Multi-Domain Active Learning for Text Classification

Lianghao Li[†] Xiaoming Jin[†] Sinno Jialin Pan[‡] Jian-Tao Sun[§]
[†]School of Software, Tsinghua University, Beijing 100084, P.R. China
[‡]Institute for Infocomm Research, Singapore 138632
[§]Microsoft Research Asia, Beijing 100080, P.R. China
lianghaoli@yahoo.com, xmjin@tsinghua.edu.cn, jspan@i2r.a-star.edu.sg,
jtsun@microsoft.com

ABSTRACT

Active learning has been proven to be effective in reducing labeling efforts for supervised learning. However, existing active learning work has mainly focused on training models for a single domain. In practical applications, it is common to simultaneously train classifiers for multiple domains. For example, some merchant web sites (like Amazon.com) may need a set of classifiers to predict the sentiment polarity of product reviews collected from various domains (e.g., electronics, books, shoes). Though different domains have their own unique features, they may share some common latent features. If we apply active learning on each domain separately, some data instances selected from different domains may contain duplicate knowledge due to the common features. Therefore, how to choose the data from multiple domains to label is crucial to further reducing the human labeling efforts in multi-domain learning. In this paper, we propose a novel *multi-domain active learning* framework to jointly select data instances from all domains with duplicate information considered. In our solution, a shared subspace is first learned to represent common latent features of different domains. By considering the common and the domain-specific features together, the model loss reduction induced by each data instance can be decomposed into a common part and a domain-specific part. In this way, the duplicate information across domains can be encoded into the common part of model loss reduction and taken into account when querying. We compare our method with the state-of-the-art active learning approaches on several text classification tasks: sentiment classification, newsgroup classification and email spam filtering. The experiment results show that our method reduces the human labeling efforts by 33.2%, 42.9% and 68.7% on the three tasks, respectively.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition, concept learning*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

KDD'12, August 12–16, 2012, Beijing, China.

General Terms

Algorithms, Experimentation

Keywords

Active Learning, Transfer Learning, Text Classification

1. INTRODUCTION

Text classification has drawn much research attention in the literature. Typically, supervised classification algorithms require sufficient labeled data to train accurate classifiers, while the data labeling cost may be expensive. Active learning has been proven to be effective in reducing the human labeling efforts by actively choosing the most informative data to label. Existing active learning work has mainly focused on training models for a single domain. But in many applications, data of interest are from multiple domains and a group of classifiers need to be trained simultaneously for all the domains. For example, Amazon.com has organized user reviews of many products. A sentiment classifier [3] of each product class (domain) is highly desirable to automatically organize reviews according to user demands. Since different words can be used to express sentiment in different domains [17], training a single classifier for all domains would not generalize well across various domains. For instance, words like "blur", "fast", "sharp" are used to comment *electronics* products, while they do not carry opinion in *books* domain. Therefore, each domain should have its own sentiment classifier. Email spam filtering is another example [8]. Since users may have different backgrounds and interests, it is reasonable to customize spam filters for individual users.

Active learning for multi-domain text classification is a novel research problem. The algorithm of selecting data instances to label is not trivial. If we simply apply active learning on each domain separately, some data instances selected from different domains may contain duplicate information due to the inherent relationship among domains. For example, in sentiment classification, reviews containing common sentiment words like "wonderful", "perfect" may be selected to label by active learners of each domain, which may cause redundant labeling efforts. On the other hand, if we apply active learning for all domains together, the query strategy may be affected by the distribution gap between different domains. Therefore, how to measure the informativeness of data instances across domains is crucial. In this paper, we propose a novel global optimization based active learning framework for multi-domain text classification. The proposed query strategy aims to select unlabeled instances

which can maximally reduce the model loss of all classifiers once labeled. In our solution, a shared subspace is first learned to represent common latent features of different domains. By splitting the feature space into a common part and a domain-specific part, the model loss reduction induced by each data candidate can be decomposed into the domain-specific loss reduction of the classifier on its corresponding domain, and the common loss reduction of the classifiers on all domains. By jointly querying instances, the common model loss of all classifiers can be reduced simultaneously, and the redundant labeling efforts can be saved.

It is worth noting that the problem setting of multi-domain classification is different from that of cross-domain classification. In cross-domain classification, data of interest are assumed to come from a source domain and a target domain. Sufficient labeled data are available in the source domain while no or few labeled data are available in the target domain. The goal is to train a classifier of the target domain by leveraging the labeled data of the source domain. In multi-domain classification, no domain is assumed to have sufficient labeled data. The goal is to simultaneously train classifiers for multiple domains by leveraging common knowledge among them. Active learning for multi-domain classification aims to jointly select data to label for training accurate classifiers on all domains.

The main contributions of our work include: 1) We studied an important practical problem for active learning in multiple domains. To the best of our knowledge, this is the first work which aims to actively build text classifiers for multiple domains simultaneously. 2) We proposed an efficient multi-domain active learning framework and showed its effectiveness on three real-world applications, i.e. sentiment classification, newsgroup classification and email spam filtering. The experiment results on the three tasks demonstrate that our proposed method can save more than 33% labeling efforts compared with the state-of-the-art active learning approaches, and save more than 50% labeling efforts compared with the random query methods.

The rest of this paper is organized as follows: we begin by reviewing the related works in the next section. After that, we describe the problem statement in Section 3, and present our solution in Section 4. The experiment results are discussed in Section 5. Finally, we conclude the paper and discuss some future work in Section 6.

2. RELATED WORK

The performance of supervised classification highly relies on labeled data. However, to collect sufficient training data is difficult and time-consuming. Active learning is an alternative learning framework which allows classification algorithms to choose the data they learn from. Existing active learning algorithms can be generally put into three categories: 1) uncertainty sampling [13, 25], which selects the data instances that are the most uncertainly predicted by the current classifier; 2) query by committee [22] selects the data instances about which the "committee" disagree most; and 3) expected error reduction [20], which aims to select the instance that can contribute the largest model loss reduction for the current classifier once labeled. Recently, Donmez and Carbonell proposed the proactive learning framework which relaxes some unrealistic assumptions of active learning in practical applications [7]. Beygelzimer *et al.* proposed an importance weighting method to avoid label-sampling bias in active learning [2]. In [15] and [5], the authors proposed the active learning methods for data with multiple views. In multi-view learning, every data instance is assumed to have several different descriptions, each of which can be used to learn concepts of interest.

