Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Terms-based discriminative information space for robust text classification

Khurum Nazir Junejo[a,b], Asim Karim[c], Malik Tahir Hassan[d,e], Moongu Jeon[e,*]

[a] *Singapore University of Technology and Design, Singapore*
[b] *Karachi Institute of Economics and Technology, Pakistan*
[c] *Department of Computer Science, SBASSE, Lahore University of Management Sciences, Lahore, Pakistan*
[d] *University of Management and Technology, Lahore, Pakistan*
[e] *School of Information and Communications, Gwangju Institute of Science and Technology, Gwangju, South Korea*

ARTICLE INFO

Article history:
Received 19 June 2015
Revised 19 July 2016
Accepted 19 August 2016
Available online 22 August 2016

Keywords:
Text classification
Discriminative term weights
Linear opinion pooling
Feature construction

ABSTRACT

With the popularity of Web 2.0, there has been a phenomenal increase in the utility of text classification in applications like document filtering and sentiment categorization. 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. In this paper, we propose a novel and efficient method using terms-based discriminative information space for robust text classification. 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 category sets. A linear opinion pool is adopted for combining the discrimination information provided by each set of terms to yield a feature space (discriminative information space) having dimensions equal to the number of classes. Subsequently, a discriminant function is learned to categorize the documents in the feature space. This classification methodology relies upon corpus information only, and is robust to distribution shifts and noise. We develop theoretical parallels of our methodology with generative, discriminative, and hybrid classifiers. We evaluate our methodology extensively with five different discriminative term weighting schemes on six data sets from different application areas. We give a side-by-side comparison with four well-known text classification techniques. The results show that our methodology consistently outperforms the rest, especially when there is a distribution shift from training to test sets. Moreover, our methodology is simple and effective for different application domains and training set sizes. It is also fast with a small and tunable memory footprint.

1. Introduction

Text classification is witnessing growing interest in recent years. This is due to the availability of digitized text such as Web pages, e-mails, blogs, digital libraries, social media, online advertisements, corporate documents, product reviews and much more [22]. Many applications based on these different data sources can be posed as text classification problems. In these problems, documents need to be categorized into predefined classes representing different semantic groups (e.g. spam and non-spam, topics, sentiments).

[*] Corresponding author.
E-mail address: mgjeon@gist.ac.kr (M. Jeon).

http://dx.doi.org/10.1016/j.ins.2016.08.073

Nomenclature

C	Set of Labels/Classes/Categories
$C\backslash k$	All Classes other than class k
L	Training Data
$Score^{k}(\mathbf{x})$	Aggregated Opinion of Terms of Set Z^{k} Present in Doc. $\mathbf{x}$
$Score^{C\backslash k}(\mathbf{x})$	Aggregated Opinion of Terms of Set $Z^{C\backslash k}$ Present in Doc. $\mathbf{x}$
T	Dictionary Size
U	Test Data
$Z^{C\backslash k}$	Set of Significant Terms for Classes other than k
Z^{k}	Set of Significant Terms for Class k
Φ	True Target Function
α^{0}	Bias Parameter for the Discriminative Model
α^{k}	Slope Parameter for the Discriminative Model
$\bar{\Phi}$	Learned Target Function
a_j	Probability of jth Term for Class k
b_j	Probability of jth Term for Class other than k
c_i	Label of ith Document
$f^{k}(.)$	Discriminant Function for Class k
k	A particular Class Label from Set C
t	Threshold Parameter
w_j	Weight of jth Term
w_j^{k}	Weight of jth Term for Class k
x_{ij}	jth attribute of the ith Document
x_i	ith Document

Text classification is challenging in the modern Internet environment. Firstly, text documents are sparsely represented in a very high dimensional term space (easily in hundreds of thousands for user generated content on the Web), making learning and generalization difficult. Secondly, due to the high cost of labeling documents, researchers are forced to rely upon small training sets or collect training data from sources different from the target domain. This results in a distribution shift between training and test data. Thirdly, documents are of varying quality, languages and lengths, making a uniform knowledge-based approach inefficient or infeasible. For example, an important domain for text classification which embodies these challenges is that of e-mail spam filtering: vocabulary of terms can be huge; users' preferences for spam and non-spam often differ; non-generic labeled collections are often not available; e-mails come in a wide variety of languages and qualities. Addressing these challenges demands a corpus-based, statistically robust, and computationally efficient text classification method.

There are numerous classification techniques available today that can be utilized for text classification as well. The naive Bayes classifier, which is a probabilistic generative method, and the support vector machine, which is a statistical discriminative method, are generally considered effective for text classification. However, the former only performs better than the latter for very small training data [61], while the latter is very sensitive to distribution shift between the training and test data [20]. A different approach to enhanced text classification is through feature engineering and semantic representations. These approaches can be corpus-based, like latent semantic indexing (LSI) [45], or knowledge-based, like WordNet-based semantic enrichment [56]. However, engineering of new features or incorporating information from external knowledge bases adds to the computational complexity of these methods. In addition, external knowledge may not be available conveniently for some domains, e.g., legal proceedings, etc., limiting the general applicability of such approaches.

Intuitively, robust classification is obtained when the classes are well separated and invariant to noise in the feature space. Constructing such a feature space is therefore a significant step. For text classification, it is desirable to have a feature space that is readily interpretable through the terms in the document collection. In Fisher linear discriminant analysis, a popular feature extraction approach, the score of a document in the feature space is a non-intuitive combination of terms. Intuitive interpretation also implies that the feature space is low-dimensional as opposed to high-dimensional that can be generated by some kernel transformations.

In this paper, we present a terms-based discriminative information space for robust text classification (DIST). This feature space is constructed from discriminative term weights and linear opinion pooling. The discriminative term weights are supervised term weights that quantify the discriminative power of each term. Linear opinion pool aggregates the discriminative powers of terms to yield discriminative information scores for each document. These scores quantify the suitability of each document for a class over the others. A standard discriminative classifier is then learned in this feature space for the final classification.

1.1. Our contribution

Specifically, we make the following contributions in this paper:

1. We present a new text classification methodology, named DIST. It combines feature space construction and linear classifier learning for robust performance in applications involving distribution shift between training and test data.
2. We propose and evaluate five supervised term weighting measures for quantifying each term's discriminative power. These measures (relative risk, log relative risk, odds, log odds, and Kullback-Leibler divergence) can be computed efficiently from the labeled training data.
3. We show that our methodology is a generalization of common generative, discriminative, and hybrid classification methods. In particular, we relate our methodology to the naive Bayes classifier and support vector machines.
4. We evaluate our methodology on six data sets belonging to different application areas. We also demonstrate the effectiveness on data sets having different training and test distributions. The results are compared with four common text classification methods. We further conduct statistical significance tests that demonstrate the overall effectiveness of our methodology with improved classification accuracies and area under the curve (AUC) values.
5. We show that our methodology is computationally efficient and suitable for modern text classification problems, especially those having (a) distribution shift between training and test data, (b) high-dimensional input (term) space, and (c) time and memory limitations.

The rest of the paper is organized as follows. We discuss the related work in Section 2. Our text classification method, DIST, is described in Section 3. Comparison of DIST with popular generative and discriminative approaches follows in Section 4. We describe the data sets, evaluation setup, and experimental results in Section 5. This section compares our results under varying distribution shift, term weighting measures, and feature selection thresholds. Section 6 analyzes the scalability of DIST, its generalization to a classifier framework, and statistical significance tests of the results. We conclude and state some promising future directions in Section 7.

2. Related work and motivation

Relevant related work on discriminative term weighting and text classifiers is presented in the following subsections.

2.1. Discriminative term weights

Term weighting is widely used in text retrieval and classification for quantifying the importance of a term in a document. Term weighting measures are either discriminative or non-discriminative in nature [14]. Non-discriminative measures are computed independent of the class information, whereas discriminative measures make use of the class information. Discriminative measures quantify the strength of the evidence that a term provides in favor of one category as opposed to the rest. This evidence is referred to as the discriminative information (or power) of a term. Recently, it has been demonstrated that the relatedness of a term to a category/topic in a document collection can be quantified by its discrimination information [9]. Various measures for quantification of discrimination information have been used for feature selection [19], association rule mining [24], hashtag recommendation [66], text classification [36] and document clustering [26]. The idea of discriminative term weights is introduced in [36] for quantifying the discrimination information provided by terms. In this work, we evaluate five such measures: relative risk (RR), log relative risk (LRR), odds ratio (OR), log odds ratio (LOR), and Kullback-Leibler divergences (KLD).

Relative risk (RR) and odds ratio (OR) have been used extensively for disease diagnosis in clinical trials [23,28,67]. RR is the risk of developing a disease relative to exposure; mathematically RR is a ratio of the probability of the event (e.g. a disease) occurring in an exposed group versus a non-exposed group. Whereas OR is the ratio of the odds of an event occurring in one group to the odds of it occurring in another group. In medical research, RR is favored for cohort studies and randomized controlled trials whereas OR is used in retrospective studies and case-control studies. Many interesting properties of OR have made it appealing to machine learning and data mining such as its symmetry under variable permutation, row/column scaling invariance, inversion invariance, null invariance and many more [59]. Usually OR has been used for feature selection [19,71] and has only occasionally been used for classification [68], mainly because its performance degrades significantly under some circumstances (as shown in Section 5.3.3). Recently, there is a growing reliance on such statistically sound measures for quantifying the relevance of patterns [41,66]. Efficient algorithms for discovering risk patterns, defined as itemsets with high relative risk, are discussed by [42]. Direct discovery of statistically sound association rules is presented by [24]. These measures have also been used in the language processing literature for quantifying term association [11], and in text clustering approaches [25]. Even though RR is a more intuitive measure, it has not received similar attention in machine learning literature. The reason being that it does not enjoy some of the interesting properties of OR. But we show in Section 5.3.3 that statistically, RR is either equal or better than the OR for text classification.

