Spectral Domain-Transfer Learning

Xiao Ling[†] Wenyuan Dai[†] Gui-Rong Xue[†] Qiang Yang[‡] Yong Yu[†]

[†]Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
{shawnling, dwyak, grxue, yyu}@apex.sjtu.edu.cn

[‡]Hong Kong University of Science and Technology, Clearway Bay, Kowloon, Hong Kong, China
qyang@cse.ust.hk

ABSTRACT

Traditional spectral classification has been proved to be effective in dealing with both labeled and unlabeled data when these data are from the same domain. In many real world applications, however, we wish to make use of the labeled data from one domain (called *in-domain*) to classify the unlabeled data in a different domain (*out-of-domain*). This problem often happens when obtaining labeled data in one domain is difficult while there are plenty of labeled data from a related but different domain. In general, this is a *transfer learning* problem where we wish to classify the unlabeled data through the labeled data even though these data are not from the same domain. In this paper, we formulate this domain-transfer learning problem under a novel spectral classification framework, where the objective function is introduced to seek consistency between the in-domain supervision and the out-of-domain intrinsic structure. Through optimization of the cost function, the label information from the in-domain data is effectively transferred to help classify the unlabeled data from the out-of-domain. We conduct extensive experiments to evaluate our method and show that our algorithm achieves significant improvements on classification performance over many state-of-the-art algorithms.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Spectral learning methods such as normalized cut [28] are increasingly being applied to many learning tasks such as document clustering and image segmentation. Exploiting the information in the eigenvectors of a data similarity matrix to find the intrinsic structure, spectral methods have been extended from unsupervised learning to supervised/semi-supervised learning [22, 19], where a unified framework is used for spectral classification (SC). The SC algorithm has been shown to be effective when the data consist of both labeled and unlabeled data.

However, a limitation of these traditional SC methods is that they only focus on the scenario that the labeled and unlabeled data are drawn from the same domain, i.e., with the same bias or feature space. Unfortunately, many scenarios in the real world do not follow this requirement. In contrast to these methods, in this paper, we aim at extending traditional spectral methods to tackle the classification problem when labeled and unlabeled data come from different domains. There are several reasons for why it is important to consider this *domain-transfer learning* problem, which is an instance of *transfer learning* [27, 30, 7]. First, the labeled information is often scarce in a target domain, while a lot of available labeled data may exist from a different but related domain. In this case, it would be desired to make maximal use of the labeled information, though their domains are different. For example, suppose that our task is to categorize some text articles, where the labeled data are Web pages and the unlabeled data are Blog entries. This task is important in practice, since there are much fewer labeled Blog articles than Web pages. These two kinds of articles may share many common terms, but the statistical observations of words may be quite different, as blog articles tend to use informal words. Second, the data distribution in many domains changes with time. Thus, classifiers trained during one time period may not be applicable to another time period again. Take spam email filtering for an example. The topics of spam/ham emails often evolve with time. Therefore the labeled data may fall into one set of topics whereas the unlabeled data other topics. Because traditional SC algorithms often fail to generalize across different domains, we must design new ways to deal with the cross-domain classification problem.

This paper focuses on transferring spectral classification models across different domains. Formally speaking, the training data are from a domain $\mathcal{D}_{in}$ and the test data are from another domain $\mathcal{D}_{out}$. $\mathcal{D}_{in}$ is called *in-domain* and $\mathcal{D}_{out}$ *out-of-domain* in order to highlight the crossing of the domains where the label set is the same. In addition, it is assumed that in-domain $\mathcal{D}_{in}$ and out-of-domain $\mathcal{D}_{out}$ are related to make the domain-transfer learning feasible. Our objective is to classify the test data from out-of-domain $\mathcal{D}_{out}$ as accurately as possible using the training data from in-domain $\mathcal{D}_{in}$.

Although several cross-domain classification algorithms have been proposed, e.g., [10, 14], they are all based on local

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

optimization. When the labeled and unlabeled data are not sufficiently large, their optimization function may have a lot of local minima and bring much difficulty for classification. In this paper, a spectral domain-transfer learning method is proposed, where we design a novel cost function from normalized cut, so that the in-domain supervision is regularized by out-of-domain structural constraints. By optimizing this cost function, two objectives are simultaneously being followed. On one hand, we seek an optimal partition of the data that respect the label information, where the labels are considered in the form of must-link constraints [31]; that is, the corresponding data points with respect to each constraint must be with the same label. On the other hand, the test data are split as separately as possible in terms of the cut size within the test set, which will facilitate the classification process. To sum up, the supervisory knowledge is used to ensure the correctness when searching for the optimal cut of all data points. At the same time, the data points in the test set are also separated with small cut sizes. To achieve this aim, a regularization form is introduced to combine both considerations, resulting in an effective transferring of the labeled knowledge towards out-of-domain $\mathcal{D}_{out}$.

We set out to test our proposed algorithm for domain-transfer learning empirically, where our algorithm is referred as the Cross-Domain Spectral Classifier (abbreviated by CDSC). In our experiments, we set up eleven domain-transfer problems to evaluate our method. Compared against several state-of-the-art algorithms, our method achieves great improvements on the competent methods.

The rest of this paper is organized as follows. Spectral methods are reviewed in Section 2. Section 3 dives into details of our method. In Section 4, our method is evaluated compared with other classifiers. Following related work discussed in Section 5, Section 6 concludes this paper with some future work discussion.