Transfer learning is another technology to save the labeling efforts for supervised learning. Dredze *et al.* developed a multi-domain learning method based on parameter combination [8]. Xie *et al.* proposed the LatentMap algorithm to leverage the shared features for transfer learning [26]. Given an oracle and a lot of labeled data from a source domain, some researchers proposed to combine active learning and transfer learning to train an accurate classifier for a target domain [18, 23]. Shi *et al.* proposed to use the source domain classifier to answer the target domain queries as often as possible, and query the oracle only when necessary [23]. In [18], Rai *et al.* considered to use the source domain classifier as an initial classifier for the target domain. And the source domain data are further used to rule out the target domain queries which appear similar to the source domain data. Different from their works, we aim to build classifiers for multiple domains together, while they targeted at training the classifier of target domain by using the knowledge from the source domain.

Our work is also related to multi-task active learning, which has been studied to solve the problem where data instances are labeled in multiple ways for different tasks. Reichart *et al.* proposed a novel active learning method to label data instances with several linguistic annotations, such as named entities, syntactic parse trees, etc. [19]. Zhang tried to solve the multi-task active learning problem where outputs of different tasks are coupled by constraints [27]. Harpale *et al.* proposed an active learning method for multi-task adaptive filtering [11]. An adaptive filtering system monitors a set of documents to find and deliver the relevant items to a particular task. Its performance is boosted with the relevance feedback received on the delivered items. In [11], the items which lead to the maximal relevance feedback will be selected to deliver. Different from their works, we focus on a single task with multiple domains. In our problem, different domains share the same target concepts but have different data distributions. In addition, our work aims at proposing an active learning method for classification problems instead of adaptive filtering or natural language annotation. This makes the optimization goal of our proposed query strategy different from theirs.

3. PROBLEM DEFINITION

In this section, we introduce some definitions and the problem statement.

Definition 1. *(Domain) A domain consists of a set of data instances which are generated from the same data distribution $P(x)$, where $x \in \mathcal{X}$ and $\mathcal{X}$ is a feature space.*

For example, a set of user reviews for electronics products can be regarded as one domain, while reviews for different types of products, such as books, movies, can be regarded as *books* and *movies* domains, respectively. Data instances from one domain are assumed to be independent and identically distributed (*i.i.d.*). But data distributions across domains may be different. In this paper, the domain each data instance belongs to is assumed to be known. The multi-domain classification problem is defined as follows:

Definition 2. *(Multi-Domain Classification) Given a set of data instances collected from K different domains, where each domain has its own data distribution. Let $\mathcal{X}$ be a feature space*[1] *and $\mathcal{Y}$ be a pre-defined label set. The task is to train K classifiers $f^\ell : \mathcal{X} \to \mathcal{Y}$, $\ell = 1, 2, \ldots, K$, for all the domains.*

Based on the definitions above, we now define the problem we aim to address in this paper as follows:

Definition 3. *(Active Learning for Multi-Domain Classification) Let $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^K\}$ be an unlabeled data pool which consists of data instances collected from K different domains. Here $\mathcal{P}^\ell = \{\mathbf{x}_1^\ell, \mathbf{x}_2^\ell, \ldots, \mathbf{x}_{N^\ell}^\ell\}$ includes N^ℓ data instances come from the ℓ'th domain. The task is to build K accurate classifiers $f^\ell : \mathcal{X} \to \mathcal{Y}$, $\ell = 1, 2, \ldots, K$, by selecting data instances to label as few as possible.*

Our active learning framework is based on pool-based sampling [13, 21]. In pool-based sampling, active learning is iteratively performed on an unlabeled data pool, which is usually assumed to be closed (i.e. stationary) [21]. Typically, in each iteration, the active learner scans the unlabeled data pool and chooses the most informative data candidates to label.

4. OUR SOLUTION

In this section, we describe our solution for multi-domain active learning. The main notations are listed in Table 1.

Table 1: Notations

Symbols	Description
K	total number of domains
$\mathbf{x}_i^\ell$	the i'th labeled data in the ℓ'th domain
y_i^ℓ	ground truth of $\mathbf{x}_i^\ell$
θ	learned shared subspace transformation matrix
$\mathbf{w}^\ell$	weight vector specific to the ℓ'th domain
$\mathbf{v}$	weight vector associated with the shared subspace
$f_\mathcal{D}^\ell$	predictive function of the ℓ'th domain
$L(\cdot)$	model loss of a classifier
λ^ℓ	domain weight specified by users
$\mathcal{L}_\mathcal{D}$	global model loss of all classifiers
$\mathcal{V}^\ell$	version space of the ℓ'th domain
$\mathcal{W}^\ell$	parameter space of the ℓ'th domain
$V_\mathcal{D}^\ell$	the size of $\mathcal{V}^\ell$

4.1 A General Optimization Framework

Recall that, in active learning for a single domain, an active learner attempts to select the most informative data instances to label in order to train an accurate classifier using as few labeling efforts as possible. In active learning for multiple domains, the goal is to choose the data instances which are not only informative for their corresponding domains but also for other domains such that all classifiers can benefit from the labeling.

Suppose that $L(f_\mathcal{D}^\ell)$ is the model loss of classifier $f_\mathcal{D}^\ell$, the global model loss of all classifiers is defined as:

$$\mathcal{L}_\mathcal{D} = \sum_{\ell=1}^{K} \lambda^\ell L(f_\mathcal{D}^\ell), \tag{1}$$

[1]In this work, we assume all domains share the same vocabulary (i.e. feature space).

where $\{\lambda^\ell\}_{\ell=1}^K$ are user specified weights for different domains. The goal of our query strategy is to select an unlabeled instance $\mathbf{x}^*$ which can maximally reduce the global model loss once labeled. The optimization objective can be formulated as:

$$\begin{aligned}\mathbf{x}^* &= \arg\max_{\mathbf{x}^*} \mathcal{L}_\mathcal{D} - \mathcal{L}_{\mathcal{D}+(\mathbf{x}^*,y^*)} \\ &= \arg\max_{\mathbf{x}^*} \sum_{\ell=1}^{K} \lambda^\ell \cdot \left(L(f_\mathcal{D}^\ell) - L(f_{\mathcal{D}+(\mathbf{x}^*,y^*)}^\ell)\right), \end{aligned} \tag{2}$$

where $\mathcal{D}+(\mathbf{x}^*, y^*)$ is the expanded training set after data instance $\mathbf{x}^*$ and its ground truth y^* are added. In some real-world applications, different domains may have different priorities. For example, users may require high classification performance or fast model convergency for some particular domains. In this case, one can assign larger weights for such domains. However, in many other scenarios, users may not have these requirements. Under such case, one can simply set the same weight for each domain. Without loss of generality, we set $\lambda^\ell = 1$ for all domains in this paper.

