修 士 学 位 論 文

Hybrid Method of Semi-supervised Learning and Feature Weighted Learning for Domain Adaptation of Document Classification

平成 27 年度

茨城大学大学院理工学研究科

情報工学専攻

XIAO LIYING

平成 27 年度 茨城大学大学院理工学研究科情報工学専攻　修士学位論文

Hybrid Method of Semi-supervised Learning and Feature Weighted Learning for Domain Adaptation of Document Classification

著者：XIAO LIYING（14NM721X）
指導教員：新納 浩幸 教授

論文要旨

In this paper, for the domain adaptation problems of document classification, we propose a hybrid method of semi-supervised learning and feature weighted learning.

In many of the tasks of natural language processing, supervised learning has been a great success. However, if we want to use a supervised learning for real problems, there is often problems in domain adaptation. In general, the supervised learning is used to create a classifier which is usually using a learning algorithm such as support vector machine (SVM) by labeled training data, then it is possible to identify the label of the test data using this classifier. In this case, the problem is that the domain of training data and test data is different, so it is a problem of domain adaptation.

In general, the method of the domain adaptation can be divided into instancebased method and feature-based method. Although these methods for domain adaptation often work well, while the differences between the domains is small, there may be counterproductive by such an method. When the difference between the domains is small, it is realistic that the problem of domain adaptation is simply regarded as data sparseness problem. In that case, the method of conventional semi-supervised learning and active learning is better.

In this paper, we are dealing with problems of the domain adaptation in document classification. Here, as described above, semi-supervised learning is available for dealing with domain adaptation that difference between domains is small. Especially as semi-supervised learning of document classification, the method using the EM algorithm based on Naive Bayes method is very famous. We refer to this method as NBEM. Here, we also use the NBEM. However, there is still room for improvement because NBEM does not employ valuable information for this task, that is the difference between source domain and target domain. Here, we use the method shown by Chen which has improved the learning of weighting feature. According to the similarity between the label distribution of the feature on source domain and the estimated label distribution of the feature on target domain, we set the weight on the feature to reconstructed training data to perform document classification by NBEM. The domain adaptation of document classification can perform more accurately by this method.

As a result of experiment by using a part of 20 Newsgroups, the effect of this method was confirmed.

平成 27 年度 茨城大学大学院理工学研究科情報工学専攻 修士学位論文

文書分類の領域適応に対する半教師あり学習と素性の重み付け学習のハイブリッド手法

著者：XIAO LIYING（14NM721X）
指導教員：新納 浩幸 教授

論文要旨

本論文では文書分類の領域適応の問題に対して、半教師あり学習と素性の重み付け学習のハイブリッド手法を提案する。

自然言語処理の多くのタスクにおいて、教師付き学習は大きな成功を収めている。ただし教師付き学習を現実の問題に利用する場合、領域適応の 問題が生じることが多い。一般に、教師付き学習ではラベル付きの訓練データから SVM などの学習アルゴリズムを用いて分類器を作成し、その分類器を用いてテストデータのラベルを識別する。この際、訓練データとテストデータの領域が異なる問題が領域適応の問題である。一般に、領域適応の手法は事例ベースの手法と素性ベースの手法に分けられる。これらの領域適応適応の手法はうまく機能する ことも多いが、領域間の違いが小さいときは、こ のような手法を利用することが逆効果になること もある。領域間の違いが小さいときは、領域適応 の問題は単にデータスパースネスの問題と捉えた 方が現実的である。その場合は、従来の半教師あり学習や能動学習の手法がそのまま利用できる。

本論文では文書分類の領域適応の問題を扱う。半教師あり学習が利用できる。特に文書分類の半教師あり学習としては、Naive Bayes 法を基本に EM アルゴリズムを用いる手法（本論文ではこの手法を NBEM と呼ぶ）が効果的であることが知られており、ここでも NBEM を用いる。ただし NBEM は訓練データとテストデータの領域の違いを利用していないために改良の余地がある。ここでは Chen が示した素性の重み付け学習を改良した手法（本論文ではこの手法を STFW と呼ぶ）を利用し、ソース領域上の素性のラベル分布と推定により得られたターゲット領域上の素性のラベル分布の類似性から素性へ重みを付けて訓練データを再構築する。この再構築された訓練データから NBEM を用いて文書分類を行う。これによって文書分類の領域適応が精度よく行うことができる。

20 Newsgroups の一部のデータを利用して実験した結果、NBEM は本タスクに対して効果的であった。また、提案手法は NBEM 法を改善できた。

Contents

Chapter 1

Introduction

In this thesis, for the domain adaption problems of document classification, we propose a hybrid method of semi-supervised learning and feature weighted learning. In many of the tasks of natural language processing, supervised learning has been a great success. However, if we want to use a supervised learning for real problems, there is often problems in domain adaptation. In general, the supervised learning is used to create a classifier which is usually using a learning algorithm such as support vector machine (SVM) by labeled training data, then it is possible to identify the label of the test data using this classifier. In this case, the problem is that the domain of training data and test data is different, so it is a problem of domain adaptation [19].

As a typical example, there is a sentiment analysis task to judge whether a review article for a commodity is positive or not [1]. For example, if we use review articles for "book" as the training data to make a classifier, the classifier can not correctly identify the review articles for "movie" which is in another domain. In addition to the emotion analysis, supervised learning such as morphological analysis [11], parsing [16], word sense disambiguation [18] [10] [9] is utilized in all tasks, it is possible that the domain adaptation problems come into being.