2. PRELIMINARIES ON SPECTRAL METHODS

Spectral clustering is aimed at minimizing the inter-cluster similarity and maximizing the intra-cluster connection. Several criteria were proposed to quantify the objective function, such as Ratio Cut [8], Normalized Cut (NCut) [28], Min-Max Cut (MCut) [15]. Using graph theory terminology, the data are modeled as vertices and the edges are valued using the similarity of the endpoints. We denote V as the universe of all examples and $V = A \cup B$ where $\{A, B\}$ is a partition of V. The goal is to find a partition that optimizes the cost function as follows:

$$F_{\text{RatioCut}} = \frac{\text{cut}(A,B)}{|A|} + \frac{\text{cut}(A,B)}{|B|}\,,$$
$$F_{\text{NCut}} = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)}\,,$$
$$F_{\text{MCut}} = \frac{\text{cut}(A,B)}{\text{assoc}(A)} + \frac{\text{cut}(A,B)}{\text{assoc}(B)}\,.$$

Here, $\text{assoc}(A,V) = \sum_{i\in A, j\in V} w_{ij}$, $\text{assoc}(A) = \text{assoc}(A,A)$, $\text{cut}(A,B) = \sum_{i\in A, j\in B, A\cap B=\emptyset} w_{ij}$, where w_{ij} represents the similarity between data points i and j. Take normalized cut as an example. The numerator $\text{cut}(A,B)$ measures how loosely the set A and B are connected, while the denominator $\text{assoc}(A,V)$ measures how compact the entire data set is. [28] presents its equivalent objective in matrix representation as

$$F_{\text{NCut}} = \frac{y^T(D-W)y}{y^T D y}\,,$$

where W is the similarity matrix, $D = diag(We)$ (e is a vector with all coordinates 1) and y is the indicator vector of the partition. Since solving the discrete-valued problem is NP-hard, y is relaxed to be continuous. Minimization of this cost function can be done via *Rayleigh quotient* [16]. Given a Laplacian ($L = D - W$) of a graph, the second smallest eigenvector y_1 meets the optimization constraint [9]. As to the discretization, linear order search [28] and other variant search methods (e.g. linkage differential order [15]) are commonly used to derive the cluster membership. Another approach was proposed in [25] which first normalizes the eigenvectors and then applies the K-Means clustering method.

3. CROSS-DOMAIN SPECTRAL CLASSIFICATION

3.1 Problem Definition

For conciseness and clarity, in this paper we mainly focus on binary classification on textual data across different domains. Extensions can be easily done for more classes and other domains. Two document sets S_{in} and S_{out} are collected from domains $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$, respectively. We also denote $S = S_{in} \cup S_{out}$. In the binary classification setting, the label set is $\{+1, -1\}$, meaning that $c(\mathbf{d}_i)$ equals $+1$ (positive) or -1 (negative) where $c(\mathbf{d}_i)$ is $\mathbf{d}_i$'s true class label. The objective is to find the hypothesis h which satisfies $h(\mathbf{d}_i) = c(\mathbf{d}_i)$ for as many $\mathbf{d}_i \in S_{out}$ as possible.

3.2 Objective Function

In our approach, the main idea is to regularize two objectives, namely, minimizing the cut size on all the data with the least inconsistency of the in-domain data, and at the same time maximizing the separation of the out-of-domain data. Intuitively, the regularization is regarded as the balance between the in-domain supervision and the out-of-domain structure.

3.2.1 Supervision from In-domain

Let $n = |S|$ be the size of the whole sample. A similarity matrix $W_{n\times n}$ is calculated according to a certain similarity measure. Then, the supervisory information is incorporated in the form of must-link constraints by building a constraint matrix U, described in more details in the next subsection. In order to measure the quality of a partition, the cost function for all the data is defined as

$$F_1 = \frac{x^T(D-W)x}{x^T D x} + \beta||U^T x||^2\,, \qquad (1)$$

where $D = diag(We)$ is defined as previously mentioned and x is the indicator vector of the partition. In Equation (1), the normalized cut is adopted for the first term and a penalty term $\beta||U^T x||^2$ is used to guarantee a good partition on the training data. The first term represents the association between two classes. The second term $\beta||U^T x||^2$ will constrain the partition of training data since any violation of constraints results in penalty regarding F_1 in Equation (1).

The parameter β controls the enforcement of constraints. This cost function is similar to that proposed in [19].

3.2.2 Structure of Out-of-domain Data

In Equation (1), F_1 mainly focuses on the labeled data. However, we wish to classify the out-of-domain test data correctly. Thus, it is important to find the optimal partition for the test data as well. The cost function for the test data alone is defined as

$$F_2 = \frac{x^T(D_s - W_s)x}{x^T D_s x}\,, \tag{2}$$

where $D_s = diag(W_s e)$, and W_s is the similarity matrix for test data only. Note that the dimension of W_s is n, similarity entries only within test data are kept, i.e. if node i and j are both in the test data then $W_{s\,(ij)} = W_{(ij)}$; other entries are set to zero.

3.2.3 Integrating In-domain and Out-of-domain via Regularization

Now a regularization parameter is introduced, incorporating Equation (1) and Equation (2) to get the unified cost function for cross-domain classification:

$$\begin{aligned} F_{CDSC} &= F_1 + \lambda F_2 \qquad (3)\\ &= \frac{x^T(D-W)x}{x^T D x} + \beta||U^T x||^2 + \lambda\frac{x^T(D_s - W_s)x}{x^T D_s x}, \end{aligned}$$

where λ is a tradeoff parameter for balancing the supervisory information (Equation (1)) and the cut size of the test data (Equation (2)). The first term F_1 ensures a good classification model should maximize the correctness of labeled data. In the domain-transfer setting, we cannot completely rely on the in-domain data. The second term F_2 can be understood as the domain-transfer fitting constraint, which means a good classification model should also keep the test data with adequately good separation. The tradeoff between these competing conditions is captured by the parameter λ, which interestingly, allows the classification model to be balanced between in-domain $\mathcal{D}_{in}$ and out-of-domain $\mathcal{D}_{out}$. In Equation (3), when $\lambda = 0$, the overall cost function degenerates into a spectral cost function over all the data in a semi-supervised manner; when λ is large enough, the overall objective is biased towards optimizing only the spectral cost function for the test data without any supervisory knowledge.

