Co-clustering based Classification for Out-of-domain Documents

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

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

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

ABSTRACT

In many real world applications, labeled data are in short supply. It often happens that obtaining labeled data in a new domain is expensive and time consuming, while there may be plenty of labeled data from a related but different domain. Traditional machine learning is not able to cope well with learning across different domains. In this paper, we address this problem for a text-mining task, where the labeled data are under one distribution in one domain known as *in-domain* data, while the unlabeled data are under a related but different domain known as *out-of-domain* data. Our general goal is to learn from the in-domain and apply the learned knowledge to out-of-domain. We propose a co-clustering based classification (CoCC) algorithm to tackle this problem. Co-clustering is used as a bridge to propagate the class structure and knowledge from the in-domain to the out-of-domain. We present theoretical and empirical analysis to show that our algorithm is able to produce high quality classification results, even when the distributions between the two data are different. The experimental results show that our algorithm greatly improves the classification performance over the traditional learning algorithms.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Induction

General Terms

Algorithms, Experimentation

Keywords

Classification, Co-clustering, Out-of-domain, Kullback-Leibler divergence

1. INTRODUCTION

Document classification plays an important role in many text processing tasks, ranging from search engines to online advertisements. Traditional document classification algorithms rely on the availability of a large amount of labeled data. In practice, labeled data are often scarce, especially for learning tasks in new domains. When a task from a new domain comes, it may be the case that we have no labeled data at all. Labeling data for classification can be expensive and time consuming in general, but there may be plenty of labeled data from a related but different domain. This may be the case when the labeled data are out of date, but the new data are obtained from fast evolving information sources. Unfortunately, traditional machine learning fails to deal with this situation, since it requires that the labeled and unlabeled data be drawn from the same distribution. This raises a critical problem on how to learn from the labeled data from one domain, and then classify the documents from another domain accurately.

In this paper, we focus on the problem of classifying documents across different domains. We have a labeled data set $\mathcal{D}_i$ from one domain which is called *in-domain*, and another data set $\mathcal{D}_o$ from a related but different domain which is called *out-of-domain*. The latter is unlabeled and to be classified. $\mathcal{D}_i$ and $\mathcal{D}_o$ are drawn from different distributions, since they are from different domains. This may be the case when we consider two related Web directories, for example, when one directory contains documents about cars, and another about trucks. We assume that, the class labels in $\mathcal{D}_i$ and the labels to be predicted in $\mathcal{D}_o$ are drawn from the same class-label set $\mathcal{C}$. Furthermore, we assume that even though the two domains are different in distributions, they are similar in the sense that similar words describe similar categories. In other words, the *true* probability of a class label given a word is very close in the two domains. This assumption is often true, as we will also demonstrate in our experiments, since $\mathcal{D}_i$ and $\mathcal{D}_o$ are related text domains, although some words in one domain may be missing in the other domain, which makes the estimated probability in the two domains to be quite different. Under such circumstances, our objective is to accurately classify the out-of-domain documents in $\mathcal{D}_o$, by making use of the in-domain data $\mathcal{D}_i$ and their labels.

We propose a novel co-clustering based classification algorithm to solve this problem, as briefly shown in Figure 1. First, the in-domain data $\mathcal{D}_i$ provide the class structure, which defines the classification task, by propagating label information. Then, co-clustering [9] is extended for out-of-domain data $\mathcal{D}_o$ to obtain out-of-domain document and word clusters, as the step 2 in Figure 1. Our key extension

KDD'07, August 12-15, 2007, San Jose, California, USA.

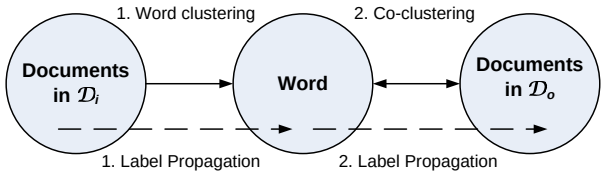

Figure 1: The model of our co-clustering based classification algorithm

to co-clustering is that class labels in $\mathcal{D}_i$ can constrain the word clusters, which is shared among the two domains, as the step 1 in Figure 1. This allows each out-of-domain cluster to be mapped to a corresponding class label based on their correlation with the document categories in $\mathcal{D}_i$, completing our classification task. A key intuition of our work is that even though the two domains may be under different distributions, we are able to identify a common part between them. In our work, this common part is the common words. Class information and knowledge passes through common words from the in-domain to the out-of-domain. Moreover, the word clustering part of co-clustering can even enrich the common words by drawing together seemingly unrelated words.

In this paper, we define a unified information theoretic formulation for the above learning task. The objective function for building the co-clustering based categorization is designed to minimize a *loss* function in mutual information between out-of-domain documents and words, and between words and class labels in the in-domain data set, simultaneously. As a result, the class category knowledge provided by in-domain data $\mathcal{D}_i$ is used as a constraint to enhance the classification of out-of-domain documents based on co-clustering.

To show that our co-clustering based classification algorithm works well, we carry out theoretical analysis to show that our algorithm increasingly optimizes the objective function as our algorithm iterates to completion, by converging quickly to a locally optimal co-clustering result. We supplement the theoretical study with extensive experiments, where we demonstrate that our algorithm is effective in making the predictions for out-of-domain documents.

The rest of the paper is organized as follows. In Section 2, we discuss the related works. In Section 3, some preliminary concepts from information theory is introduced. The problem formulation is presented in Section 4. Section 5 proposes our co-clustering based classification algorithm. The empirical analysis is presented in Section 6. Section 7 concludes the whole paper and give some future works. Some detailed proofs of the theoretical conclusions are given in the Appendix.

2. RELATED WORK

In this section, we review several prior works mostly related to our work, including traditional classification, multi-task and multi-domain learning, and semi-supervised clustering.

2.1 Classification Learning

The traditional classification formulation assumes that the class labels are given for training data under the same distribution as the test data. Two schemes are generally considered, where one is *supervised classification* and the other is *semi-supervised classification*. Supervised classification focuses on the case where the labeled data are sufficient, and where the learning objective is to estimate a function that maps examples to class labels using the labeled training instances. Naive Bayes Classifiers [20] and Support Vector Machines [3] are known as two of the most effective methods for document classification.

Semi-supervised classification [28] addresses the problem that the labeled data are too few to build a good classifier. It makes use of a large amount of unlabeled data, together with a small amount of the labeled data to enhance the classifiers. Many semi-supervised learning techniques have been proposed, e.g., co-training [2], EM-based methods [23], cluster-based methods [27], transductive learning [15] etc.

Both supervised and semi-supervised classification assume that the distributions of the labeled and unlabeled data should be identical. However, in our problem, the labeled data are from in-domain, while the unlabeled data are from out-of-domain. The distributions of the labeled and unlabeled data are different from each other. This violates the basic assumption of traditional supervised and semi-supervised classification algorithms.

2.2 Multi-task and Multi-domain Learning

Another related learning research area is multi-task learning, where the domain-specific information in related tasks (training and test data sets) are jointly trained in a way that can benefit each other [4]. A shared representation is exploited while the extra tasks can be used as an inductive bias during learning. By defining the common knowledge carefully, it is possible to allow the knowledge learned from each task to help the learning of other tasks.