Like OR and RR, measures such as Kullback-Leibler divergence, chi-square statistic, information gain, mutual information, Hellinger distance and many more have also been used for feature selection. Some have even been used for text classification [19,68,73], but none have given consistent results. We overcome this deficiency by transforming these term weights to a robust two dimensional feature space by means of term partitioning followed by linear opinion pooling. To the best of our knowledge, RR, OR, KL or other weighing measures have not been used for feature construction that is given as input to a classifier.

External knowledge bases have also been used to compute semantically enriched term weights. [56] uses WordNet-based semantic enrichment, while [70] uses Wikipedia concepts and categories. These approaches may result in performance improvement. However, engineering of new features or incorporating information from external knowledge bases adds to the computational complexity of these methods. In addition, external knowledge may not be available conveniently for some domains, e.g., legal proceedings, etc., limiting the general applicability of such approaches.

These discriminative term weights can be used to identify the common, and domain-specific sets of words for multi-domain active learning approach for text classification proposed by [43]. The most discriminative terms can help identify examples that differentiate one domain from the rest, thus corresponding to the domain-specific part. Whereas, the least discriminative terms can help build the common part by identifying examples that are common across domains. However, [43] is not comparable to our work as active learning is a semi-supervised learning approach with the goal of reducing the number of examples needed to train a classifier by intelligently selecting examples from a large pool of unlabelled records. In contrast, we are proposing a supervised classification approach with the aim of enhancing classifier performance rather than optimizing the number of examples required for classification.

The distinction between our discriminative term weights (DTWs) and the (document) term weights (TWs) such as TF-IDF, etc., is worth pointing out. Firstly, DTWs quantify the discrimination information that terms provide for classification while TWs quantify the significance of a term within a document. Secondly, DTWs are defined globally for every term in the vocabulary while TWs are defined locally for every term in a document. Thirdly, DTWs are computed in a supervised fashion rather than the usual unsupervised computation of TWs. DTWs are not a substitute for TWs; they are defined for a different purpose of classification rather than representation.

2.2. Text classifiers

The most common generative classifier is the naive Bayes (NB) classifier [49]. This classifier results from the application of the Bayes rule with the assumption that each variable is independent of the others given the class label. Three types of naive Bayes classifiers have been used for text classification: Multi-variate Bernoulli, multi-nomial with term frequencies (TF) as weights, and multi-nomial with Boolean attributes. Multi-variate Bernoulli model penalizes the absence of a term in a document during classification. This has led to its relatively poor performance than multi-nomial variants for sparsely represented documents in a high dimensional space [49]. Since multi-nomial NB with TF weights contains more information than the one with Boolean attributes, it is quite reasonable to assume that it would perform better, but [49] shows otherwise. Their finding has been verified by [62] with experiments in the spam filtering domain. [53] gives a comparison of the above mentioned variants of the naive Bayes with the addition of two more, namely, multi-variate Gaussian NB and flexible Bayes, both of which assume an underlying probability distribution for the terms in the documents. [53] conclude that multi-nomial NB with Boolean attributes to be the better of the five variation of NB method for text classification. We compare our results to this variant of NB. Henceforth, our usage of the term NB in rest of the paper refers to multi-nomial NB with Boolean attributes.

Another successful probabilistic classifier is maximum entropy [57]. It estimates the joint probability distribution by maximizing the entropy constrained by the empirical distribution. It is commonly used as an alternative to the naive Bayes classifier because it does not assume statistical independence of the features. However, it assumes collinearity to be relatively low because it becomes difficult to differentiate between the impact of several features if they are highly correlated. Learning a model of maximum entropy is slower than that of a naive Bayes classifier but it has been shown to sometimes perform significantly better than naive Bayes [57].

The most popular discriminative text classifier is support vector machine (SVM) [32]. It is based on statistical learning theory and structural risk minimization. It learns a maximum margin linear discriminant in a (possibly) high dimensional feature space. According to [32], SVM does not require feature selection as it tends to be robust to over-fitting and can scale up to considerable dimensions, thus making it suitable for textual data. It is further emphasized that no human and machine effort in parameter tuning on validation set is needed because there is a theoretically motivated default choice of parameter settings. Nonetheless, we experiment to find the best values for the parameters to compare it with our technique.

Another well known discriminative classifier for computing good class boundaries is Rocchio method [8]. The centroid of a class is computed as the center of mass of its members. The boundary between two classes is then the set of points with equal distance from the two centroids. As a result, the boundaries of the class regions are hyper planes. The classification rule is quite simple: a point is classified in accordance with the region it falls into. It is as computationally efficient as NB classifier, in addition it has relevance feedback mechanism but suffers from low classification accuracy. Furthermore, the classes must be approximate spheres with similar radii, a condition rarely met in text classification. It also performs poorly for multi-modal classes and is often surpassed by SVM [1].

A more suitable choice is the k nearest neighbor (k-NN) classifier as it determines the decision boundary locally. This makes it more applicable with classes that have non-spherical, disconnected or other irregular shaped boundaries. However it is expensive to train because it requires an order of k passes to find the best value of k. Being a memory based lazy learning approach, it keeps all the training examples in memory for classification thus rendering it infeasible for large datasets

[69]. Since it operates without pre-modeling, it incurs a high cost to classify new documents when the training set is large [54].

Another widely used discriminative classifier is the decision tree classifier. It is simple to understand and interpret, however, it does not work well when the number of distinguishing features is large, as is the case in text classification [1,63].

The balanced winnow is another example of a discriminative classifier that learns a linear discriminant in the input space by minimizing the mistakes made by the classifier [13]. It is very similar to the perceptron algorithm; however, the perceptron algorithm uses an additive weight-update scheme, whereas balanced winnow uses a multiplicative scheme. This allows it to move quickly to the desired weight vector despite of the large size of the available features. [10] shows that its performance is comparable and sometimes even better than the linear SVM.

3. DIST: Discriminative information space for text classification

In this section, we describe our text classification methodology, DIST, based on terms-based discriminative information space construction and linear classification. DIST addresses the key issues of high dimensionality, term weighting and selection, and feature construction faced by supervised text classification methods. It uses statistical, information theoretic and probabilistic techniques in a hybrid generative-discriminative model of the classification problem. It is efficient, robust, scalable, and simple – characteristics that are much desired for today's text applications. In the following subsections, we define the text classification problem, followed by the presentation of our DIST classification methodology.

3.1. Problem statement

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 $\mathbf{x}_i$, $|C|$ 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 $\Phi^{'}: U \to \{1, 2, \ldots, |C|\}$ by the classifier function $\Phi: U \to \{1, 2, \ldots, |C|\}$ such that the number of documents in U for which $\Phi(\mathbf{x}_j) \neq \Phi^{'}(\mathbf{x}_j)$ is minimized. 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 unique 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.

Aforementioned document representation is the most commonly used representation for textual documents in machine learning methods. It is also known as "bag-of-words" representation or "vector space model" [14]. It 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 (TF-IDF) [14]. We restrict ourselves to the binary term vector representation that has been shown to produce more accurate classifiers [6].

3.2. Discriminative term weighting

Discriminative term weighting is a supervised term weighting strategy that computes weights for each term in the vocabulary from the labeled training data. These weights quantify the discriminative information provided by a term for a specific category over the others. Introduced by [36], discriminative term weights (DTWs) differ from traditional term weights (TWs) like TF-IDF. DTWs are meant for enhanced representation in classification problems rather than for general-purpose representation. In this work, we present and evaluate five discriminative term weighting measures – relative risk (RR), log relative risk (LRR), odds ratio (OR), log odds ratio (LOR), and Kullback-Leibler divergence (KLD) – for feature space construction in text classification.

Intuitively, a document $\mathbf{x}$ containing a term j (i.e. $x_j = 1$) is more likely to belong to category k if the occurrence of term j in documents in k is higher than its occurrence in documents not in k. We want to quantify this discriminative information that the term j provides regarding category k over rest of the categories ($C\backslash k$). We do this by associating a weight, w_j^k, for each term $j = 1, 2, \ldots, |T|$ and every category $k \in C$.

One way of quantifying w_j^k is through the relative risk (RR):

$$w_j^k = \begin{cases} a_j/b_j & \text{when } a_j > b_j \\ b_j/a_j & \text{otherwise} \end{cases} \tag{1}$$

where $a_j = p(x_j = 1|c = k)$ and $b_j = p(x_j = 1|c \in C\backslash k)$. Notice that the discriminative information the term j provides for categories $C\backslash k$ over category k is b_j/a_j. Thus, the smallest weight assigned by Eq. (1) is one and the highest value assigned could be infinity which is avoided by smoothing.

The log of the relative risk, or log relative risk (LRR), can be used to quantify discriminative term weights. Using this measure the weight for term j for category k is defined as

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

Unlike RR, the smallest value assigned by LRR is zero which is consistent with no discrimination information. Eqs. (1) and (2) are monotonically related, with the former always giving a larger value than the latter. This difference in values becomes greater with increasing difference between a_j and b_j.