3.3 Incorporating Constraints

In Equation (3), a penalty for violations [31] of the supervisory constraints is introduced. In the binary classification setting, assume there are n_1 positive data and n_2 negative data in the training set. The constraint matrix U is constructed as follows:

$$U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]\,, \tag{4}$$

where each $\mathbf{u}_i$ is an n-dimentional vector (same row index as W) with two non-zero entries. Each column $\mathbf{u}_k$ has an entry of $+1$ in the ith row, -1 in the jth row and the rest are all zero, which represents a pairwise constraint (data i and data j must be with the same label). Therefore U has $m = n_1 \times (n_1-1)/2 + n_2 \times (n_2-1)/2$ columns (constraints).

The detailed construction of the constraint matrix U is presented in Algorithm 1. It is easily seen with the indicator vector x that

$$U^T x = 0\,, \tag{5}$$

when x satisfies all the constraints. Adding this constraint component into the Normalized Cut criterion [28], the cost function becomes Equation (1).

One problem of the constraint matrix U is that the matrix U (with m rows) is greatly oversized, which makes it hard to compute $U' = UU^T$ in Equation (3) ($||U^T x||^2 = x^\top UU^\top x$). To alleviate this oversize problem, U' can be directly built by considering the pairwise property of the constraints. Notice that U'_{ij} is the inner product of ith row and jth row of U. Then U'_{ij} has four cases:

$$U'_{ij} = \begin{cases} n_1 - 1, & i = j \text{ and they are both in positive class;}\\ n_2 - 1, & i = j \text{ and they are both in negative class;}\\ -1, & i \neq j \text{ and } i,\, j \text{ are in the same class;}\\ 0, & \text{otherwise.} \end{cases}$$

where n_1 is the size of positive data and n_2 is the size of negative data.

Algorithm 1 FormConstraintMatrix

Input : the size of positive data n_1, the size of negative data n_2 and $n = n_1 + n_2$; here, without loss of generality, we assume the first n_1 examples are positive, and the next n_2 examples are negative.
Output : Constraint Matrix U
Let U be a $n \times \frac{n_1(n_1-1)+n_2(n_2-1)}{2}$ matrix.
Let $colNum = 1$.
Construct the matrix column by column.
for $i \leftarrow 1$ to n_1 **do**
  for $j \leftarrow i+1$ to n_1 **do**
    $U(i, colNum) = 1$
    $U(j, colNum) = -1$
    $colNum = colNum + 1$
  end for
end for
for $i \leftarrow n_1 + 1$ to $n_1 + n_2$ **do**
  for $j \leftarrow i+1$ to $n_1 + n_2$ **do**
    $U(i, colNum) = 1$
    $U(j, colNum) = -1$
    $colNum = colNum + 1$
  end for
end for
return U

3.4 Optimization

In this section, the optimization of the overall function (Equation (3)) is addressed.

Since Equation (3) is difficult to optimize, we have to seek an approximation. In this work, we use $\frac{x^T(D_s-W_s)x}{x^T Dx}$ instead of $\frac{x^T(D_s-W_s)x}{x^T D_s x}$ in F_{CDSC}. Usually, $\frac{x^T(D_s-W_s)x}{x^T Dx}$ might mislead the normalized cut on S_{out}. However, in F_{CDSC}, when F_1 is sufficiently optimized, the partition of in-domain training data will be more or less balanced due to the constraint $\beta||U^T x||^2$, and thus the balancing functionality of the denominator $x^T Dx$ is reduced on only out-of-domain test data

(refer to $x^T D_s x$). Then, we have

$$F_{CDSC} \approx \frac{x^T(D-W)x}{x^T Dx} + \beta||U^T x||^2 + \lambda\frac{x^T(D_s-W_s)x}{x^T Dx}$$
$$= \frac{x^T[(D-W)+\lambda(D_s-W_s)]x}{x^T Dx} + \beta||U^T x||^2 . \quad (6)$$

The similarity matrix is thus modified by amplifying the similarity inside the test data submatrix. In the interpretation through random walk [24], this modification can be seen as increasing the transition probability inside the test data.

Replacing $y^T = x^T D^{1/2}/||x^T D^{1/2}||$,

$$\frac{x^T(D-W)x}{x^T Dx} = y^T D^{-1/2}(D-W)D^{-1/2}y . \quad (7)$$

Similarly,

$$\frac{x^T(D_s-W_s)x}{x^T Dx} = y^T D^{-1/2}(D_s-W_s)D^{-1/2}y . \quad (8)$$

With Equation (5),

$$U^T D^{-1/2} y = 0 . \quad (9)$$

Combining Equations (7), (8) and (9), we obtain

$$\begin{aligned} & F_{CDSC} \\ &= \frac{x^T[(D-W)+\lambda(D_s-W_s)]x}{x^T Dx} + \beta||U^T x||^2 \\ &= y^T D^{-1/2}[(D-W+\lambda(D_s-W_s)]D^{-1/2}y \\ &\quad + \beta||U^T D^{-1/2} y||^2 \\ &= y^T D^{-1/2}[(D-W)+\beta UU^T+\lambda(D_s-W_s)]D^{-1/2}y \\ &= \frac{x^T[(D-W)+\beta UU^T+\lambda(D_s-W_s)]x}{x^T Dx} \\ &= \frac{x^T Tx}{x^T Dx} , \end{aligned} \quad (10)$$

where $T = (D-W)+\beta UU^T+\lambda(D_s-W_s)$. Then, F_{CDSC} can be minimized by solving an eigen-system:

$$Tx = dDx , \quad (11)$$

where d is the eigenvalue. Moreover, Equation (11) can also be rewritten into

$$D^{-1/2}TD^{-1/2}y = dy . \quad (12)$$

Similar to other spectral methods, y is relaxed to be a real-valued vector. To this end, our problem has been transformed into the minimization of $\frac{x^T(D^{-1/2}TD^{-1/2})x}{x^T x}$, which is called *Rayleigh Quotient*. In [16], we have

LEMMA 1 (RAYLEIGH QUOTIENT). *Let A be a real symmetric matrix. Under the constraint that x is orthogonal to the $j-1$ smallest eigenvectors $x_1, \ldots, x_{j-1}$, the quotient $\frac{x^T Ax}{x^T x}$ is minimized by the next smallest eigenvector x_j and its minimum value is the corresponding eigenvalue d_j.*

Furthermore, we can prove

LEMMA 2. *$T' = D^{-1/2}TD^{-1/2}$ is symmetric and its eigenvectors are orthogonal.*

PROOF. Since $D-W$, D_s-W_s and UU^T are all symmetric, $T = (D-W)+\beta UU^T+\lambda(D_s-W_s)$ is therefore symmetric. With the diagonal matrix $D^{-1/2}$, T' is also symmetric.

