2008 Eighth IEEE International Conference on Data Mining

A Robust Discriminative Term Weighting based Linear Discriminant Method for Text Classification

Khurum Nazir Junejo and Asim Karim
Dept. of Computer Science
LUMS School of Science and Engineering
Lahore, Pakistan
{junejo, akarim}@lums.edu.pk

Abstract

Text classification is widely used in applications ranging from e-mail filtering to review classification. Many of these applications demand that the classification method be efficient and robust, yet produce accurate categorizations by using the terms in the documents only. We present a supervised text classification method based on discriminative term weighting, discrimination information pooling, and linear discrimination. Terms in the documents are assigned weights according to the discrimination information they provide for one category over the others. These weights also serve to partition the terms into two sets. A linear opinion pool is adopted for combining the discrimination information provided by each set of terms yielding a two-dimensional feature space. Subsequently, a linear discriminant function is learned to categorize the documents in the feature space. We provide intuitive and empirical evidence of the robustness of our method with three term weighting strategies. Experimental results are presented for data sets from three different application areas. The results show that our method's accuracy is higher than other popular methods, especially when there is a distribution shift from training to testing sets. Moreover, our method is simple yet robust to different application domains and small training set sizes.

1 Introduction

Automatic content based text classification into predefined categories is becoming extremely useful with the increasing availability of text documents in digital formats such as Web pages, e-mails, Web blogs, digital libraries, and corporate text databases. A common application of text classification is information filtering where a stream of documents (e.g. e-mails) is classified before or after reaching its destination. Other applications of text classification include document organization, web page categorization, and query classification. More recently, text classification has been used for semantic analysis of documents such as review documents' categorization as positive or negative and word sense disambiguation. The scope and scale of text classification applications is bound to increase in the future as more text documents in digital formats become available.

The prototypical text classification problem can be defined as follows. Given a set of labeled text documents $L = \{\langle \mathbf{x}_i, c_i \rangle\}_{i=1}^{|L|}$ where $c_i \in C = \{1, 2, \ldots, |C|\}$ denotes the category of document $\mathbf{x}_i$ and $|C|$ and $|L|$ are the total number of predefined categories and labeled documents; learn a classifier that assigns a category label from 1 to $|C|$ to each document in the set $U = \{\langle \mathbf{x}_i \rangle\}_{i=1}^{|U|}$. This is a supervised learning setting in which it is assumed that the joint probability distribution of documents and categories is identical in sets U and L (although this is not guaranteed in practice for some applications). In other words, the task is to learn to approximate the unknown target function $\Phi^{'} : U \rightarrow \{1, 2, \ldots, |C|\}$ by the classifier function $\Phi : U \rightarrow \{1, 2, \ldots, |C|\}$ such that the number of documents in U for which $\Phi(\mathbf{x}_j) \neq \Phi^{'}(\mathbf{x}_j)$ is a minimum. A document is represented as a 0/1 vector $\mathbf{x}_i = \langle x_{i1}, x_{i2}, \ldots, x_{i|T|} \rangle$ where $x_{ij} \in \{0, 1\}$ indicates whether the term (typically a word) j exists in document i or not. The integer $|T|$ is the number of terms in the dictionary of L and U (after standard preprocessing of stop word removal and stemming). The terms and categories are assumed to be just symbolic labels without semantics and that no additional knowledge of a procedural or declarative nature is available.

Text classification, as defined above, is challenging for two reasons. First, the dimensionality of the term space ($|T|$) is large and the set of documents L is sparsely represented in it. Second, selecting the relevant terms and their weights (relative importance) for the classification. Different methods for text classification address these challenges

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.26

in different ways. Besides these challenges, text classification methods must be efficient in order to handle large volumes of data in various applications. Furthermore, text classification methods must be robust to small sizes of labeled sets and differing distributions of labeled and unlabeled sets.

In this paper, we present a robust and efficient text classification method based on discriminative term weighting, discrimination information pooling, and linear discrimination in a two-dimensional feature space. Three term weighting strategies are investigated for discriminating and partitioning the terms: odds, log-odds, and Kullback-Leibler divergence. These strategies weigh the terms based on the discrimination information they provide for one category over the others. A two-dimensional one-category-versus-others feature space is constructed as the weighted sum of terms. This transformation is based on a technique for combining experts' opinions known as linear opinion pool. The classification is then learned in the feature space by a simple linear discriminant function. Our method is a hybrid generative-discriminative method where the term weights represent a generative model and the linear discriminant represents a discriminative model of the classification problem. We evaluate our method on three data sets belonging to three different application areas - spam filtering, movie review, and SRAA. The results are compared with four common text classification methods, demonstrating the overall effectiveness of our method with improved classification accuracies.

The rest of the paper is organized as follows. We present the motivation and related work in Section 2. Our text classification method, DTWC, is described in Section 3, including a comparison with the naive Bayes classifier. We describe the data sets and evaluation setup in Section 4. Section 5 presents the results of our evaluations and comparisons with other methods. We conclude in Section 6.