In contrast to multi-task learning, our problem should be considered as single-task learning, since the class labels for in-domain and out-of-domain are from the same class label set. However, our classification problem crosses different domains. This problem can be referred to as *multi-domain learning*, or *cross-domain learning*. [25] studied on cross-domain learning in neural network, while we focus on the cross-domain text classification. [7] studied a similar problem where they investigated how to learn a general model from the in-domain and out-of-domain labeled data to train a statistical classifier for a natural language *Mention Type Classification* and *Tagging* problem. In contrast, in our work, we assume that the out-of-domain data are completely unlabeled.

2.3 Semi-supervised Clustering

Semi-supervised clustering [11] builds clusters under some additional constraints provided by a few labeled data, in the form of must-links (two examples must in the same cluster) and cannot-links (two examples cannot in the same cluster). It finds a balance between satisfying these constraints and optimizing the original clustering objective function. Several semi-supervised clustering algorithms have been proposed, including [1, 5, 12, 10].

Semi-supervised clustering provides a good method to make use of a few labeled data in clustering. However, the must-link and cannot-link constraints must be available for clustering to work. When the labeled data are few, the same-distribution requirement can be relaxed. This fact makes it feasible to extend semi-supervised clustering for different distribution data sets.

In this paper, we consider a co-clustering based classification algorithm which extends the information theoretic co-clustering approach of [9], where constraints given by in-domain data is added to the word clusters [8] to provide a class structure and partial categorization knowledge. Our algorithm is essentially a classification algorithm using the co-clustering technique. It will be shown theoretically and empirically that our algorithm works well for classifying out-of-domain documents.

In addition to building clusters, we are interested in using the class-label knowledge gained from in-domain data to help classify out-of-domain problems, which is not solvable by traditional semi-supervised clustering algorithms alone. As we will present later, our algorithm adds constraints on the word clusters to help assign labels to co-clustering results. The class structures on word clusters are passed on from the in-domain data to the out-of-domain data, which makes classification possible.

3. PRELIMINARIES

In this section, we introduce some preliminary concepts from information theory that will be used frequently in this paper. For more details, please refer to [6]

Let X and Y be random variable sets with a joint distribution $p(X, Y)$ and marginal distributions $p(X)$ and $p(Y)$. The *mutual information* $I(X;Y)$ is defined as

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \quad (1)$$

The mutual information is a measure of the dependency between random variables. It is always non-negative, and it is zero if and only if the variables are statistically independent. Higher mutual information values indicate more certainty that one random variable depends on another.

The use of mutual information can also be motivated using the *Kullback-Leibler (KL) divergence* or *relative entropy* measures, defined for two probability mass functions $p(x)$ and $q(x)$,

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}. \quad (2)$$

KL-divergence can be considered as a kind of a distance between the two probability distributions, although it is not a real distance measure because it is not symmetric. Besides, KL-divergence is always non-negative [6].

4. PROBLEM FORMULATION

Let $\mathcal{D}_i$ be the set of in-domain data with labels, $\mathcal{D}_o$ be the set of out-of-domain data without labels. $\mathcal{D}_i$ and $\mathcal{D}_o$ can also be considered as random variable sets that take the in-domain and out-of-domain instances as random variables, respectively. From the in-domain data $\mathcal{D}_i$, we are able to get a set of class labels $\mathcal{C}$ which is the class structure information. The labels (which are unknown and to be predicted) of out-of-domain data $\mathcal{D}_o$ are also drawn from the same label set $\mathcal{C}$. From $\mathcal{D}_i$ and $\mathcal{D}_o$, the word set $\mathcal{W}$ can be obtained from the word occurrences in $\mathcal{D}_i$ and $\mathcal{D}_o$.

In our approach, we take co-clustering as a bridge to propagate the knowledge from the in-domain to out-of-domain. Co-clustering on out-of-domain data aims to simultaneously cluster the out-of-domain documents $\mathcal{D}_o$ and words $\mathcal{W}$ into $|\mathcal{C}|$ document clusters and k word clusters, respectively. Let $\hat{\mathcal{D}}_o$ denote the out-of-domain document clustering, and $\hat{\mathcal{W}}$ denote the word clustering, where $|\hat{\mathcal{W}}| = k$. The document cluster-partition function $C_{\mathcal{D}_o}$ and the word cluster-partition function $C_{\mathcal{W}}$ can be defined as

$$C_{\mathcal{D}_o}(d) = \hat{d}, \text{ where } d \in \hat{d} \wedge \hat{d} \in \hat{\mathcal{D}}_o \quad (3)$$

$$C_{\mathcal{W}}(w) = \hat{w}, \text{ where } w \in \hat{w} \wedge \hat{w} \in \hat{\mathcal{W}} \quad (4)$$

where $\hat{d}$ represents the document cluster that d belongs to and $\hat{w}$ represents the word cluster that w belongs to. Then, the co-clustering can be represented by $(C_{\mathcal{D}_o}, C_{\mathcal{W}})$ or $(\hat{\mathcal{D}}_o, \hat{\mathcal{W}})$.

In order to measure the quality of a co-clustering, we define the loss for co-clustering in mutual information as

$$I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}). \quad (5)$$

This form of loss function is the same as that used in [9]. From Equation (5), we know that co-clustering aims to minimize the loss in mutual information between documents and words before and after clustering.

Since our problem is to classify out-of-domain documents $\mathcal{D}_o$, a key point is to add the knowledge about classes to the co-clustering process, which is extracted from in-domain data $\mathcal{D}_i$. In this paper, we use the relationship between word clusters and class labels to apply class label information to the co-clustering. We define the loss in mutual information for a word clustering as

$$I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}}). \quad (6)$$

This form of loss function is the same as that used in [8]. Equation (6) indicates that a word clustering should minimize the loss in mutual information between class labels $\mathcal{C}$ and words $\mathcal{W}$ before and after clustering, for in-domain data.

Integrating Equations (5) and (6), the loss function for co-clustering based classification can be obtained:

$$I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda \cdot (I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}})) \quad (7)$$

where λ is a trade-off parameter that balances the effect to word clusters from co-clustering (see Equation (5)) and word clustering (see Equation (6)). The objective is to find a co-clustering that minimizes the function value of Equation (7).

With Equation (7), we can now describe our process of classifying out-of-domain documents through co-clustering. Since $I(\mathcal{D}_o; \mathcal{W})$ and $I(\mathcal{C}; \mathcal{W})$ are fixed, minimizing Equation (7) equivalents maximizing $I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}})$ and $I(\mathcal{C}; \hat{\mathcal{W}})$ simultaneously – maximizing $I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda \cdot I(\mathcal{C}; \hat{\mathcal{W}})$. Based on the definition of mutual information in Section 3, in order to maximize $I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}})$ and $I(\mathcal{C}; \hat{\mathcal{W}})$, $\hat{\mathcal{D}}_o$ should depend on $\hat{\mathcal{W}}$, and $\hat{\mathcal{W}}$ should depend on $\mathcal{C}$. Under our problem assumption, $\hat{\mathcal{D}}_o$ would depend on $\mathcal{C}$, which indicates that the clusters in $\hat{\mathcal{D}}_o$ should be rely on the classes in $\mathcal{C}$. We can let the number of document clusters be the same as the number of class labels to enable a 1-1 mapping between them. That is, we let $|\hat{\mathcal{D}}_o| = |\mathcal{C}|$, and build a mapping between $\hat{\mathcal{D}}_o$ and $\mathcal{C}$ based on the dependence between each $\hat{d} \in \hat{\mathcal{D}}_o$ and each $c \in \mathcal{C}$. Then, using the co-clustering based classification approach that optimizes Equation (7), the documents in $\mathcal{D}_o$ will be assigned to their corresponding classes according to cluster membership, which enables our co-clustering based classification approach.