Specifically, let $\mathbf{v}, \mathbf{w}$ be arbitrarily two different eigenvectors of T and d_v, d_w be corresponding eigenvalues which are thus different.

$$d_v \mathbf{v}^T\mathbf{w} = (T'\mathbf{v})^T\mathbf{w} = \mathbf{v}^T(T'\mathbf{w}) = d_w\mathbf{v}^T\mathbf{w} .$$

Since $d_v \neq d_w$, $\mathbf{v}^T\mathbf{w}$ should be equal to 0. This implies that $\mathbf{v}$ and $\mathbf{w}$ are orthogonal. □

By Lemma 1 and Lemma 2, the k smallest orthogonal eigenvectors of $T' = D^{-1/2}TD^{-1/2}$ are used after row normalization. Each data point is represented by the corresponding row.

Algorithm 2 Cross-Domain Spectral Classification

Input : training data (n_1 positive instances, n_2 negative instances and $n = n_1 + n_2$) and test data, parameters $\{\lambda, \beta, k\}$ and a reasonable classifier $\mathcal{F}$.

Output : class predictions for test data

1: Construct the similarity matrix $W_{n\times n}$ given both training and test data and W_s for only test data, where n to be the number of all the data.
2: Let $D = diag(We)$, $D_s = diag(W_s e)$ and $U =$ **FormConstraintMatrix**(n_1, n_2, n).
3: Find the k smallest eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k$ of $T' = D^{-1/2}(D-W+\beta UU^T+\lambda(D_s-W_s))D^{-1/2}$ and construct a matrix $X = D^{-1/2}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k)$.
4: Normalize X by row into Y where $Y_{ij} = X_{ij}/\sqrt{\sum_{l=1}^k X_{il}^2}$.
5: Call $\mathcal{F}$ with input of the eigenvectors to obtain the classification result.

In Algorithm 2, we firstly prepare the data matrix and constraint matrix. Then the cost function is optimized by solving an eigen-system (Equation (11)). Finally, we use a traditional classifier for the final prediction, which is similar to the procedure in [22].Empirically, our algorithm improves several other state-of-the-art classifiers as will be shown in the experiment part (Section 4).

The major computational cost of the above algorithm is for computing the eigenvectors. The eigenvectors can be obtained by Lanczos method, whose computational cost is proportional to the number of nonzero elements of the target matrix. Thus the cost of our algorithm is $O(kN_L nnz(T'))$, where k denotes the number of eigenvectors desired, N_L is the number of Lanczos iteration steps and $nnz(T')$ is the number of non-zero entries in T'.

3.5 Case Study

Figure 1 plots the *rec* vs *talk* data (data details will be presented in Section 4.1) represented by the two smallast eigenvectors using our algorithm CDSC. The data points in the figure are sufficiently separated for classification since the eigenvectors contain the needed structural information. Moreover, the training and test data are similar in terms of Euclidean distance. In this way, the approximate decision boundary can be easily detected (the dashed line) and, as a result, good performance is obtained using our method.

	Data Set		Positive (250 in all)	Negative (250 in all)
SRAA	auto vs aviation	train	sim-auto	sim-aviation
		test	real-auto	real-aviation
	real vs simulated	train	real-aviation	sim-aviation
		test	real-auto	sim-auto
20NG	rec vs talk	train	rec.{autos, motorcycles}	talk.{politics.guns, politics.misc}
		test	rec.{sport.baseball, sport.hockey}	talk.{politics.mideast, religion.misc}
	rec vs sci	train	rec.{autos, sport.baseball}	sci.{med, space}
		test	rec.{motorcycles, sport.hockey}	sci.{crypt, electronics}
	comp vs talk	train	comp.{graphics, windows.x, sys.mac.hardware}	talk.{politics.mideast, religion.misc}
		test	comp.{os.ms-windows.misc, sys.ibm.pc.hardware}	talk.{politics.guns, politics.misc}
	comp vs sci	train	comp.{graphics,os.ms-windows.misc}	sci.{crypt, electronics}
		test	comp.{sys.mac.hardware, windows.x, sys.ibm.pc.hardware}	sci.{med, space}
	comp vs rec	train	comp.{graphics, sys.mac.hardware, sys.ibm.pc.hardware}	rec.{motorcycles, sport.hockey}
		test	comp.{os.ms-windows.misc, windows.x}	rec.{autos, sport.baseball}
	sci vs talk	train	sci.{electronics, med}	talk.{politics.misc, religion.misc}
		test	sci.{crypt, space}	talk.{politics.guns, politics.mideast}
Reuters	orgs vs places	train	orgs.{...}	places.{...}
		test	orgs.{...}	places.{...}
	people vs places	train	people.{...}	places.{...}
		test	people.{...}	places.{...}
	orgs vs people	train	orgs.{...}	people.{...}
		test	orgs.{...}	people.{...}

Table 1: The composition of all the data sets. Since there are too many subcategories in Reuters-21578, we omit the composition details of last three data sets here.