In practice, we do not know ground truth y^* of data instance $\mathbf{x}^*$ before querying. Therefore, we are not able to estimate the model loss in (2) directly. Instead, we use the expectation loss over all possible labels to approximate the true model loss. As a result, we can replace (2) by the following objective:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}^*} \sum_{y\in\mathcal{Y}} \hat{P}(y|\mathbf{x}^*) \sum_{\ell=1}^{K} \left(L(f_\mathcal{D}^\ell) - L(f_{\mathcal{D}+(\mathbf{x}^*,y)}^\ell)\right), \tag{3}$$

where $\hat{P}(y|\mathbf{x}^*)$ is the conditional probability of label y given data instance $\mathbf{x}^*$ estimated by the current classifier.

4.2 Multi-Domain Classification with SVM

Before describing our solution for multi-domain active learning, we first present an SVM-based multi-domain classification method which is used as the classification model in our optimization framework.

Support Vector Machines (SVMs) have been widely used for text classification [12, 25]. In this paper, we incorporate a shared subspace to represent common latent features into SVM for multi-domain classification. The predictive function $f_\mathcal{D}^\ell$ of the ℓ'th ($\ell \in \{1, ..., K\}$) domain is defined as:

$$f_\mathcal{D}^\ell(\mathbf{x}^\ell) = \mathbf{w}^\ell \cdot \Phi(\mathbf{x}^\ell) + \mathbf{v} \cdot \Phi(\theta\mathbf{x}^\ell), \tag{4}$$

which consists of two parts: one is performed on the original feature space, and the other is derived for the shared subspace. Here $\mathcal{D}$ is a training set, Φ is a feature map, $\mathbf{w}^\ell$ and $\mathbf{v}$ are two weight vectors, θ is a learned transformation matrix to map the original feature space to the shared low-dimensional subspace. The shared parameters $\mathbf{v}$ and θ are leveraged to capture the common latent features across domains. Note that the idea of the formulation above is similar to that in multi-task learning [1, 9]. However, in this paper, we focus on proposing a novel *active learning* framework for multi-domain classification instead of a novel multi-domain classification method. In [1], Ando and Zhang proposed to learn the parameters $\{\mathbf{w}^\ell\}$'s, $\mathbf{v}$ and θ jointly by updating them iteratively. In each iteration, the singular value decomposition is required to update θ, which is not efficient, especially for active learning.

We propose to learn the parameters in two steps. In the first step, we apply Spectral Feature Alignment (SFA) [17],

which is an unsupervised shared subspace learning method, to estimate θ. Note that besides SFA, many other effective approaches to shared subspace learning can be integrated into our framework, such as Structural Correspondence learning (SCL) [4], Maximum Mean Discrepancy Embedding (MMDE) [16], etc. In SFA, a set of domain independent features are firstly identified, and a bipartite graph is constructed to model the co-occurrence between the domain-independent features and the domain-specific features. Then a spectral clustering algorithm is adapted on the bipartite graph to co-align the two kinds of features into unified clusters. The space spanned by the unified clusters is then considered as the shared subspace across domains. In the second step, we estimate $\{\mathbf{w}^\ell\}$'s and $\mathbf{v}$ by solving the SVM optimization problem as follows[2]:

$$\min_{\{\mathbf{w}^\ell\}\text{'s},\mathbf{v}} \quad \frac{1}{2}\|\mathbf{v}\|^2 + \frac{1}{2}\sum_{\ell=1}^{K}\|\mathbf{w}^\ell\|^2, \tag{5}$$
$$s.t. \quad y_i^\ell(\mathbf{w}^\ell \cdot \Phi(\mathbf{x}_i^\ell) + \mathbf{v} \cdot \Phi(\theta\mathbf{x}_i^\ell)) \geq 1, \;\; \ell = 1, ..., K.$$

Note that for text classification, data instances are often linearly-separable due to the high dimensionality of its feature space. Therefore, in this paper, we present our framework in the linearly-separable manner and leave the non-separable case to our future work. It can be shown that the optimization problem (5) can be directly linked to a standard SVM problem with a proper feature map [9] and solved by a standard SVM solver.

In our approach, the weight vector $\mathbf{v}$ is derived from the shared subspace, and learned from all training data across domains. Therefore, it can reflect the common discriminative information of all domains. The weight vectors $\{\mathbf{w}^\ell\}$'s are only affected by the training data in the corresponding domain, which implies that they should reflect the domain-specific discriminative information. By splitting the feature space into the two parts, we can measure both the common and the domain-specific model loss reduction induced by each data instance.

4.3 Multi-Domain Active Learning

In this section, we describe our solution for the proposed optimization framework (3) based on the multi-domain SVM. According to (1), the global model loss can be decomposed into the model loss of the classifier in each domain. So our problem becomes to measure the model loss reduction $\{L(f_\mathcal{D}^\ell) - L(f_{\mathcal{D}+(\mathbf{x}^*,y)}^\ell)\}_{\ell=1}^K$ of each classifier. As suggested by Tong and Koller [24], we can measure the model loss of each classifier by the size of version space. A version space $\mathcal{V}$ is a set of hypotheses that are consistent with the current training data instances [14]. For the ℓ'th domain, the version space $\mathcal{V}^\ell$ is defined as:

$$\mathcal{V}^\ell = \left\{ \frac{\mathbf{u}^\ell}{\|\mathbf{u}^\ell\|} \,\middle|\, \mathbf{u}^\ell \in \mathcal{W}^\ell, \forall i \; y_i^\ell(\mathbf{w}^\ell\Phi(\mathbf{x}_i^\ell) + \mathbf{v}\Phi(\theta\mathbf{x}_i^\ell)) > 0 \right\}, \tag{6}$$

where $\mathbf{u}^\ell = [\mathbf{w}^\ell, \mathbf{v}]$ and $\mathcal{W}^\ell$ is the parameter space. Since we can simply multiply a non-zero scale to a consistent hypothesis to get another one, we normalize the weight vectors to eliminate this freedom.