In the rest of this section, we will rewrite the objective function in Equation (7) into another form that is represented by KL-divergence. Before rewriting the objective function, let us first define some probability mass functions.

Definition 1. Let $f(\mathcal{D}_o, \mathcal{W})$ denote the joint probability distribution of $\mathcal{D}_o$ and $\mathcal{W}$. That is

$$f(d, w) = p(d, w). \tag{8}$$

$\hat{f}(\mathcal{D}_o, \mathcal{W})$ denotes the joint probability distribution of $\mathcal{D}_o$ and $\mathcal{W}$ under co-clustering $(\hat{\mathcal{D}}_o, \hat{\mathcal{W}})$ that

$$\hat{f}(d, w) = p(\hat{d}, \hat{w})p(d|\hat{d})p(w|\hat{w}) = p(\hat{d}, \hat{w})\frac{p(d)}{p(\hat{d})}\frac{p(w)}{p(\hat{w})}, \tag{9}$$

where $d \in \hat{d}$ and $w \in \hat{w}$, where $\hat{d}$ is a document cluster, and $\hat{w}$ is a word cluster, respectively.

Similarly, $g(\mathcal{C}, \mathcal{W})$ denotes the joint probability distribution of $\mathcal{C}$ and $\mathcal{W}$ that

$$g(c, w) = p(c, w). \tag{10}$$

$\hat{g}(\mathcal{C}, \mathcal{W})$ denotes the joint probability distribution of $\mathcal{C}$ and $\mathcal{W}$ under the word clustering $\hat{\mathcal{W}}$ that

$$\hat{g}(c, w) = p(c, \hat{w})p(w|\hat{w}) = p(c, \hat{w})\frac{p(w)}{p(\hat{w})}, \tag{11}$$

where $w \in \hat{w}$.

The marginal and conditional probability distributions for f, $\hat{f}$, g and $\hat{g}$ can be defined naturally. For example,

$$\hat{f}(d) = \sum_w \hat{f}(d, w), \text{ and } \hat{f}(d|w) = \frac{\hat{f}(d, w)}{\hat{f}(w)}. \tag{12}$$

LEMMA 1. *For a fixed co-clustering $(\hat{\mathcal{D}}_o, \hat{\mathcal{W}})$, we can write the loss in mutual information as*

$$\begin{aligned} &I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda \cdot (I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}})) \\ &= D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) + \lambda \cdot D(g(\mathcal{C}, \mathcal{W})||\hat{g}(\mathcal{C}, \mathcal{W})). \end{aligned} \tag{13}$$

where $D(\cdot||\cdot)$ is defined in Equation (2).

PROOF.

$$\begin{aligned} &I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda \cdot (I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}})) \\ &= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{d \in \hat{d}} \sum_{w \in \hat{w}} p(d, w) \log \frac{p(d, w)}{p(d)p(w)} \\ &\quad - \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{\hat{w} \in \hat{\mathcal{W}}} \left(\sum_{d \in \hat{d}} \sum_{w \in \hat{w}} p(d, w) \right) \log \frac{p(\hat{d}, \hat{w})}{p(\hat{d})p(\hat{w})} \\ &\quad + \lambda \sum_{c \in \mathcal{C}} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{w \in \hat{w}} p(c, w) \log \frac{p(c, w)}{p(c)p(w)} \\ &\quad - \lambda \sum_{c \in \mathcal{C}} \sum_{\hat{w} \in \hat{\mathcal{W}}} \left(\sum_{w \in \hat{w}} p(c, w) \right) \frac{p(c, \hat{w})}{p(c)p(\hat{w})} \end{aligned} \tag{14}$$

$$= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{d \in \hat{d}} \sum_{w \in \hat{w}} p(d, w) \log \frac{p(d, w)}{p(\hat{d}, \hat{w})\frac{p(d)}{p(\hat{d})}\frac{p(w)}{p(\hat{w})}} + \lambda \sum_{c \in \mathcal{C}} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{w \in \hat{w}} p(d, w) \log \frac{p(d, w)}{p(d, \hat{w})\frac{p(w)}{p(\hat{w})}} \tag{15}$$

$$= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{d \in \hat{d}} \sum_{w \in \hat{w}} f(d, w) \log \frac{f(d, w)}{\hat{f}(d, w)} + \lambda \sum_{c \in \mathcal{C}} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{w \in \hat{w}} g(d, w) \log \frac{g(d, w)}{\hat{g}(d, w)} \tag{16}$$

$$= D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) + \lambda \cdot D(g(\mathcal{C}, \mathcal{W})||\hat{g}(\mathcal{C}, \mathcal{W})) \tag{17}$$

□

Equation (13) shows that the loss in mutual information in the objective function equals to the sum of KL-divergence between f and $\hat{f}$ and KL-divergence between g and $\hat{g}$. To minimize the objective function in Equation (7), we need only to minimize the KL-divergence between f and $\hat{f}$, and the KL-divergence between g and $\hat{g}$.

5. CO-CLUSTERING BASED CLASSIFICATION

We now describe the co-clustering based classification algorithm for classifying the out-of-domain data, which minimizes the objective function in Equation (13). The objective function given in Equation (13) is a multi-part function which is hard to be optimized. Therefore, we should find a way to make the optimization easier. Lemmas 2 and 3 show an alternative approach, which allows us to iteratively reduce the divergence values.

LEMMA 2.

$$\begin{aligned} &D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) \\ &\quad = \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{d \in \hat{d}} f(d) D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d})), \end{aligned} \tag{18}$$

$$\begin{aligned} &D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) \\ &\quad = \sum_{\hat{w} \in \hat{W}} \sum_{w \in \hat{W}} f(w) D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w})). \end{aligned} \tag{19}$$

PROOF.

$$\begin{aligned} &D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) \\ &= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{d \in \hat{d}} \sum_{w \in \hat{w}} f(d, w) \log \frac{f(d, w)}{\hat{f}(d, w)} \end{aligned} \tag{20}$$

$$= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{d \in \hat{d}} \sum_{w \in \hat{w}} f(d) f(w|d) \log \frac{f(d) f(w|d)}{f(d) \hat{f}(w|\hat{d})} \tag{21}$$

$$= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{d \in \hat{d}} f(d) \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{w \in \hat{w}} f(w|d) \log \frac{f(w|d)}{\hat{f}(w|\hat{d})} \tag{22}$$

$$= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{d \in \hat{d}} f(d) D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d})) \tag{23}$$

Note that Equation (21) is based on

$$\hat{f}(d, w) = p(\hat{d}, \hat{w})p(d|\hat{d})p(w|\hat{w}) = p(d)\frac{p(\hat{d}, \hat{w})}{p(\hat{d})}\frac{p(w)}{p(\hat{w})} \tag{24}$$

$$= p(d)p(\hat{w}|\hat{d})p(w|\hat{w}) = f(d)\hat{f}(w|\hat{d}) \tag{25}$$

The last equality follows by $p(d) = f(d)$ and $p(\hat{w}|\hat{d})p(w|\hat{w}) = \hat{f}(w|\hat{d})$.