Besides relative risk, the related measure of odds ratio is utilized to quantify discriminative term information. In statistics, odds ratio is defined as the ratio of the odds of an event occurring in one group to the odds of it occurring in another group, where odds is the relative likelihood of the event. Mathematically, the discriminative term weight for term j and category k quantified by odds ratio (OR) and log odds ratio (LOR) are defined as

$$w_j^k = \begin{cases} \dfrac{a_j}{1-a_j} \Big/ \dfrac{b_j}{1-b_j} & \text{when } a_j > b_j \\ \dfrac{b_j}{1-b_j} \Big/ \dfrac{a_j}{1-a_j} & \text{otherwise} \end{cases} \tag{3}$$

and

$$w_j^k = \begin{cases} \log\left(\dfrac{a_j}{1-a_j} \Big/ \dfrac{b_j}{1-b_j}\right) & \text{when } a_j > b_j \\ \log\left(\dfrac{b_j}{1-b_j} \Big/ \dfrac{a_j}{1-a_j}\right) & \text{otherwise} \end{cases} \tag{4}$$

Like RR and LRR, the minimum weight using OR and LOR is one and zero, respectively. Eqs. (1) and (3) are monotonically related to their logarithm versions of (2) and (4) respectively.

The fifth measure for discriminative term weights used is the information theoretic measure Kullback-Leibler (KL) divergence. The KL divergence of probability distribution $p(x)$ from probability distribution $q(x)$ is defined as

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

The KL divergence can 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 $p(x_j|c \in C\backslash k)$ where x_j can take on values of zero and one. In this context, the expected discrimination information provided by knowledge of term j for category k over other categories is given by the KL divergence (KLD) as

$$w_j^k = D_{KL}(p(x_j|c=k)\|p(x_j|c \in C\backslash k) \tag{5}$$

$$= a_j \log \frac{a_j}{b_j} + (1-a_j) \log \frac{1-a_j}{1-b_j} \tag{6}$$

Except for RR and LRR, the remaining three DTW measures consider both the occurrence and the absence of a term. Eq. 5 is also monotonically related with the other four measures but is not symmetric. Even though RR asymptotically approaches OR for small probabilities, both are quite different. If values of a_j and b_j are significantly small, then Eqs. (3) and (4) approximate Eqs. (1) and (2) respectively. The distinction between the two becomes pronounced in cases of medium to high probabilities. For example, if a term occurs in category k and in categories $C\backslash k$ with 0.999 and 0.99 probability, respectively, the relative risk is just over 1 while the odds ratio is more than 10 times higher.

In any case, all five measures quantify the discrimination information provided by term j for discrimination between category k and categories $C\backslash k$, with larger weights signifying higher 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). This prevents the weight of a term to become infinity, a situation that arises because of division by zero during weight computation.

3.3. Term space partitioning and term selection

The DTWs described in the previous section are then used for term space partitioning and term selection. Our weighting strategies naturally partition the terms into two sets: the first set identified by the index set Z^k, contains terms for which $a_j > b_j$. The second 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 by their discriminative term weights. In the next subsection, we describe how we use this partitioning to create a feature space for the classification problem.

Our weighting strategies also provide a natural way of selecting highly discriminating and relevant terms. A term j is deemed significant if

$$|a_j - b_j| \geq t$$

where $j \in \{Z^k, Z^{C\setminus k}\}$ and 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 or no discrimination information. This is a supervised and more direct approach for term selection as compared to the common techniques used in practice like information gain and principal component analysis. It is reasonable to expect that the classification accuracy will be the highest when all features are selected i.e. $t = 0$; however, because of feature information noise and variance the highest accuracy is often achieved with a reduced feature set. DIST does not require term selection and dimensionality reduction as it efficiently transforms the input terms to a two-dimensional feature space (described in the next subsection). However, term selection may be necessary for large scale online applications like personalized spam filtering by e-mail service providers [35]. For such applications, DIST's accuracy can be traded off with its space and time complexity by varying the value of t.

3.4. Linear opinion pool and linear discrimination in feature space

The two-set partitioning of the term space, i.e. Z^k and $Z^{C\setminus k}$, is used to construct a two-dimensional feature space. Each term j in the document $\mathbf{x}$ expresses an opinion regarding the document's categorization. This opinion is captured by the discriminative term weights w_j^k and $w_j^{C\setminus k}$. The terms in the set Z^k give their opinion regarding the document's membership to the category k, Terms in the set $Z^{C\setminus k}$ give their opinion regarding the document's membership to the category $C\setminus k$. The aggregated opinion of all these terms is obtained as the linear combination of individuals' opinions:

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

This equation follows from a linear opinion pool or an ensemble average, which is a statistical technique for combining experts' opinions [31]. Each opinion (w_j^k) is weighted by the normalized term occurrence ($x_j/\Sigma x_j$), and all weighted opinions are summed yielding an aggregated discrimination score of the document for category k ($Score^k(\mathbf{x})$). 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 $j \in Z^{C\setminus k}$ as

$$Score^{C\setminus k}(\mathbf{x}) = \frac{\sum_{j\in Z^{C\setminus k}} x_j w_j^{C\setminus k}}{\sum_j x_j}. \tag{8}$$

These two scores, $Score^k(\mathbf{x})$ and $Score^{C\setminus k}(\mathbf{x})$ define the two-dimensional feature space . In this space, documents are well separated and discriminated, as illustrated for a spam classification data (Fig. 1). The documents belonging to category k and category $C\setminus k$ are nicely aligned with the y-axis and x-axis respectively. Each discrimination score in the transformed space is a consolidation of the opinion of discriminative terms of its corresponding category. These scores can be treated as confidence values of document's membership in the respective categories. A decision rule could be to assign the document to the category for which it has the highest discrimination score. Although this simple rule works well in some situations it is not robust enough for broader applications. In particular, it fails when the distribution of categories in a data set are very different or when there is a significant class imbalance. We solve this problem by learning the categorization in this new feature space by a linear discriminant function:

$$f^k(\mathbf{x}) = \alpha^k \cdot Score^k(\mathbf{x}) - Score^{C\setminus k}(\mathbf{x}) + \alpha^0 \tag{9}$$

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 (Fig. 1). For a $|C|$ category classification problem, we learn $|C|$ 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 [46]. The category with the highest score is assigned to the document i.e. DIST's overall classifier function is defined as

$$\Phi(\mathbf{x}) = \text{argmax}_k\ f^k(\mathbf{x}). \tag{10}$$

DIST 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 DIST efficient, in terms of both time and space, and robust to noise and changing data distributions. DIST contains three key steps: (1) discriminative term weight computation, which can be done in a single pass over the labeled data set, (2) constructing the two-dimensional feature space, and (3) learning the parameters of the discriminating line which can be done efficiently using common optimization algorithms. The DIST algorithm is given in Algorithm 1.

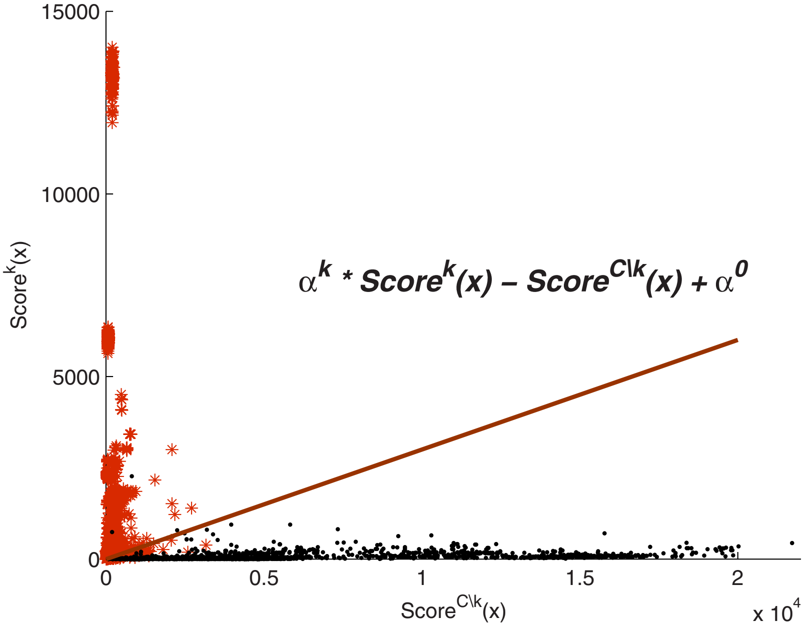

Fig. 1. The two-dimensional feature space and the linear discriminant function for a spam classification problem.

Algorithm 1 DIST

```
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 $w_j^{C\backslash k}$ (Eqs. 1, 2, 3, 4 or 5)
  end for
  compute $Score^k(\mathbf{x})$ and $Score^{C\backslash k}(\mathbf{x})$ (Eqs. 7 and 8)
  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. 7 and 8)
  compute $f^k(\mathbf{x})$ (Eq. 9)
end for
output $k = argmax_k f^k(\mathbf{x})$ (Eq. 10)
```

4. Interpretations and comparisons

In this section, we provide a broader interpretation of DIST by comparing it with generative, discriminative, and hybrid classifiers.