2 Related Work and Motivation

Text classification has been studied extensively in the literature. A comprehensive review of text classification methods is given in [28]. Here we focus on document representation, feature selection, supervised methods, and generative-discriminative methods. Many text classification methods use the "bag-of-words" representation that describes a document by a term vector where each term (typically a word) is given a weight (term position information is not preserved). The common weighting techniques include term occurrence (binary), term frequency, and term-frequency-inverse-document frequency [27, 23]. Our method can work with any weighting technique as long as a term vector representation is used. In this paper, however, we restrict ourselves to the binary term vector representation that has been shown to produce more accurate classifiers in some settings [16].

Regardless of the document representation approach, the dimensionality of the term space is very large for text classification problems. Feature selection and dimensionality reduction for text classification has been studied extensively [5, 11, 7]. Techniques for feature selection and dimensionality reduction can be supervised or unsupervised depending on whether they require class information. However, for the text classification problem setting discussed in this paper, supervised techniques are more commonly used [2, 8, 4, 9]. These techniques rely on class information and information theoretic measures, such as entropy, to identify high relevance terms. Our term weighting and selection technique belongs to this latter category of techniques. In particular, we weigh each term by the discrimination information it provides for discriminating between one category and the rest. The weights also serve to partition the terms into two sets, and they can be thresholded for term selection and dimensionality reduction. A novel information pooling technique is adopted to aggregate the discrimination information of each set to form a two-dimensional feature space in which a linear discriminant function is learned. The approach of learning terms' weights from training data based on their distributions in the two categories appears to have been first proposed by [8]. They present information theoretic functions to replace the IDF component of the TFIDF term weighting strategy and use these weights in the classification model. In this work, we focus on discrimination information measures of weighting the terms for both selection and classification.

Supervised text classification methods can be based on a generative or discriminative model of the problem. The most common generative methods are naive Bayes and maximum entropy [29, 17, 24]. The naive Bayes classifier results from the application of the Bayes rule with the assumption that each term is independent of the others given the category label, while the maximum entropy method estimates the class conditional distributions by maximizing the entropy among them. The most popular discriminative method for text classification is support vector machine (SVM) [14]. SVM, which is based on statistical learning theory and structural risk minimization, learns a maximum margin linear discriminant in a high dimensional feature space. The balanced winnow method is another example of a discriminative method that learns a linear discriminant in the term space by minimizing the mistakes made by the classifier [6].

There has been continuing interest in hybrid generative-discriminative methods [12, 26, 19, 21]. These methods try to exploit the strengths of generative and discriminative methods by first learning the data distribution and then building a discriminative classifier using the learned distribution. Several variants of this general concept have been explored with promising results. Our method can also be categorized as a hybrid generative-discriminative method. However, our method is simpler and efficient requiring fewer parameters and learns faster.

3 DTWC – Our Text Classification Method

In this section, we describe our text classification method based on discriminative term weighting and linear discrimination. Our method, subsequently referred to as DTWC, addresses the key issues of high dimensionality, term weighting and selection, and feature enhancement faced by supervised text classification methods. DTWC uses statistical and probabilistic techniques in a hybrid generative-discriminative model of the classification problem. DTWC is efficient and robust – characteristics much desired for today's text classification applications. And, as demonstrated by our evaluations, DTWC's classification accuracy is higher than other well-known methods for text classification. In the remaining subsections, we present our discriminative term weighting strategies, term space partitioning and term selection strategy, discrimination information pooling, linear discriminant learning in the feature space, and relation of DTWC with the naive Bayes classifier.

3.1 Discriminative Term Weighting

In the literature, term weighting has often been employed for effective document representations whereby the frequency of the term or a derived measure like term-frequency-inverse-document-frequency (TFIDF) is used to weigh the term. In this paper, we represent a document as a term occurrence binary vector $\mathbf{x} = \langle x_1, \ldots, x_{|T|} \rangle$ and view term weighting as a measure of the relevance of the term for the classification problem. As such, term weights are properties of the individual terms and are derived from the training data L and not from an individual document only. This view will also help us in term space partitioning and term selection, as described in the next subsection. Each term is weighed by the discrimination information it provides for a specific category over the others. We present three discriminative term weighting strategies: odds, log-odds, and KL divergence.

If a document $\mathbf{x}$ contains a term j (i.e. $x_j = 1$) then it is more likely to belong to category k if $p(x_j = 1|c = k, L)$ is greater than $p(x_j = 1|c = C\backslash k, L)$, where notation $C\backslash k$ denotes all categories but k. Equivalently, a document $\mathbf{x}$ is likely to belong to category k if the odds for category k are greater than one:

$$\frac{p(c = k|x_j = 1)}{p(C\backslash k|x_j = 1)} = \frac{p(x_j = 1|c = k)p(c = k)}{p(x_j = 1|c = C\backslash k)p(c = C\backslash k)} > 1, \tag{1}$$

In the above and subsequent equations, the conditioning on the labeled set L has been omitted for brevity. We would like to quantify the discriminative information that a term j provides regarding category k over categories $C\backslash k$. One way of doing this is to weigh the term by its odds of occurring in documents belonging to category k over documents of categories $C\backslash k$:

$$w_j^k = \begin{cases} a_j/b_j & \text{when } a_j > b_j \\ b_j/a_j & \text{otherwise} \end{cases} \tag{2}$$

where $a_j = p(x_j = 1|c = k)$ and $b_j = p(x_j = 1|c = C\backslash k)$. Notice that the discrimination information that term j provides for categories $C\backslash k$ over category k is b_j/a_j. Thus, the smallest weight assigned by Eq. 2 is one.