Using the same argument, we can prove that

$$D(f(\mathcal{D}_o,\mathcal{W})||\hat{f}(\mathcal{D}_o,\mathcal{W})) = \sum_{\hat{w}\in\hat{\mathcal{W}}}\sum_{w\in\hat{w}} f(w)D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w})) \quad (26)$$

□

Lemma 2 tells us that minimizing $D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d}))$ corresponding to a single document d can reduce the global objective function value given in Equation (13). The same conclusion can be derived for minimizing $D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w}))$ corresponding to a single word w.

LEMMA 3.

$$D(g(\mathcal{C},\mathcal{W})||\hat{g}(\mathcal{C},\mathcal{W})) = \sum_{\hat{w}\in\hat{\mathcal{W}}}\sum_{w\in\hat{w}} g(w)D(g(\mathcal{C}|w)||\hat{g}(\mathcal{C}|\hat{w})). \quad (27)$$

The proof of Lemma 3 is omitted, and it can be derived using the similar argument to Lemma 2. From Lemma 3, we can obtain the similar conclusion with that in Lemma 2.

According to Lemmas 2 and 3, our co-clustering based classification algorithm, called CoCC, is derived. This algorithms iteratively searches a co-clustering for the out-of-domain data, and then assigns class labels to the document clusters to complete the classification task.

Algorithm 1 The Co-clustering based Classification (CoCC) Algorithm

Input: A labeled in-domain data set $\mathcal{D}_i$; an unlabeled out-of-domain data set $\mathcal{D}_o$; a set $\mathcal{C}$ of all the class labels; a set $\mathcal{W}$ of all the word features; initial co-clustering $(C^{(0)}_{\mathcal{D}_o}, C^{(0)}_{\mathcal{W}})$; the number of iterations T.

Initialize the joint probability distribution f, $\hat{f}$, g and $\hat{g}$ based on Equations (8), (9), (10) and (11), respectively.

For $t \leftarrow 1, 3, 5, \ldots, 2T+1$

1: Compute the document cluster:

$$C^{(t)}_{\mathcal{D}_o}(d) = \arg\min_{\hat{d}} D(f(\mathcal{W}|d)||\hat{f}^{(t-1)}(\mathcal{W}|\hat{d})) \quad (28)$$

2: Update the probability distribution $\hat{f}^{(t)}$ based on $C^{(t)}_{\mathcal{D}_o}$, $C^{(t-1)}_{\mathcal{W}}$, and Equation (9). $C^{(t)}_{\mathcal{W}} = C^{(t-1)}_{\mathcal{W}}$ and $\hat{g}^{(t)} = \hat{g}^{(t-1)}$.

3: Compute the word cluster:

$$C^{(t+1)}_{\mathcal{W}}(w) = \arg\min_{\hat{w}} \; f(w)D(f(\mathcal{D}_o|w)||\hat{f}^{(t)}(\mathcal{D}_o|\hat{w})) + \lambda\cdot g(w)D(g(\mathcal{C}|w)||\hat{g}^{(t)}(\mathcal{C}|\hat{w})) \quad (29)$$

4: Update the probability distribution $\hat{g}^{(t+1)}$ based on $C^{(t+1)}_{\mathcal{W}}$, and Equation (11). $\hat{f}^{(t+1)} = \hat{f}^{(t)}$ and $C^{(t+1)}_{\mathcal{D}_o} = C^{(t)}_{\mathcal{D}_o}$.

End For

Output: the partition functions $C^{(T)}_{\mathcal{D}_o}$ and $C^{(T)}_{\mathcal{W}}$.

As shown in Algorithm 1, in each iteration, the algorithm chooses the best document cluster $\hat{d}$ for each d to minimize the function $D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d}))$ (see Equation (28)). As we discussed above, this can reduce the global objective function value in Equation (13). Then, in each iteration, the algorithm chooses the best word cluster $\hat{w}$ to minimize the function $D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w}))$ and $D(g(\mathcal{C}|w)||\hat{g}(\mathcal{C}|\hat{w}))$ simultaneously (see Equation (29)). This can reduce the global objective function value too. We will prove the monotonically decreasing property of the objective function in the following theorem:

THEOREM 4. *The algorithm CoCC in Algorithm 1 monotonically decreases the objective function in Lemma 1.*

$$D(f(\mathcal{D}_o,\mathcal{W})||\hat{f}^{(t)}(\mathcal{D}_o,\mathcal{W})) + \lambda\cdot D(g(\mathcal{C},\mathcal{W})||\hat{g}^{(t)}(\mathcal{C},\mathcal{W})) \geq D(f(\mathcal{D}_o,\mathcal{W})||\hat{f}^{(t+1)}(\mathcal{D}_o,\mathcal{W})) + \lambda\cdot D(g(\mathcal{C},\mathcal{W})||\hat{g}^{(t+1)}(\mathcal{C},\mathcal{W})). \quad (30)$$

The detailed proof of Theorem 4 is given in the Appendix. Note that, although the algorithm is able to minimize the objective function value in Equation (13), it is only able to find a locally minimal one. Finding the global optimal co-clustering is NP-hard.

COROLLARY 5. *Algorithm 1 converges in a finite number of iterations.*

PROOF. Since the total number of co-clusterings is finite, the corollary can be derived straightforward from Theorem 4. □

Regarding the computational complexity, suppose the total number of document-word co-occurrences in $\mathcal{D}_o$ is N. For each iteration, updating $C_{\mathcal{D}_o}$ takes $O(|\mathcal{C}|\cdot N)$, while updating $C_{\mathcal{W}}$ takes $O((|\mathcal{C}|+|\hat{\mathcal{W}}|)\cdot N)$. The number of iterations is T. Therefore, the time complexity of our co-clustering based classification algorithm is $O((|\mathcal{C}|+|\hat{\mathcal{W}}|)\cdot T\cdot N)$. In the experiments, it is shown that $T = 10$ is enough for convergence. Considering space complexity, our algorithm need to store all the document-word co-occurrences and their corresponding probabilities. Thus, the space complexity is $O(N)$.

6. EXPERIMENTS

In order to evaluate the properties of our algorithm, we perform the experiments in this section. In the experiments, we focus on the binary classification. Moreover, the data sets are all balanced between the class labels. Note that the binary classifiers can be easily extended for multiple class.

6.1 Data Sets

We conducted experiments on three data sets, 20 Newsgroups [18], SRAA [21] and Reuters-21578 [19]. In order to make the data set satisfy our problem setting, we split the original data in a way to make the labeled and unlabeled data drawn from related but different domains, as follows.