4.1. Relation to naive Bayes classifier

Here, we develop the naive Bayes classifier and show its relation to DIST. The relative risk that a document **x** belongs to category k rather than categories $C\backslash k$ can be written as

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

Assuming that the occurrence of each term is independent of others given the category, the document's relative risk on the right hand side becomes a product of terms' relative risks. The naive Bayes classification of a document **x** is category k

when

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

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

$$\log\frac{p(c=k)}{p(c\in C\backslash k)}+\sum_j x_j\log\frac{p(x_j|c=k)}{p(x_j|c\in C\backslash k)}>0 \tag{11}$$

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. (11), with DIST yields some interesting observations. The discriminative model of DIST is similar to Eq. (11) in that the structure of the discrimination score computation (Eqs. (7) and (8)) is similar to the summation in Eq. (11) and the bias parameter α^0 corresponds to the first term in Eq. (11). The log relative risk for term j in Eq. (11) corresponds to the log relative risk discriminative weighting measure in DIST.

However, there are also significant differences between DIST and naive Bayes. (1) The discrimination scores in DIST are normalized (for each document) using the ℓ_1 norm. Document length normalization is typically not done in naive Bayes classification, and when it is, the ℓ_2 norm is used. (2) DIST 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) DIST allows the use of different discriminative term weighting measures as long as they quantify the discrimination information that a term provides for one category over the others. (4) DIST does not require the naive Bayes assumption of conditional independence of the terms given the category.

DIST will be identical to naive Bayes when the log relative risk term weighting measure is used, discrimination scores are not normalized, the slope parameter α^k is equal to one, and the bias parameter α^0 is equal to the first term in Eq. (11).

4.2. Discriminative classifiers

Popular discriminative classifiers learn a hyperplane or linear discriminant in the space representing the objects to be classified (documents in our case). Let ϕ: $X \to C$ be the function that maps a document $\mathbf{x} \in X$ from the T-dimensional input space to $\mathbf{v}$ in a d-dimensional feature space. Then, a hyperplane in the feature space is defined by

$$\sum_{j=1}^{d}\alpha_j v_j+\alpha_0=0 \tag{12}$$

where $\alpha_j (j = 1, 2, \ldots, d)$ are the parameters of the hyperplane.

DIST's discriminative model is a linear discriminant. However, this discriminant function is learned in a two-dimensional feature space defined by scores (Eqs. (7) and (8)) and has only two parameters (Eq. (9)). Input-to-feature space transformation is typically not done for discriminative classifiers like balanced winnow/perceptron and logistic regression. In SVM, this transformation is done implicitly through the inner product kernel $k(\mathbf{x},\mathbf{x}') = \langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle$, where $\boldsymbol{\phi}(\,\cdot\,)$ is the function that maps from input to feature space.

The input-to-feature space transformation in DIST can be written as

$$\phi(\mathbf{x})=[\phi_1(\mathbf{x})\ \phi_2(\mathbf{x})]^T=\left[Score^k(\mathbf{x})\ Score^{C\backslash k}(\mathbf{x})\right]^T \tag{13}$$

where the scores are defined in Eqs. (7) and (8). This represents a linear mapping from a T-dimensional input space to a two-dimensional feature space. The kernel is then defined as follows (after substitution and using vector notations):

$$k(\mathbf{x},\mathbf{x}')=\phi^T(\mathbf{x})\phi(\mathbf{x}')=\bar{\mathbf{x}}^T\mathbf{W}^+\bar{\mathbf{x}}'+\bar{\mathbf{x}}^T\mathbf{W}^-\bar{\mathbf{x}}' \tag{14}$$

where $\mathbf{W}^+ = \mathbf{w}^+\mathbf{w}^{+T}$ and $\mathbf{W}^- = \mathbf{w}^-\mathbf{w}^{-T}$ are $T \times T$-dimensional matrices and $\bar{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}\|_{L1}$, $\bar{\mathbf{x}}' = \mathbf{x}'/\|\mathbf{x}'\|_{L1}$. The elements of vector $\mathbf{w}^+$ (vector $\mathbf{w}^-$) are equal to w_j (Eq. (1)) when $j \in Z^k$ ($j \in Z^C\backslash k$) and zero otherwise. Noting that terms in the vocabulary are hard-partitioned, we can write

$$k(\mathbf{x},\mathbf{x}')=\bar{\mathbf{x}}^T\mathbf{W}\bar{\mathbf{x}}' \tag{15}$$

where $\mathbf{W} = \mathbf{W}^+ + \mathbf{W}^-$.

The following observations can be made from the above discussion. (1) DIST performs a linear transformation from the input space to a *lower* dimension feature space. This feature space is formed in such a way that the discrimination between categories k and $C\backslash k$ is enhanced. Recently, it has been shown that feature space representations are critical to making classifiers robust for domain adaptation [2,5]. (2) The smoothing matrices $\mathbf{W}$, $\mathbf{W}^+$, and $\mathbf{W}^-$ are symmetric and the associated kernel is positive semi-definite. (3) The transformation is supervised, requiring information about class labels.

4.3. Hybrid classifiers

Multiple classifiers can be combined to form a single classifier. This technique is referred to as classifier ensembles, combiners, hybrid, or committees. They aggregate several classifiers by combining their individual predictions through voting or averaging [58]. They generally provide better performance than their constituent models (also known as base models) [27], but are generally slower because multiple models are learned.

[30] discusses a hybrid classifier in which the kernel function is derived from a generative model. The input-to-feature space transformation is based on the Fisher score and the resulting kernel is a Fisher kernel. Our input-to-feature space transformation is based on discrimination scores computed from discrimination information provided by the terms in the text document. [60] presents another hybrid classifier in which the input space (defined by the terms) is partitioned into two sets (based on domain knowledge). The weights for each set of class conditional distributions (learned generatively) are learned discriminatively. Discriminative model parameters of DIST are similar in purpose. However, the two sets of class conditional distributions in DIST correspond to the sets of significant terms of the two classes, which are determined from labeled documents. [37] extends naive Bayes by splitting features into two sets. One of the feature set is maximized through discriminant model and the other through generative model. [7] proposes a method to trade-off generative and discriminative modeling. It is similar to multi-conditional learning because it maximizes a weighted combination of two likelihood terms using one set of parameters. They learn a discriminative model and a generative model and present a combined objective function. [29] uses naive Bayes approach to vectorize a document according to a probability distribution reflecting the probable categories that the document may belong to. SVM is then used to classify the documents in this vector space on a multi-dimensional level. The results in [29] show that this hybrid approach performs better than naive Bayes on most of the data sets. Other works have been done by [38] (Multi-conditional Learning), [3] (Hybrid Markov/Semi-Markov Conditional Random Fields) [65] (Structured Output Learning (SOL)), and [44] (Bayes Perceptron Model).

5. Evaluation setup

We evaluate DIST on six commonly-used text classification data sets – personalized spam filtering (ECML), movie review (Movies), 20 Newsgroups (20NG), Simulated Real Auto Annealing (SRAA), ECUE and PU email data set. Three of these data sets have more than one train-test data pairs. Not only the data sets are from a varied domain but have different underlying characteristics, with some data sets having distribution shift while others don't, some are two class problems while others are multiclass, some having small training set while others have a large one. We compare DIST performance with four other classifiers – naive Bayes (NB), balanced winnow (BW), maximum entropy (ME), and support vector machine (SVM). DIST's performance with relative risk (DIST-RR), log relative risk (DIST-LRR), odds (DIST-OR), log odds (DIST-LOR), and KL divergence (DIST-KL) discriminative term weighting strategies is reported. For naive Bayes, maximum entropy, and balanced winnow we use the implementation provided by the Mallet toolkit [51]. For SVM, we use the implementation provided by *SVM^{Light}* [33]. We report the classification accuracy for all the data sets, and the mean of classification accuracy for movie and SRAA data sets calculated over 5 runs of the algorithms. The AUC (area under the ROC curve) value is considered as a more robust measure of classifier performance [12], therefore we also compare the classifiers based on their AUC values.

5.1. Data sets

In all the six data sets, a documents is represented as a bag-of-words or -terms. In [19], it has been shown that term occurrence vectors perform slightly better than term frequency vectors (more than 0.3% in AUC value), therefore we convert the data sets to the former format. Stop words, HTML tags, and message headers are also removed.

5.1.1. 20 News groups

20 Newsgroups (20NG) data set collected by Ken Lang [39] is a very popular data set for text classification in the data mining and machine learning community. It is a collection of about 20,000 newsgroup documents from around 20 newsgroups, each corresponding to a different topic. Some of these topics are related and can be loosely grouped in six different topic categories e.g. autos, motorcycles are related topics. There are three different versions of 20 Newsgroups data set available.[1] The first one is the original data set that has 19,997 documents, the second version called the "bydate" version is sorted by date into training (60%) and testing (40%) with duplicates and header information removed. The third version called "18828" does not include duplicates, and includes the "Form" and "Subject" headers. We chose the "bydate" version for three reasons: firstly, the duplicates are removed, secondly newsgroup identifying information (header information) is left out, and lastly there is no randomness in the selection of training and test set which makes it more realistic.

5.1.2. ECML

The personalized spam filtering data set, henceforth identified as the ECML data set, captures the e-mail classification problem in which individual user's e-mails are labeled as either spam or non-spam. A common general labeled training set

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

is used for each of the user's. This data set corresponds to data set A provided by the 2006 ECML/PKDD Discovery Challenge [6]. It contains a labeled training set of 4000 e-mails and three unlabeled users inboxes of 2500 e-mails each, i.e. ECML-1, ECML-2 and ECML-3. 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).

5.1.3. Movie review

The movie review data set[2], 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). It consist of 2000 positive and 2000 negative reviews. We remove the stop words/terms using the mallet toolkit [51]. We holdout 400 examples of each class for testing and randomly select different numbers of examples for training.