A second strategy for discriminative term weighting is to use the log-odds for the term. Using this strategy, the weight for term j is defined as

$$w_j^k = \begin{cases} \log(a_j/b_j) & \text{when } a_j > b_j \\ \log(b_j/a_j) & \text{otherwise} \end{cases} \tag{3}$$

Following this strategy, the smallest weight is zero which is consistent with no discrimination information. Nonetheless, Eqs. 2 and 3 are monotonically related with Eq. 2 always giving a larger value than Eq. 3. This difference in values becomes greater with increasing difference between a_j and b_j.

A third strategy for discriminative term weighting is to use the information theoretic measure known as Kullback-Leibler (KL) divergence. The KL divergence of probability distribution $p(x)$ from $q(x)$ is defined as

$$D_{KL}(p(x)\|q(x)) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

The KL divergence can also be interpreted as the expected discrimination information for $p(x)$ over $q(x)$. In our context, the two probability distributions are $p(x_j|c = k)$ and $p(x_j|c = C\backslash k)$ where x_j can take on values of zero and one. Then, the expected discrimination information provided by knowledge of term j for category k over other categories is given by the KL divergence as

$$\begin{aligned} w_j^k &= D_{KL}(p(x_j|c = k)\|p(x_j|c = C\backslash k) && (4) \\ &= a_j \log \frac{a_j}{b_j} + (1 - a_j) \log \frac{1 - a_j}{1 - b_j} && (5) \end{aligned}$$

Unlike the previous two strategies, this term weighting strategy considers both the occurrence and the absence of a term. Eq. 4 is also monotonically related with Eqs. 2 and 3. However, unlike Eqs. 2 and 3, Eq. 4 is not symmetric. In any case, all three equations quantify the discrimination information provided by term j for discriminating between category k and categories $C\backslash k$ with larger weights signifying larger discrimination information.

The probabilities a_j and b_j are estimated from the training data L by maximum likelihood estimation. A Laplacian prior is used for each event for smoothing (add-one smoothing).

3.2 Term Space Partitioning and Term Selection

The discriminative term weighting strategies described in the previous section can be used for term space partitioning and discriminating term selection. Our weighting strategies naturally partitions the terms into two sets: one set, identified by the index set Z^k, contains terms for which $a_j > b_j$ and the other set, identified by the index set $Z^{C\backslash k}$, contains the remaining terms. All terms $j \in Z^k$ provide evidence for category k over the rest, and this evidence is quantified in their weights in the form of discrimination information. In the next subsection, we describe how we use this partitioning to create a discriminative model of the classification problem.

Our weighting strategies also provide a natural way of selecting highly discriminating and relevant terms. A term j is selected as relevant for the k versus $C\backslash k$ classification problem if

$$w_j^k \geq t$$

where t is a positive valued threshold. All terms that do not satisfy this condition are discarded from the classification model. By increasing the value of t, the number of relevant terms can be reduced by eliminating terms that provide little discrimination information.

DTWC does not require term selection and dimensionality reduction as it transforms the input terms to a two-dimensional feature space (described in the next subsection). However, term selection may be necessary for large scale applications like personalized spam filtering by e-mail service providers [16]. For such applications, DTWC's accuracy can be traded off with its space complexity by varying the value of t.

3.3 Linear Opinion Pool and Linear Discrimination in Feature Space

We use the two set partitioning of the term space, which is based on discrimination information, to form a two-dimensional feature space. Consider a document $\mathbf{x}$. Each term $j \in Z^k$ in the document expresses an opinion regarding the document's categorization. This opinion is captured by the discriminative term weight w_j^k. The aggregated opinion of all these terms is obtained as the linear combination of individuals' opinions:

$$Score^k(\mathbf{x}) = \frac{\sum_{j \in Z^k} x_j w_j^k}{\sum_j x_j} \tag{6}$$

This equation follows from a linear opinion pool or an ensemble average, which is a statistical technique for combining experts' opinions [13, 1]. Each opinion (w_j^k) is weighted by the normalized term occurrence ($x_j / \sum x_j$) and all weighted opinions are summed yielding an aggregated discrimination score for category k ($Score^k(\mathbf{x})$) of the document. If a term i does not occur in the document (i.e. $x_i = 0$) then it does not contribute to the pool. Also, terms that do not belong to set Z^k do not contribute to the pool. Similarly, an aggregated discrimination score can be computed for all terms $j \in Z^{C\backslash k}$ as

$$Score^{C\backslash k}(\mathbf{x}) = \frac{\sum_{j \in Z^{C\backslash k}} x_j w_j^{C\backslash k}}{\sum_j x_j}. \tag{7}$$