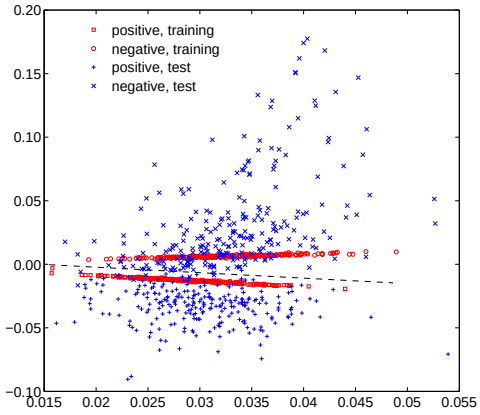


Figure 1: Projected data of *rec* vs *talk* in 2-dimensional eigen-space.

4. EXPERIMENTS

Our method is evaluated extensively on several data sets with the training and test data from different domains. As we will show later, our method outperforms several state-of-the-art classifiers in all the tasks.

4.1 Data Sets

The cross-domain data sets are generated in specific strategies using 20 Newsgroups[1], Reuters-21578[2] and SRAA[3]. The basic idea of our design is utilizing the hierarchy of the data sets to distinguish domains. Specifically, the task is defined as top-category classification. Each top category is split into two disjoint parts with different sub-categories, one for training and the other for test. Because the training and test data are in different subcategories, they are across domains as a result. To reduce the computational burden, we sampled 500 training and 500 test examples for each task.

[1]http://people.csail.mit.edu/jrennie/20Newsgroups/
[2]http://www.daviddlewis.com/resources/testcollections/
[3]http://www.cs.umass.edu/~mccallum/data/sraa.tar.gz

4.1.1 20 Newsgroups

The 20 Newsgroups is a text collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups nearly evenly. Six different data sets are generated for evaluating cross-domain classification algorithms. For each data set, two top categories[4] are chosen, one as positive and the other as negative. Then, the data are split based on sub-categories. Different sub-categories can be considered as different domains, while the task is defined as top category classification. The splitting strategy ensures the domains of labeled and unlabeled data related, since they are under the same top categories. Besides, the domains are also ensured to be different, since they are drawn from different sub-categories. Table 1 shows how the data sets are generated in our experiments.

4.1.2 Reuters-21578

Reuters-21578 is one of the most famous test collections for evaluation of automatic text categorization techniques. It contains 5 top categories. Among these categories, *orgs*, *people* and *places* are three big ones. For the category *places*, all the documents about the USA are removed to make the three categories nearly even, because more than a half of the documents in the corpus are in the USA sub-categories. Reuters-21578 corpus also has hierarchical structure. We generated three data sets *orgs vs people*, *orgs vs places* and *people vs places* for cross-domain classification in a similar way as what have been done on the 20 Newsgroups. Since there are too many sub-categories, the detailed description cannot be listed here.

4.1.3 SRAA

SRAA is a Simulated/Real/Aviation/Auto UseNet data set for document classification. 73,218 UseNet articles are collected from four discussion groups about simulated autos (*sim-auto*), simulated aviation (*sim-aviation*), real autos (*real-auto*) and real aviation (*real-aviation*). Consider the task that aims to predict labels of instances between *real* and *simulated*. The documents in *real-auto* and *sim-*

[4]Three top categories, *misc*, *soc* and *alt* are removed, because they are too small.

Data Set	Verification of Data Set			Traditional Classification				Cross-Domain Classification		
	$\mathcal{D}_{in}$–$\mathcal{D}_{out}$	$\mathcal{D}_{out}$–CV	$\mathcal{D}_{in}$–CV	SVM	TSVM	SGT	SC	CoCC	KDE	CDSC
real vs simulated	0.330	0.032	0.030	0.330	0.316	0.276	0.278	0.250	0.330	**0.188**
auto vs aviation	0.252	0.033	0.048	0.252	0.188	0.208	0.160	0.142	0.248	**0.120**
comp vs sci	0.380	0.012	0.016	0.380	0.334	0.428	0.270	0.192	0.380	**0.098**
rec vs talk	0.316	0.003	0.002	0.316	0.118	0.190	0.428	**0.092**	0.324	**0.092**
rec vs sci	0.234	0.007	0.003	0.234	0.162	0.160	0.192	0.160	0.234	**0.124**
sci vs talk	0.198	0.009	0.006	0.198	0.148	0.114	0.362	0.100	0.194	**0.044**
comp vs rec	0.142	0.008	0.003	0.142	0.104	0.044	0.086	0.090	0.142	**0.042**
comp vs talk	0.098	0.005	0.005	0.098	**0.024**	0.030	0.042	0.042	0.100	**0.024**
orgs vs people	0.306	0.106	0.020	0.306	0.294	0.288	0.276	**0.232**	0.298	**0.232**
orgs vs places	0.428	0.085	0.093	0.428	0.424	0.456	0.386	0.400	0.418	**0.318**
people vs places	0.262	0.113	0.017	0.262	0.256	0.216	0.230	0.226	0.262	**0.202**
average	0.268	0.038	0.022	0.268	0.215	0.219	0.246	0.175	0.266	**0.135**

Table 2: The error rate given by each classifier. Under the column "Verification of Data Set", "$\mathcal{D}_{in}$–$\mathcal{D}_{out}$" means training on in-domain $\mathcal{D}_{in}$ and testing on out-of-domain $\mathcal{D}_{out}$; "$\mathcal{D}_{out}$–CV" and "$\mathcal{D}_{in}$–CV" means 10-fold cross-validation on out-of-domain $\mathcal{D}_{out}$ and in-domain $\mathcal{D}_{in}$. Note that, the experimental results given by CoCC here are somewhat different from those presented in the original paper, since we sampled only 500 examples from each original data set.

auto are used as in-domain data, while *real-aviation* and *sim-aviation* as out-of-domain data. Then, the data set *real vs sim* is generated as shown in Table 1. Therefore all the data in the in-domain data set are about *auto*, while all the data in the out-of-domain set are about *aviation*. The *auto vs aviation* data set is generated in the similar way as shown in Table 1.