For SVM, we can use the margin of SVM as an indicator of the size of version space. Suppose we have a pool of unlabeled instances, we can evaluate each candidate by adding it into $\mathcal{D}$ and re-training an SVM based on (5) to estimate the new margin. We then select the data candidate which contributes the largest reduction of all version spaces to label. However, this process is very expensive in computation, especially when the candidate pool is large. To make it more practical, we apply a heuristic idea as proposed in [24](cf. page 34) to simplify the computation by mapping the size of new version space to the size of current version space. Denote $V_\mathcal{D}^\ell$ the size of current version space, the size of new version space (i.e. $V_{\mathcal{D}+(\mathbf{x}^*,y)}^\ell$) after adding $(\mathbf{x}^*, y)$ into the training set can be approximated as:

$$V_{\mathcal{D}+(\mathbf{x}^*,y)}^\ell \approx \frac{1 + y f_\mathcal{D}^\ell(\mathbf{x}^*)}{2} V_\mathcal{D}^\ell. \tag{7}$$

Based on the approximation above, the model loss reduction of each classifier in (3) can be rewritten as:

$$\begin{aligned} L(f_\mathcal{D}^\ell) - L(f_{\mathcal{D}+(\mathbf{x}^*,y)}^\ell) &= V_\mathcal{D}^\ell - V_{\mathcal{D}+(\mathbf{x}^*,y)}^\ell \\ &\approx \frac{1 - y f_\mathcal{D}^\ell(\mathbf{x}^*)}{2} V_\mathcal{D}^\ell. \end{aligned} \tag{8}$$

An intuitive explanation for the above estimation is that if data candidate $\mathbf{x}^*$ can be correctly predicted by the current model, that is $y = \text{sgn}(f_\mathcal{D}^\ell(\mathbf{x}^*))$, then the smaller the value of $\|f_\mathcal{D}^\ell(\mathbf{x}^*)\|$ is, the less confidence on $\mathbf{x}^*$ the current model has. As a result, data candidate $\mathbf{x}^*$ tend to be queried for labeling. On the other hand, if data candidate $\mathbf{x}^*$ cannot be correctly predicted, then the larger the value of $\|f_\mathcal{D}^\ell(\mathbf{x}^*)\|$ is, the more errors the current model makes. In this case, querying $\mathbf{x}^*$ can greatly improve the current model.

Recall that, given classifier $f_\mathcal{D}^\ell$ of the ℓ'th domain, if data candidate $\mathbf{x}^*$ is not from the ℓ'th domain, then $\mathbf{x}^*$ can only affect the version space of the ℓ'th domain via the shared subspace when queried. Correspondingly, classifier $f_\mathcal{D}^\ell$ can only make prediction on data candidate $\mathbf{x}^*$ through the common weight vector $\mathbf{v}$. So we propose to use the following predictive function $f_\mathcal{D}^\ell(\mathbf{x}^*)$ to calculate the model loss reduction in (8),

$$f_\mathcal{D}^\ell(\mathbf{x}^*) = \begin{cases} \mathbf{w}^\ell \cdot \Phi(\mathbf{x}^*) + \mathbf{v} \cdot \Phi(\theta\mathbf{x}^*) & \mathbf{x}^* \in \mathcal{P}^\ell, \\ \mathbf{v} \cdot \Phi(\theta\mathbf{x}^*) & \mathbf{x}^* \notin \mathcal{P}^\ell. \end{cases}$$

Therefore, the model loss reduction induced by each data candidate is decomposed into two parts: 1) the version space reduction of its corresponding domain in the whole feature space, and 2) the version space reduction of other domains in the shared subspace. In this way, the common model loss of all classifiers can be reduced together, and more labeling efforts can be saved. Since we learn all the classifiers jointly, there is no guarantee that the solutions of (5) can lead to the maximal margin solution for the classifier of each domain. However, because the low-dimensional subspace is shared by all domains, the hyperplane learned onto it should be consistent with the data instances from all domains. Therefore, the hyperplane of each domain learned by (5) is a good approximation of the hyperplane learned on the labeled data only from its corresponding domain.

By using the size of SVM margin as the indicator of $V_\mathcal{D}^\ell$, and substitute (8) into (3), our final query strategy for multi-domain active learning can be written as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}^*} \sum_{y=\pm 1} \hat{P}(y|\mathbf{x}^*) \sum_{\ell=1}^{K} \frac{1 - y f_\mathcal{D}^\ell(\mathbf{x}^*)}{\|\mathbf{u}^\ell\|}. \tag{9}$$

[2]Here we introduce $\Phi_0(\mathbf{x}) = 1$ to replace the bias parameter of SVM.

Algorithm 1: Multi-Domain Active Learning

Input : (1) A pool $\mathcal{P}$ of unlabeled instances which are collected from K domains, (2) Number of initial training data in each domain M, (3) Number of iterations T, (4) Number of queried instances per iteration S

Output: K classifiers

Randomly label M data instances of each domain, and form the initial training set $\mathcal{D}$;

Learn the low-dimensional shared subspace using SFA;

for $t \leftarrow 1$ **to** T **do**

  Train K classifiers in the training set $\mathcal{D}$ using (5);

  foreach $\mathbf{x}_n^\ell \in \mathcal{P}$ **do**

    Estimate the global model loss reduction via (9);

  end

  Query the labels Y^* of S unlabeled instances U^* which have the largest global model loss reduction;

  Update the training set by $\mathcal{D} \leftarrow \mathcal{D} \cup (U^*, Y^*)$, and remove U^* from $\mathcal{P}$;

end

In order to calculate $\hat{P}(y|\mathbf{x}^*)$ in (9), we train a Logistic Regression classifier on all training data by maximizing the log-likelihood $J(\mathbf{w}^1, \cdots, \mathbf{w}^K, \mathbf{v}) = \sum_{\ell,i} \log \sigma(y_i^\ell(\mathbf{w}^\ell \mathbf{x}_i^\ell + \mathbf{v}\theta\mathbf{x}_i^\ell))$, and use it to estimate the probabilities. The complete process of our proposed method is summarized in Algorithm 1. The proposed method is very efficient because it only needs to learn one SVM per iteration, and in each iteration, it estimates the global model loss reduction induced by each candidate efficiently via (9).