The two-dimensional feature space is defined by the two scores $Score^k(\mathbf{x})$ and $Score^{C\backslash k}(\mathbf{x})$. In this space, documents are well separated and discriminated, as illustrated for a spam classification data (Figure 1). We learn the categorization in this space by a linear discriminant function:

$$f^k(\mathbf{x}) = \alpha^k \cdot Score^k(\mathbf{x}) - Score^{C\backslash k}(\mathbf{x}) + \alpha^0 \tag{8}$$

where α^k and α^0 are the slope and bias parameters, respectively. The discriminating line is defined by $f^k(\cdot) = 0$. If $f^k(\cdot) > 0$ then the document $\mathbf{x}$ is likely to belong to category k (Figure 1). For a $|C|$ category classification problem, we learn $|C| - 1$ discriminant functions each with two parameters. In practice, however, the bias parameter set to zero often yields better results, leaving only the slope parameter to be learned. The discriminative model parameters are learned by minimizing the classification error over the labeled training set L. This represents a straightforward optimization problem that can be solved by any iterative optimization technique [20]. DTWC's overall classifier function is defined as

$$\Phi(\mathbf{x}) = \text{argmax}_k \, f^k(\mathbf{x}). \tag{9}$$

DTWC derives its strength from the discrimination information based term weighting, discrimination information pooling to form a two-dimensional feature space, and a simple linear discriminative model for classification. These characteristics make DTWC efficient, in terms of both time and space, and robust to noise and changing data distributions. DTWC contains three key steps: (1) discriminative term weight computation, which can be done in one pass over the labeled data set, (2) forming the two-dimensional feature space, and (3) learning the parameters of the discriminating line which can be done efficiently using straightforward optimization algorithms. The DTWC algorithm is given in Algorithm 1.

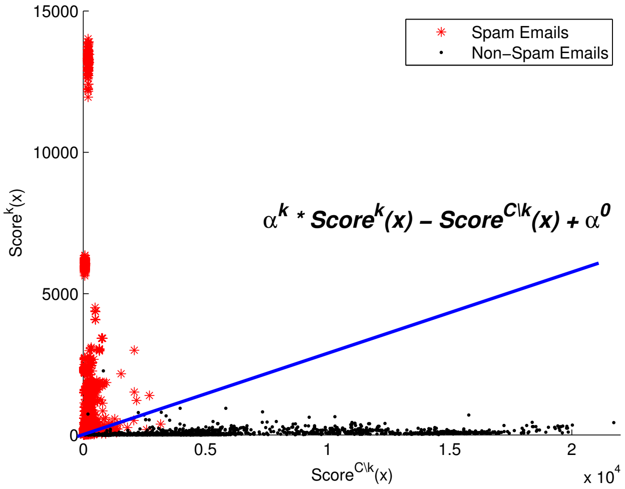

Figure 1. The two-dimensional feature space and the linear discriminant function for a spam classification problem

```
Algorithm 1 DTWC
Input: set of labeled documents $L$, set of unlabeled documents $U$
Output: labels for documents in $U$

On training data $L$
for $k = 1$ to $|C| - 1$ do
  for $j = 1$ to $|T|$ do
    compute $w_j^k$ and $w_j^{C\backslash k}$ (Eq. 2, 3, or 4)
  end for
  compute $Score^k(\mathbf{x})$ and $Score^{C\backslash k}(\mathbf{x})$ (Eqs. 6 and 7)
  learn parameters $\alpha^k$ and $\alpha^0$
end for

On test data $U$
for $k = 1$ to $|C| - 1$ do
  compute $Score^k(\mathbf{x})$ and $Score^{C\backslash k}(\mathbf{x})$ (Eqs. 6 and 7)
  compute $f^k(\mathbf{x})$ (Eq. 8)
end for
output $k = argmax_k f^k(\mathbf{x})$ (Eq. 9)
```

3.4 Relation to Naive Bayes Classifier

In this section, we develop the naive Bayes classifier and show its relation to DTWC. The odds that a document $\mathbf{x}$ belongs to category k rather than categories $C\backslash k$ can be written as

$$\frac{p(c = k|\mathbf{x})}{p(c = C\backslash k|\mathbf{x})} = \frac{p(\mathbf{x}|c = k)p(c = k)}{p(\mathbf{x}|c = C\backslash k)p(c = C\backslash k)}$$

Assuming that the occurrence of each term is independent of others given the category, the document odds on the right-hand side becomes a product of terms' odds. The naive Bayes classification of document $\mathbf{x}$ is category k when

$$\frac{p(c = k)}{p(c = C\backslash k)} \prod_j \left(\frac{p(x_j|c = k}{p(x_j|c = C\backslash k)} \right)^{x_j} > 1$$

Equivalently, taking the log of both sides, the above expression can be written as

$$\log \frac{p(c = k)}{p(c = C\backslash k)} + \sum_j x_j \log \frac{p(x_j|c = k)}{p(x_j|c = C\backslash k)} > 0 \quad (10)$$

This equation computes a non-negative score, and when this score is greater than zero the naive Bayes classification for the document $\mathbf{x}$ is k. Notice that only those terms are included in the summation for which $x_j = 1$.