4.1.4 Verification of Data Sets

To verify our data design, the error rates are recorded using the SVM classifier in the scenario of domain-transfer learning ($\mathcal{D}_{in}$–$\mathcal{D}_{out}$) as well as the single-domain classification case within the out-of-domain and within the in-domain, respectively. Under the column "SVM" in Table 2, the three groups of classification results are displayed in the sub-columns. The column "$\mathcal{D}_{in}$–$\mathcal{D}_{out}$" means that the classifier is trained on in-domain data and tested on out-of-domain data. The next two columns "$\mathcal{D}_{out}$–CV" and "$\mathcal{D}_{in}$–CV" show the best results by the SVM classifier obtained during 10-fold cross validation. In these two experiments, the training and test data are extracted from the same domain, out-of-domain $\mathcal{D}_{out}$ and in-domain $\mathcal{D}_{in}$ respectively. Note that the error rates under the $\mathcal{D}_{in}$–$\mathcal{D}_{out}$ column is much worse than the ones under $\mathcal{D}_{out}$–CV and $\mathcal{D}_{in}$–CV. This implies that our data sets are not applicable for traditional classification.

4.2 Comparison Methods

To verify the effectiveness of our classifier, the supervised learner SVM is set as the baseline method. Our method is also compared to several semi-supervised classifiers, including Transductive SVM (TSVM) [20], Spectral Graph Transducer (SGT) [21] and Spectral Classifier (SC) [22]. Note that [22] is approximately a special case CDSC with $\lambda = 0$. We also compare to the co-clustering based classification (CoCC) [10] as the state-of-the-art domain-transfer learning algorithm and one representative selection bias correction (KDE) [29]. CoCC builds connection between in-domain and out-of-domain through feature clustering, and is formulated under the co-clustering framework. KDE corrects the domain bias in the in-domain, and then adapts the in-domain classification model to out-of-domain. We use test error rate as the evaluation measure.

4.3 Implementation Details

On the textual data designed in Section 4.1, we have conducted preprocessing procedures including tokenizing text into bag-of-words, converting text into low-case words, stopword removal and stemming using the Porter stemmer [26]. Each document $\mathbf{d}_i$ in S is represented by a feature vector using *Vector Space Model*. Each feature represents a term, which is weighted by its *tf-idf* value. Feature selection is carried out by thresholding Document Frequency [34]. In our experiments, Document Frequency threshold is set to 3, and the final result is not sensitive to it. The cosine similarity measure $\frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|}$ is adopted when constructing the similarity matrix.

The comparison methods are implemented by SVM^{light}[5] and SGT^{light}[6]. All parameters are set default by the software. The Spectral Classifier (SC) is implemented according to [22]. CoCC uses the same initialization and parameters in [10]. KDE is implemented according to [29, 35].

4.4 Experimental Results

4.4.1 Performance

By comparing with the traditional supervised classifier, it is observed that the cross-domain data present much difficulty in classification, where SVM (training on in-domain $\mathcal{D}_{in}$ and testing on out-of-domain $\mathcal{D}_{out}$) made more than 20% average prediction errors. In Table 2, we observe that the TSVM and SGT always outperformed the supervised classifier SVM. The semi-supervised classifiers worked better since they used the unlabeled data in the classification process, so that they captured more information in the out-of-domain. However, semi-supervised learning still works under the identical-domain assumption, and thus its improvement is limited. The situations are similar in SC. CoCC improves a lot over the traditional classification algorithm, since CoCC is a cross-domain classification algorithm, and it effectively transfers knowledge across different domains. KDE shows few improvement against SVM in our experiments, although it can effectively correct selection bias between two different domains. In our opinion, KDE fails to improve much in domain-transfer learning because the domain difference may be affected by the selection bias very few. In general, our algorithm CDSC is a spectral domain-transfer learning method, and achieves the best performance against all the comparison methods. Compared to the state-

[5]Software available at http://svmlight.joachims.org.
[6]Software available at http://sgt.joachims.org.

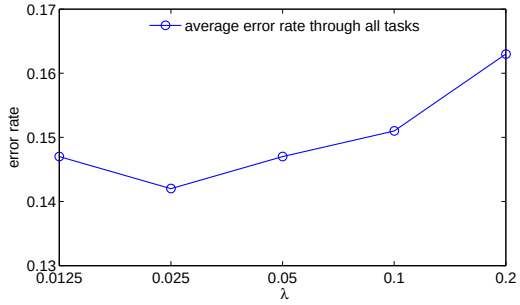


Figure 2: The average error rate curve of λ when fixing β at 15.

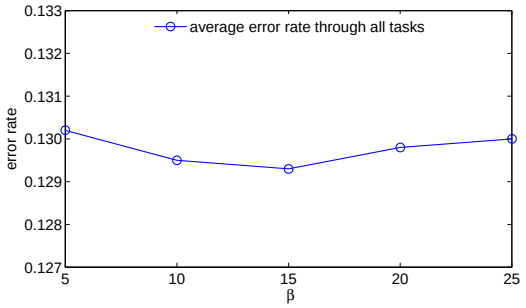


Figure 3: The average error rate curve of β when fixing λ at 0.025.

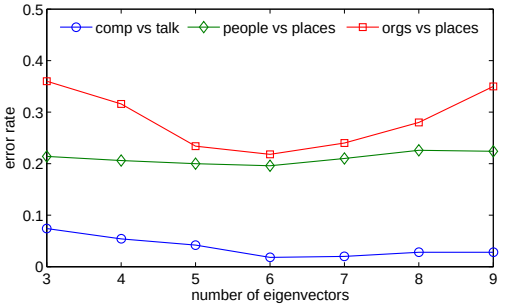


Figure 4: The error rates against the number of eigenvectors.