For the classification problem having more than two categories, one simple and effective way is to use the one-vs-all technique. Suppose we have C classes, we can train C binary classifiers $\{f_{\mathcal{D}}^{\ell,c}\}_{c=1}^C$, where the classifier $f_{\mathcal{D}}^{\ell,c}$ is used to predict whether an instance belongs to the c'th class or not. Our multi-domain active learning method can be applied accordingly.

5. EXPERIMENTS

In this section, we conduct experiments on three real-world applications (i.e., sentiment classification, newsgroup classification and email spam filtering) to evaluate the effectiveness of our method.

5.1 Datasets

5.1.1 Multi-Domain Sentiment Dataset

The Multi-Domain Sentiment Dataset [3] has been widely used as a benchmark dataset for domain adaptation and sentiment analysis. It contains a collection of product reviews from Amazon.com. The reviews are about four product domains: *Book* (**B**), *DVD* (**D**), *Electronics* (**E**) and *Kitchen* (**K**). Each review has been annotated as positive or negative sentiment polarity according to users' rating scores. The summary of this dataset is described in Table 2.

From this dataset, we construct five multi-domain sentiment classification tasks: **B+D+E**, **B+D+K**, **B+E+K**, **D+E+K** and **B+D+E+K**, where each boldfaced letter corresponds with a domain. For example, **B+D+E** denotes sentiment classification in *Book*, *DVD* and *Electronics* domains.

Table 2: Summary of Multi-Domain Sentiment Dataset

Domain	# Reviews	# Pos	# Neg	# Features
Book	6,465	3,264	3,201	17,465
DVD	5,585	2,807	2,778	15,437
Electronics	7,677	3,853	3,824	13,687
Kitchen	7,945	3,954	3,991	12,439

5.1.2 20Newsgroups

The 20Newsgroups dataset[3] has been widely used for newsgroup classification and cross-domain text classification. As in the previous work [6], we generate four newsgroup domains from the dataset by utilizing its hierarchical structure. Table 3 shows the generated newsgroup domains. For example, domain NG-1 contains documents from four sub-categories, which are under four top-categories, respectively. The classification task is defined in the top-category level, where our goal is to classify documents into one of the four top-categories: *comp*, *rec*, *sci* and *talk*. This domain generation strategy can ensure the domains are different but related, because different domains consist of documents in different sub-categories, but are under the same top-categories.

Table 3: Four Domains Generated from 20Newsgroups

Domain	Newsgroups	
NG-1	comp.graphics sci.crypt	rec.autos talk.politics.guns
NG-2	comp.os.ms-windows.misc sci.electronics	rec.motorcycles talk.politics.mideast
NG-3	comp.sys.ibm.pc.hardware sci.med	rec.sport.baseball talk.politics.misc
NG-4	comp.sys.mac.hardware sci.space	rec.sport.hockey talk.religion.misc

By using the generated domains, we construct four multi-domain newsgroup classification tasks: **NG-123**, **NG-124**, **NG-134** and **NG-234**, where each digit denotes a domain. For example, **NG-123** denotes the multi-domain newsgroup classification in NG-1, NG-2 and NG-3 domains.

5.1.3 Email Spam Filtering Dataset

The email spam filtering dataset[4] released by ECML/PKDD 2006 discovery challenge contains 15 separate inboxes for users u00∼u14, where "u∗∗" is a user id. For each inbox, there are 200 spam and 200 non-spam emails. In our experiments, each inbox is regarded as a domain and the learning task is to train a spam filter for each user to classify whether a new mail is a spam or not. From this dataset, we construct four multi-domain spam filtering tasks: **u00-u04**, **u05-u09**, **u10-u14** and **u00-u14**. For example, **u00-u04** denotes the email spam filtering in u00∼u04 domains.

5.2 Comparison Methods

In order to test the effectiveness of our method (which is referred to as MultiAL), we compare it with several active learning approaches. The first method is to perform a single-domain active learning for each domain independently. We call it SingleAL. The second method is to merge all domain data into a unified pool and perform active learning in the unified pool to train a single classifier for prediction. We call this approach UnifiedAL. In addition, once the shared subspace is identified, we can embed data instances from all domains into the shared subspace and generate a new

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

[4]http://www.ecmlpkdd2006.org/challenge.html

unified domain. We perform active learning in the new unified domain and train a single classifier for prediction. We call this method EmbedAL. We also test the Random query method which chooses unlabeled instances to label at random. In this paper, the SVM-based Simple-Margin active learning method proposed in [25] is adopted as the basic active learner for SingleAL, UnifiedAL and EmbedAL.

An alternative solution for multi-domain active learning is to apply existing active learning algorithms in each domain independently, and then apply existing transfer learning techniques to train more accurate classifiers by leveraging labeled data among domains. Here, we adopt the multi-domain SVM described in (5) as the classification method for SingleAL and Random to get another two comparison methods. We call them SingleAL+ and Random+, respectively.

5.3 Experiment Settings

For data preprocessing, we convert all words to lower cases and remove the stop words. Term frequency is used for feature weighting in all methods. Linear kernel is used as the feature map for SVM because of its good performance in text classification [12]. LIBLINEAR SVM [10] is used as the base classifier for all methods, and all parameters are set to their default values. In using SFA to learn the shared subspace for the multi-domain SVM, we adopt the same parameter setting adopted in the original paper [17]. Specifically, we set the number of domain-independent features to 500, and the dimensionality of shared subspace to 100.

The classification accuracy is adopted as the evaluation criteria. It is defined as:

$$\text{Accuracy} = \frac{\left|\{\mathbf{x}|\mathbf{x} \in \mathcal{D}^{\ell}_{tst} \cap c^{\ell}(\mathbf{x}) = y(\mathbf{x})\}\right|}{\left|\{\mathbf{x}|\mathbf{x} \in \mathcal{D}^{\ell}_{tst}\}\right|},$$

where $\mathcal{D}^{\ell}_{tst}$ denotes test data, $y(\mathbf{x})$ is the ground truth and $c^{\ell}(\mathbf{x})$ is the predicted label. For evaluating the overall classification performance on all domains, we adopt the domain average accuracy as the evaluation measure. All experiments are run on a machine with a 2.4GHz Intel Xeon processor and 16G RAM. The average results of 20 random runs are reported.

