Actively Transfer Domain Knowledge

Xiaoxiao Shi[1], Wei Fan[2], and Jiangtao Ren[1,⋆]

[1] Department of Computer Science
Sun Yat-sen University, Guangzhou, China
{isshxx,issrjt}@mail.sysu.edu.cn
[2] IBM T.J.Watson Research, USA
weifan@us.ibm.com

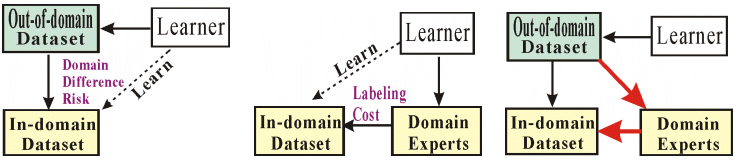
Abstract. When labeled examples are not readily available, active learning and transfer learning are separate efforts to obtain labeled examples for inductive learning. Active learning asks domain experts to label a small set of examples, but there is a cost incurred for each answer. While transfer learning could borrow labeled examples from a different domain without incurring any labeling cost, there is no guarantee that the transferred examples will actually help improve the learning accuracy. To solve both problems, we propose a framework to actively transfer the knowledge across domains, and the key intuition is to use the knowledge transferred from other domain as often as possible to help learn the current domain, and query experts only when necessary. To do so, labeled examples from the other domain (out-of-domain) are examined on the basis of their likelihood to correctly label the examples of the current domain (in-domain). When this likelihood is low, these out-of-domain examples will not be used to label the in-domain example, but domain experts are consulted to provide class label. We derive a sampling error bound and a querying bound to demonstrate that the proposed method can effectively mitigate risk of domain difference by transferring domain knowledge only when they are useful, and query domain experts only when necessary. Experimental studies have employed synthetic datasets and two types of real world datasets, including remote sensing and text classification problems. The proposed method is compared with previously proposed transfer learning and active learning methods. Across all comparisons, the proposed approach can evidently outperform the transfer learning model in classification accuracy given different out-of-domain datasets. For example, upon the remote sensing dataset, the proposed approach achieves an accuracy around 94.5%, while the comparable transfer learning model drops to less than 89% in most cases. The software and datasets are available from the authors.

1 Introduction

Supervised learning methods require sufficient labeled examples in order to construct accurate models. However, in real world applications, one may easily

⋆ The author is supported by the National Natural Science Foundation of China under Grant No. 60703110.

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pp. 342–357, 2008.



(a) Transfer learning (b) Active learning (c) Actively transfer

Fig. 1. Different models to resolve label deficiency

encounter those situations in which labeled examples are deficient, such as data streams, biological sequence annotation or web searching, etc. To alleviate this problem, two separate approaches, transfer learning and active learning, have been proposed and studied. Transfer learning mainly focuses on how to utilize the data from a related domain called out-of-domain, to help learn the current domain called in-domain, as depicted in Fig 1(a). It can be quite effective when the out-of-domain dataset is very similar to the in-domain dataset. As a different solution, active learning does not gain knowledge from other domains, but mainly focuses on selecting a small set of essential in-domain instances for which it requests labels from the domain experts, as depicted in Fig 1(b). However, both transfer learning and active learning have some practical constraints. For transfer learning, if the knowledge from the out-of-domain is essentially different from the in-domain, the learning accuracy might be reduced, and this is called the "domain difference risk". For active learning, the obvious issue is the cost associated with the answer from domain experts.

Our Method. To mitigate domain difference risk and reduce labeling cost, we propose a framework that can actively transfer the knowledge from out-of-domain to in-domain, as depicted in Fig 1(c). Intuitively, in daily life, we normally first try to use our related knowledge in learning, but if the related knowledge is unable to guide, we turn to teachers. For example, when learning a foreign language, one normally associates it with the mother tongue. This transfer is easy between, for example, Spanish and Portuguese. But it is not so obvious between Chinese and English. In this situation, one normally pays a teacher instead of picking up himself. The proposed framework is based on these intuitions. We first select an instance that is supposed to be essential to construct an inductive model from the new or in-domain dataset, and then a transfer classifier, trained with labeled data from in-domain and out-of-domain dataset, is used to predict this unlabeled in-domain example. According to defined transfer confidence measure, this instance is either directly labeled by the transfer classifier or labeled by the domain experts if needed. In order to guarantee the performance when "importing" out-of-domain knowledge, the proposed transfer classifier is bounded to be no worse than an instance-based ensemble method in error rate (Section 3 and Theorem 1).

Contributions. The most important contributions of the proposed approach can be summarized as follows:

Table 1. Symbol definition

Symbol	Definition
$\mathcal{U}$	Unlabeled in-domain dataset
$\mathcal{L}$	Labeled in-domain dataset
$\mathcal{O}$	Labeled out-of-domain dataset
$\mathbf{M}^{\mathcal{L}}$	The in-domain classifier that is trained on $\mathcal{L}$
$\mathbf{M}^{\mathcal{O}}$	The out-of-domain classifier that is trained on $\mathcal{O}$
$\mathbf{T}$	The transfer classifier (Fig 3 and Equ. 1)
$\mathbb{F}(\mathbf{x})$	A decision function (Equ. 3 and Equ. 4)
ℓ	The actively transfer learner (Algorithm 1)
	The following are some symbols only used in Fig 3
$\mathcal{L}^+$	$\mathcal{L}^+ = \{\mathbf{x} \vert \mathbf{x} \in L \wedge \mathbf{M}^{\mathcal{O}}(\mathbf{x}) = \text{“}+''\}$
$\mathcal{L}^-$	$\mathcal{L}^- = \{\mathbf{x} \vert \mathbf{x} \in L \wedge \mathbf{M}^{\mathcal{O}}(\mathbf{x}) = \text{“}-''\}$
$\mathbf{M}^{\mathcal{L}^+}$	The classifier that is trained on $\mathcal{L}^+$
$\mathbf{M}^{\mathcal{L}^-}$	The classifier that is trained on $\mathcal{L}^-$

– We propose a framework that can transfer the knowledge across domains actively. We derive the bounds in Theorem 2 and Theorem 3 to prove that the proposed framework not only can mitigate the domain difference risk by transferring out-of-domain knowledge only when they are useful, but also reduce labeling cost by querying fewer examples labeled by experts as compared with traditional active learners.
– We also propose a transfer classifier whose error is bounded.
– Most of the previous active learners can be directly adopted in the proposed framework without changing their original preferences and strategies to select essential examples.