In general, the method of the domain adaptation can be divided into instance-based method and feature-based method [13]. Instance-based method is a method of learning using weighted training data. Learning under covariate shift [20] is typical in this method. The covariate shift means the assumption that $P_S(\boldsymbol{x}) \neq P_T(\boldsymbol{x})$, $P_S(y|\boldsymbol{x}) = P_T(y|\boldsymbol{x})$. Learning under covariate shift is regarded as weighted learning, where the weight is set to the probability density ratio $P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$. The feature-based method is a method that maps the source and target features spaces to a common features space to maintain important characteristics of both

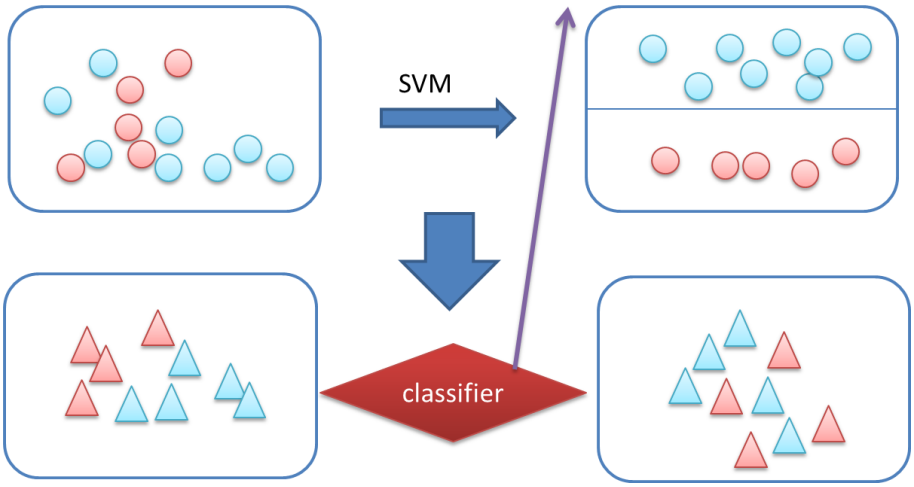

Figure 1.1: Example

domains by reducing the difference between domains. The paper [2] proposed the dimension reduction method called structural correspondence learning (SCL).

The paper [6] offered a weighting system for features . In this study, vector $\boldsymbol{x}_s$ of the training data in the source domain is mapped to an augmented input space $(\boldsymbol{x}_s, \boldsymbol{x}_s, \mathbf{0})$, and vector $\boldsymbol{x}_t$ of the training data in the target domain is mapped to an augmented input space $(\mathbf{0}, \boldsymbol{x}_t, \boldsymbol{x}_t)$. The classifier learned from the augmented vectors solves the classification problem. Daumé's method assumes that an effect can be determined by overlapping the characteristics that are common to the source and target domains.

Although these methods for domain adaption often work well, while the differences between the domains is small, there may be counterproductive by such an method. When the difference between the domains is small, it is realistic that the problem of domain adaption is simply regarded as data sparseness problem. In that case, the method of conventional semi-supervised learning [3] and active learning [17] [14] is better.

In this thesis, we are dealing with problems of the domain adaption in document classification. Here, as described above, semi-supervised learning is available for dealing with domain adaption that difference between domains is small. Especially as semi-supervised learning of document classification, the method using the EM algorithm based on Naive Bayes method is very famous [12]. In this

thesis, we refer to this method as NBEM. Here, we also use the NBEM. However, there is still room for improvement because NBEM does not employ valuable information for this task, that is the difference between source domain and target domain. Here, we use the method shown by Chen [4] which has improved the learning of weighting feature. This method is named as Self-Training Feature Weight, called STFW for short. STFW uses self-learning to estimate the label distribution of features on target domain, but we use NBEM to do it in STFW. The original STFW can be applied to only a binary classification task. For the multi-class classification, we improve STFW. Finally, we use the combination of NBEM and STFW. The domain adaption of document classification can perform more accurately by this. As for the experiment we used the 20 Newsgroups data[1] to construct the domain A and the domain B, and then domain adaption experiments were conducted from domain A to domain B and from domain B to domain A. As a result, NBEM was effective for our task. And the proposed method was able to improve NBEM.

1.1 The Configuration Of the Paper

In this thesis, we will introduce the method we propose and the theory we use, also conduct experiment to compare with the one that use thesaurus which has already existed. Chapter 2 will introduce the method of semi-supervised learning in general and comparation with supervised learning. Chapter 3 will introduce the general domain adaptation and previous research of it. Chapter 4 will explain the hybrid method of NBEM and STFW we proposed in detail. The way of experiment and the result will be written in Chapter 5. In Chapter 6 we will discuss the result of experiment and have a comparison with other method. Chapter 7 is the conclusion.

1.2 Related works

There are some researches using NBEM for domain adaptation of document classification. The Naive Bayes Transfer Classifier (NBTC) modifies EM parts in NBEM to adapt to a target domain [5]. NBTC needs the probability that a test document appears in the source domain. NBTC estimates this probability by using KL divergence between the source domain and the target domain, and empirical

[1] tt http://qwone.com/˜jason/20Newsgroups/

parameters. The Adapting Naive Bayes (ANB) also modifies EM parts in NBEM like NBEM [21]. AMB uses the mixture distribution of the source domain and the target domain as the document generative model. The weight of the source domain is reduced according to EM iterations. As a result, both of NBEM and ANB gives weight to a feature through the class distribution of target domain. On the other hand, our method is based on the idea that the feature must be weighted if the class distribution of a feature in the target domain are similar.

Chapter 2

Semi-supervised Learning of Document Classification

Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent or a physical experiment. The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value.(wiki)

In many of the tasks of natural language processing, supervised learning has been a great success. However, if we want to use a supervised learning for real problems, there is often problems in domain adaptation. And as the two of the most famous method of semi-supervised learning, here we introduce Naive Bayes Classifer and Transductive support vector machines(TSVM).

2.1 Naive Bayes Classifer

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence

assumptions between the features.