6.1.1 20 Newsgroups

The 20 Newsgroups [18] is a text collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups nearly evenly. We generated six different data sets for evaluating cross-domain classification algorithms. For each data set, two top categories[1] are chosen,

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

Data Set	$\mathcal{D}_i$	$\mathcal{D}_o$
comp vs sci	comp.graphics	comp.sys.ibm.pc.hardware
	comp.os.ms-windows.misc	comp.sys.mac.hardware
	sci.crypt	comp.windows.x
	sci.electronics	sci.med
		sci.space
rec vs talk	rec.autos	rec.sport.baseball
	rec.motorcycles	rec.sport.hockey
	talk.politics.guns	talk.politics.mideast
	talk.politics.misc	talk.religion.misc
rec vs sci	rec.autos	rec.motorcycles
	rec.sport.baseball	rec.sport.hockey
	sci.med	sci.crypt
	sci.space	sci.electronics
sci vs talk	sci.electronics	sci.crypt
	sci.med	sci.space
	talk.politics.misc	talk.politics.guns
	talk.religion.misc	talk.politics.mideast
comp vs rec	comp.graphics	comp.os.ms-windows.misc
	comp.sys.ibm.pc.hardware	comp.windows.x
	comp.sys.mac.hardware	rec.autos
	rec.motorcycles	rec.sport.baseball
	rec.sport.hockey	
comp vs talk	comp.graphics	comp.os.ms-windows.misc
	comp.sys.mac.hardware	comp.sys.ibm.pc.hardware
	comp.windows.x	talk.politics.guns
	talk.politics.mideast	talk.politics.misc
	talk.religion.misc	

Table 1: The description of 20 Newsgroups data sets for cross-domain classification.

Data Set	$\mathcal{D}_i$	$\mathcal{D}_o$
auto vs aviation	sim-auto & sim-aviation	real-auto & real-aviation
real vs simulated	real-aviation & sim-aviation	real-auto & sim-auto

Table 2: The description of SRAA data sets for cross-domain classification.

one as positive and the other as negative. Then, we split the data 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 we generated the data sets in our experiments.

6.1.2 SRAA

SRAA [21] 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*. We use the documents in `real-auto` and `sim-auto` as in-domain data, while `real-aviation` and `sim-aviation` as out-of-domain data. Then, the data set `real vs simulated` is generated as shown in Table 2. As a result, all the data in the in-domain data set are about autos, 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 2.

6.1.3 Reuters-21578

Reuters-21578 [19] 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`, we removed all the documents about the USA 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 we have done on the 20 Newsgroups and SRAA corpora. Since there are too many sub-categories, we can not list the detailed description here.

Data Set	K-L	Documents			SVM	
		$\lvert\mathcal{D}_i\rvert$	$\lvert\mathcal{D}_o\rvert$	$\lvert\mathcal{W}\rvert$	$\mathcal{D}_i$–$\mathcal{D}_o$	$\mathcal{D}_o$–CV
real vs simulated	1.161	8,000	8,000	14,433	0.266	0.032
auto vs aviation	1.126	8,000	8,000	14,433	0.228	0.033
rec vs talk	1.102	3,669	3,561	19,412	0.233	0.003
rec vs sci	1.021	3,961	3,965	18,152	0.212	0.007
comp vs talk	0.967	4,482	3,652	17,918	0.103	0.005
comp vs sci	0.874	3,930	4,900	18,379	0.317	0.012
comp vs rec	0.866	4,904	3,949	18,903	0.165	0.008
sci vs talk	0.854	3,374	3,828	20,057	0.226	0.009
orgs vs places	0.329	1,079	1,080	4,415	0.454	0.085
people vs places	0.307	1,239	1,210	4,562	0.266	0.113
orgs vs people	0.303	1,016	1,046	4,771	0.297	0.106

Table 3: Description of the data sets for cross-domain text classification, including errors given by SVM. "$\mathcal{D}_i$–$\mathcal{D}_o$" means training on $\mathcal{D}_i$ and testing on $\mathcal{D}_o$; "$\mathcal{D}_o$–CV" means 10-fold cross-validation on $\mathcal{D}_o$. The performances are in test error rate.

6.1.4 Properties of the Data Sets

Table 3 shows the description of all the data sets. The first three columns of the table show the statistical properties of the data sets. The first two data sets are from SRAA corpus. The next six are generated using 20 Newsgroups data set. The last three are from Reuters-21578 test collection. KL-divergence values calculated by $D(\mathcal{D}_i||\mathcal{D}_o)$ on all the data set are presented in the second column in the table, sorted in decreasing order from top down. It can be seen that the KL-divergence values for all the data sets are much larger than the identical-distribution case which has a KL value of nearly zero. The next column titled "Documents" shows the size of the data sets and the vocabulary set used. Under the column titled "SVM", we show two groups of classification results in two sub-columns. First, "$\mathcal{D}_i$–$\mathcal{D}_o$" denotes the test error rate obtained when a classifier is trained based on the in-domain data set $\mathcal{D}_i$ and applied to the out-of-domain data set $\mathcal{D}_o$. The column titled "$\mathcal{D}_o$–CV" denotes the best-case obtained by the corresponding classifier, where the best case is to conduct a 10-fold cross-validation on the out-of-domain data set $\mathcal{D}_o$ using that classifier. Note in obtaining the best case for each classifier, the training part is labeled data from $\mathcal{D}_o$ and the test part is also from $\mathcal{D}_o$, according to different folds, which gives the best possibly result for that classifier. It can be found that the test error rates, given by SVM, in the case of "$\mathcal{D}_i$–$\mathcal{D}_o$" is much worse than those in the case of "$\mathcal{D}_o$–CV". This indicates that our data sets are not suitable for traditional supervised classification algorithms.

Figure 2 shows the document-word co-occurrence distribution on the `auto vs aviation` data set. In this figure, documents 1 to 8000 are from $\mathcal{D}_i$, while documents 8001 to 16000 are from $\mathcal{D}_o$. The documents are ordered first by their domains ($\mathcal{D}_i$ or $\mathcal{D}_o$), and second by their categories (positive or negative). The words are sorted by $n_+(w)/n_-(w)$, where $n_+(w)$ and $n_-(w)$ represent the number of word positions w appears in positive and negative documents, respectively. From Figure 2, it can be found that the distributions of in-domain and out-of-domain data are somewhat different, however the figure also shows large commonness exists between the two domains. In our algorithm, the class infor-

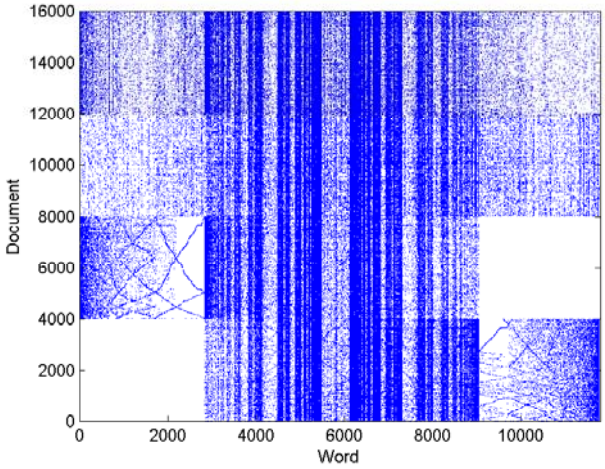

Figure 2: Document-word co-occurrence distribution on the `auto vs aviation` data set

mation and knowledge passes through these common information from the in-domain to the out-of-domain. Moreover, the word clustering part in the co-clustering can even enrich the common part to further propagate knowledge between different domains.