2 Actively Transfer

The main flow of proposed approach AcTraK (**Ac**tively **Tra**nsfer **K**nowledge) is summarized in Fig 2 and Algorithm 1, and the most important symbols are in Table 1. The key idea is to use the out-of-domain data to predict in-domain data as often as possible. But when the prediction confidence is too low, in-domain experts are consulted to provide the label. To do so, the algorithm first applies a traditional active learner to select a critical instance $\mathbf{x}$ from the in-domain dataset, then a transfer classifier is trained and used to classify this selected example. According to the prediction confidence of the transfer classifier, the algorithm decides how to label the instance, either using the predicted label given by the transfer classifier or asking domain experts to label. Then, the process is iteratively performed to select and label important examples. Shown as Fig 2, the essential elements of the proposed algorithm are the “transfer classifier” and the “decision function”, as described below.

Transfer Classifier. Given an unlabeled in-domain dataset $\mathcal{U}$, a small set of labeled in-domain examples $\mathcal{L}$, as well as a labeled out-of-domain dataset $\mathcal{O}$, a transfer classifier is constructed from both $\mathcal{O}$ and $\mathcal{L}$ to classify unlabeled examples in $\mathcal{U}$. In previous work on transfer learning, out-of-domain dataset $\mathcal{O}$ is assumed to share similar distributions with in-domain dataset $\mathcal{U} \cup \mathcal{L}$ ([3][16]).

Input: Unlabeled in-domain dataset: $\mathcal{U}$;
Labeled in-domain dataset: $\mathcal{L}$;
Labeled out-of-domain dataset: $\mathcal{O}$;
Maximum number of examples labeled by experts: N.
Output: The actively transfer learner: ℓ

1 Initial the number of examples that have been labeled by experts: $n \leftarrow 0$;
2 **repeat**
3 $\quad$ $\mathbf{x} \leftarrow$ select an instance from $\mathcal{U}$ by a traditional active learner;
4 $\quad$ Train the transfer classifier $\mathbf{T}$ (Fig 3);
5 $\quad$ Predict $\mathbf{x}$ by $\mathbf{T}(\mathbf{x})$ (Fig 3 and Equ. 1);
6 $\quad$ Calculate the decision function $\mathbb{F}(\mathbf{x})$ (Details in Equ. 3 and Equ. 4);
7 $\quad$ **if** $\mathbb{F}(\mathbf{x}) = 0$ **then**
8 $\quad\quad$ y $\leftarrow$ label by $\mathbf{T}(\mathbf{x})$;
9 $\quad$ **else**
10 $\quad\quad$ y $\leftarrow$ label by the experts;
11 $\quad\quad$ $n \leftarrow n + 1$;
12 $\quad$ **end**
13 $\quad$ $\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathbf{x}, y)\}$;
14 **until** $n \geq N$
15 Train the learner ℓ with $\mathcal{L}$
16 Return the learner ℓ.

Algorithm 1. Framework

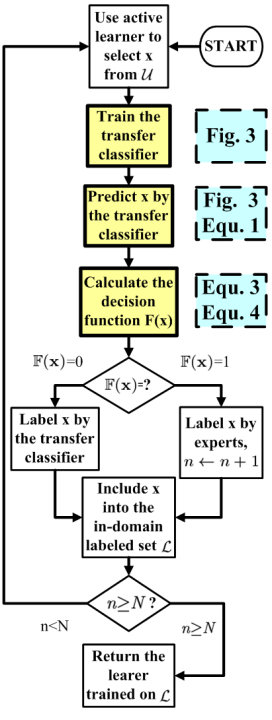


Fig. 2. Algorithm flow

Thus, exploring the similarities and exploiting them is expected to improve accuracy. However, it is unclear on how to formally determine whether the out-of-domain dataset shares sufficient similarity with the in-domain dataset, and how to guarantee transfer learning improves accuracy. Thus, in this paper, we propose a transfer learning model whose expected error is bounded.

Intuitively, if one uses the knowledge in out-of-domain dataset $\mathcal{O}$ to make prediction for an in-domain example $\mathbf{x}$, and then double check the predicted label with an in-domain classifier to see if the prediction is the same, it is more likely that the predicted label is correct. Before discussing in detail, we define some common notations. Let $\mathbf{M}^{\mathcal{O}}$ denote the out-of-domain classifier trained on the out-of-domain dataset $\mathcal{O}$. Also, let $\mathcal{L}_t$ denote a set of labeled data from in-domain, but they are labeled as y_t by the out-of-domain classifier $\mathbf{M}^{\mathcal{O}}$ (y_t is the label of the tth class). Formally, $\mathcal{L}_t = \{\mathbf{x} | \mathbf{x} \in \mathcal{L} \wedge \mathbf{M}^{\mathcal{O}}(\mathbf{x}) = y_t\}$. Note that the true labels of examples in $\mathcal{L}_t$ are not necessarily y_t, but they just happen to be labeled as class y_t by the out-of-domain classifier. We illustrate the transfer classifier for a binary-class problem in Fig 3. There are two classes, "+" and "-", and $\mathcal{L}^+ = \{\mathbf{x} | \mathbf{x} \in \mathcal{L} \wedge \mathbf{M}^{\mathcal{O}}(\mathbf{x}) = \text{"+"}\}$, and $\mathcal{L}^- = \{\mathbf{x} | \mathbf{x} \in \mathcal{L} \wedge \mathbf{M}^{\mathcal{O}}(\mathbf{x}) = \text{"-"}\}$. The transfer classifier $\mathbf{T}(\mathbf{x})$ executes the following steps to label an instance $\mathbf{x}$:

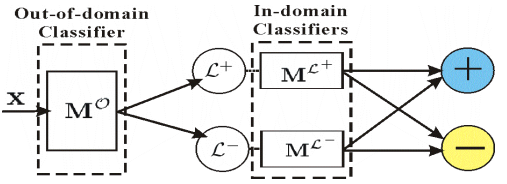


Fig. 3. Transfer classifier in Algorithm 1

1. Classify $\mathbf{x}$ with the out-of-domain classifier $\mathbf{M}^{\mathcal{O}}$ to obtain $P(\mathcal{L}^+|\mathbf{x}, \mathbf{M}^{\mathcal{O}})$ and $P(\mathcal{L}^-|\mathbf{x}, \mathbf{M}^{\mathcal{O}})$.
2. Classify $\mathbf{x}$ with the in-domain classifiers $\mathbf{M}^{\mathcal{L}+}$ and $\mathbf{M}^{\mathcal{L}-}$ to obtain $P(+|\mathbf{x}, \mathbf{M}^{\mathcal{L}^+})$ and $P(+|\mathbf{x}, \mathbf{M}^{\mathcal{L}^-})$.
3. The transfer probability for $\mathbf{x}$ being "+" is
   $P_T(+|\mathbf{x})$
   $= P(\mathcal{L}^+|\mathbf{x}, \mathbf{M}^{\mathcal{O}}) \times P(+|\mathbf{x}, \mathbf{M}^{\mathcal{L}^+}) + P(\mathcal{L}^-|\mathbf{x}, \mathbf{M}^{\mathcal{O}}) \times P(+|\mathbf{x}, \mathbf{M}^{\mathcal{L}^-})$