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes ' theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes ' theorem states the following relationship:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{2.1}$$

Using the naive independence assumption that

$$P(x_i|y, x_1, ..., x_i - 1, x_i + 1, ..., x_n) = P(x_1|y) \tag{2.2}$$

for all ı this relationship is simplified to

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, ...x_n)} \tag{2.3}$$

Since $P(x_1, ..., x_n)$ is constant given the input, we can use the following classification rule:

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \tag{2.4}$$

$$\hat{y} = argmaxP(y)\prod_{i=1}^{n} P(x_i|y) \tag{2.5}$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i|y)$; the former is then the relative frequency of class in the training set. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i|y)$.

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features.

2.2 Transductive Method

Like semi-supervised learning,transductive learning is another mothod using unlabeled data in improve the classifier learned through labeled data. And then as a representative method of transductive learning, there is Transductive-SVM(TSVM).

Transductive support vector machines(TSVM) has been widely used as a means of treating partially labeled data in semi-supervised learning.TSVM seeks the largest separation between labeled and unlabeled data through regularization. In em-pirical studies, it performs well in text classification(Joachims, 1999) but can perform substantially worse than its supervised counterpart SVM (Cortes and Vap-nik, 1995) in other applications (Wu, Bennett, Cris-tianini and Shawe-Taylor, 1999). This unstable per-formance has been criticized.

In the section below, we will introduce the result of experiment compared with NBEM.

2.3 EM algorithm in semi-supervised learning

In the semi-supervised setting with labeled and unlabeled data, we would still like to find MAP parameter estimates, as in the supervised setting above. Because there are no labels for the unlabeled data, the closed-form equations from the previous section are not applicable. However, using the Expectation-Maximization (EM) technique, we can find locally MAP parameter estimates for the generative model.

Using X_u to refer to the unlabeled example, and X_l to refer to the examples for which labela are given. This algorithm is summarized as following.

1. Inputs: Collections X_l of documents and X_u of unlabeled documents.

2. Build an initial naive Bayes classifier, $\hat{(}\theta)$,from the labeled documents, X_l, only. Use maximum a posteriori parameter estimation to find $\hat{(}\theta) = argmaxP(X_l|\theta)P(\theta)$. The equations is,

$$\hat{(}\theta)_{w_t|c_j} \equiv \frac{1 + \sum_{x_i \in X} \delta_{ij} x_{it}}{|\chi| + \sum_{s=1}^{|\chi|} \sum_{x_i \in X}^{|\chi|} \delta_{ij} x_{is}} \tag{2.6}$$

3. Loop while classifier parameters improve, as measured by the change in $l(\theta|X, Y)$, the log probability of the labeled and unlabeled data, and the piror, the equation is

$$l(\theta|X,Y) = log(P(\theta)) + \sum_{x_i \in X_u} log \sum_{j \in [M]} P(c_j|\theta)P(x_i|c_j;\theta) + \sum_{x_i \in X_l} log(P(y_i = c_j|\theta)P(x_i|y_i = c_j;)) \tag{2.7}$$

3.1(E-step)Use the current classifer, $\hat{(}\theta)$,given the estimate component membership of each unlabeled document,i.e.,the probability that each mixture component (and class) generated each document, $P(c_j|x_i;\theta)$.

The equaption is,

$$P(y_i = c_j|x_i;\hat{(}\theta)) = \frac{P(c_j|\hat{(}\theta)P(x_i|c_j;\hat{(}\theta))}{P(x_i|\hat{(}\theta)} = \frac{P(c_j|\hat{(}\theta))\prod_{w_t \in \chi} P(w_t|c_j;\hat{(}\theta))^x_{it}}{\sum_{k=1}^{M} P(c_k|\hat{(}\theta))\prod_{w_t \in X} P(w_t|c_k;\hat{(}\theta))^x_{it}} \tag{2.8}$$

3.2 (M-step)Re-estimate the classier,$\hat{(}\theta)$,given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{(}\theta) = argmaxP(X,Y|\theta)P(\theta)$.(See the equaptions above)

4. Output:A classifier,$\hat{\theta}$, that tasks an unlabeled document and predicts a class label.

Chapter 3

Domain adaptation

Domain Adaptation is a field associated with machine learning and transfer learning. This scenario arises when we aim at learning from a source data distribution a well performing model on a different (but related) target data distribution. For instance, one of the tasks of the common spam filtering problem consists in adapting a model from one user (the source distribution) to a new one who receives significantly different emails (the target distribution). Note that, when more than one source distribution is available we talked about multi-source domain adaptation.

Although supervised learning technique can be used in many tasks of natural language, there are problems of domain adaptation existing. The problem of domain adaptation is that the domain of source data which is used to learn as the training data is different from the domain of target data which the classifier get from learning applies in. research has been conducted Actively in recent years.

In general, the method of the domain adaptation can be divided into instance-based method and feature-based method [13]. Instance-based method is a method of learning using weighted training data. Learning under covariate shift [20] is typical in this method. The covariate shift means the assumption that $P_S(\boldsymbol{x}) \neq P_T(\boldsymbol{x})$, $P_S(y|\boldsymbol{x}) = P_T(y|\boldsymbol{x})$. Learning under covariate shift is regarded as weighted learning, where the weight is set to the probability density ratio $P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$. The feature-based method is a method that maps the source and target features spaces to a common features space to maintain important characteristics of both domains by reducing the difference between domains. The paper [2] proposed the dimension reduction method called structural correspondence learning (SCL). The paper [6] offered a weighting system for features . In this study, vector $\boldsymbol{x}_s$ of the training data in the source domain is mapped to an augmented input space $(\boldsymbol{x}_s, \boldsymbol{x}_s, \boldsymbol{0})$, and vector $\boldsymbol{x}_t$ of the training data in the target domain is mapped to

an augmented input space $(\mathbf{0}, \boldsymbol{x}_t, \boldsymbol{x}_t)$. The classifier learned from the augmented vectors solves the classification problem. Daumé's method assumes that an effect can be determined by overlapping the characteristics that are common to the source and target domains.