Comparing the naive Bayes classifier, as expressed by Eq. 10, with DTWC yields some interesting observations. The discriminative model of DTWC is similar to Eq. 10 in that the structure of the discrimination score computation (Eqs. 6 and 7) is similar to the summation in Eq. 10 and the bias parameter α^0 corresponds to the first term in Eq. 10. The log-odds for term j in Eq. 10 corresponds to the log-odds discriminative weighting strategy in DTWC.

However, there are also significant differences between DTWC and naive Bayes. (1) The discrimination scores in DTWC are normalized (for each document) using the $L1$ norm. Document length normalization is typically not done in naive Bayes classification, and when it is, the $L2$ norm is used. Recently, it has been shown that performing $L1$ document length normalization improves the precision of naive Bayes for text classification [17]. (2) DTWC partitions the summation into two, based on discrimination information, and then learns a linear discriminative model of the classification. Naive Bayes, on the other hand, is a purely generative model with no discriminative learning of parameters. (3) DTWC allows the use of different discriminative term weighting strategies as long as they quantify the discrimination information that a term provides for one category over the others. (4) DTWC does not require the naive Bayes assumption of conditional independence of the terms given the category.

DTWC will be identical to naive Bayes when the log-odds term weighting strategy is used, discrimination scores are not normalized, and the slope parameter α^k is equal to one.

4 Evaluation Setup

We evaluate DTWC on three commonly-used text classification data sets – personalized spam filtering, movie review, and SRAA – and compare its performance with four other classifiers – Naive Bayes (NB), Maximum Entropy (ME), Balanced Winnow (BW), and Support Vector Machine (SVM). The performance of DTWC with odds (DTWC-O), log-odds (DTWC-LO), and KL divergence (DTWC-KL) discriminative term weighting strategies is reported. For naive Bayes, Maximum Entropy, and Balanced Winnow we use the implementation provided by the Mallet toolkit [22]. For SVM, we use the implementation provided by SVM^{Light} [15]. We report the classification accuracy for spam data set, and the mean and standard deviation of classification accuracy for movie and SRAA data sets calculated over 5 runs of the algorithms.

4.1 Data Sets

In all three data sets, documents are represented as bag-of-words/terms. We convert them to formats in which documents are represented by term frequency vectors and term occurrence vectors. Where applicable stop words, HTML tags, and message headers are removed from the data sets.

The personalized spam filtering data set, henceforth identified as the Spam data set, captures the e-mail classification problem in which individual user's e-mails are labeled as either spam or non-spam (2 categories) after learning from a general labeled training set. This data set corresponds to data set A provided by the 2006 ECML/PKDD Discovery Challenge [3]. It contains a labeled training set of 4000 e-mails and three unlabeled users inboxes of 2500 e-mails each. The composition of the training set is: 50% spam e-mails sent by blacklisted servers of the Spamhaus project (http://www.spamhaus.org), 40% non-spam e-mails from the SpamAssassin corpus, and 10% non-spam e-mails from about 100 different subscribed English and German newsletters. The composition of e-mails in users inboxes is more varied with 50% non-spam e-mails of distinct Enron employees from the Enron corpus and 50% spam e-mails from various sources. Low frequency terms have already been removed. A key characteristic of this data set is that the distribution of e-mails in the training set is different from those in the users' inboxes (test sets).

The movie review data set, henceforth identified as the Movie data set, captures the sentiment classification problem in which movie reviews from IMDB (Internet Movie Database) are labeled as either positive or negative (2 categories). This data set is obtained from http://www.cs.cornell.edu/people/pabo/movie-review-data. It consist of 2000 positive and 2000 negative reviews. We remove the stop words/terms using the Mallet toolkit [22]. We holdout 400 examples of each class for testing and randomly select different numbers of examples for training.

The SRAA (Simulated/Real/Aviation/Auto) data set [1] is a collection of 73,218 documents from four newsgroups (simulated-aviation, simulated-auto, real-aviation, and real-auto), representing a 4 category classification problem. We remove the HTML header and the stop words using the Mallet toolkit [22]. We holdout 1000 examples of each class for testing and randomly select different numbers of examples for training.

4.2 Tuning the Algorithms

Documents are represented by term frequency vectors for the NB, ME, BW, and SVM classifiers. For DTWC, however, we use term occurrence vectors for document representation. An extensive evaluation of DTWC with different document vector representations is beyond the scope of this paper, although we do find that the term occurrence representation outperforms the term frequency representation on the Spam data set. The default algorithm settings provided by Mallet are adopted for NB, ME, and BW.

The SVM (using SVM^{Light}) is tuned for each data set by evaluating its performance on a validation set that is a 30% holdout of the training set. The SVM^{Light} parameter C that controls the trade-off between classification error and margin width is tuned for each data set. Similarly, we evaluate the performance of SVM with both linear and non-linear kernels and find the linear kernel to be superior. This observation is consistent with that reported in the literature [18, 10, 30]. We perform document length normalization using $L2$ (Euclidean) norm. This improves performance slightly from the non-normalized case, as observed by others as well [25, 10, 30]. We keep the remaining parameters of SVM^{Light} at default values. There are no tunable parameters in DTWC (we keep the threshold $t = 0$, unless mentioned otherwise).