Omitting the explicit dependency on models the above formula can be simplified as:

$$P_T(+|\mathbf{x}) = P(+|\mathcal{L}^+, \mathbf{x}) \times P(\mathcal{L}^+|\mathbf{x}) + P(+|\mathcal{L}^-, \mathbf{x}) \times P(\mathcal{L}^-|\mathbf{x}) \tag{1}$$

Under 0-1 loss, when $P_T(+|\mathbf{x}) > 0.5$, $\mathbf{x}$ is classified as "+". This transfer classifier just described has the following important property.

Theorem 1. *Let ε_1 and ε_2 denote the expected error of the in-domain classifiers $\mathbf{M}^{\mathcal{L}+}$ and $\mathbf{M}^{\mathcal{L}-}$ respectively, and let ε_t denote the expected error of the transfer classifier $\mathbf{T}(\mathbf{x})$. Then,*

$$\min(\varepsilon_1, \varepsilon_2) \leq \varepsilon_t \leq \frac{1}{2}(\varepsilon_1 + \varepsilon_2) \tag{2}$$

Proof. $\forall \mathbf{x} \in \mathcal{U}$, we consider the situations in which the transfer classifier $\mathbf{T}(\mathbf{x})$ assigns it the wrong label. Let the true label of $\mathbf{x}$ be "+". Further assume that "+" examples are more likely classified into $\mathcal{L}^+$ or $P(\mathcal{L}^+|\mathbf{x}) \geq P(\mathcal{L}^-|\mathbf{x})$. Thus, the probability that $\mathbf{x}$ is mistakenly labeled as "-" is:
$\varepsilon_t(\mathbf{x}) = P(-|\mathbf{x})$
$= P(\mathcal{L}^-|\mathbf{x}) \times P(-|\mathcal{L}^-, \mathbf{x}) + (1 - P(\mathcal{L}^-|\mathbf{x})) \times P(-|\mathcal{L}^+, \mathbf{x})$
$= P(-|\mathcal{L}^+, \mathbf{x}) + P(\mathcal{L}^-|\mathbf{x}) \times (P(-|\mathcal{L}^-, \mathbf{x}) - P(-|\mathcal{L}^+, \mathbf{x}))$

Since $P(-|\mathcal{L}^-, \mathbf{x}) = \frac{P(\mathbf{x}|\mathcal{L}^-,-)P(\mathcal{L}^-,-)}{P(\mathcal{L}^-|\mathbf{x})P(\mathbf{x})} > \frac{P(\mathbf{x}|\mathcal{L}^+,-)P(\mathcal{L}^+,-)}{P(\mathcal{L}^+|\mathbf{x})P(\mathbf{x})} = P(-|\mathcal{L}^+, \mathbf{x})$, then $P(-|\mathbf{x}) \geq P(-|\mathcal{L}^+, \mathbf{x}) = \min(P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^+}), P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^-}))$. In addition, since $P(\mathcal{L}^+|\mathbf{x}) \geq P(\mathcal{L}^-|\mathbf{x})$, we have $0 \leq P(\mathcal{L}^-|\mathbf{x}) \leq \frac{1}{2}$. Then, $P(-|\mathbf{x}) \leq P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^+}) + \frac{1}{2}(P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^-}) - P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^+})) = \frac{1}{2}(P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^+}) + P(-|\mathbf{x}, \mathbf{M}^{\mathcal{L}^-}))$. Thus, we have $\min(\varepsilon_1, \varepsilon_2) \leq \varepsilon_t \leq \frac{1}{2}(\varepsilon_1 + \varepsilon_2)$. □

Hence, Theorem 1 indicates that if the out-of-domain knowledge is similar to the current domain, or $P(\mathcal{L}^-|\mathbf{x})$ is small, the model obtains the expected error close to $\varepsilon_t = \min(\varepsilon_1, \varepsilon_2)$. When the out-of-domain knowledge is different, the expected error is bounded by $\frac{1}{2}(\varepsilon_1 + \varepsilon_2)$. In other words, in the worst case, the performance of the transfer classifier is no worse than the average performances of the two in-domain classifiers $\mathbf{M}^{\mathcal{L}^+}$ and $\mathbf{M}^{\mathcal{L}^-}$.

Decision Function. After the transfer classifier $\mathbf{T}(\mathbf{x})$ predicts the selected example $\mathbf{x}$, a decision function $\mathbb{F}(\mathbf{x})$ is calculated and further decides how to label the example. In the following situations, one should query the experts for the class label in case of mislabeling.

- When the transfer classifier assigns $\mathbf{x}$ with a class label that is different from that given by an in-domain classifier.
- When the transfer classifier's classification is low in confidence.
- When the size of the labeled in-domain dataset $\mathcal{L}$ is very small.

Recall that $\mathbf{M}^{\mathcal{L}}$ is the in-domain classifier trained on labeled in-domain dataset $\mathcal{L}$. According to the above considerations, we design a "querying indicator" function $\theta(\mathbf{x})$ to reflect the necessity to query experts.

$$\begin{aligned} \theta(\mathbf{x}) &= \Big(1 + \alpha(\mathbf{x})\Big)^{-1} \\ \alpha(\mathbf{x}) &= \Big(1 - [\![\mathbf{M}^{\mathcal{L}}(\mathbf{x}) \neq \mathbf{T}(\mathbf{x})]\!]\Big) \cdot \mathbf{P}_T(\mathbf{T}(\mathbf{x}) = y|\mathbf{x}) \cdot \exp(-\frac{1}{|\mathcal{L}|}) \end{aligned} \tag{3}$$