5.4 Results and Discussions

5.4.1 Results on Sentiment and Newsgroups Datasets

In this section, we conduct experiments on the multi-domain classification tasks constructed from Multi-Domain Sentiment Dataset and 20Newsgroups. In the experiment on each task, we first randomly select 100 labeled instances from each domain to form an initial training set, and use the remaining data instances to form an unlabeled pool. Active learning is iteratively performed several iterations until the learner achieves a sufficient accuracy. In each iteration, every active learner labels 30 data instances from unlabeled pool and move them to the training set. Once the labeled instances are incorporated, each active learner re-trains classifiers on the expanded training set and its performance is evaluated on the remaining unlabeled instances.

Figure 1 shows the overall performance of each method on sentiment classification task **B**+**D**+**K**. As can be seen, MultiAL consistently outperforms each comparison method when increasing number of new labeled instances are added. This result suggests that our method can take advantage of the multi-domain structure for querying, and effectively optimize all domain classifiers together. From the figure, we can also observe that the transfer learning baselines SingleAL+ and Random+ perform much better than the non-transfer baselines SingleAL and Random, respectively. The improvement is large especially when only a few data instances are queried to be labeled. However, as more new labeled instances are added, the performance of SingleAL+ and SingleAL becomes close, while MultiAL constantly outperforms both SingleAL+ and SingleAL. In addition, MultiAL increasingly outperforms EmbedAL when the number of queried data instances increases. It implies that the classifier trained in the shared subspace alone may not be able to generalize well across different domains. The overall performance on other sentiment classification tasks is presented in Figure 2. From the figures, we can observe the similar trends on all tasks.

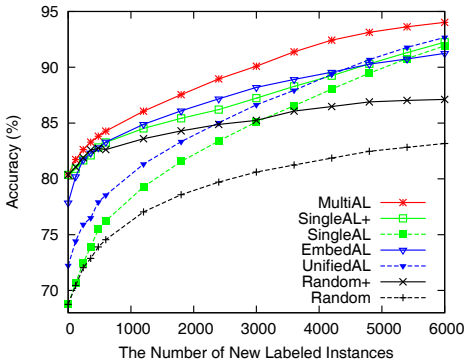
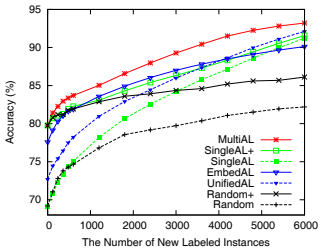
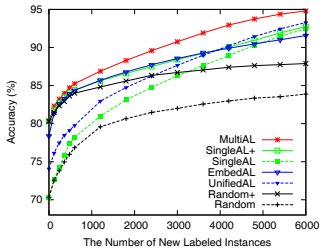


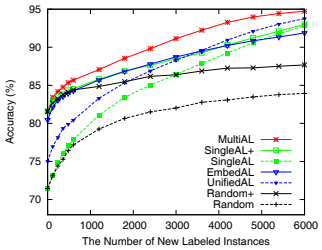
Figure 1: The sentiment classification results on **B**+**D**+**K**



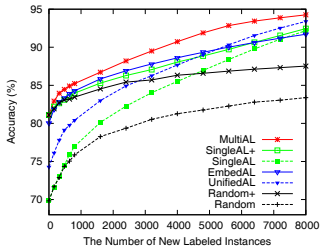
(a) Results on **B**+**D**+**E**



(b) Results on **B**+**E**+**K**



(c) Results on **D**+**E**+**K**



(d) Results on **B**+**D**+**E**+**K**

Figure 2: The sentiment classification results on the tasks constructed from Multi-Domain Sentiment Dataset

Table 4 summarizes the classification accuracy on each domain of task **B**+**D**+**K**. From the table, we can observe that MultiAL outperforms all comparison methods on each individual domain.

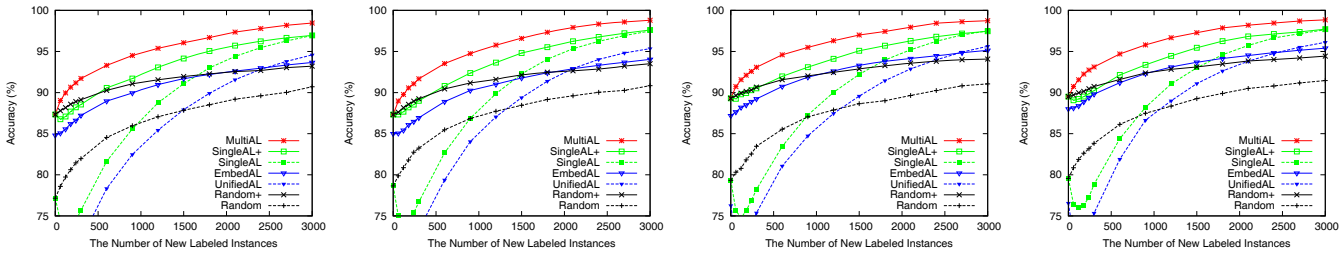
Furthermore, one more practical and interesting question is that how many human labeling efforts can be saved by using MultiAL? Table 5 shows how many new labeled instances

Table 4: The Classification Accuracy (%) on Task **B+D+K** after 4,000 New Labeled Instances Added

Domain	Random	Random+	UnifiedAL	EmbedAL	SingleAL	SingleAL+	MultiAL
Book	79.73±0.60	84.95±0.41	86.63±0.33	88.48±0.34	85.19±0.25	86.62±0.31	90.95±0.19
DVD	79.81±0.50	85.01±0.47	87.42±0.80	88.12±0.10	85.85±0.29	87.56±0.23	90.33±0.36
Kitchen	85.30±0.31	89.11±0.23	92.39±0.22	91.34±0.12	91.38±0.29	92.24±0.32	94.80±0.15
Average	81.62±0.26	86.36±0.32	88.81±0.33	89.31±0.10	87.47±0.12	88.81±0.14	92.03±0.14

Table 5: The Number of New Labeled Instances Needed for Each Learner to Achieve 90% Accuracy on Sentiment Classification

Task	Random	Random+	UnifiedAL	EmbedAL	SingleAL	SingleAL+	MultiAL
B+D+E	>6,000	>6,000	4,860	5,820	5,460	5,100	3,360
B+D+K	>6,000	>6,000	4,500	4,560	5,100	4,680	3,000
B+E+K	>6,000	>6,000	4,140	4,320	4,680	4,200	2,700
D+E+K	>6,000	>6,000	3,780	4,020	4,560	4,080	2,520
B+D+E+K	>8,000	>8,000	5,440	5,600	6,560	5,920	3,600
Average	>6,400	>6,400	4,544	4,864	5,272	4,796	3,036