5.1.4. SRAA

The SRAA (Simulated/Real/Aviation/Auto) data set[3] 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. We holdout 1000 examples of each class for testing and randomly select different numbers of examples for training.

5.1.5. PU1 And PU2 data sets

The PU1 and PU2 data sets contain e-mails received by a particular user [4]. The order in which the e-mails are received is not preserved in these data sets. Moreover, only the earliest five non-spam e-mails from each sender are retained in the data sets. Attachments, HTML tags, and duplicate spam e-mails received on the same day are removed before preprocessing. The PU1 data set is available in four versions depending on the preprocessing performed. We use the version with stop words removed. The PU2 data set is available in the bare form only, i.e., without stop word removal and lemmatization.

The PU1 data set contains 481 spam and 618 non-spam e-mails available in 10 partitions or folders. We select the first 7 folders for training and the last 3 for testing. Within the training and test sets we retain 362 and 145 e-mails, respectively, of each class for our evaluation. The PU2 data set is also available in 10 folders with the first 7 folders selected for training and the last 3 for testing. There are 497 training e-mails (399 are non-spam) and 213 test e-mails (171 are non-spam). For this data set, we do not sample the e-mails to achieve even proportions of spam and non-spam e-mails because doing so produces very small training and test sets.

5.1.6. ECUE-1 And ECUE-2 data sets

The ECUE-1 and ECUE-2 data sets are derived from the ECUE concept drift 1 and 2 data sets, respectively [16]. Each data set is a collection of e-mails received by one specific user over the period of one year. The order in which the e-mails are received is preserved in these data sets. The training sets contain 1000 e-mails (500 spam and 500 non-spam) received during the first three months. The test sets contain 2000 e-mails (1000 spam and 1000 non-spam) randomly sampled from the e-mails received during the last nine months. As such, concept drift exists from training to test sets in these data sets.

These data sets are not preprocessed for stop word, stemming, or lemmatization. E-mail attachments are removed before parsing but any HTML text present in the e-mails is included in the tokenization. A selection of header fields, including the Subject, To and From, are also included in the tokenization. These data sets contain three types of features: (a) word features, (b) letter or single character features, and (c) structural features, e.g., the proportion of uppercase or lowercase characters. See [16] for further details.

5.2. Tuning the algorithms

Documents are represented by term frequency vectors for the NB, ME, BW, and SVM classifiers. For DIST, however, we use term occurrence vectors for document representation. An extensive evaluation of DIST with different document vector representations is beyond the scope of this paper. 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 25% 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 nonlinear kernels and find the linear kernel to be superior. This observation is consistent with that reported in the literature [18,72]. We perform document length normalization using $L2$ (Euclidean) norm. This improves performance slightly from the non-normalized case, as observed by others as well [18,72]. We keep the remaining parameters of SVM^{Light} at default values. The parameters of DIST are tuned on the training set for each data set.

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data.

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

Table 1
Comparison of DIST variants using the five discriminative term weighting strategies. Mean accuracy computed over 5 runs with randomly drawn training sets of sizes specified in the *Tr.* column and randomly selected test sets of sizes 800 and 4000 is reported for Movie and SRAA data sets respectively. For ECML, ECUE, PU and 20 Newsgroup data sets, test set or inbox is specified in *Ts.* column. Average accuracy values are reported in italic and top values are reported in bold typeface.

Data	Tr.	DIST-RR	DIST-LRR	DIST-KL	DIST-OR	DIST-LOR
Movie	600	80.90	81.35	**82.32**	81.07	81.59
	500	82.47	82.40	**83.24**	82.69	82.20
	400	79.84	81.42	**81.52**	80.27	81.57
	300	79.64	80.07	**82.27**	79.96	79.55
	200	78.02	79.30	**80.87**	78.17	79.54
	Avg.	*80.17*	*80.91*	***82.04***	*80.43*	*80.89*
SRAA	1500	**93.41**	91.93	88.61	93.40	91.92
	1000	**92.94**	91.14	88.50	92.85	91.12
	500	91.26	88.48	87.50	**91.28**	88.40
	250	**88.88**	83.60	85.52	88.79	83.95
	150	86.63	78.12	83.74	**86.77**	78.39
	Avg.	***90.62***	*86.65*	*86.77*	***90.62***	*86.76*
	Ts.					
ECML	1	91.00	**91.12**	79.88	90.64	90.04
	2	**92.36**	91.96	82.24	91.96	92.00
	3	87.52	**88.60**	68.88	87.24	88.04
	Avg.	*90.29*	***90.56***	*77.00*	*89.95*	*90.03*
ECUE	1	**92.20**	91.95	83.05	81.25	86.95
	2	83.45	82.65	**89.25**	88.05	84.45
	Avg.	***87.83***	*87.30*	*86.15*	*84.65*	*85.70*
PU	1	98.27	98.62	97.58	98.62	**98.96**
	2	**97.18**	94.83	94.36	96.71	95.30
	Avg.	***97.73***	*96.73*	*95.97*	*97.67*	*97.13*
20NG	-	**78.73**	35.13	66.59	78.59	34.73
Overall	*Avg.*	***87.56***	*79.55*	*82.42*	*86.99*	*79.21*

5.2.1. Parameter estimation

DIST 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 is a convex optimization problem, as empirically verified from the error versus slope parameter graph (not shown here). 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. Results and discussion

In this section we first demonstrate the results of using different discriminative term weighting strategies with our classifier DIST. The most consistent measure is then chosen and compared with other methods. Other experiments and discussions on the results are also provided.

5.3.1. Classification performance

Table 1 shows the classification accuracy values of the five DIST variants on Movie, SRAA, ECML, ECUE, PU and 20NG data sets. The results for DIST with the five measures; relative risk, log relative risk, odds, log odds, and KL divergence, are identified by DIST-RR, DIST-LRR, DIST-OR, DIST-LOR and DIST-KL, respectively. DIST using relative risk (DIST-RR) achieves the highest and the most consistent results overall. Thus we select DIST-RR for further comparisons with other competitor methods, i.e., naive Bayes (NB), maximum entropy (ME), balanced winnow (BW), and SVM, in Table 2. The training and test splits for Movie and SRAA data sets, were chosen randomly, therefore we give the mean of the classification accuracies over five runs of the classifier. For the ECML data set, we report results for each user inbox separately.

DIST performs exceptionally good for data sets that have a distribution shift between the training and test data such as the ECML data set. The distribution of training and test sets are similar for the Movie and the SRAA data sets. For these data sets also, DIST outperforms all the other algorithms, however by a lesser margin as compared to that for the ECML data set. Notice that the performance of DIST degrades gracefully as the number of examples in the training set is reduced. The results obtained by NB, ME,and SVM are comparable to those reported in [18,50]. DIST's performance appears slightly

Table 2
Comparison of our method DIST (using relative risk, i.e., DIST-RR) with other methods. Mean accuracy computed over 5 runs with randomly drawn training sets of sizes specified in the *Tr.* column and randomly selected test sets of sizes 800 and 4000 is reported for Movie and SRAA data sets respectively. For ECML, ECUE, PU and 20 Newsgroup data sets, test set or inbox is specified in *Ts.* column. Average accuracy values are reported in italic and top values are reported in bold typeface.

Data	Tr.	DIST-RR	NB	ME	BW	SVM
Movie	600	80.90	79.25	**82.14**	78.89	81.85
	500	**82.47**	80.74	81.32	77.92	81.35
	400	**79.84**	79.17	79.62	78.34	79.65
	300	**79.64**	77.57	77.97	76.09	78.52
	200	**78.02**	76.42	76.32	74.12	76.10
	Avg.	***80.17***	*78.63*	*79.47*	*77.07*	*79.49*
SRAA	1500	**93.41**	92.72	90.53	88.23	91.54
	1000	**92.94**	92.10	89.12	87.54	89.34
	500	**91.26**	90.59	86.75	85.01	86.73
	250	**88.88**	88.05	83.28	81.94	84.52
	150	**86.63**	85.69	81.87	79.97	83.58
	Avg.	***90.62***	*89.83*	*86.31*	*84.54*	*87.14*
	Ts.					
ECML	1	**91.00**	81.24	62.20	61.00	64.40
	2	**92.36**	83.80	68.16	64.76	69.56
	3	87.52	**87.88**	78.92	73.44	80.24
	Avg.	***90.29***	*84.31*	*69.76*	*66.40*	*71.40*
ECUE	1	**92.20**	50.05	78.30	83.05	83.30
	2	**83.45**	50.00	79.50	77.50	76.95
	Avg.	***87.83***	*50.03*	*78.90*	*80.28*	*80.13*
PU	1	**98.27**	96.55	96.89	97.24	96.21
	2	**97.18**	87.32	94.36	90.61	88.26
	Avg.	***97.73***	*91.94*	*95.63*	*93.93*	*92.24*
20NG	-	**78.73**	73.67	71.20	60.32	77.32
Overall	*Avg.*	***87.56***	*78.07*	*80.21*	*77.09*	*81.29*

lesser than that of multi-conditional learning reported in [50]; however, their exact evaluation and data set is not known so a direct comparison is not possible.