where $[\![\pi]\!] = 1$ if π is true. And $\mathbf{P}_T(\mathbf{T}(\mathbf{x}) = y|\mathbf{x})$ is the transfer probability given by the transfer classifier $\mathbf{T}(\mathbf{x})$. Thus, $0 \leq \alpha(\mathbf{x}) \leq 1$ and it reflects the confidence that the transfer classifier has correctly labeled the example $\mathbf{x}$: it increases with the transfer probability $\mathbf{P}_T(\mathbf{T}(\mathbf{x}) = y|\mathbf{x})$, and $\alpha(\mathbf{x}) = 0$ if the two classifiers $\mathbf{M}^{\mathcal{L}}(\mathbf{x})$ and $\mathbf{T}(\mathbf{x})$ have assigned different labels to $\mathbf{x}$. Hence, the larger of $\alpha(\mathbf{x})$, the less we need to query the experts to label the example. Formally, the "querying indicator" function $\theta(\mathbf{x})$ requires $\theta(\mathbf{x}) \propto \alpha(\mathbf{x})^{-1}$. Moreover, because mislabeling of the first few selected examples can exert significant negative effect on accuracy, we further set $\theta(\mathbf{x}) = \Big(1+\alpha(\mathbf{x})\Big)^{-1}$ so as to guarantee the possibility (necessity) to query experts is greater than 50%. In other words, labeling by the experts is the priority and we trust the label given by the transfer classifier only when its confidence reflected by $\alpha(\mathbf{x})$ is very high. Thus, the proposed algorithm asks the experts to label the example with probability $\theta(\mathbf{x})$. Accordingly, with the value of $\theta(\mathbf{x})$, we randomly generate a real number R within 0 to 1, and then the decision function $\mathbb{F}(\mathbf{x})$ is defined as

$$\mathbb{F}(\mathbf{x}) = \begin{cases} 0 & if \;\; R > \theta(\mathbf{x}) \\ 1 & otherwise \end{cases} \tag{4}$$

According to Eq. 4, if the randomly selected real number $R > \theta(\mathbf{x})$, $\mathbb{F}(\mathbf{x}) = 0$, and it means Algorithm 1 labels the example by the transfer classifier; otherwise, the example is labeled by the domain experts. In other words, AcTraK labels the example $\mathbf{x}$ by transfer classifier with probability $1 - \theta(\mathbf{x})$.

2.1 Formal Analysis of AcTraK

We have proposed the approach AcTraK to transfer knowledge across domains actively. Now, we formally derive its sampling error bound to demonstrate its ability to mitigate domain difference risk, which guarantees that out-of-domain examples are transferred to label in-domain data only when they are useful. Additionally, we prove its querying bound to validate the claim that AcTraK can reduce labeling cost by querying fewer examples labeled by experts by any based level active learner incorporated into the framework.

Theorem 2. *In the algorithm* **AcTraK** *(Algorithm 1), let ε_t denote the expected error of the transfer classifier, and let N denote the maximum number of examples labeled by experts, then the sampling error ε for* **AcTraK** *satisfies*

$$\varepsilon \leq \frac{\varepsilon_t^2}{1 + (1 - \varepsilon_t) \times \exp(-|N|^{-1})} \tag{5}$$

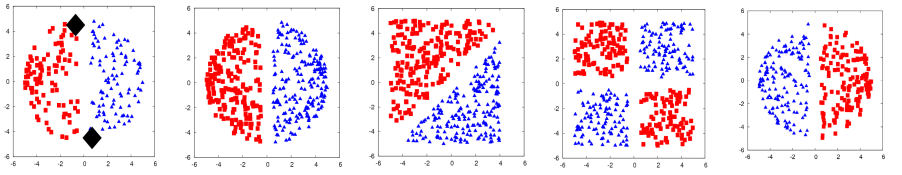
Proof. The proof for Theorem 2 is straightforward. According to the transfer classifier $\mathbf{T}(\mathbf{x})$ and the decision function $\mathbb{F}(\mathbf{x})$ described above, **AcTraK** makes wrong decision only when both the transfer classifier $\mathbf{T}(\mathbf{x})$ and the in-domain classifier $\mathbf{M}^{\mathcal{L}}$ agree on the wrong label. And in this case, **AcTraK** has probability $1 - \theta(\mathbf{x})$ to trust the classification result given by $\mathbf{T}(\mathbf{x})$, where $\theta(\mathbf{x})$ is defined in Eq. 3. Thus, the sampling error for **AcTraK** can be written as $\varepsilon \leq (\varepsilon_t)^2(1-\theta(\mathbf{x}))$. Moreover, in this situation, $\theta(\mathbf{x}) = \frac{1}{1+(1-\varepsilon_t)e^{-\frac{1}{|\mathcal{L}|}}} \geq \frac{1}{1+(1-\varepsilon_t)e^{-\frac{1}{N}}}$. Thus, $\varepsilon \leq \varepsilon_t^2(1-\theta(\mathbf{x})) \leq \frac{\varepsilon_t^2 \times (1-\varepsilon_t) \times \exp(-|N|^{-1})}{1+(1-\varepsilon_t)\times\exp(-|N|^{-1})} \leq \frac{\varepsilon_t^2}{1+(1-\varepsilon_t)\times\exp(-|N|^{-1})}$. □

Theorem 3. *In the algorithm* **AcTraK** *(Algorithm 1), let ε_t and ε_i denote the expected error of the transfer classifier and in-domain classifier respectively. And let $\alpha = \varepsilon_t + \varepsilon_i$. Then for an in-domain instance, the probability that* **AcTraK** *queries the label from the experts (with cost) satisfies:*

$$P[Query] \leq \alpha + \frac{1 - \alpha}{1 + (1 - \varepsilon_t) \times \exp(-\frac{1}{|N|})} \tag{6}$$

Proof. According to the labeling-decision function $\mathbb{F}(\mathbf{x})$, **AcTraK** will query the experts to label the instance when $\mathbf{T}(\mathbf{x})$ and $\mathbf{M}^{\mathcal{L}}$ hold different predictions on the classification result. Even when the two classifiers agree on the result, it still has probability $\theta(\mathbf{x})$ to query the experts. Thus, $P[Query] = \varepsilon_i(1 - \varepsilon_t) + \varepsilon_t(1 - \varepsilon_i) + [\varepsilon_t\varepsilon_i + (1 - \varepsilon_t)(1 - \varepsilon_i)]\theta(\mathbf{x}) = \theta(\mathbf{x}) + (\varepsilon_t + \varepsilon_i - 2\varepsilon_t\varepsilon_i)(1 - \theta(\mathbf{x})) \leq \alpha + (1 - \alpha)\theta(\mathbf{x}) \leq \alpha + \frac{1-\alpha}{1+(1-\varepsilon_t)\times\exp(-\frac{1}{|N|})}$. □

From Theorem 2, we can find that the sampling error of the proposed approach AcTraK is bounded by $O(\frac{\varepsilon_t^2}{1-\varepsilon_t})$, where ε_t is the expected error of the transfer classifier, and ε_t is also bounded according to Theorem 1. Thus, the proposed



(a) $\mathcal{U}$ (b) $\mathcal{O}_1$ (c) $\mathcal{O}_2$ (d) $\mathcal{O}_3$ (e) $\mathcal{O}_4$

Fig. 4. Synthetic dataset

method AcTraK can effectively bound the sampling error to reduce the domain difference risk. In addition, we derive Theorem 3 to understand why AcTraK can query fewer examples labeled by experts as compared with traditional active learners. From Theorem 3, we can see that the querying probability of AcTraK is bounded, and the querying bound decreases with the decreasing ε_t. In other words, the more accurate of the transfer classifier, the less likely will AcTraK query the experts to label the instance. Hence, one can perceive that AcTraK can actively decide how to gain its knowledge.