6.2 Comparison Methods

Since our co-clustering based classification algorithm (CoCC) is a classification algorithm essentially, we should compare CoCC with the existing classification methods to show the advantages of our algorithm. We take the supervised classification algorithms to be the baseline methods. Naive Bayes Classifier (NBC) [20] and Support Vector Machines (SVM) [3] are introduced in the experiments. Transductive Support Vector Machines (TSVM) [15] and Spectral Graph Transducer (SGT) are also introduced as comparison semi-supervised learning methods.

6.3 Implementation Details

Data preprocessing has been applied to the raw data. First, we converted all the letters in the text to lower case, and stemmed the words using the Porter stemmer [24]. Besides, stop words were removed. We used a simple feature selection method, Document Frequency (DF) Thresholding [26], to cut down the number of features, and speed up the classification. Based on [26], DF thresholding, which has comparable performance with Information Gain (IG) or CHI, is suggested since it is simplest with lowest cost in computation. In our experiments, we set the DF threshold to 3.

TF-IDF is used for feature weighting when training Support Vector Machines (SVM) [3, 15] and Spectral Graph Transducer (SGT) [16]. TF is used for feature weighting when training Naive Bayes Classifier (NBC) [20] and our co-clustering based classification (CoCC) algorithm.

SVM and TSVM are implemented by SVM^{light} [14] with default parameters (linear kernel). For more details about SVM and TSVM, please refer to [3] and [15]. SGT is implemented by SGT^{light} [13] with default parameters ($k = 50$, $d = 80$ and $c = 100$). For more details about SGT, please refer to [16].

The initialization of CoCC is important, since different initialization will lead to different local optimal co-clustering. In the experiments, we assign the initial document clustering by NBC. NBC is trained using $\mathcal{D}_i$, and then predicts the labels of $\mathcal{D}_o$. Then, the documents in $\mathcal{D}_o$ are assigned to the clusters based on their prediction labels. The initial word clustering is derived by CLUTO [17] with default parameters.

Another important issue to be mentioned is that, in order to avoid infinity values for $D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d}))$ in Equation (28), and $D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w}))$ and $D(g(\mathcal{C}|w)||\hat{g}(\mathcal{C}|\hat{w}))$ in Equation (29), Laplacian smoothing [22] is applied to estimate the probabilities.

Finally, after co-clustering, we assign each document d to the class c by

$$c = \arg\min_{c \in \mathcal{C}} D(\hat{g}(\mathcal{W}|c)||\hat{f}(\mathcal{W}|\hat{d})). \tag{31}$$

Equation (31) indicates that we always assign the document d to the class c which is most relevant to $\hat{d}$. Note that, our objective function in Equation (13) ensures that $\mathcal{C}$ and $\hat{\mathcal{D}}_o$ are highly dependent, and hence the assignment makes sense.

6.4 Evaluation Metrics

The performance of the proposed methods was evaluated by test error rate. Let C be the function which maps from document d to its true class label $c = C(d)$, and F be the function which maps from document d to its prediction label $c = F(d)$ given by the classifiers. Test error rate is defined as

$$\epsilon = \frac{|\{d|d \in \mathcal{D}_o \wedge C(d) \neq F(d)\}|}{|\mathcal{D}_o|}. \tag{32}$$

6.5 Experimental Results

6.5.1 *Performance*

Table 4 presents the performance on each data set given by NBC, SVM, TSVM, SGT and our algorithm CoCC in test error rate. The implementation details of the algorithms have already been presented in the last subsection, and the parameter setting for CoCC will be given later.

From the table, we can see that CoCC always give the best performances. Besides, it seems that NBC is better for classifying out-of-domain documents than SVM, although SVM is known as a stronger classifier than NBC. In our opinion, SVM is a relatively strong classifier for traditional classification problem, compared with NBC, but NBC is more general for out-of-domain data. TSVM and SGT give better performance than NBC and SVM on most data sets, since they utilize the information given by the unlabeled data in $\mathcal{D}_o$, although their basic assumption is violated. However, they still fail sometimes, e.g. TSVM on the `orgs vs places` data set, and SGT on the `orgs vs people` data set.

Figure 3 presents the test error rates on different sizes of the *labeled in-domain data* . The labeled data are randomly chosen from $\mathcal{D}_i$ in the `auto vs aviation` data set by different proportion. It can be seen that CoCC gives comparable performance even when there is only 10% of the in-domain data, while NBC gets quickly worse when the proportion of in-domain data is less than 20%.

Data Set	NBC	SVM	TSVM	SGT	CoCC
real vs simulated	0.259	0.266	0.130	0.130	**0.120**
auto vs aviation	0.150	0.228	0.102	0.087	**0.068**
rec vs talk	0.235	0.233	0.040	0.091	**0.035**
rec vs sci	0.165	0.212	0.062	0.062	**0.055**
comp vs talk	0.024	0.103	0.097	0.028	**0.020**
comp vs sci	0.207	0.317	0.183	0.279	**0.130**
comp vs rec	0.072	0.165	0.098	0.047	**0.042**
sci vs talk	0.226	0.226	0.108	0.083	**0.054**
orgs vs places	0.377	0.454	0.436	0.385	**0.320**
people vs places	0.216	0.266	0.231	0.192	**0.174**
orgs vs people	0.289	0.297	0.297	0.306	**0.236**

Table 4: Test error rate for each classifier on each data set

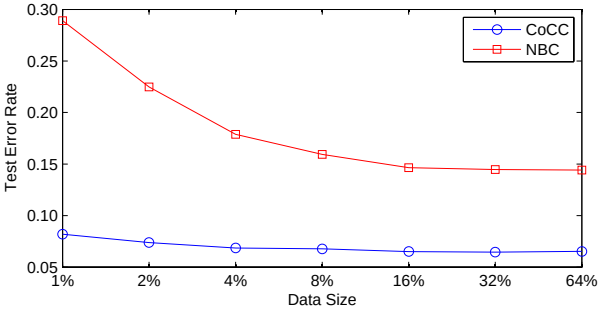

Figure 3: Test error rate curve on different size of `auto vs aviation` data set

6.5.2 *Convergence*

Since our algorithm CoCC is an iterative algorithm, an important issue for CoCC is the convergence property. Theorem 4 has already proven the convergence of CoCC theoretically. Now, let us empirically show the convergence property of CoCC. Figure 4 shows the test error rate curves as functions for each iteration on three data sets, `real vs simulated`, `rec vs sci` and `orgs vs places`. From the figure, it can be seen that CoCC always achieves almost convergence points within 5 iterations. This indicates that CoCC converges very fast. We believe that 10 iterations is enough for CoCC.

6.5.3 *Parameters Tuning*

There are two parameters in our algorithm. One is the trade-off parameter λ in Equation (7); the other is the number of word clusters. We perform the parameter tuning on the `auto vs aviation` data set. When tuning the parameter λ, we tried three different numbers of word clusters – 16, 64 and 128. The error rate for each λ from 0.003125 to 8 is given in Figure 5. According to the figure, we set λ to 0.125 in our experiments. When tuning the number of word clusters, we tried different λs which are 1, 0.5 and 0.25. The error rate for each number of word clusters from 2 to 512 is given in Figure 6. According to the figure, we set the number of word clusters to 128 in our experiments.