Even though the results of NB, ME, BW and SVM are comparable, but each one of them performs significantly poorly on at least one of the data sets. For example, NB for ECUE-1 and ECUE-2 data sets, BW for 20NG and ECML data set, ME for ECUE-1 and ECML data sets, and SVM for the ECML data set. DIST fares as the most consistent classifier of the lot, and we show in Section 6.2 that its performance is statistically better than the above mentioned classifiers. One surprising result is that DIST-LRR performs poorly on the 20NG as compared to its non-log variant, this is also true for DIST-LOR. One reason for this is that taking log of the ratio reduces the magnitude of the discrimination value, and this effect becomes more visible in multi-class data sets.

The AUC values for the above data sets are reported in Figs. 2 and 3. Both of the figures do not show the result for SRAA and 20NG data sets because the AUC measure is defined for two class problems only. The Fig. 3 shows that in terms of AUC, DIST consistently outperforms all the other algorithms by a big margin. Fig. 2 compares the performance of the aforementioned weighting strategies. DIST-LRR and DIST-LOR perform slightly better than their without log versions often. The worst performer is the DIST-KL but it catches up with the rest on the PU and Movies data sets. This characteristic of the DIST-KL can be credited to the penalizing of absence of a term by the Kullback-Leibler divergence measure. Since the distribution shift is largest for the ECML data set, many terms that occur in the training set are absent in the test set, as a result the performance of DIST-KL is the worst for this data set. Because of the lesser distribution shift in the ECUE data set (only temporal shift), DIST-KL is not far away from the rest of the weighting schemes. As for PU and Movies 600 which do not have this distribution shift, DIST-KL performs equally better as the rest of the schemes. In Section 5.3.2, we further explore the quantification and effects of distribution shift on these algorithms.

5.3.2. Varying distribution shift

We also evaluate the performance of DIST, NB, ME, BW, and SVM under varying distribution shift between training and test data. This evaluation is performed on ECML data set by swapping varying numbers of e-mails between training and test (user) data. By increasing the number of e-mails swapped, the distribution shift between training and test data reduces. To illustrate the evaluation procedure, suppose 100 randomly selected e-mails from user 1 are moved to the training data and 100 randomly selected e-mails from the training data are moved to user 1 e-mails. The filters are then trained and tested

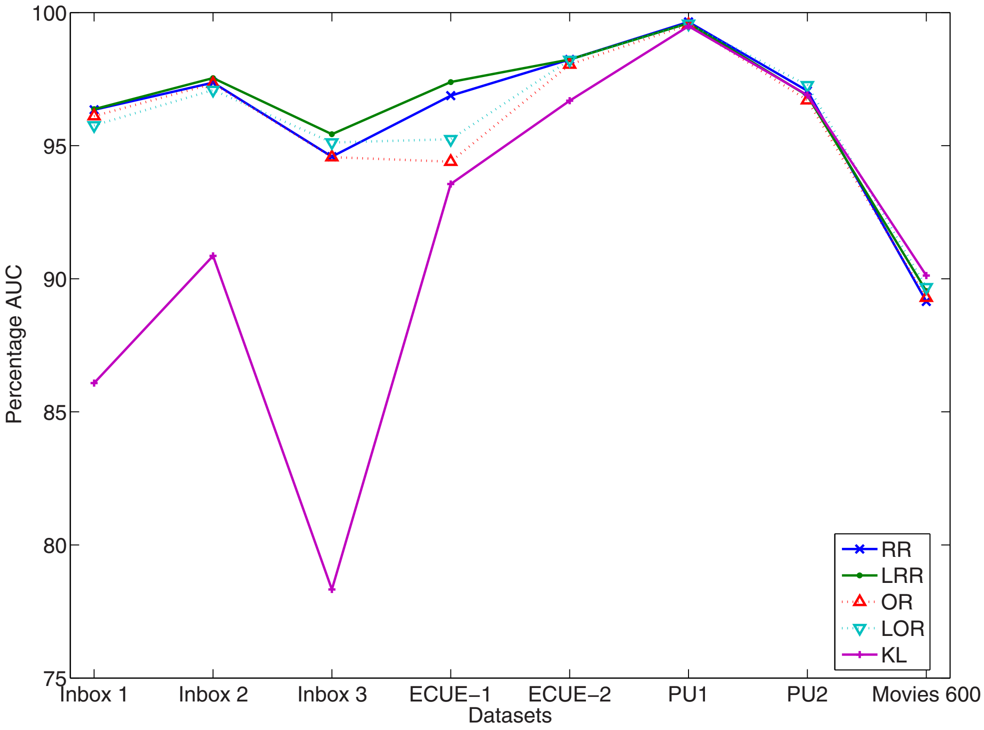

Fig. 2. Comparison between different weighting schemes using area under the curve (AUC).

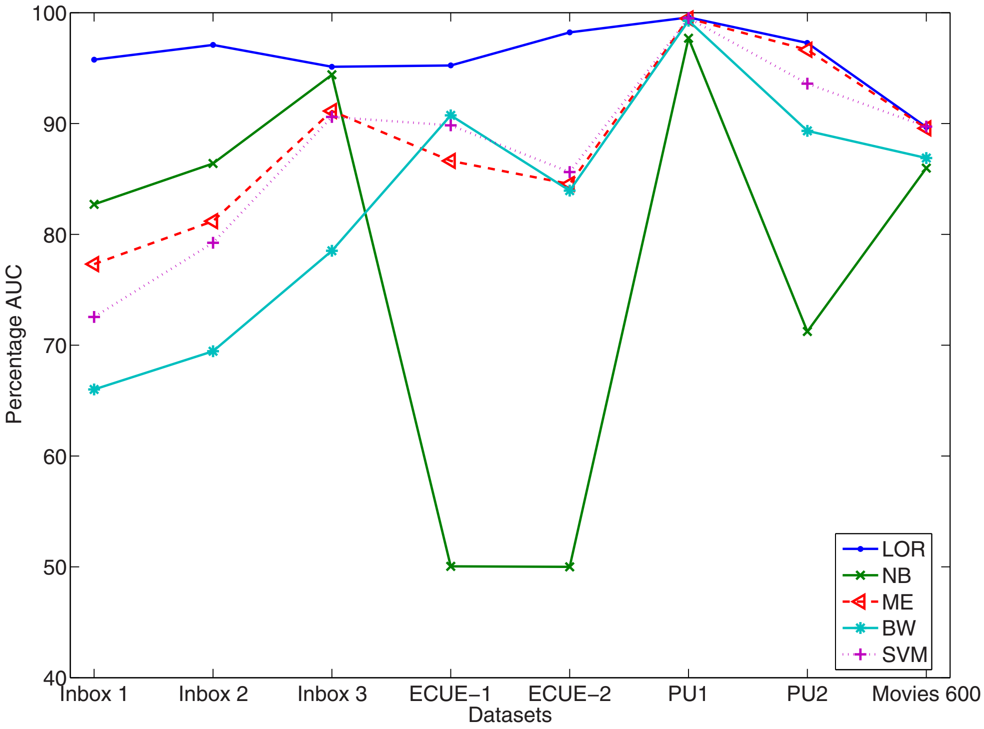

Fig. 3. Comparison of DIST with other methods in terms of area under the curve (AUC).

Table 3
Performance under varying distribution shift. Average percentage accuracy and AUC values are given for ECML data set.

#	D_{KL}	D_{TV}	DIST-RR		NB		ME		BW		SVM	
	$(\times 10^{-7})$	$(\times 10^{-7})$	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
0	717	444	**90.29**	**96.10**	84.30	87.83	69.76	83.21	66.40	71.33	71.69	80.79
100	517	384	**93.30**	**98.04**	92.08	95.84	87.90	94.53	75.53	82.73	88.12	94.96
250	477	336	**94.68**	**98.72**	93.70	97.05	92.80	97.36	82.13	88.87	92.08	97.70
500	346	259	**96.60**	**99.38**	95.57	98.01	96.18	98.68	87.29	92.91	95.70	99.17
1500	157	127	97.63	99.55	96.39	97.82	**97.97**	99.01	95.08	98.11	97.30	**99.68**
Avg			***94.50***	***98.35***	*92.40*	*95.31*	*88.92*	*94.55*	*81.28*	*86.79*	*88.97*	*94.46*

Table 4
Classification results through selection of Maximum Scores. Values are percentage accuracies.

Data	RR	LRR	OR	LOR	KL
ECML	64.64	60.52	64.53	60.36	**66.21**
ECUE-1	50.00	50.00	50.00	50.00	50.00
ECUE-2	50.05	**50.10**	50.05	50.00	50.00
PU1	**97.93**	95.86	**97.93**	96.89	83.10
PU2	82.62	80.28	**83.09**	81.22	81.69
Movies 600	67.79	**69.82**	68.04	69.39	67.32
Avg.	*68.84*	*67.76*	***68.94***	*67.98*	*66.39*

using the modified training and test data. This procedure is repeated for each user and for different numbers of e-mails swapped.

To quantify the distribution shift between training and test data we adapt the KL-divergence and total variation distance as follows:

$$D_{KL}(L,U) = \frac{1}{2T}\left[\sum_j p_L(x_j|+)\log\frac{p_L(x_j|+)}{p_U(x_j|+)} + \sum_j p_L(x_j|-)\log\frac{p_L(x_j|-)}{p_U(x_j|-)}\right]$$

$$D_{TV}(L,U) = \frac{1}{2T}\left[\sum_j |p_L(x_j|+) - p_U(x_j|+)| + \sum_j |p_L(x_j|-) - p_U(x_j|-)|\right]$$

where $D_{KL}(\cdot,\cdot)$ and $D_{TV}(\cdot,\cdot)$ denote the adapted KL-divergence and total variation distance, respectively. L and U identify training and test data, respectively, and T is the total number of distinct terms in the training and test data. The quantity D_{KL} (D_{TV}) is computed as the average of the KL-divergence (total variation distance) for spam and non-spam conditional distributions normalized by T. The normalization ensures that these quantities range from 0 to 1 for all training-test data pairs irrespective of the numbers of terms in them.