On the one hand, the problem of domain adaptation can also be regarded as the problem of the sparsity of training data. Because of this, self-training, semi-supervised learning as well as active learning can also be used in the domain adaptation. Especially Self-Training is very useful, because it is possible that in unsupervised learning of adaptation domain it is unnecessary to label the data.

While use Self-Training in domain adaptation, Chen has propose the method to improve the effect of learning by setting the weight to feature.

3.1 Self-training

Self-Training is useful as a method of unsupervised learning of adaptation domain that it is not necessary to label the data. Now, we set labeled data to L and unlabeled data to U. So in the initial state, L is labeled data of source domain, and U is unlabeled data of target domain.

At each step of the Self-Training, the classifier h is learned from the training data L, and applied to unlabeled data U. Here we give the reliability to each data of U. Assign the label identified by h to c pieces of high reliability. And then add the c pieces of data to L. Or this step make U empty, and ends when reliability for all data comes to below a certain threshold.

In Self-Training, the way to assign reliability and the setting of c is necessary. We take advantage of SVM as a learning algorithm. And assign reliability by the distance to the separating hyperplane obtained by SVM.

3.2 Self-training with feature weighted

As the weight of the feature, Chen used the feature and class label and their correlation coefficient. First, we set the value of feature f of data $\boldsymbol{x}$ to x_f, set the class of data $\boldsymbol{x}$ to y_x. We regard the correlation coefficient of x_f and y_x as $\rho_S(x_f, y_x)$ for labeled data in source domain. About the data $\boldsymbol{x}$ in target domain, its class is substituted for the class which estimated by self-learning y'_x, and we obtain the correlation coefficient $\rho_T(x_f, y'_x)$ of x_f and y'_x. Then the weight $w(f)$ of feature f is defined as the following.

$$w(f) = \frac{1 + \rho_S(x_f, y_x)\rho_T(x_f, y'_x)}{2} \tag{3.1}$$

Then we use $w(f)$ to get the new x_f.

$$x_f ? x_f + \gamma_n w(f) \tag{3.2}$$

Here n means to the steps of Self-Training, γ is satisfied to $gamma_n$. $gamma_n$ is defined as following.

$$\gamma_n = 0.01\frac{(}{N} - n)N \tag{3.3}$$

N here is the maximum value of steps of Self-Training. And after updating x_f for each feature f, normalize $x$to 1.

3.3 Extension of Self-training with feature weighted

Here we explain the new method of weighting using the distribution of the weight of the feature for the class distribution. The weight, in both of the source domain and the target domain, will be large if there is some feature used in a same way, Conversely if it is different, we will set it small. As for conventional method, it is not able to extend in the task of multi-class classification, in the method we proposed can be adapted in the multi-class classification.

In this thesis, we define the $w(f)$ by the following procedure.

$$\hat{i} = argmax P_S(x, y_i) \tag{3.4}$$

$$\hat{j} = argmax P_T(x, y_j) \tag{3.5}$$

If $\hat{i}$ and $\hat{j}$ are equal,

$$w(f) = max(P_S(x, y_i), P_T(x, y_i)) \tag{3.6}$$

If they are not equal,

$$w(f) = 0 \tag{3.7}$$

Here in data x of source domain S, the set of data x which class is i is written as S_i, P_S is defined as following.

$$P_S(x, y_i) = \frac{\sum_{xinS_i} x_f}{\sum_{x \in S} x_f} \tag{3.8}$$

About data x of target domain T, their class are unknown. So we think the class we get by classifier that time if the class of x. In T, the set of data x which class is i is written as T_i, and P_T is defined as following.

$$P_T(x, y_i) = \frac{\sum_{xinT_i} x_f}{\sum_{xinT} x_f} \tag{3.9}$$

3.4 The Problem of Sparsness

This part describes the method of increasing the accuracy by the external resources to the domain adaptation. Instead of problem of adaptation, in many tasks of natural language processing, there is a sparseness problem that the number of dimensions of training data is less than the number of training cases. It is often not able to build an accurate model because of this problem. In order to solve the sparseness problem, thesaurus is used in general. The thesaurus is the things which built by hand such as classification vocabulary table and which built from corpus automatically. The former has a high quality and exists the problem of field dependence. The latter one ' s quality is not so high, but has an advantage that can be constructed for each sector. In this thesis, we do the experiment using things built by hand such vocabulary classification and classify automatically and its comparative experiments.

Chapter 4

Hybrid method of NBEM and STFW

4.1 NBEM

NBEM is one of the semi-supervised learning for learning a classifier from a little labeled training data and much unlabeled data. Generally speaking, it is an method that learn the classifier of Naive Bayes from labeled training data, and use a large amount of unlabeled data and EM algorithm to improve this classifier.

In a classification problem, let $C = \{c_1, c_2, \cdots, c_m\}$ be a set of classes. An instance x is represented as a feature list

$$\boldsymbol{x} = (f_1, f_2, \cdots, f_n). \tag{4.1}$$

We can solve the problem classification by estimating the probability $P(c|x)$. Actually, the class c_x of $\boldsymbol{x}$, is given by

$$c_x = \arg\max_{c \in C} P(c|\boldsymbol{x}). \tag{4.2}$$

Bayes theorem shows that

$$P(c|\boldsymbol{x}) = \frac{P(c)P(\boldsymbol{x}|c)}{P(\boldsymbol{x})}. \tag{4.3}$$

As a result, we get

$$c_x = \arg\max_{c \in C} P(c)P(\boldsymbol{x}|c). \tag{4.4}$$

In the equation above, $P(c)$ could be estimated easily; the question is how to estimate $P(\boldsymbol{x}|c)$. Naive Bayes models assume the following:

$$P(\boldsymbol{x}|c) = \prod_{i=1}^{n} P(f_i|c). \tag{4.5}$$

The estimation of $P(f_i|c)$ is easy, so we can estimate $P(\boldsymbol{x}|c)$.