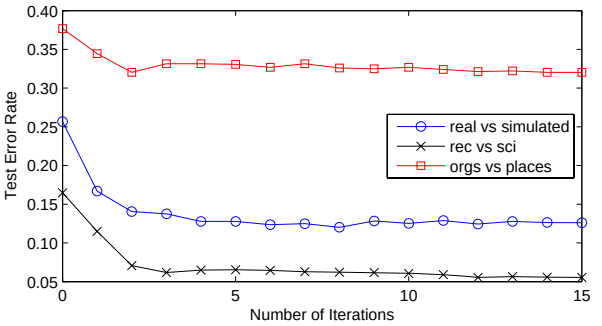

Figure 4: Test error rate curves after each iteration

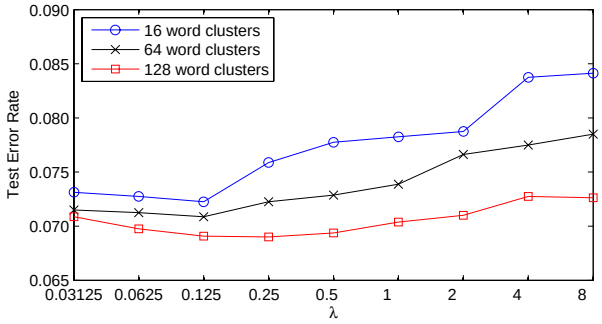

Figure 5: Test error rate curve on different λ

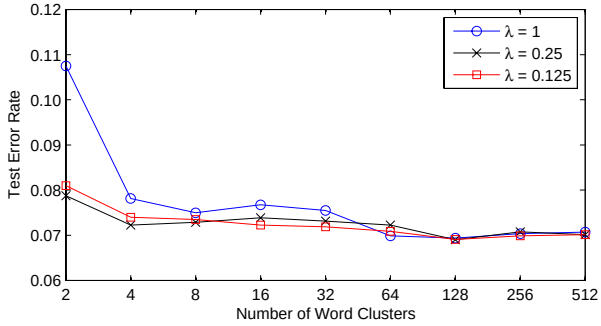

Figure 6: Test error rate curve on different number of word clusters

6.5.4 *KL-divergence and Improvement*

We test how the difference in the distribution between $\mathcal{D}_i$ and $\mathcal{D}_o$ influence the performance of CoCC. The KL-divergence and relative improvements by test error rate reduction between CoCC and NBC, and between CoCC and SVM are calculated for each data set in Figure 7. The data sets have been sorted by KL-divergence in decreasing order from left to right. In this figure, when the KL-divergence is small, the relative improvement is not much significant. The improvement becomes great when KL-divergence values become large, in general. But, there are still some exceptional points.

7. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a novel co-clustering-based classification algorithm (CoCC) to classify out-of-domain documents. The class structure passes through word clusters from the in-domain data to the out-of-domain data. Additional class-label information given by the in-domain data is extracted and used for labeling the word clusters for

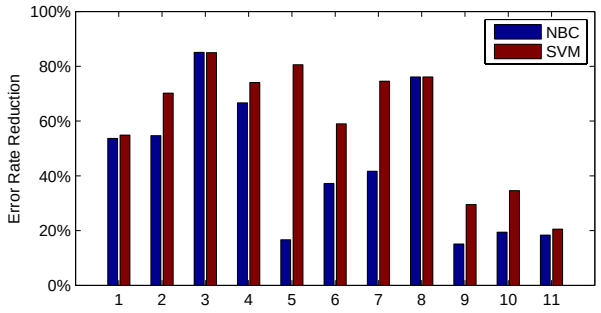

Figure 7: Test error rate reductions against NBC and SVM on all data sets sorted by KL divergence in descending order from left to right

out-of-domain documents. We formulate the problem under an information-theoretic scheme, and designed an objective function to minimize the loss in mutual information before and after co-clustering based categorization. Our theory shows that CoCC can monotonically reduce the objective function value. The empirically results also support our theoretical analysis. In our experiment, it is shown that CoCC greatly outperforms traditional supervised and semi-supervised classification algorithms when classifying out-of-domain documents.

In CoCC, the number of word clusters are quite large (128 clusters in the experiments) to obtain good performance. Since the time complexity of CoCC depends on the number of word clusters, it can inefficient. In the future, we will try to speed up the algorithm to make it more scalable for large data set. Moreover, the parameters in CoCC are tuned manually. In the future, we will investigate automatic approaches to tune the parameters.

8. REFERENCES

[1] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *Proceedings of Ninteenth International Conference on Machine Learning*, 2002.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998.

[3] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992.

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

[5] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. Technical Report TR2003-1892, Cornell University, 2003.

[6] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 1991.

[7] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

[8] I. S. Dhillon, S. Mallela, and R. Kumar. Enhanced word clustering for hierarchical text classification. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[9] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[10] J. Gao, P.-N. Tan, and H. Cheng. Semi-supervised clustering with partial background information. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006.

[11] N. Grira, M. Crucianu, and N. Boujemaa. Unsupervised and semi-supervised clustering: a brief survey, 2005. In *A Review of Machine Learning Techniques for Processing Multimedia Content*, Report of the MUSCLE Eurepean Network of Excellence (6th Framework Programme).

[12] X. Ji, W. Xu, and S. Zhu. Document clustering with prior knowledge. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[13] T. Joachims. SGT^{light}. `http://sgt.joachims.org/`.

[14] T. Joachims. SVM^{light}. `http://svmlight.joachims.org/`.

[15] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of Sixteenth International Conference on Machine Learning*, 1999.

[16] T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of Twentieth International Conference on Machine Learning*, 2003.

[17] G. Karypis. Cluto – software for clustering high-dimensional datasets. `http://glaros.dtc.umn.edu/gkhome/views/cluto`.

[18] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[19] D. D. Lewis. Reuters-21578 test collection. `http://www.daviddlewis.com/`.

[20] D. D. Lewis. *Representation and learning in information retrieval*. PhD thesis, Amherst, MA, USA, 1992.

[21] A. K. McCallum. Simulated/real/aviation/auto usenet data. `http://www.cs.umass.edu/~mccallum/code-data.html`.

[22] T. M. Mitchell. *Machine Learning*, chapter 6, page 179. McGraw Hill, 1997.

[23] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2-3):103–134, 2000.

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

[25] S. Swarup and S. R. Ray. Cross-domain knowledge transfer using structured representations. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 2006.

[26] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of Fourteenth International Conference on Machine Learning*, 1997.

[27] H.-J. Zeng, X.-H. Wang, Z. Chen, H. Lu, and W.-Y. Ma. Cbc: Clustering based text classification requiring minimal labeled data. In *Proceedings of the third IEEE International Conference on Data Mining*, 2003.

[28] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin–Madison, 2006.