(a) Results on **NG-123** (b) Results on **NG-124** (c) Results on **NG-134** (d) Results on **NG-234**

Figure 3: The newsgroup classification results on the tasks constructed from 20Newsgroups

Table 6: The Number of New Labeled Instances Needed for Each Learner to Achieve 95% Accuracy on Newsgroup Classification

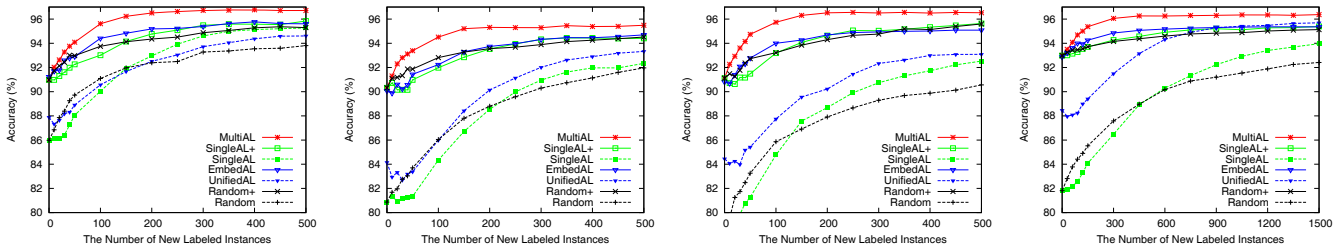
Task	Random	Random+	UnifiedAL	EmbedAL	SingleAL	SingleAL+	MultiAL
NG-123	>3,000	>3,000	>3,000	>3,000	2,280	1,800	1,080
NG-124	>3,000	>3,000	2,820	>3,000	2,040	1,620	1,020
NG-134	>3,000	>3,000	2,760	2,880	2,100	1,500	780
NG-234	>3,000	>3,000	2,520	2,580	1,920	1,380	720
Average	>3,000	>3,000	>2,775	>2,865	2,085	1,575	900

are needed for each active method to achieve a satisfactory classification accuracy (i.e., 90%). From Table 5, we can find that MultiAL saves at least 33.2% labeling efforts on average compared with all comparison methods. For example, on task **B+D+E+K**, MultiAL only needs to label 3,600 data instances to achieve 90% classification accuracy, while the best active learning baseline UnifiedAL requires 5,440 new labeled instances. The random query methods Random and Random+ cannot achieve the desired classification accuracy even if 8,000 new labeled instances are added. The result suggests that our method can effectively save the redundant labeling efforts by optimizing the classifiers of all domains together.

In the following, we report the experiment results on the newsgroup classification tasks. Figure 3 illustrates the overall performance on the four multi-domain newsgroup classification tasks. As can be seen from the figures, MultiAL consistently outperforms the comparison methods on all tasks. In addition, we can find that UnifiedAL always performs worse than SingleAL. An explanation is that when the domain gap is large, the query strategy would be affected by the inherent difference between domains. Table 6 shows the number of new labeled instances needed for each method to achieve 95% newsgroup classification accuracy. As presented in the table, MultiAL saves more than 42.9% labeling efforts compared with all comparison methods.

5.4.2 *Results on Email Spam Filtering Dataset*

In this section, we discuss our experiments on the email spam filtering dataset. The experiments are conducted in the inductive setting. For each task, we randomly select 10 emails of each user to form an initial training set, and select 100 emails of each user to form a test set. The remaining emails are used to form an unlabeled pool. Active learning is iteratively performed several iterations with 5 new labeled emails are added per iteration. Figures 4 shows the classification results on the test sets for the four spam filtering tasks. As illustrated in the figures, the accuracy curves of MultiAL grow very fast and achieve satisfactory performance after a few iterations. But other methods need much more querying iterations to obtain the comparable performance. Table 7 shows how many new labeled instances are needed for each method to achieve 95% classification accuracy on each task. From the table, we can observe that MultiAL saves more than 68.7% labeling efforts on average compared with all baseline methods.



(a) Results on **u00-u04** (b) Results on **u05-u09** (c) Results on **u10-u14** (d) Results on **u00-u14**

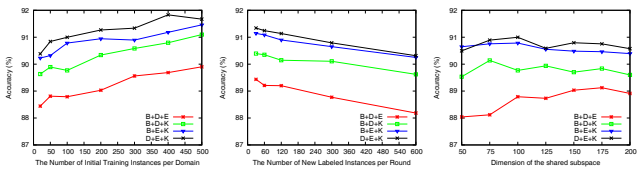
Figure 4: The email spam filtering results on the tasks constructed from Email Spam Filtering Dataset

Table 7: The Number of New Labeled Instances Needed for Each Learner to Achieve 95% Accuracy on Email Spam Filtering

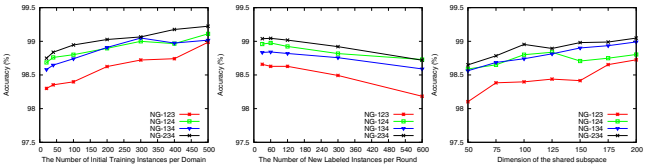
Task	Random	Random+	UnifiedAL	EmbedAL	SingleAL	SingleAL+	MultiAL
u00-u04	>500	310	>500	190	360	230	80
u05-u09	>500	>500	>500	>500	>500	>500	150
u10-u14	>500	340	>500	290	>500	250	70
u00-u14	>1,500	1,140	780	360	>1,500	720	120
Average	>750	>572	>570	>335	>715	>425	105

5.4.3 *Parameter Sensitivity Analysis*

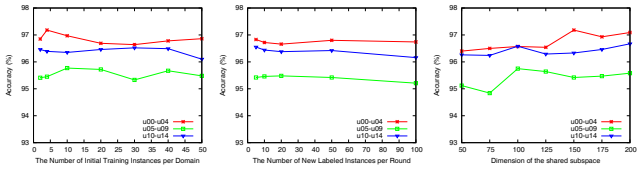
There are two important parameters for active learning methods: 1) the size of initial training set, and 2) the number of queried instances per iteration. In addition, the dimensionality of the shared space is an important parameter of the multi-domain classification method in our proposed framework. In this section, we test the sensitivity of these parameters. When testing a specific parameter, we fix other parameters and vary the value of the parameter of our interest. For example, when testing the influence of initial training set size, we set the dimensionality of shared subspace to 100 for all tasks, and set the number of queried instances per iteration to 300 and 5 for the sentiment/newsgroup classification and the spam filtering tasks, respectively.