We can use the EM method if we use Naive Bayes for classification problems. In this thesis, we show only key equations and the key algorithm of this method [12].

Basically the method computes $P(f_i|c_j)$ where f_i is a feature and c_j is a class. This probability is given by [1]

$$P(f_i|c_j) = \frac{1 + \sum_{k=1}^{|D|} N(f_i, d_k) P(c_j|d_k)}{|F| + \sum_{m=1}^{|F|} \sum_{k=1}^{|D|} N(f_m, d_k) P(c_j|d_k)}. \tag{4.6}$$

D**:** all data consisting of labeled data and unlabeled data
d_k**:** an element in D
F**:** the set of all features
f_m**:** an element in F
$N(f_i, d_k)$**:** the number of f_i in the instance d_k.

In our problem, $N(f_i, d_k)$ is 0 or 1, and almost all of them are 0. If d_k is labeled, $P(c_j|d_k)$ is 0 or 1. If d_k is unlabeled, $P(c_j|d_k)$ is initially 0, and is updated to an appropriate value step by step in proportion to the iteration of the EM algorithm.

By using equation 4.6, the following classifier is constructed:

$$P(c_j|d_i) = \frac{P(c_j) \prod_{f_n \in K_{d_i}} P(f_n|c_j)}{\sum_{r=1}^{|C|} P(c_r) \prod_{f_n \in K_{d_i}} P(f_n|c_r)}. \tag{4.7}$$

In this equation, K_{d_i} is the set of features in the instance d_i.

$P(c_j)$ is computed by

$$P(c_j) = \frac{1 + \sum_{k=1}^{|D|} P(c_j|d_k)}{|C| + |D|}. \tag{4.8}$$

[1] This equation is smoothed by taking into account the frequency 0.

The EM algorithm computes $P(c_j|d_i)$ by using equation 4.7 (E-step). Next, by using equation 4.6, $P(f_i|c_j)$ is computed (M-step). By iterating E-step and M-step, $P(f_i|c_j)$ and $P(c_j|d_i)$ converge. In our experiment, when the difference between the current $P(f_i|c_j)$ and the updated $P(f_i|c_j)$ comes to less than $8{\cdot}10^{-6}$ or the iteration number reaches 10 times, we judge that the algorithm has converged.

4.2 STFW

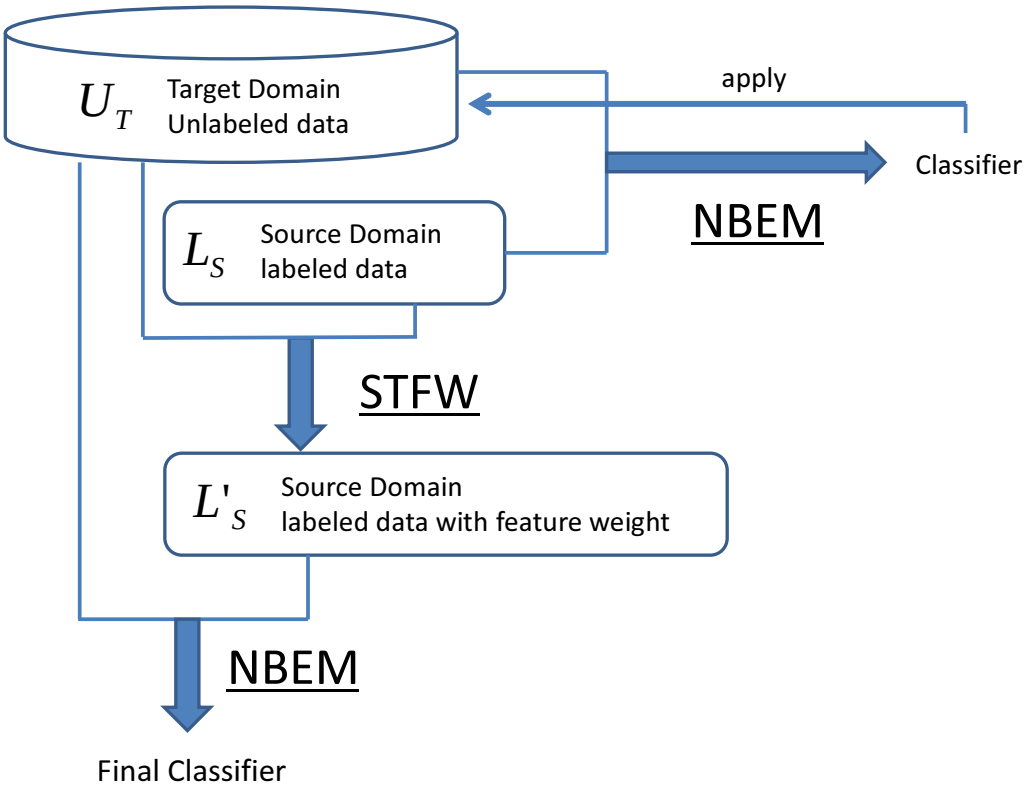

Figure 4.1: Hybrid method of NBEM and STFW

In this thesis, we improved STFW proposed by Chen. STFW is a feature-based method which is effective in domain adaption. In essence, feature-based method can be regarded as a method which maps the common space of feature

between the space of target domain and the source domain. As for the operation, we corresponds to weighting the feature, so intuitively, it is also considered as an method that set a weight to feature that is effective to identification in both domains of the source domain and the target domain. Chen set weight to the feature in the following ways. First, we set the value of feature f of data $\boldsymbol{x}$ to x_f, set the class of data $\boldsymbol{x}$ to y_x. We regard the correlation coefficient of x_f and y_x as $\rho_S(x_f, y_x)$ for labeled data in source domain. About the data $\boldsymbol{x}$ in target domain, its class is substituted for the class which estimated by self-learning y'_x, and we obtain the correlation coefficient $\rho_T(x_f, y'_x)$ of x_f and y'_x. Then the weight $w(f)$ of feature f is defined as the following.