APPENDIX

A. PROOF OF THEOREM 4

We split Theorem 4 into two lemmas, Lemma 6 and Lemma 7. Lemma 6 proves Theorem 4 for $t = 1, 3, \ldots, 2T+1$. Lemma 7 proves Theorem 4 for $t = 2, 4, \ldots, 2T+2$. Combine the two lemmas, Theorem 4 is proved.

LEMMA 6. *Theorem 4 holds when $t = 1, 3, \ldots, 2T+1$.*

PROOF. For $t = 1, 3, \ldots, 2T+1$, since $C_{\mathcal{W}}^{(t)} = C_{\mathcal{W}}^{(t+1)}$, we need only to prove

$$D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}^{(t)}(\mathcal{D}_o, \mathcal{W})) \geq D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}^{(t+1)}(\mathcal{D}_o, \mathcal{W})).$$

$$\begin{aligned}
&D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}^{(t)}(\mathcal{D}_o, \mathcal{W})) \\
&= \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t)}(d)|d\in\mathcal{D}_o\}} \sum_{d\in\hat{d}} f(d) \\
&\quad \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t)}(w)|w\in\mathcal{W}\}} \sum_{w\in\hat{w}} f(w|d) \log \frac{f(w|d)}{\hat{f}^{(t)}(w|\hat{d})} && (33) \\
&\geq \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t)}(d)|d\in\mathcal{D}_o\}} \sum_{d\in\hat{d}} f(d) \\
&\quad \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t)}(w)|w\in\mathcal{W}\}} \sum_{w\in\hat{w}} f(w|d) \log \frac{f(w|d)}{\hat{f}^{(t)}(w|C_{\mathcal{D}_o}^{(t+1)}(d))} && (34) \\
&= \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \sum_{d\in\hat{d}} f(d) \\
&\quad \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \sum_{w\in\hat{w}} f(w|d) \log \frac{f(w|d)}{\hat{f}^{(t)}(w|\hat{d})} && (35) \\
&= \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \\
&\quad \sum_{d\in\hat{d}} \sum_{w\in\hat{w}} f(d)f(w|d) \log \frac{f(w|d)}{\hat{f}^{(t)}(\hat{w}|\hat{d})\hat{f}^{(t)}(w|\hat{w})} && (36) \\
&\geq \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \\
&\quad \sum_{d\in\hat{d}} \sum_{w\in\hat{w}} f(d)f(w|d) \log \frac{f(w|d)}{\hat{f}^{(t+1)}(\hat{w}|\hat{d})\hat{f}^{(t+1)}(w|\hat{w})} && (37) \\
&= D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}^{(t+1)}(\mathcal{D}_o, \mathcal{W})) && (38)
\end{aligned}$$

Note: Equation (33) is based on Lemma 2; Equation (34) is based on Equation (28); Equation (35) follows by rearranging the sum and $C_{\mathcal{W}}^{(t)} = C_{\mathcal{W}}^{(t+1)}$; Equation (36) follows by rearranging the sum and total probability theorem ($f(w|\hat{w}) \neq 0$ only for $w \in \hat{w}$); Equation (37) is due to $\hat{f}^{(t)}(w|\hat{w}) = \hat{f}^{(t+1)}(w|\hat{w})$ and

$$\begin{aligned}
&\sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \\
&\quad \sum_{d\in\hat{d}} \sum_{w\in\hat{w}} f(d)f(w|d) \log \frac{1}{\hat{f}^{(t)}(\hat{w}|\hat{d})} \\
&= \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \\
&\quad \left(\sum_{d\in\hat{d}} \sum_{w\in\hat{w}} f(d)f(w|d)\right) \log \frac{1}{\hat{f}^{(t)}(\hat{w}|\hat{d})} && (39) \\
&= \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \hat{f}^{(t+1)}(\hat{d}) \\
&\quad \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \hat{f}^{(t+1)}(\hat{w}|\hat{d}) \log \frac{1}{\hat{f}^{(t)}(\hat{w}|\hat{d})} && (40) \\
&\geq \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t+1)}(d)|d\in\mathcal{D}_o\}} \hat{f}^{(t+1)}(\hat{d}) \\
&\quad \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t+1)}(w)|w\in\mathcal{W}\}} \hat{f}^{(t+1)}(\hat{w}|\hat{d}) \log \frac{1}{\hat{f}^{(t+1)}(\hat{w}|\hat{d})} && (41)
\end{aligned}$$

Equation (41) follows by the non-negativity of the Kullback-Leibler divergence. Note that

$$\begin{aligned}
&\sum_{\hat{w}} \hat{f}^{(t+1)}(\hat{w}|\hat{d}) \log \frac{1}{\hat{f}^{(t)}(\hat{w}|\hat{d})} - \sum_{\hat{w}} \hat{f}^{(t+1)}(\hat{w}|\hat{d}) \log \frac{1}{\hat{f}^{(t+1)}(\hat{w}|\hat{d})} \\
&= \sum_{\hat{w}} \hat{f}^{(t+1)}(\hat{w}|\hat{d}) \log \frac{\hat{f}^{(t+1)}(\hat{w}|\hat{d})}{\hat{f}^{(t)}(\hat{w}|\hat{d})} = D(\hat{f}^{(t+1)}(\hat{\mathcal{W}}|\hat{d})||\hat{f}^{(t)}(\hat{\mathcal{W}}|\hat{d})) && (42)
\end{aligned}$$

□

LEMMA 7. *Theorem 4 holds when $t = 2, 4, \ldots, 2T+2$.*

PROOF. For $t = 2, 4, \ldots, 2T+2$,

$$\begin{aligned}
&D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}^{(t)}(\mathcal{D}_o, \mathcal{W})) + \lambda \cdot D(g(\mathcal{C}, \mathcal{W})||\hat{g}^{(t)}(\mathcal{C}, \mathcal{W})) \\
&= \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t)}(w)|w\in\mathcal{W}\}} \sum_{w\in\hat{w}} \\
&\quad \left(f(w) \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t)}(d)|d\in\mathcal{D}_o\}} \sum_{d\in\hat{d}} f(d|w) \log \frac{f(d|w)}{\hat{f}^{(t)}(d|\hat{w})} \right. \\
&\quad \left. + \lambda \cdot g(w) \sum_{c\in\mathcal{C}} g(c|w) \log \frac{g(c|w)}{\hat{g}^{(t)}(c|\hat{w})} \right) && (43) \\
&\geq \sum_{\hat{w}\in\{C_{\mathcal{W}}^{(t)}(w)|w\in\mathcal{W}\}} \sum_{w\in\hat{w}} \\
&\quad \left(f(w) \sum_{\hat{d}\in\{C_{\mathcal{D}_o}^{(t)}(d)|d\in\mathcal{D}_o\}} \sum_{d\in\hat{d}} f(d|w) \log \frac{f(d|w)}{\hat{f}^{(t)}(d|C_{\mathcal{W}}^{(t+1)}(w))} \right. \\
&\quad \left. + \lambda \cdot g(w) \sum_{c\in\mathcal{C}} g(c|w) \log \frac{g(c|w)}{\hat{g}^{(t)}(c|C_{\mathcal{W}}^{(t+1)}(w))} \right) && (44)
\end{aligned}$$

Equation (44) is based on Equation (29). Using the same argument as Lemma 6, we can prove this lemma. □