Table 3 shows the average performance for the three test sets in ECML data set. The table shows that as the number of e-mails swapped between training and test sets (given in the first column of Table 3) increases, the distribution shift between the sets decreases, as quantified by the values of D_{KL} and D_{TV}. More interestingly, it is observed that as the distribution shift decreases the performance gap between DIST and the other algorithms narrows down. The performance of all the algorithms improves with the decrease in distribution shift, especially for ME, BW, and SVM. For example, the average accuracy of ME jumps up by 28.21% from the case when no e-mails are swapped to the case when 1500 e-mails are swapped. Our supervised spam filter, DIST, comprehensively outperforms the other algorithms when distribution shift is large, while its performance compares well with the others at low distribution shift.

5.3.3. Comparison of feature weighing schemes

The results of five feature weighting schemes (RR, LRR, OR, LOR and KL-Divergence) as a part of DIST were compared in Section 5.3.1. Here we compare them on the basis of the generative model (i.e. class with the maximum score is assigned as the label) to determine which weighting measure captures the discriminative information more effectively. This comparison is presented in Table 4. Even though there is no clear winner in this table, a few interesting observations can be drawn from it. The logarithmic variants (LRR and LOR), on average, fare worse than their non-logarithmic variants. Secondly, there is no significant difference between the performance OR and RR (same is the case for LOR and LRR). Thirdly, KL is on average 2.5% behind RR and OR.

5.3.4. Term selection

The threshold t, introduced in Section 3.3 earlier, can be used to trade-off DIST's space requirement and accuracy performance. This is evident from Fig. 4 which shows the variation of the number of selected terms with threshold t for the

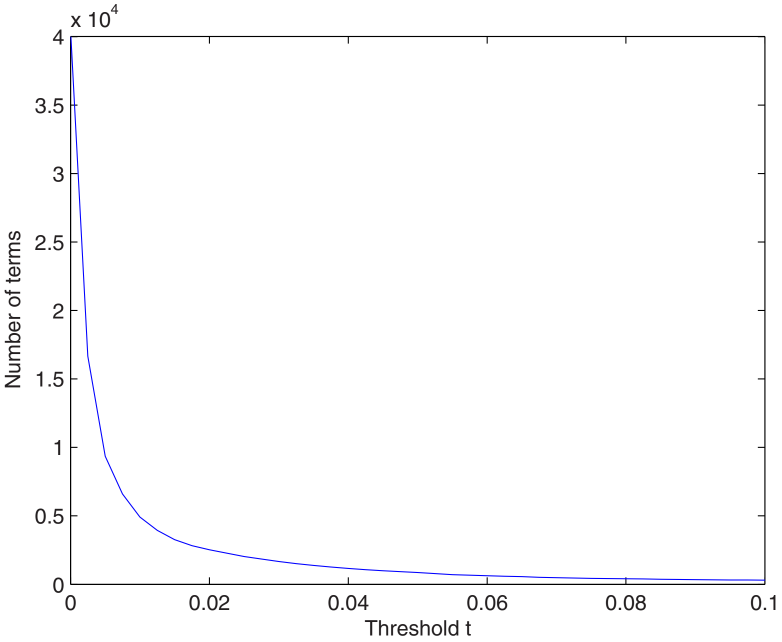

Fig. 4. Number of terms selected versus threshold *t* for ECML data set (DIST-RR).

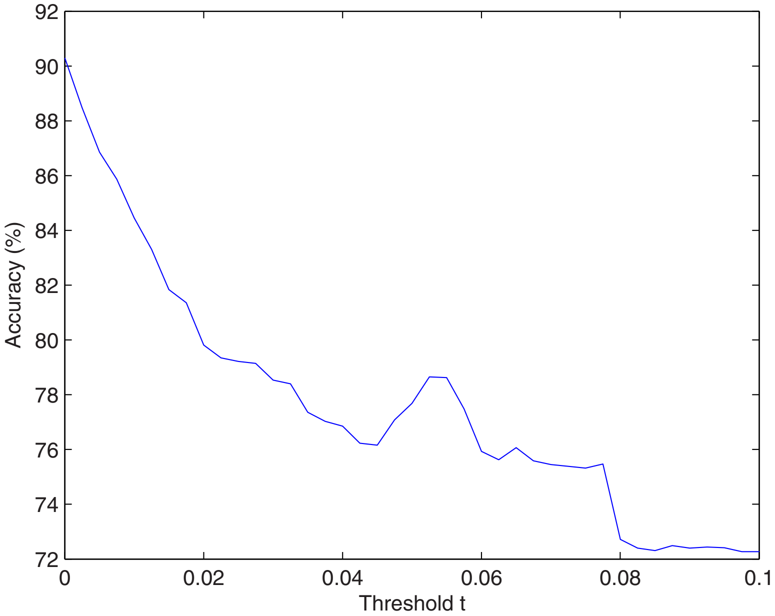

Fig. 5. Average accuracy versus threshold *t* for ECML data set (DIST-RR).

ECML data set (using DIST-RR). The number of selected terms drops significantly with only a small increase in *t*. Remarkably, however, the classification accuracy does not decrease drastically (Fig. 5). Table 5 shows the number of terms and the average accuracy (averaged over the 3 inboxes) of DIST-RR for ECML data set. It is seen that even when the number of terms is reduced by one-eighth (from 40,516 to 4913 terms) the average accuracy value for DIST 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 service side personalized spam filtering by e-mail service providers [35].

Table 5
Number of selected terms and resulting accuracy for different values of the threshold t (DIST-RR on ECML data set).

Threshold	Terms	Accuracy
0.0000	40516	90.29
0.0025	16666	88.48
0.0050	9333	86.85
0.0075	6608	85.86
0.0100	4913	84.45

6. Classifier properties and generalizations

Here we analyze the scalability aspect of DIST by discussing its asymptotic time and space complexity, and some limitations. We then do statistical significance tests to demonstrate that the performance of DIST is significantly better than the compared algorithms. Finally we elaborate on a classifier framework that results as a generalization of the DIST classifier followed by some limitations of DIST.

6.1. Scalability and complexity analysis

DIST is highly efficient in terms of time and space requirements. It only requires a single pass over the labeled data to compute the discriminative term weights (the generative model parameters). Its discriminative model parameters are also obtained by solving a straightforward optimization problem. No further references to the training data are required once the model is learned. Time taken to train the generative model of DIST is $\mathcal{O}(|L|a)$, where $|L|$ is the number of documents, $|T|$ is the size of the dictionary and a is the average length of the document such that $a \ll |T|$. This complexity is linear in terms of the size of the data set. The discriminant model is learned on top of the generative model i.e. in a two dimensional space. Depending on which discriminant model we are using, it can also be learned in linear time. This is the fastest (asymptotically) that we can get for any text classification algorithm.

Event though DIST is trained in linear time, it takes longer to train than NB because it has an additional step of learning a discriminative classifier. The real computational advantage that DIST enjoys is that it classifies new documents asymptotically faster than NB. The time taken by DIST for classifying each document is $\mathcal{O}(|T|)$ because in order to classify each document we have to calculate the $|C|$ scores. These scores are the sums of the weights of $\mathcal{O}(|T|)$ terms that have been partitioned into $|C|$ different mutually exclusive sets. This is faster than NB as it requires $\mathcal{O}(|C||T|)$ time to classify a single document. This is because to calculate the class conditional probabilities, NB calculates the product of class conditional probabilities of each of the $|T|$ terms. Therefore it takes $\mathcal{O}(|T|)$ time to calculate a single class conditional probability and there are $|C|$ such classes.

Similarly, the generative model of DIST requires only $\mathcal{O}(|T|)$ space as compared to the $\mathcal{O}(|C||T|)$ space required by the NB classifier. DIST owes this advantage to its partitioning of the terms into $|C|$ mutually exclusive sets, whereas NB stores the probabilities of all the terms for each class. SVM tends to be quite effective for text classification but at times its training model size becomes very prohibitive. Even though it can be trained in linear time [34], it tends to be slower than NB [47]. For the 20 Newsgroups data set, the size of the SVM model reached more than 1.5 GB with the regularization parameter Y converging at 2.5 million. As for DIST the training model for this data set has a total number of 95,663 features, for each of them we only store their id (integer), weight (float) and the class number (integer). For the discriminative model we only store two parameters for each class. If the integer and float are taken to be of 4 bytes each, then the total size of our model is less than 1.1 mega byte and still DIST performs better than SVM.

Generally $|C| \ll |T|$ and $|C| \ll |L|$, but for problems such as Wikipedia article categorization, 2.9 million articles are assigned to 1.5 million categories [15]. In problems such as targeted advertisement, the number of classes (advertisements) could be asymptotically equal to the number of examples (users). In problems like personalized spam filtering, where millions of users receive more than 100 e-mails per day (both spam and non-spam), the filter size and response time is very crucial. The saving of space per user filter and steps per classification of each email by $|C|$ times is very significant for large e-mail service providers.