$$w(f) = \frac{1 + \rho_S(x_f, y_x)\rho_T(x_f, y'_x)}{2} \tag{4.9}$$

A new value v_{new} of the feature come to be obtained by multiplying the weight:

$$v_{new} = w(f) \cdot v_{old} \tag{4.10}$$

Note $v_{new} = 0$ if $v_{old} = 0$ in the equation 4.10.

Chen's method uses a correlation coefficient $\rho_S(x_f, y_x) \lesseqgtr \rho_T(x_f, y'_x)$ to define the weight. Because the label is a categorical value, in fact, only binary classification can be targeted. Based on Chen's method here, it is defined of weighting that it also can be used in the multi-class classification. The weight Chen defined can be regarded that measured the similarity of the label distribution P_s of feature f in source domain and label distribution P_t of feature f in target domain. The P_s is the distribution of the following set:

$$\{y_x | x \ in \ Source \ data \ set, x_f > 0\}. \tag{4.11}$$

The P_t can be defined by the same way.

Therefore, thesis, first, define the distance $d(f)$ between P_s and P_t as following:

$$d(f) = |P_s - P_t|. \tag{4.12}$$

Then set the weight by using $d(f)$. However, our task is document classification. We use Naive Bayes as a learning algorithm, so the value of feature becomes frequency. Therefore, the value of feature (i.e. the weight) is desirably an integer of 0 or more. As a result, we define the new value v_{new} of the feature as follows:

$$v_{new} = \begin{cases} v_{old} + 1 & if \quad d(f) < \theta_1, v_{old} > 0 \\ v_{old} - 1 & if \quad d(f) > \theta_2, v_{old} > 0 \\ v_{old} & if \quad others \end{cases}$$

However, if v_{new} is a negative number after minus 1, $v_{new} = 0$. In the experiments of this paper, the parameter θ_1 and θ_2 was set to 0.2 and 1.5 respectively. These values were obtained through some experiments [2].

Also because there is no label of the data in target domain, P_t can not simply obtained. Chen labeled the data in target domain by self-learning, and seeking P_t only on reliable data. In this thesis, we do not use self-learning, but the classifier learned by NBEM. And it is not only limited to those reliable data, all of the data will be used to estimate P_t.

4.3 Combination of NBEM and STFW

In this thesis we propose an method that uses a combination of NBEM and STFW, referring to Figure 6.2.

First, we learn a classifier by using the NBEM against labeled training data L_S of the source domain and unlabeled data U_T of the target domain. Use this classifier to estimate the label of U_T.

Using this label estimated, we set a weight to the feature of L_s by STFW, and construct new training data L'_S.

[2]The parameter θ_1 and θ_2 depend on the number of classes. In the experiments of this paper, all of the number of classes are three.

Chapter 5

Experiment

Table 5.1: Experimental results (%)

	NB (S-only)	NBEM	NBEM+STFW	NB (T-only)
X → Y	72.83	90.00	**92.33**	94.67
Y → X	81.17	82.67	**82.83**	90.00

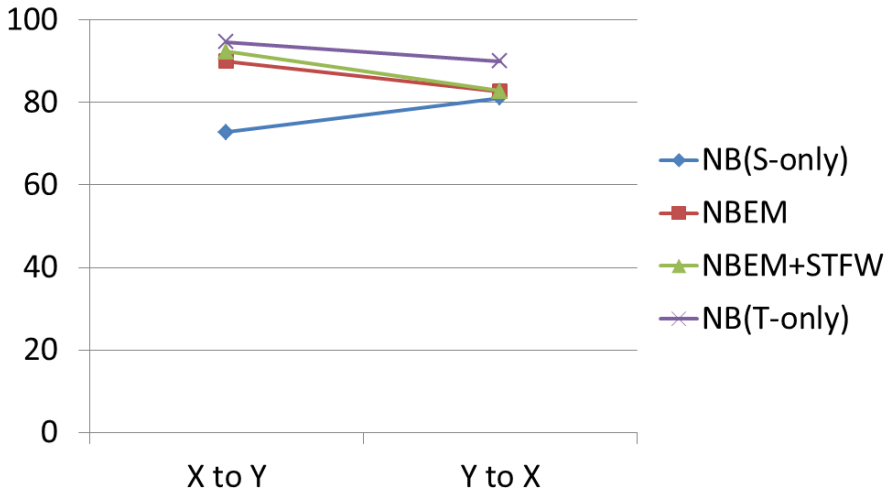

Figure 5.1: The graph of Experiment result

It took out a 20 Newsgroups data set [1] from the document group of following six categories in our experiment. Symbols in parentheses refer to the class name.

```
A: comp.sys.ibm.pc.hardware (comp)
B: rec.sport.baseball  (rec)
C: sci.electronics (sci)
D: comp.sys.mac.hardware  (comp)
E: rec.sport.hockey  (rec)
F: sci.med  (sci)
```

We suppose the dataset of `(A,B,C)` to domain X, and the dataset of `(D,E,F)` to domain Y. Each domain has become a dataset of the document classification that `L = {comp, rec, sci}` is the class label set.