5 Results and Discussion

5.1 Classification Accuracy

Tables 1, 2, and 3 show the classification accuracies of DTWC, naive Bayes (NB), maximum entropy (ME), balanced winnow (BW), and SVM on Spam, Movie, and SRAA data sets, respectively. The results for DTWC with odds, log-odds, and KL divergence discriminative term weighting strategies are identified by DTWC-O, DTWC-LO, and DTWC-KL, respectively. For Movie and SRAA data sets, we give the mean and standard deviation of the classification accuracies over five runs of the classifiers with

[1] http://www.cs.umass.edu/ mccallum/code-data.html

Table 1. Accuracy results for Spam data set. The training set and each user's inbox contain 4000 and 2500 e-mails, respectively.

Inbox	*DTWC-O*	*DTWC-KL*	*DTWC-LO*	*NB*	*ME*	*BW*	*SVM*
Inbox 1	91.00	79.88	91.12	81.24	62.20	61.00	64.40
Inbox 2	92.36	82.24	91.80	83.80	68.16	64.76	69.56
Inbox 3	87.52	68.88	88.60	87.88	78.92	73.44	80.24
Avg	90.29	77.00	90.56	84.30	69.76	66.40	71.40

Table 2. Accuracy results for Movie data set. Means plus/minus standard deviations are computed from 5 runs with randomly drawn training sets of sizes specified in the first column and randomly selected test sets of size 800.

Ex.	*DTWC-O*	*DTWC-KL*	*DTWC-LO*	*NB*	*ME*	*BW*	*SVM*
600	80.90 ± 1.13	82.32 ± 0.83	81.35 ± 0.83	79.25 ± 1.15	82.14 ± 0.50	78.89 ± 0.86	81.85 ± 0.91
500	82.47 ± 1.50	83.24 ± 1.14	82.40 ± 1.21	80.74 ± 0.37	81.32 ± 0.50	77.92 ± 2.31	81.35 ± 1.78
400	79.84 ± 1.94	81.52 ± 0.79	81.42 ± 0.80	79.17 ± 1.08	79.62 ± 1.28	78.34 ± 1.58	79.65 ± 1.07
300	79.64 ± 1.00	82.27 ± 1.36	80.07 ± 0.91	77.57 ± 1.01	77.97 ± 1.56	76.09 ± 1.36	78.52 ± 1.33
200	78.02 ± 1.46	80.87 ± 1.28	79.30 ± 1.90	76.42 ± 1.57	76.32 ± 0.92	74.12 ± 1.86	76.10 ± 1.22
Avg	80.17	82.04	80.91	78.63	79.47	77.07	79.49

each run using randomly chosen examples for training and testing. For Spam data set, we give classification accuracies for each user inbox.

The results show that for all the runs, DTWC outperforms all the other classification algorithms. This is also true for averaged results. The average performance of DTWC on the Spam data set is impressive (90.56% by DTWC-LO) with the second best performance being over 6% lower. The personalized spam filtering problem is challenging because the distributions of e-mails in the training set and the users' inboxes are quite different. As such, although ME, BW, and SVM can achieve a high accuracy on the training set, their generalization onto the unseen data is very poor. Usually for such differing distribution classification problem settings, techniques that make use of the users' inboxes (unlabeled data) during learning, i.e., transductive or semi-supervised learning, will achieve better results [16]. Nonetheless, DTWC, which uses supervised learning, appears to be little affected by this change in distribution of the two sets. The naive Bayes classifier appears to be second least affected by this change. The SVM performs poorly on this data set. The superior performance of DTWC can be attributed to the discriminative term-based model of spam and non-spam and the simple and generalized discriminative model. An interesting observation for the Spam data set is that DTWC-KL's performance is significantly lower than that of DTWC-O and DTWC-LO. DTWC-KL uses the KL divergence as the discriminative term weighting strategy as opposed to the odds and log-odds strategies used by the other two. This observation needs further investigation. Here we only conjecture that this may be related to the consideration of both presence and absence of terms in the context of personalized spam classification.

The distribution of training and testing sets are similar for the Movie and the SRAA data sets. For these data sets also, DTWC outperforms the other algorithms. The improvement, however, is less significant as compared to that for the Spam data set. DTWC-KL is the best performer for the Movie data set, while DTWC-O is the best performer for the SRAA data set. The results obtained by NB, ME, and SVM are comparable to those reported in [10, 21]. DTWC's performance appears slightly lesser than that of multi-conditional learning reported in [21]; however, their exact evaluation and data set up is not known so a direct comparison is not possible. Notice that the performance of DTWC degrades gracefully as the number of examples in the training set is reduced.

5.2 Parameter Estimation

DTWC uses a set of generative model parameters – the discriminative term weights – and $|C| - 1$ discriminative model parameters – the slope α^k and bias α^0. The weights are computed from the labeled training set by maximum likelihood estimation. This is a straightforward computation requiring a single pass over the training set. The discriminative model parameters are learned by minimizing the classification error over the labeled training set. This

Table 3. Accuracy results for SRAA data set. Means plus/minus standard deviations are computed from 5 runs with randomly drawn training sets of sizes specified in the first column and randomly selected test sets of size 4000.