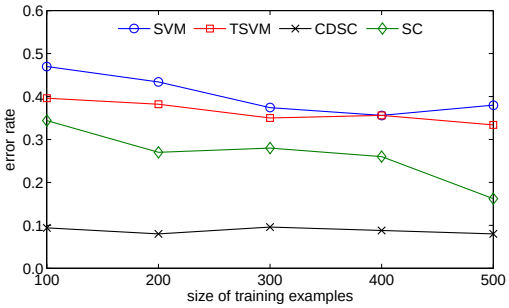


Figure 5: The error rate curve on the data set *comp* vs *sci* against different sizes of training examples.

of-the-art domain-transfer learning algorithm CoCC, CDSC also shows superiority in this experiments. We believe, it is because the data size in our experiment is not so large, and spectral learning is much more superior in learning with small data than many other learning methods.

However, in some data sets the performance is not satisfactory. For example, this can be observed in *orgs* vs *places*. This can be attributed to less common knowledge between in-domain and out-of-domain data. Our method requires that the in-domain and out-of-domain should be related, namely that they share some knowledge. If this condition cannot be satisfied, the quality of transferred knowledge will not be guaranteed. As to the tasks derived from the 20 Newsgroups, the in-domain and out-of-domain data may share a large amount of common knowledge which leads to better performance, despite the fact that other methods failed in most cases. In general, our algorithm can alleviate the classification difficulty better when the in-domain and out-of-domain are not the same albeit related.

4.4.2 Parameter Tuning

There are two parameters in our method: β adjusts the enforcement of supervisory constraints; λ represents the trade-off of transferring knowledge into the target domain. We tested 5 different values of β when λ is fixed. λ is enumerated from 0.0125 to 0.2 with 5 log-scale values with fixing β. We use the average error rate through 11 tasks for evaluation. From Figure 2, it can be seen that, empirically the best λ is between $[0.0125, 0.05]$, and we set $\lambda = 0.025$ in our experiments. From Figure 3, the performance of CDSC is not very sensitive to β, and we set $\beta = 15$ in the experiments.

4.4.3 Eigenvectors

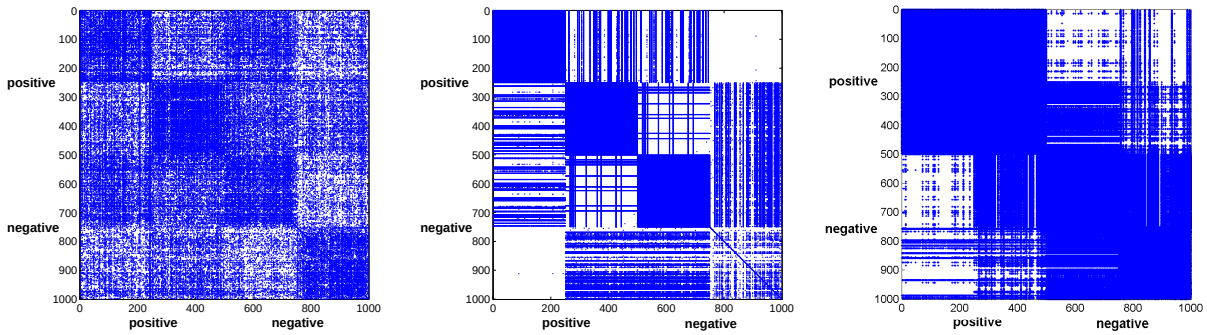
The eigenvectors obtained in the classification process represent the original information approximately in a different feature space. In this work, the optimal number is found by enumerating the number of used eigenvectors empirically. Figure 4 illustrates the error rates of several data sets against different numbers of eigenvectors used for classification. From the figure, it can be seen that, generally, the classification on 6 eigenvectors shows the best performance.

4.4.4 Varying the Size of the Training Data

We have also investigated the influence by the size of training examples. Take *comp* vs *sci* data set for example (Figure 5). We chose a portion of examples in the training data randomly ranging from 100 examples to all of the samples (500). We observe that SVM, TSVM and SC often performed, in general, increasingly worse when the number of training examples decreases. In contrast to these baselines, the error rate curve of our algorithm is generally stable. This indicates our algorithm CDSC can better deal with the data sparsity problem. More importantly, CDSC tops the performance over almost all trials.

4.4.5 Similarity Pattern

Spectral methods promise to draw the similar data points nearer by representing the original data in the eigen-space. But how does this projection work on cross-domain data? To answer this question, we illustrate the similarity pattern of the original data, the projected data in Spectral Classifier (SC) [22] and the projected data in our method (CDSC). Take the data set *rec* vs *talk* for example. The data are indexed firstly by category and secondly by training and test, namely positive training, positive test, negative training and negative test in order. Figure 6(a) displays the document-document similarity matrix of the original data valued by the cosine measure, which has a threshold by the mean of this matrix. The latter two patterns are similarly thresholded. In Figure 6(b), it is shown that SC fails to draw the



(a) original data (b) projected data by SC (c) projected data by CDSC

Figure 6: The similarity pattern on the data set *rec vs talk*. The data are indexed firstly by category and secondly by training and test, namely positive training, positive test, negative training and negative test in order. The document-document similarity matrix of the original data valued by the cosine measure, which has a threshold by the mean of this matrix.

data within the same category more similar. Figure 6(c) plots the similarity matrix of the projected data using our method. The projected data show more obvious block-like behavior within the same category. On the contrary to the SC pattern, the data from same category become similar while the data from different categories become dissimilar although the whole data are across domains. It is mainly attributed to our novel objective function, which also considers the out-of-domain separation. This block-like behavior indicates that the supervisory knowledge from another domain can be used directly and effectively. In this way, the classifier will find the decision boundary more easily and more accurately and hence perform better.