The document number (the number of data) of each document group is shown in Table 5.2. Although the class distribution of labeled training data is uniform in each domain, Class distribution of the test data which can fit the problem of reality was set to be different in each domain.

On the one hand, in domain adaption which is from domain X to domain Y, labeled data of A, B, C becomes training data (a total of 300 documents), and the unlabeled data of D, E, F is unlabeled data(a total of 900 documents) which can be used. Then the test data of D,E,F is used as test data (a total of 600 documents). On the other hand, in domain adaption which is from domain Y to domain X, labeled data of D, E, F becomes training data (a total of 300 documents), and the unlabeled data of A, B, C is unlabeled data(a total of 900 documents) which can be used. Then the test data of A, B, C is used as test data (a total of 600 documents).

Table 5.2: Number of data of each document group

	Labeled data	Unlabeled data	Test data
A	100	400	300
B	100	300	200
C	100	200	100
D	100	200	100
E	100	400	300
F	100	300	200

[1] http://qwone.com/~jason/20Newsgroups/

The results of the experiment is shown in table5.1.

The column of NB (S-Only) learns the classifier only from the training data of the source domain by Naive Bayes, has been written of the accuracy rate of test data identified. The column of NBEM is the accuracy rate using the training data and unlabeled data by NBEM, the column of NBEM+STFW is accuracy rate by hybrid method of NBEM and STFW proposed in this thesis. The effect of the method proposed in Table5.1 can be confirmed. Also as reference accuracy rate that it learn the classifier from training data of target domain by Naive Bayes is shown in NB (T-Only). These values have shown the accuracy rate of supervised learning in the case of the usual problems of domain adaption have not occurred.

Chapter 6

Discussion

6.1 Comparison with transductive method

Like semi-supervised learning, transductive learning is another method using unlabeled data in order to improve the classifier learned through labeled data. And then as a representative method of transductive learning, there is Transductive-SVM (TSVM) [7].

In this thesis, although we use NBEM of semi-supervised learning, it is also possible to use the TSVM instead of NBEM.

Table 6.1: Another method using unlabeled data

	NB	NBEM	SVM	TSVM
X → Y	72.83	90.00	75.83	66.50
Y → X	81.17	82.67	71.16	70.83

Generally SVM has a higher accuracy than NB. However, NB sometimes has high accuracy in the case of document classification. In fact, in domain adaption of Y → X, NB is better than SVM. When using NB for document classification, it is better that documents simply represent by a bag of words. Thus, using SVM, it becomes necessary to make some processing. In the experiment using SVM above, we set the vector value by TF*IDF, and finally normalize the size of the vector to 1.

TSVM does not improve the accuracy of the SVM, conversely the accuracy

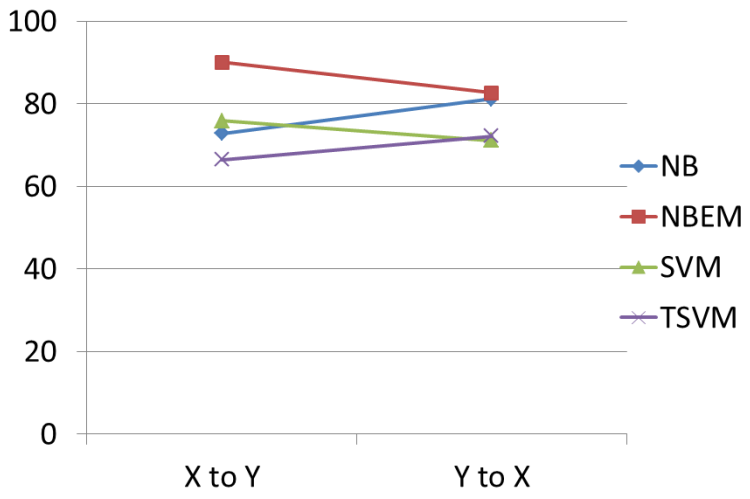

Figure 6.1: The graph of Experiment result comparing with another method

become lower. It is because that TSVM assumes that the class distribution of test data and training data is the same, but this assumption is not satisfied in our experiments.

6.2 Comparison with other methods of domain adaption

The method of domain adaption can be classified to feature-based method and instance-based method. In this section we apply a feature-based method and an instance-based method, and compare them with our proposed method.

As a feature-based method, we use the structural correspondence learning (SCL) [2]. This is the representative feature-base method. On the other hand, the typical instance-based method is learning by covariate shift. In learning by covariate shift, the calculation of the probability density ratio become the key point. Here we use a density calculation method named Unconstrained Least Squares Importance Fitting (uLSIF) [8].

The result of experiment is shown in Table 6.2. NBEM+STFW in the table is the our proposed method.

Table 6.2: Other domain adaptation methods

	NBEM+STFW	SVM	SCL	uLSIF
X → Y	**92.33**	75.83	74.33	73.67
Y → X	**82.83**	71.16	71.83	72.17

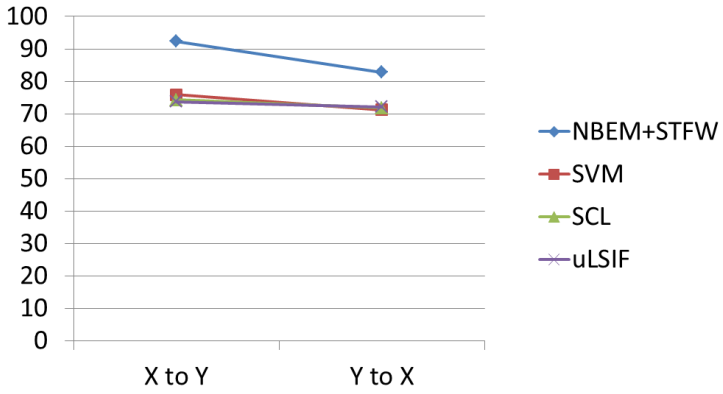

Figure 6.2: The graph of Experiment result comparing with otherdomain adaptation method