Ex.	*DTWC-O*	*DTWC-KL*	*DTWC-LO*	*NB*	*ME*	*BW*	*SVM*
1500	93.41 ± 0.30	88.61 ± 0.66	91.93 ± 0.10	92.72 ± 0.31	90.53 ± 0.58	88.23 ± 0.46	91.54 ± 0.36
1000	92.94 ± 0.14	88.50 ± 0.37	91.14 ± 0.37	92.10 ± 0.67	89.12 ± 0.26	87.54 ± 0.37	89.34 ± 0.30
500	91.26 ± 0.51	87.50 ± 0.81	88.48 ± 0.61	90.59 ± 0.67	86.75 ± 0.60	85.01 ± 0.82	86.73 ± 1.36
250	88.88 ± 0.42	85.52 ± 0.72	83.60 ± 1.41	88.05 ± 0.92	83.28 ± 0.17	81.94 ± 0.54	84.52 ± 0.37
150	86.63 ± 0.22	83.74 ± 0.96	78.12 ± 1.31	85.69 ± 0.69	81.87 ± 1.02	79.97 ± 1.01	83.58 ± 1.17
Avg	90.62	86.77	86.65	89.83	86.31	84.54	87.14

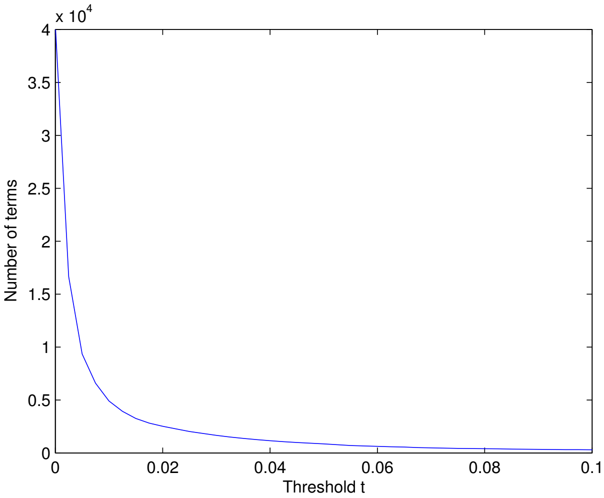

Figure 2. Number of terms selected versus threshold t for Spam data set (DTWC-O)

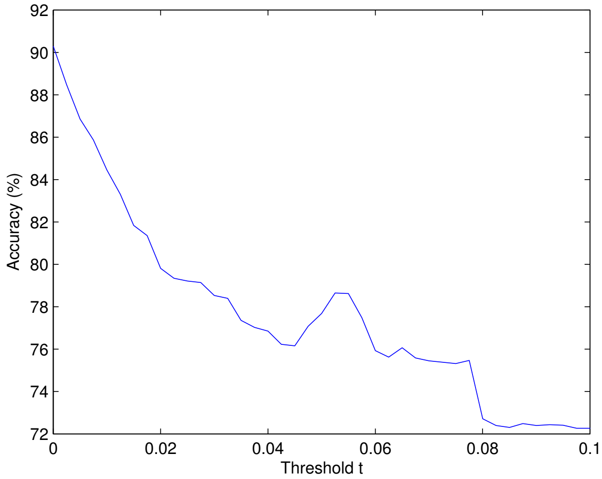

Figure 3. Average accuracy versus threshold t for Spam data set (DTWC-O)

is a convex optimization problem, as empirically verified from the error versus slope parameter graph (Figure 4). The bias parameter, which is usually close to zero in our evaluations, can be determined after learning the slope parameter. The optimization problems can be solved efficiently by an iterative optimization technique or by grid search.

5.3 Term Selection

The threshold t can be used to trade-off DTWC's space requirement and accuracy performance. This is evident from Figure 2 which shows the variation of the number of selected terms with threshold t for the Spam data set (using DTWC-O). The number of selected terms drops significantly with only a small increase in t. Remarkably, however, the classification accuracy does not decrease drastically (Figure 3). Table 4 shows the number of terms and the average accuracy (averaged over the 3 inboxes) of DTWC-O for Spam data set. It is seen that even when the number of terms is reduced by one-eighth (from 40516 to 4913 terms) the average accuracy value for DTWC is still higher than the second best performer, i.e., naive Bayes. This result demonstrates the robustness and scalability of our algorithm, and its suitability for application like personalized spam filtering by e-mail service providers.