5. RELATED WORK

5.1 Spectral Methods

In addition to these unsupervised learning methods (described in Section 2), [19] developed a semi-supervised spectral clustering algorithm by incorporating the prior knowledge. In a supervised manner, [22] designs a spectral learning framework for classification. To represent the supervisory knowledge, the similarity between two same-label data points is set to 1. An one-nearest-neighbor classifier is applied to the data represented by eigenvectors. [22] also showed how to make spectral classification to achieve better performance by adding more labeled and unlabeled data. Compared to these methods in [25, 19], our method is derived from spectral clustering, but the eigenvectors are used for classification instead of clustering. A difference from [22] is that our method classifies the unlabeled data based on the label information from a different domain, while [22] focuses on learning within a single domain.

5.2 Transfer Learning

Transfer learning has been introduced to handle the learning problem where learning and prediction are in different scenarios. The idea of transfer learning is inspired by the intuition that humans often learn better when they have learned well in related domains. For instance, a good checker player may find it easier to learn to play chess. Previous works in transfer learning include "learning how to learn" [27] ,"learning one more thing" [30] and "multi-task learning" [7], which laid the initial foundations. [2] presented the notion of relatedness between learning tasks, which provided theoretical justifications for transfer learning. In our problem setting, we aim to accomplish the same task (i.e. learn with the same label set) in different domains, which is called *multi-domain* or *domain-transfer learning* – a special case of transfer learning.

The domain-transfer learning can be classified into two categories according to whether the out-of-domain supervision is given. [32] investigated how to exploiting auxiliary data in k-Nearest-Neighbors and SVM algorithm. They used the term "auxiliary data" to refer to the in-domain data and their experiments have demonstrated that the learning performance can be significantly improved with the help of auxiliary data. [14] utilized additional "in-domain" labeled data to train a statistical classifier under the *Conditional Expectation-Maximization* framework. Those "in-domain" data play a role as auxiliary data in tackling the scarcity of "out-of-domain" training data. In these works [32, 23, 14, 13, 12], auxiliary data serve as a supplement to the ordinary training data. In contrast to these works, our work focuses on the second category of domain-transfer learning, where the problem is classification *without* any training examples in the *out-of-domain*. Note that, in our problem, the in-domain and out-of-domain data are assumed to be relevant, in order to make the domain-transfer learning feasible. In the past, [10] proposed a co-clustering based algorithm to overcome the domain difference. In this paper, we use both in-domain supervision and out-of-domain structural information to handle the domain-transfer problem through spectral learning. As we showed in the experiments, our algorithm shows superiority over [10], when the data size is not sufficiently large. Other work includes [6, 1, 11, 33, 5].

Covariate shift [29] (or *sample selection bias* [35]) is a similar problem which occurs when samples are selected non-randomly. Originated from the Nobel-prize work in 2000, [17] made his contributions on correction of *sample selection bias* in econometrics. Recent researches on covariate shift include [35, 29, 18, 4, 3]. They used the instance weighting method to correct the bias. Although correcting sample selection bias [35] can solve the classification when training and test data are governed by different selection bias, it still mainly focuses on learning within a single domain. Our experiments in Section 4 show that correcting sample selection bias can only improve very little in domain-transfer learning.

6. CONCLUSION AND FUTURE WORK

In this paper, a novel spectral classification based method CDSC is presented where an objective function is proposed for domain-transfer learning. In the domain-transfer setting, the labeled data from the in-domains are available for training and the unlabeled data from out-of-domains are to be classified. Based on the normalized cut cost function, supervisory knowledge is transferred through a constraint matrix, and the regularized objective function (see Equation (10)) finds the consistency between the in-domain supervision and the out-of-domain intrinsic structure. The original data are then represented by a set of eigenvectors, to which a linear classifier is applied to get the final predictions. Several domain-transfer learning tasks are used to evaluate our learning method, where experimental results justify that our method is effective on handling this cross-domain classification problem.

There are several directions for future work. The CDSC is given in batch style in this paper. In the future, we would like to extend CDSC to an online cross-domain classifier. It is also important to investigate when negative transfer (domains are sufficiently dissimilar) would happen in domain-transfer learning.

7. ACKNOWLEDGMENTS

Qiang Yang would like to thank the support of Hong Kong RGC Grant 621307. Gui-Rong Xue would like to thank Microsoft Research Asia for their support to the MSRA-SJTU joint lab project "Transfer Learning and its application on the Web". We also thank the anonymous reviewers for their valuable comments.

8. REFERENCES

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.

[2] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *COLT*, 2003.

[3] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML*, 2007.

[4] S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In *NIPS*, 2007.

[5] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *NIPS*, 2008.

[6] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, 2006.

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

[8] C.-K. Cheng and Y.-C. A. Wei. An improved two-way partitioning algorithm with stable performance [VLSI]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(12):1502–1511, 1991.

[9] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[10] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *SIGKDD*, 2007.

[11] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Transferring naive bayes classifiers for text classification. In *AAAI*, 2007.

[12] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *ICML*, 2007.

[13] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[14] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *JAIR*, 1:1–15, 2006.

[15] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. Spectral min-max cut for graph partitioning and data clustering. In *ICDM*, 2001.

[16] G. H. Golub and C. F. Van Loan. *Matrix Computation*. The Johns Hopkins University Press Baltimore, 1996.

[17] J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–162, 1979.

[18] J. Huang, A. J. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2007.

[19] X. Ji and W. Xu. Document clustering with prior knowledge. In *SIGIR*, 2006.

[20] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[21] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, 2003.

[22] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *IJCAI*, 2003.

[23] X. Liao, Y. Xue, and L. Carin. Logistic regression with an auxiliary data source. In *ICML*, 2005.

[24] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics*, 2001.

[25] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[26] M. Porter. An algorithm for suffix stripping program. *Program*, 14(3):130–137, 1980.

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

[28] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[29] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[30] S. Thrun and T. Mitchell. Learning one more thing. In *IJCAI*, 1995.

[31] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, 2000.

[32] P. Wu and T. G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *ICML*, 2004.

[33] D. Xing, W. Dai, G.-R. Xue, and Y. Yu. Bridged refinement for transfer learning. In *PKDD*, 2007.

[34] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, 1997.

[35] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML*, 2004.