3 Experiments

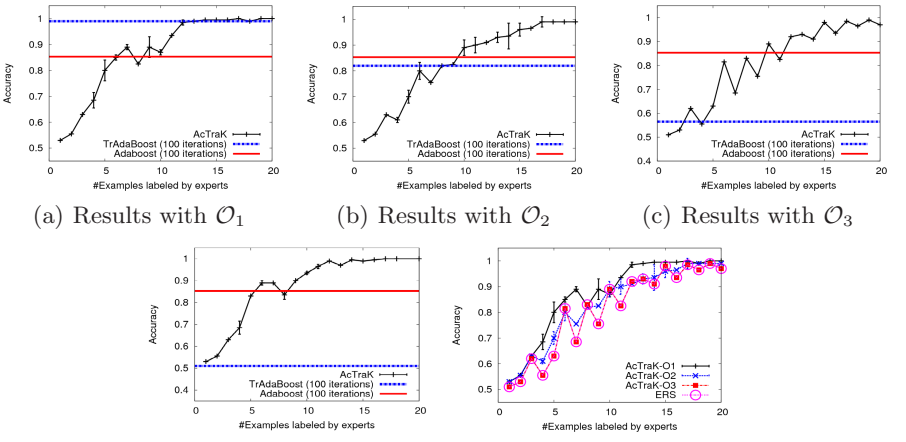
We first design synthetic datasets to demonstrate how AcTraK mitigates the domain difference risk that can make transfer learning fail, and then study how out-of-domain knowledge can help AcTraK to query fewer examples labeled by experts, as compared with traditional active learner. Then several transfer learning problems composed from remote sensing and text classification datasets are used for evaluation. We use SVM as the basic learners, and logistic regression to simulate the classification probabilities. Furthermore, for active learner employed in AcTraK, we adopt ERS (Error Reduction Sampling method [11]). AcTraK is compared with both a transfer learning model TrAdaBoost ([4]) and the active learning model ERS ([11]). These are some of the most obvious choices, commonly adopted in the research community.

3.1 Synthetic Datasets

We design synthetic datasets to empirically address the following questions:

1. Domain difference risk: can AcTraK overcome domain difference if the out-of-domain knowledge is significantly different from the current domain?
2. Number of examples labeled by experts: do experts label fewer examples in AcTraK under the help of out-of-domain knowledge?

We generate five two-dimensional datasets shown in Fig 4 (electronic copy of this paper contains color figures). Fig 4(a) draws the in-domain dataset $\mathcal{U}$, which has two labeled examples highlighted by "♦". Importantly, four out-of-domain datasets with different distributions are plotted in Fig 4(b)∼Fig 4(e).



(a) Results with $\mathcal{O}_1$ (b) Results with $\mathcal{O}_2$ (c) Results with $\mathcal{O}_3$

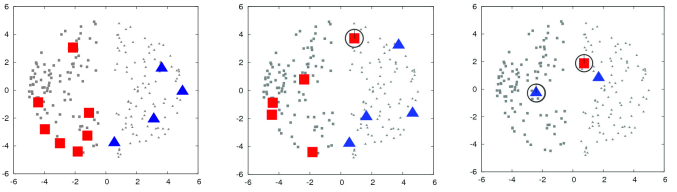
(d) Results with $\mathcal{O}_4$ (e) AcTraK VS. ERS

Fig. 5. Results on synthetic dataset

Note: To resolve label deficiency, TrAdaBoost does not query experts but passively trains with the in-domain and all the labeled out-of-domain data. Thus, the its learning curve is straight line.

Fig 4(b) presents a dataset $\mathcal{O}_1$, similarly distributed as the in-domain dataset $\mathcal{U}$. But the dataset $\mathcal{O}_2$ is clearly different from $\mathcal{U}$ though they may still share some similarity. Fig 4(d) plots an "XOR" dataset $\mathcal{O}_3$. In addition, we include the dataset $\mathcal{O}_4$ depicted as Fig 4(e), which has a similar "shape" but totally reversed class labels with the in-domain dataset $\mathcal{U}$. Thus, four experiments are conduced by using the same in-domain dataset $\mathcal{U}$ but different out-of-domain datasets $\mathcal{O}_1 \sim \mathcal{O}_4$. We vary the number of examples labeled by experts up to 50. Moreover, to reveal how domain difference affects transfer learning, we also run on AdaBoost, or boosting without transferring knowledge from other domains. Each experiment is repeated 10 times and average results are reported. For both TrAdaBoost and AdaBoost, the iteration is set to be 100. For the sake of clarity, we plot the most distinctive results in Fig 5.

Can AcTraK Overcome Domain Difference? Fig 5(a) to Fig 5(d) plot the performance comparison of AcTraK vs. TrAdaBoost as they are given the four out-of-domain datasets. The result given by AdaBoost is to compare with TrAdaBoost to study effect of domain difference. It is important to note that, to resolve label deficiency, TrAdaBoost does not query the experts but trains on in-domain and all labeled out-of-domain data for many iterations (100 in our experiment). Thus, its result is a straight line. From Fig 5, TrAdaBoost is effective when the out-of-domain dataset is $\mathcal{O}_1$ or Fig 4(b), which shares similar distribution with the in-domain dataset. In this case, TrAdaBoost obviously outperforms the original AdaBoost. However, when the out-of-domain dataset distributes differently from the in-domain dataset, the classification accuracy given by TrAdaBoost significantly drops: 0.82 when the out-of-domain dataset is $\mathcal{O}_2$; 0.57 when the



(a) With $\mathcal{O}_1$ (b) With $\mathcal{O}_2$ (c) With $\mathcal{O}_3$

Fig. 6. Examples (in $\mathcal{U}$) labeled by the transfer classifier

out-of-domain dataset is $\mathcal{O}_3$ and only 0.51 when with $\mathcal{O}_4$ (the last two results are just slightly better than random guessing). Importantly, these numbers are even worse than the original AdaBoost that achieves an accuracy of 0.85 without using the knowledge from $\mathcal{O}_2$, $\mathcal{O}_3$ or $\mathcal{O}_4$. It is obvious that the culprit is the domain differences from these datasets.