(a) Parameter sensitivity on the sentiment datasets



(b) Parameter sensitivity on the newsgroups datasets



(c) Parameter sensitivity on the email spam datasets

Figure 5: Parameter sensitivity of: 1) numbers of initial training instances per domain, 2) numbers of new labeled instances per iteration, 3) dimensionality of shared subspaces

Figure 5 shows the parameter sensitivity of our method after 3000, 3000 and 500 new labeled data instances are added on the sentiment classification, the newsgroup classification and the spam filtering tasks, respectively. As presented in the figures, the performance of MultiAL is stable and consistent under different parameter settings. In general, the performance of MultiAL improves when more initial training data instances are available. And when the total number of queried instances are fixed, the performance of MultiAL drops when the number of queried instances per iteration increases. The reason may be that for each iteration, the more instances the active learner queries, the more duplicate information the active learner may get. Finally, we can also observe that MultiAL works well and stably when the dimensionality of the shared subspace ranges from 50 to 200.

5.4.4 *Scalability*

In this section, we investigate the scalability of the proposed method. In our experiment, we use the re-sampling strategy on both Multi-Domain Sentiment Dataset and 20Newsgroups to construct a set of data pools for experiments. In the experiment, we fix the number of iterations to 10, and fix the number of queried instances per iteration to 400. Figure 6 demonstrates the different running time when the size of unlabeled data pool varying from 20,000 to 1,000,000. From the figure, we can observe that the running time linearly increases under varying sizes of the unlabeled data pool. The result suggests that our proposed method is efficient and capable of dealing with large-scale applications.

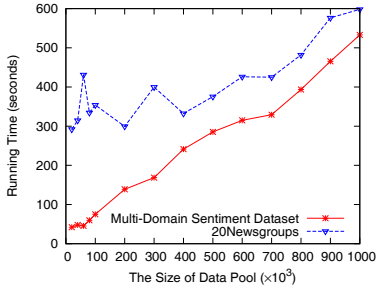


Figure 6: Running time under different size of data pool

6. CONCLUSIONS AND FUTURE WORK

In this work, we aim to solve a novel active learning problem for building classifiers of multiple domains simultaneously. Different from conventional active learning algorithms which focus on improving a single domain classifier, the proposed method aims to query the data instance which can not only improve the classifier of its corresponding domain but also improve the classifiers of other domains. The experiment results on three real-world applications show that our method respectively reduce the human labeling efforts by 33.2%, 42.9% and 68.7% on these applications. In addition, the proposed approach has been verified to be efficient and easily applied to large-scale applications. In the future, we plan to extend our work in the following directions: 1) In this work, we use a score function to rank unlabeled data instances for querying. However, this criteria can be biased by some data instances which contain rare patterns and are far away from existing labeled instances. It is not clear whether we can correct such label-sampling bias with importance weighting. 2) Given a large number of domains, some features may be shared by a subset of domains instead of all domains. It is interesting to jointly query instances under a hierarchical structure among domains. 3) With the increasing number of new labeled instances, it would be helpful to re-build the shared subspace after each iteration of active learning. 4) How to apply our active learning framework to other classification methods is also an interesting problem.

7. ACKNOWLEDGMENT

The work was supported by National Natural Science Foundation of China (60973103,90924003) and HGJ National Key Project (2010ZX01042-002-002).

References

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, Dec. 2005.

[2] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of ICML*, pages 49–56, Montreal, Quebec, Canada, 2009. ACM.

[3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, Prague, Czech Republic, 2007. ACL.

[4] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of EMNLP*, pages 120–128, Sydney, Australia, 2006. ACL.

[5] N. Cebron and M. R. Berthold. Active learning in parallel universes. In *Proceedings of CIKM*, pages 1621–1624, New York, NY, USA, 2010. ACM.

[6] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of ICML*, pages 193–200, Corvalis, Oregon, USA, 2007. ACM.

[7] P. Donmez and J. G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceeding of CIKM*, pages 619–628, Napa Valley, California, USA, 2008. ACM.

[8] M. Dredze, A. Kulesza, and K. Crammer. Multi-domain learning by confidence-weighted parameter combination. *Mach. Learn.*, 79(1-2):123–149, May 2010.

[9] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of KDD*, pages 109–117, Seattle, WA, USA, 2004. ACM.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

[11] A. Harpale and Y. Yang. Active learning for multi-task adaptive filtering. In *Proceedings of ICML*, pages 431–438. Omnipress, 2010.

[12] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML*, pages 137–142, Berlin, Germany, 1998. Springer.

[13] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR*, pages 3–12, Dublin, Ireland, 1994. Springer-Verlag New York.

[14] T. M. Mitchell. Generalization as search. *Artif. Int.*, 18(2):203 – 226, 1982.

[15] I. Muslea, S. Minton, and C. A. Knoblock. Active learning with multiple views. *J. Artif. Int. Res.*, 27(1):203–233, Oct. 2006.

[16] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of AAAI*, pages 677–682, Chicago, Illinois, USA, 2008. AAAI.

[17] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of WWW*, pages 751–760, Raleigh, North Carolina, USA, 2010. ACM.

[18] P. Rai, A. Saha, H. Daumé, III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010*, pages 27–32, Los Angeles, California, USA, 2010. ACL.

[19] R. Reichart, K. Tomanek, U. Hahn, and A. Rappoport. Multi-task active learning for linguistic annotations. In *Proceedings of ACL-08: HLT*, pages 861–869, Columbus, Ohio, USA, June 2008. ACL.

[20] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of ICML*, pages 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[21] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[22] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of COLT*, pages 287–294, Pittsburgh, Pennsylvania, USA, 1992. ACM.

[23] X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *Proceedings of ECML/PKDD*, pages 342–357, Antwerp, Belgium, 2008. Springer-Verlag.

[24] S. Tong. *Active learning: theory and applications*. PhD thesis, 2001.

[25] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, Mar. 2002.

[26] S. Xie, W. Fan, J. Peng, O. Verscheure, and J. Ren. Latent space domain transfer between high dimensional overlapping distributions. In *Proceedings of WWW*, pages 91–100. ACM, 2009.

[27] Y. Zhang. Multi-task active learning with output constraints. In *Proceedings of AAAI*, Atlanta, Georgia, USA, 2010. AAAI.