6 Conclusion

In this paper, we describe a new text classification method, named DTWC, based on discriminative term weighting, discrimination information aggregation, and linear discrimination in a two-dimensional feature space. Each term in the classification problem is assigned a weight that

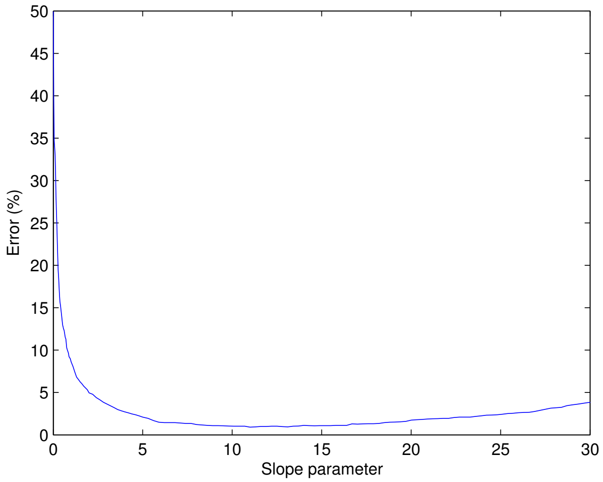

Figure 4. Classification error versus slope parameter curve

Table 4. Selected terms and accuracy at different values of threshold t for Spam data set (DTWC-O)

Threshold	*Terms*	*Accuracy*
0	40516	90.29
0.0025	16666	88.48
0.005	9333	86.85
0.0075	6608	85.86
0.01	4913	84.45

quantifies the discrimination information it provides for category k over the others. These discriminative term weights are then used to transform the input term space into a two-dimensional feature space. The transformation is based on a statistical model of opinion pooling. Category k and the rest are discriminated in the feature space by a straight line. As such, the discriminative model has only two parameters, the slope and the bias of the line, which can be computed efficiently by an iterative optimization algorithm. DTWC is simple, efficient, effective, and robust. All these characteristics make it suitable for many application areas, including personalized spam filtering where scalability and robustness are essential.

DTWC is evaluated on spam filtering, movie review, and simulaed/real/aviation/auto data sets. Its classification accuracy is compared with that of four other classifiers – naive Bayes, maximum entropy, balanced winnow, and SVM. DTWC outperforms all classifiers in all settings. Its performance is substantially better in situations where the training and testing sets follow different distributions. We also discuss the efficiency and robustness characteristics of DTWC by evaluating its performance against term selection.

Acknowledgment

The first author gratefully acknowledges support from Lahore University of Management Sciences (LUMS) and Higher Education Commission (HEC) of Pakistan for this research.

References

[1] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2004.

[2] L.D. Baker and A. McCallum. Distributional clustering of words for text classification. In *Proceedings of SIGIR'98*. ACM, 1998.

[3] Steffen Bickel. Ecml/pkdd: Discovery challenge. In *Proceedings of ECML/PKDD discovery challenge*, 2006.

[4] B. Bigi. Using kullback-leibler distance for text categorization. In *Proceedings of ECIR*. Springer, 2003.

[5] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.

[6] I. Dagan, Y. Karov, and D. Roth. Mistake driven learning in text categorization. In *Proceedings of EMNLP-97, 2nd Conference on Empirical Methods in Natural Language Processing*, pages 55–63, 1997.

[7] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney. Feature selection methods for text classification. In *Proceedings of KDD-07, 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2007.

[8] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, pages 784–788, Melbourne, US, 2003. ACM Press, New York, US.

[9] I.S. Dhillon, S. Mallela, and R. Kumar. A divisive information theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003.

[10] G. Druck, C. Pal, A. McCallum, and X. Zhu. Semi-supervised classification with hybrid generative/discriminative methods. In *Proceedings of KDD-07, 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.

[11] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, pages 1289–1305, 2003.

[12] T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, 1998.

[13] R.A. Jacobs. Methods for combining experts' probability assessments. *Neural Computation*, 7:867–888, 1995.

[14] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1998.

[15] T. Joachims. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999.

[16] K.N. Junejo and A. Karim. Pssf: A novel statistical approach for personalized service-side spam filtering. In *Proceeding of WI-07, IEEE / WIC / ACM International Conference on Web Intelligence*, Sillicon Valley, USA, 2007.

[17] A. Kolcz and W. Yih. Raising the baseline for high-precision text classifiers. In *Proceedings of KDD 2007*, 2007.

[18] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, 2004.

[19] J. Liu, J.-Q. Song, and Y.-L. Huang. A generative/discriminative hybrid model: Bayes perceptron classifier. In *Proceedings of ICMLC-07, International Conference on Machine Learning and Cybernetics*, 2007.

[20] D.G. Luenberger. *Linear and Nonlinear Programming*. Reading, Mass: Addison-Wesley, 2nd edition, 1984.

[21] A. McCallum, C. Pal, G. Druck, and X. Wang. Multi-conditional learning: generative/discriminative training for clustering and classification. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[22] A.K. McCallum. A machine learning for language toolkit, 2002.

[23] E. Montanes, I. Diaz, J. Ranilla, E.F. Combarro, and J. Fernandez. Scoring and selecting terms for text categorization. *Intelligent Systems*, 20(3):40–47, 2005.

[24] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[25] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[26] R. Raina, Y. Shen, and A.Y. Ng. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems*, 2004.

[27] G. Salton and C. Buckley. Term weighting approaches in automated text retrieval. Technical Report 87-881, Dept. of Computer Science, Cornell University, 1987.

[28] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[29] A.K. Seewald. An evaluation of naive bayes variants in content-based learning for spam filtering. *Intelligent Data Analysis*, 11(5):497–524, 2007.

[30] L. Zhang, J. Zhu, and T. Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269, 2004.