Importantly under these same challenging situations, from Fig 5(b) to Fig 5(d), this domain difference does not significantly affect the proposed algorithm AcTraK. The classification accuracies of AcTraK with different out-of-domain datasets are over 0.9 when the number of examples labeled by experts is 12, demonstrating its ability to overcome domain difference risk. It is interesting to notice that when the out-of-domain dataset is $\mathcal{O}_4$, AcTraK acts similar as that with $\mathcal{O}_1$. This is because that the transfer classifier described in Fig 3 is not sensitive to the actual "name" of the labels of the out-of-domain dataset. For example, if $\mathcal{L}^+$ in Fig 3 actually includes most examples with class label $-$, the term $P(-|\mathbf{x}, \mathcal{L}^+)$ will be likely large and make the final classification result more likely to be $-$. Thus, with respect to their similar structure, $\mathcal{O}_4$ is homogeneous with $\mathcal{O}_1$ to some extend. Hence, we do not consider $\mathcal{O}_4$ but $\mathcal{O}_3$ as the most different distributed dataset with the in-domain dataset $\mathcal{U}$ in this experiment. And owing to the limited space and the homogeneity of $\mathcal{O}_1$ and $\mathcal{O}_4$ to AcTraK, the result of $\mathcal{O}_4$ is omitted in the following of the paper.

Importantly, Fig 5 shows that, even with the dataset $\mathcal{O}_2$ and $\mathcal{O}_3$, AcTraK can ensure the accuracy. It is mainly due to the actively transfer strategy: it does not all depend on the out-of-domain dataset passively. To further reveal this active strategy in AcTraK, we plot the examples labeled by the transfer classifier in Fig 6. The examples mislabeled by the transfer classifier are marked with circles. Shown in Fig 6, when the out-of-domain dataset is $\mathcal{O}_1$, the most similar to the in-domain dataset, the transfer classifier help label more examples than those with $\mathcal{O}_2$ or $\mathcal{O}_3$. Especially when the out-of-domain dataset is $\mathcal{O}_3$, the transfer classifier help label only 3 examples and this is due to domain difference.

The sampling error bound of AcTraK under domain difference is derived in Theorem 2. We calculate these bounds and compare them with the actual sampling errors in Table 2. It is important to mention that the actual sampling error or sampling error bound discussed here is the labeling error for the selected examples, but not the accuracy results given in Fig 5, which is the accuracy on the whole in-domain dataset. To calculate the actual sampling error of AcTraK, for example, when the out-of-domain dataset is $\mathcal{O}_2$, there are a total of 9 examples

Table 2. Sampling error bound and querying bound on synthetic dataset

Datasets	Error of $\mathbf{T}(\mathbf{x})$	Sampling error	Sampling error bound	Querying rate	Querying bound
$\mathcal{O}_1$	0.00	0.000	0.000	71.43%	75.25%
$\mathcal{O}_2$	0.18	0.017	0.017	84.75%	85.45%
$\mathcal{O}_3$	0.34	0.037	0.070	94.34%	93.72%

labeled by the transfer classifier with one mislabeled, as depicted in Fig 6(b). Thus, the actual sampling error of AcTraK is $\frac{1}{50+9} = 0.017$, and we compare it with the bound calculated according to Theorem 2. The results are summarized in Table 2. The sampling error bound given in Theorem 2 is obviously tight for the synthetic dataset. Moreover, it is evident that AcTraK can effectively reduce sampling error. For instance, when the true error of the transfer classifier $\mathbf{T}(\mathbf{x})$ is 0.34 with the dataset $\mathcal{O}_3$, AcTraK bounds its sampling error as 0.07 and gets the actual error 0.04. Thus, it can be concluded that AcTraK can effectively resolve domain difference by bounding the sampling error shown as Theorem 2.

Do Experts Label Fewer Examples in AcTraK? In the proposed approach AcTraK, knowledge transferred from other domain is used to help label the examples. In other words, it saves the number of examples to ask the experts. Thus, compared with traditional active learner, AcTraK is expected to reduce the number of examples labeled by experts, thus to reduce labeling cost. We present Fig 5(e) to demonstrate this claim by comparing AcTraK with the traditional active learner ERS. It is important to note that the original ERS only works on the in-domain dataset. Thus, there is only one result plotted for ERS in Fig 5(e) but three for AcTraK with different out-of-domain datasets. From Fig 5(e), we can see that in most cases, "AcTraK-$\mathcal{O}_1$" and "AcTraK-$\mathcal{O}_2$" can evidently outperform ERS by reaching the same accuracy but with fewer examples labeled by experts. And this is because that some of the examples have been labeled by the transfer classifier under the help of the out-of-domain datasets. Additionally, the out-of-domain dataset $\mathcal{O}_1$ seems more helpful than $\mathcal{O}_2$ to AcTraK, due to the similar distribution between $\mathcal{O}_1$ and the in-domain dataset.

When we use the XOR dataset $\mathcal{O}_3$ to be the out-of-domain dataset, the learning curve of AcTraK overlaps with that of ERS depicted as Fig 5(e). It implies that the transfer learning process is unable to help label examples in this case, and both AcTraK and ERS select the same examples and label them all by experts. Depicted as Fig 4(a) and Fig 4(d), the distribution of $\mathcal{O}_3$ is significantly different from the in-domain dataset $\mathcal{U}$, and thus AcTraK judiciously drops the knowledge transferred from $\mathcal{O}_3$ but queries the experts instead in order to avoid mislabeling. This is formally discussed in Theorem 3, in which we have shown that the bound of the probability to query experts increases when the transfer classifier can not confidently label the examples. We also calculate these querying bounds and the actual querying rates in Table 2. The querying bound given in Theorem 3 is tight. Moreover, we can clearly see that AcTraK queries the experts with probability 94% when the out-of-domain dataset is $\mathcal{O}_3$. It explains why AcTraK can not outperform ERS with $\mathcal{O}_3$ in Fig 5(e): the transfer classifier has too little chance $(1 - 94\% = 6\%)$ to label the examples. Additionally, the

querying bound of $\mathcal{O}_1$ is less than that of $\mathcal{O}_2$. In other words, AcTraK may label more examples by the transfer classifier when the out-of-domain dataset is $\mathcal{O}_1$. It explains why $\mathcal{O}_1$ is more helpful than $\mathcal{O}_2$ to AcTraK in Fig 5(e).