As a result of SCL and uLSIF has not changed a lot that both of them is based of SVM, there is a high overwhelmingly accuracy toward NBEM+STFW. Here we can see the great difference of the results is because that whether the base of the learning algorithm is SVM or NB. NB made a higher accuracy than SVM just in our task. Both of SCL and uLSIF are transductive method, although the test data in target domain is used in the process of learning, the unlabeled data are not used. On the other hand, NBEM+STFW does not use test data, but unlabeled data. Test data is also unlabeled data, but the former is smaller than the latter. In this experiment, the amount of unlabeled data is 1.5 times of the amount of test data. Therefore it can be considered one reason that NBEM+STFW is better than SCL and uLSIF.

6.3 Weighting to feature

In this thesis we give a weight to the feature likely to be valid for identification in domain adaption, subtract the weight of the feature likely to make an adverse effect on identification.

Here we examined the points following:

- Weighting to Test Data
- Size of the Added Weight
- Negative Weights

We show results of the experiment in turn below.

Weighting to Test Data

In this thesis we set the weight to features of training data only, but it is also conceivable to the test data. The result of the experiment is shown in Table 6.3.

Table 6.3: Weighting to Test Data (TW)

	NBEM+STFW (without TW) - our method -	NBEM+STFW (with TW)
X → Y	**92.33**	91.17
Y → X	82.83	**83.00**

Weighting to the test data is effective to domain adaption of Y → X, but it is not effective of X → Y.

Size of the Added Weight

In this thesis, giving a weight means to plus 1, here we change it to plus 2, and the result of the experiment is shown in Table 6.4.

Table 6.4: Change the Size of the Added Weight

	NBEM+STFW (+1) - our method -	NBEM+STFW (+2)
X → Y	92.33	**93.33**
Y → X	82.83	82.83

While we make the twice of the weight, it is effective in domain adaption of X → Y, but it is not effective in Y → X.

Negative Weights

In domain adaption, there may be some labeled data which creates an adverse result in learning. This is called 'negative transfer' [15]. Our method is designed on the based on 'negative transfer.' That is, if the difference between class distributions of feature on the source domain and the target domain is quite big, we assign the feature negative weight (-1), In order to investigate the effect of negative weights here, we make an experiment which did not assign negative weight. And its result is shown in Table6.5.

Table 6.5: The Effect of Negative Weight (NW)

	NBEM+STFW (with NW) - our method -	NBEM+STFW (without NW)
X → Y	92.33	**93.00**
Y → X	**82.83**	82.67

Without negative weight, although it is effective in domain adaption of X → Y, it is not effective of Y → X.

It can be confirmed that the accuracy is subtly changed by the way of setting weight and its value.

Chapter 7

Conclusion

In this thesis, for the domain adaption problems of document classification, we proposed a hybrid method of semi-supervised learning and feature weighted learning. NBEM is used to learn a classifier, and then the learned classifier and SFTW reconstruct training data, and then the final classifier is learned by using the reconstruct training data and NBEM again. As a result of experiment by using a part of 20 Newsgroups, the effect of our method was confirmed. As for challenges in the future, we need to discover an more appropriate setting way and a better size of weight.

Acknowledgments

My deepest gratitude goes first and foremost to Professor Shinnou, my supervisor, for his guidance and encouragement. He has walked me through all the stages of the writing of this thesis. Without his consistent and illuminating instruction, this thesis could not have reached its present form. I also owe my sincere gratitude to the members in our research room who gave me the help and time in listening to me and helping me work out my problems during the difficult course of the thesis.

Bibliography

[1] John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

[2] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *EMNLP-2006*, pages 120–128, 2006.

[3] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.

[4] Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *NIPS-2014*, pages 2456–2464, 2011.

[5] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring Naive Bayes Classifiers for Text Classification. In *AAAI-2007*, 2007.

[6] Daumé III, Hal. Frustratingly Easy Domain Adaptation. In *ACL-2007*, pages 256–263, 2007.

[7] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[8] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

[9] Kanako Komiya and Manabu Okumura. Automatic Determination of a Domain Adaptation Method for Word Sense Disambiguation using Decision Tree Learning. In *IJCNLP-2011*, pages 1107–1115, 2011.

[10] Kanako Komiya and Manabu Okumura. Automatic Domain Adaptation for Word Sense Disambiguation Based on Comparison of Multiple Classifiers. In *PACLIC-2012*, pages 75–85, 2012.

[11] Shinsuke Mori. Domain adaptation in natural language processing (in japanese). *The Japanese Society for Artificial Intelligence*, 27(4):365–372, 2012.

[12] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[13] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[14] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

[15] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, volume 898, 2005.

[16] Kenji Sagae and Jun'ichi Tsujii. Dependency parsing and domain adaptation with lr models and parser ensembles. In *EMNLP-CoNLL-2007*, pages 1044–1050, 2007.

[17] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[18] Hiroyuki Shinnou, Yoshiyuki Onodera, Minoru Sasaki, and Kanako Komiya. Active Learning to Remove Source Instances for Domain Adaptation for Word Sense Disambiguation. In *PACLING-2015*, pages 156–162, 2015.

[19] Anders Søgaard. *Semi-Supervised Learning and Domain Adaptation in Natural Language Processing*. Morgan & Claypool, 2013.

[20] Masashi Sugiyama and Motoaki Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation* . MIT Press, 2011.

[21] Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In *Advances in Information Retrieval*, pages 337–349. 2009.