, 2011.

[20] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classif ers. In *Proceedings of the 26th ICML*, pages 289–296, 2009.

[21] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, pages 1041–1048, 2009.

[22] Liang Ge, Jing Gao, Hung Ngo, Kang Li, and Aidong Zhang. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, pages 254–271, 2014.