3.2 Real World Dataset

We use two sets of real world datasets, remote sensing problem as well as text classification problem, to empirically evaluate the proposed method. But we first employ an evaluation metric to compare two active learners. One mainly cares about the performance with the increasing number of examples labeled by experts, shown by the learning curves of $\mathcal{A}_1$ and $\mathcal{A}_2$ in Fig 7. A superior active learner is supposed to gain a higher accuracy under the same number of queried examples, or reach the same classification accuracy with fewer labeled examples. This is shown by n_1 vs. n_2 in Fig 7. Thus, the superiority of $\mathcal{A}_1$ compared with $\mathcal{A}_2$ can be reflected by the area surrounded by the two learning curves, highlighted by dotted lines in Fig 7. In order to qualify this difference, we employ an evaluation metric IEA(*) (Integral Evaluation on Accuracy), and apply it to evaluate the proposed method in the following experiments.

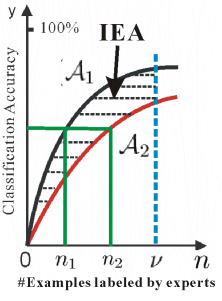


Fig. 7. IEA

Definition 1. *Given a classifier* $\mathbf{M}$*, two active learners* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*, let* $\mathcal{A}(n)$ *denote the classification accuracy of* $\mathbf{M}$ *trained on the dataset selected by the active learner* $\mathcal{A}$ *when the number of examples labeled by experts is* n*. Then,*

$$\mathbf{IEA}(\mathcal{A}_1, \mathcal{A}_2, \nu) = \int_0^{\nu} (\mathcal{A}_1(n) - \mathcal{A}_2(n)) d\mathbf{n} = \sum_{n=0}^{\nu} (\mathcal{A}_1(n) - \mathcal{A}_2(n)) \Delta n \qquad (7)$$

Remote Sensing Problem. The remote sensing problem is based on data collected from real landmines[1]. In this problem, there are a total of 29 sets of data, collected from different landmine fields. Each data is represented as a 9-dimensional feature vector extracted from radar images, and the class label is true mine or false mine. Since each of the 29 datasets are collected from different regions that may have different types of ground surface conditions, these datasets are considered to be dominated by different distributions. According to [17], datasets 1 to 10 are collected at foliated regions while datasets 20 to 24 are collected from regions that are bare earth or desert. Then, we combine the datasets 1 to 5 as the unlabeled in-domain dataset, while the datasets 20 to 24 as the labeled out-of-domain dataset respectively. Furthermore, we also combine datasets 6 to 10 as another out-of-domain dataset that is assumed to have a very similar distribution with the in-domain dataset. We conduct the experiment 10 times and

[1] http://www.ee.duke.edu/∼lcarin/LandmineData.zip

Table 3. Accuracy comparisons on remote sensing (landmine) dataset

Out-of-domain Dataset	SVM	TrAdaBoost(100 iterations)	AcTraK	IEA(AcTraK, ERS, 100)
Dataset 20	57%	89.76%	**94.49%**	+0.101
Dataset 21	57%	86.04%	**94.48%**	+0.108
Dataset 22	57%	90.5%	**94.49%**	+0.103
Dataset 23	57%	88.42%	**94.49%**	+0.123
Dataset 24	57%	90.7%	**94.49%**	+0.12
Dataset 6-10	57%	**94.76%**	94.49%	+0.134

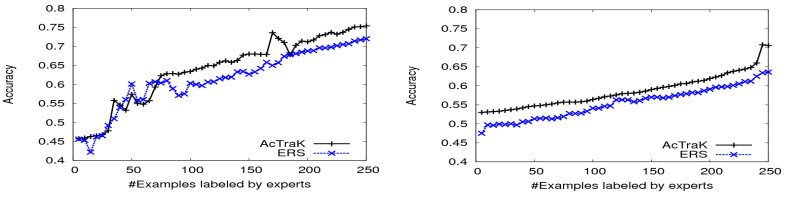
Note: The results of AcTraK under the 4th column is when only one example is labeled by experts.

randomly select two examples (one with label "true" while the other with label "false") as the initial training set each time. The experiment results are averaged and summarized in Table 3.

The first 5 rows of Table 3 show that AcTraK outperforms TrAdaBoost when the in-domain dataset is not so similar to the out-of-domain dataset (Dataset 20 to Dataset 24). Moreover, AcTraK also outperforms the active learner ERS in all cases. When the in-domain dataset is similar with the out-of-domain dataset (Dataset 6-10), AcTraK achieves the highest gain on ERS, demonstrating domain transfer can improve learning and reduce number of examples labeled by experts.

Text Classification Problem. Another set of experiment on text classification problem uses the 20 Newsgroups. It contains approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. We generate 6 cross-domain learning tasks. 20 Newsgroups has a two-level hierarchy so that each learning task involves a top category classification problem but the training and test data are drawn from different sub categories. For example, the goal is to distinguish documents from two top newsgroup categories: rec and talk. So a training set involves documents from "rec.autos", and "talk.politics.misc" whereas the test set includes sub-categories "rec.sport.baseball" "talk.religions.misc", etc. The strategy is to split the sub-categories among the training and the test sets so that the distributions of the two sets are similar but not exactly the same. The tasks are generated in the same way as in [4] and more details can be found there. Similar to other experiments, each of the in-domain datasets has only 2 randomly selected labeled examples, one positive and another negative. Reported results in Table 4 are averaged over 10 runs. The results of the first two datasets are also plotted in Fig 8.

It is important to note that the classification results of AcTraK shown in Fig 4 is when the number of examples labeled by experts is 250. It is relatively small in size since each dataset in our experiment has 3500 to 3965 unlabeled documents ([4]). As summarized in Table 4, TrAdaBoost can increase the learning accuracy in some cases, such as with the dataset "comp vs. talk". However, one can hardly guarantee that the exclusive use of transfer learning is enough to learn the current task. For example, when the dataset is "comp vs. sci", TrAdaBoost does not increase the accuracy significantly. But the proposed algorithm AcTraK can achieve an accuracy 78% compared with 57.3% given by TrAdaBoost. It implies the efficiency of AcTraK to actively gain its knowledge both from transfer



(a) rec vs. talk (b) rec vs. sci

Fig. 8. Comparisons with ERS on 20 Newsgroups dataset

Table 4. Accuracy comparisons on 20 Newsgroups dataset

Dataset	SVM	TrAdaBoost(100 iterations)	AcTraK	IEA(AcTraK, ERS, 250)
rec vs. talk	60.2%	72.3%	**75.4%**	+0.91
rec vs. sci	59.1%	67.4%	**70.6%**	+1.83
comp vs. talk	53.6%	74.4%	**80.9%**	+0.21
comp vs. sci	52.7%	57.3%	**78.0%**	+0.88
comp vs. rec	49.1%	77.2%	**82.1%**	+0.35
sci vs. talk	57.6%	71.3%	**75.1%**	+0.84

learning and domain experts, while TrAdaBoost adopts the passive strategy to thoroughly depend on transfer learning. In addition, from Table 4 and Fig 8, we find that AcTraK can effectively reduce the number of examples labeled by experts as compared with ERS. For example, upon the dataset "rec vs. talk", to reach the accuracy 70%, AcTraK is with 160 examples labeled by experts while ERS needs over 230 such examples.

4 Related Work

Transfer learning utilizes the knowledge from other domain(s) to help learn the current domain so as to resolve label deficiency. One of the main issues in this area is how to resolve the different data distributions. One general approach proposed to solve the problem with different data distributions is based on instance weighting (e.g., [2][4][5][10][7]). The motivation of these methods are to "emphasize" those "similar" and discriminated instances. Another line of work tries to change the representation of the observation $\mathbf{x}$ by projecting them into another space in which the projected instances from different domains are similar to each other (e.g., [1][12]). Most of the previous work assume that the knowledge transferred from other domain(s) can finally help the learning. However, this assumption can be easily violated in practice. The knowledge transferred from other domains may reduce the learning accuracy due to implicit domain differences. We call it the domain difference risk, and we effectively solve the problem by actively transfer the knowledge across domains to help the learning only when they are useful.

Active learning is another way to solve label deficiency. It mainly focuses on carefully selecting a few additional examples for which it requests labels, so as to increase the learning accuracy. Thus, different active learners use different

selection criteria. For example, *uncertainty sampling* (e.g., [9][18]) selects the example on which the current learner has lower certainty; *Query-by-Committee* (e.g., [6][13]) selects examples that cause maximum disagreement amongst an ensemble of hypotheses, etc. There are also some other topics proposed in recent years to resolve different problems in active learning, such as the incorporation of ensemble methods (e.g., [8]), the incorporation of model selection (e.g., [15]), etc. It is important to mention that the examples selected by the previous active learners are more or less uncertain to be labeled directly by the in-domain classifier. Then in this paper, we use the knowledge transferred from other domain to help label these selected examples, so as to reduce labeling cost.

5 Conclusion

We propose a new framework to actively transfer the knowledge from other domain to help label the instances from the current domain. To do so, we first select an essential example and then apply a transfer classifier to label it. But if the classification given by the transfer classifier is of low confidence, we ask domain experts instead to label the example. We develop Theorem 2 to demonstrate that this strategy can effectively resolve domain difference risk by transferring domain knowledge only when they are useful. Furthermore, we also derive Theorem 3 to prove that the proposed framework can reduce labeling cost by querying fewer examples labeled by experts, as compared with traditional active learners. In addition, we also propose a new transfer learning model adopted in the framework, and this transfer learning model is bounded to be no worse than an instance-based ensemble method in error rate, proven in Theorem 1. There are at least two important advantages of the proposed approach. First, it effectively solves the domain difference risk problem that can easily make transfer learning fail. Second, most of previous active learning models can be directly adopted in the framework to reduce the number of examples labeled by experts.

We design a few synthetic datasets to study how the proposed framework resolves domain difference and reduce the number of examples labeled by experts. The proposed sampling error bound in Theorem 2 and querying bound in Theorem 3 are also empirically demonstrated to be tight bounds in this experiment. Furthermore, two categories of real world datasets, including remote sensing and text classification datasets have been used to generate several transfer learning problems. Experiment shows that the proposed method can significantly outperform the comparable transfer learning model by resolving domain difference. For example, with one of the text classification datasets, the proposed method achieves an accuracy 78.0%, while the comparable transfer learning model drops to 57.3%, due to domain difference. Moreover, the proposed method can also effectively reduce labeling cost by querying fewer examples labeled by experts as compared with the traditional active learner. For instance, in an experiment on the text classification problem, the comparable active learner requires over 230 examples labeled by experts to gain the accuracy 70%, while the proposed method is with at most 160 such examples to reach the same accuracy.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Proc. of NIPS 2006 (2007)
2. Bickel, S., Brückner, M., Scheffer, T.: Discriminative learning for differing training and test distributions. In: Proc. of ICML 2007 (2007)
3. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research 26, 101–126 (2006)
4. Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for transfer learning. In: Proc. of ICML 2007 (2007)
5. Fan, W., Davidson, I.: On Sample Selection Bias and Its Efficient Correction via Model Averaging and Unlabeled Examples. In: Proc. of SDM 2007 (2007)
6. Freund, Y., Seung, H., Shamir, E., Tishby, N.: Selective sampling using the Query By Committee algorithm. Machine Learning Journal 28, 133–168 (1997)
7. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: Proc. of NIPS 2006 (2007)
8. Körner, C., Wrobel, S.: Multi-class Ensemble-Based Active Learning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212. Springer, Heidelberg (2006)
9. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: Proc. of SIGIR 1994 (1994)
10. Ren, J., Shi, X., Fan, W., Yu, P.: Type-Independent Correction of Sample Selection Bias via Structural Discovery and Re-balancing. In: Proc. of SDM 2008 (2008)
11. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proc. of ICML 2001 (2001)
12. Satpal, S., Sarawagi, S.: Domain adaptation of conditional probability models via feature subsetting. In: Kok, J.N., Koronacki, J., López de Mántaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702. Springer, Heidelberg (2007)
13. Senung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proc. 5th Annual ACM Workshop on Computational Learning Theory (1992)
14. Shimodaira, H.: Inproving predictive inference under covariate shift by weighting the log-likehood function. Journal of Statistical Planning and Inference 90, 227–244 (2000)
15. Sugiyama, M., Rubens, N.: Active Learning with Model Selection in Linear Regression. In: Proc. of SDM 2008 (2008)
16. Xing, D., Dai, W., Xue, G., Yu, Y.: Bridged refinement for transfer learning. In: Kok, J.N., Koronacki, J., López de Mántaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702. Springer, Heidelberg (2007)
17. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with dirichlet process priors. Journal of Machine Learning Research 8, 35–63 (2007)
18. Xu, Z., Tresp, Y.K., Xu, V., Wang, X.: Representative sampling for text classification using support vector machines. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633. Springer, Heidelberg (2003)