In text classification, documents are sparsely represented in a $|T|$ dimensional space. Approaches dependent on document incidence matrix data structure require an $|L|$x$|T|$ dimensional matrix, which results in a huge memory and computation cost. DIST enjoys the benefit of being implemented for a document incidence matrix and as well as for the hash table data structure for which it only takes $\mathcal{O}(|L|a)$ time for training. Using the hash table data structure, the object corresponding to each term (containing id, weight, etc.) can be retrieved and stored in constant time. Hash table data structure is the natural choice for DIST because, a) search and update, which are the most frequently performed operations only take constant time, and b) DIST does not require access to terms in any specific order e.g. sorting of terms on term id's (or weights) or finding

Table 6
Friedman's test with stepwise step-down post-hoc analysis of filters' performances (accuracy).

Homogeneous subsets			
		Subsets	
		1	2
Classifier[1]	BW	1.90	
	NB	2.70	
	ME	2.80	
	SVM	2.90	
	DIST		4.7
Test Statistic		3.00	[2]
Sig. (2-sided test)		0.392	
Adjusted Sig. (2-sided test)		0.392	

[1] Each cell shows the classifier's average rank.

[2] Unable to compute (subset has only one sample)

the term with minimum or maximum term id (or weight). This computational complexity is the best for text classification problems.

6.2. Statistical significance testing

Here we present the results of statistical significance tests that compare the performance of DIST with the other classifiers on all the data sets used. Statistical analysis is necessary to ascertain the consistency of the observed performances and to reject the possibility that they are produced purely by chance. Although numerous statistical tests have been proposed, [17] recommends the non-parametric Wilcoxon signed-ranks test and the non-parametric Friedman's test (with post-hoc analysis) for comparing two or more classifiers on multiple data sets. In our analysis, we are comparing 5 classifiers (DIST, NB, ME, BW, SVM) on 10 data sets (3 ECML-A, 2 PU, 1 20NG, 1 Movies, 1 SRAA, and 2 ECUE data sets). These tests are applied on performances measured through accuracy values. These tests are performed using the SPSS software version 19.

The Wilcoxon signed-ranks test compares the performance of two classifiers on multiple data sets by ranking their absolute difference in performance on the data sets. The null-hypothesis is that the observed differences in performance are insignificant. It is rejected when the smaller of the sum of ranks for positive and negative differences is less than a critical value for a specified confidence level. We find that DIST's performance is significantly different (better) than the others, in fact the null hypothesis is rejected for NB, ME, BW and SVM at the significance level of 0.007, 0.007, 0.005, and 0.007, respectively, with 99% confidence interval.

The Friedman's test evaluates multiple classifiers on multiple data sets by comparing the average ranks of the classifiers on the data sets (higher average ranks are considered better). The null-hypothesis that the classifiers are equivalent is rejected when the Friedman statistic is greater than a critical value at a specified confidence level. We use the Friedman's test with stepwise step-down post-hoc analysis which finds subsets of classifiers that are statistically equivalent (homogeneous subsets). Table 6 shows that, DIST's performance is statistically better than all other classifiers evaluated as it is placed in a subset that contains none of the other classifiers.

These statistical analyses provide evidence for the overall superiority of our algorithm for text classification. They also validate the robustness of our algorithms across different text classification problems.

6.3. Classifier framework

A general framework emerges when the specific decisions made for DIST are generalized. There are five major steps in building DIST, with each step having a variety of options to choose from. For example, in the first step we have used a bag-of-words representation (or descriptors) of the documents, aka vector space model. In this representation each word is a term (attribute) and the order of terms is not preserved 3.1. This is also referred to as a uni-gram representation. Bi-grams, an alternative approach, models attribute as a combination of two words. Hence for $|T|$ unique words in the dataset, the size of the feature vector now becomes $|T|^2$, but some order and semantics of the words are preserved. These bi-grams have shown to outperform the uni-gram on large data sizes [48]. In character gram representation, an attribute is constructed by consecutive characters rather than words. [21] show that in some cases character grams outperform the word bi-gram representations. TF-IDF is another well know term weighting scheme that can be used to weight the above mentioned word grams, and character grams mentioned in Section 3.1. [40] propose an alternative to bag of words representation. They use paragraph vector instead of the word vector, which they show to perform better on their datasets. Named entity recognition, part of speech tagging, and syntactic n-grams are just a few more examples of other representations [52,64]. External knowledge bases have also been used to compute semantically enriched term weights. [56] uses WordNet-based semantic enrichment, while [70] uses Wikipedia concepts and categories. [55] used a combination of more than one of the aforementioned representation to train a SVM classifier to win the first position amongst forty four teams at an international competition organized by Conference on Semantic Evaluation Exercises (SemEval-2013). In the first step of DIST, any of the aforementioned representations can be used without any modification to later stages, because we do not make any assumptions about the representation.

In the second step, we quantify the discrimination information of each of the descriptors. We compared the discrimination information of five information theoretic measures (RR, LRR, OR, LOR, and KL-Divergence) in Table 4. Other measures such as information gain, chi-square or conditional probabilities can serve as a good discrimination information measure. There is a vast number of statistical divergence measures that can be used in our framework, e.g. information theoretic measures such as Hellinger distance, total variation distance, Renyi's divergence, Jensen-Shannon divergence, etc. Probability based measures such as Bhattacharyya distance, Levy-Prokhorov metric and Wasserstein metric (also known as earth movers distance) might capture the discriminative information more effectively for text classification problems. Other approaches common in computer vision literature such as Mahalanobis distance can also be used here as well. Each of these weighting measures have distinct properties that can significantly impact the classifier's outcome. Only few of these measures have been explored in the literature for feature selection let alone feature weighting and feature transformation [19].

Similarly, in the third step we can use arithmetic, harmonic mean, or any other ensemble approach to combine the discrimination information of the descriptors instead of linear opinion pooling. In step four we could do a non-linear transform instead of a linear transform. Lastly, instead of learning a line in the transformed space, we can learn different classifiers such as SVM, neural network (NN), logistic regression etc.

Choosing different options at each step of this framework would result in a different classifier, some of which would be more robust than others. Specifically, the classifier framework contains the following components: (1) Identification of descriptors or experts (we use terms). (2) Quantification of descriptors' (or experts') discrimination opinions (we use terms' RR, LRR, OR, LOR, and KLD). (3) Combining experts' opinions (we use a linear opinion pool). (4) Transformation from input to feature space (we do a linear transformation). (5) Discriminative classification in the feature space (we use a linear discriminant function).

6.4. Limitations

Event though DIST classifies new examples asymptotically faster than NB, it takes longer to train because it has an additional step of learning a discriminative classifier. The time still remains linear because the linear discriminant is learned in linear time. Furthermore, additional pass over the training data is also not required, as this discriminant is learned in the transformed two dimensional space rather than the original term space.

We learn a simple linear discriminant in the discriminative information space. Since there can be infinite many lines to choose from, the learned discriminant function may not be optimal, a problem also faced by neural networks. A maximum margin classifier such as linear SVM, can learn an optimal line in this discriminative space that could give a better generalization onto the unseen data. Doing so would increase the training time, but since it would be learning in a two dimensional space, it is likely to converge quickly than the original term space consisting of thousands of attributes.

As we showed in Section 5.3.2, DIST looses its performance superiority as the distribution shift between the training and test data decreases. Therefore, when the training and test data follow the same distribution it only gives a marginal improvement (Table 2). In this case if training time is of primary concern then it might be better to learn a NB classifier.

7. Conclusion and future direction

In this paper, we present a new text classification methodology, named DIST, based on discriminative information space construction and discrimination information pooling. Each term in the classification problem is assigned a weight that quantifies the discrimination information it provides for category k over the rest. These discriminative term weights are then used to transform the input term space into a new two-dimensional feature space. The transformation is based on a statistical model of opinion pooling. A simple linear discriminant function is then learned in this feature space for final classification.

DIST is evaluated on ten different training and test pairs from six different data sets belonging to news article classification, sentiment analysis, and spam filtering. Its classification accuracy and area under the ROC curve value is compared with that of four other classifiers – naive Bayes, maximum entropy, balanced winnow, and support vector machines. Statistical significance tests show that DIST outperforms the aforementioned classifiers in all settings. Its performance is substantially better in situations where the training and test set follow different distributions. DIST classification time per document is asymptotically $\mathcal{O}(|T|)$ (i.e. independent of the number of categories in the problem), as compared to $\mathcal{O}(|C||T|)$ of naive Bayes. The performance of DIST with one tenth of the features is still better than that of naive Bayes with the full feature set on some data sets. Furthermore, contrary to the popular belief our results also showed that relative risk and its logarithmic variant is either equivalent or even better than odds ratio and its logarithmic variant respectively for text classification. A theoretical comparison of DIST with naive Bayes, discriminative, and hybrid classifiers is also presented.

DIST is efficient, effective, robust, and simple. All these characteristics make it suitable for many text classification problems, especially those having (a) distribution shift between training and test data, (b) high dimensional feature space, and (c) time and memory limitations. DIST is based on a general classifier framework. This framework emerges when the specific decisions made for DIST are generalized. The framework is very rich and diverse with the potential of generating dozens of different classifiers. We have only tested some of these variants. This is a rich framework worth exploring in future.

Acknowledgment

This work is in part supported by the ICT R&D program of MSIP/IITP. [B0101-16-0525, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis], South Korea; Lahore University of Management Sciences (LUMS), University of Management and Technology (UMT), and Higher Education Commission (HEC) of Pakistan.

References

[1] C.C. Aggarwal, C. Zhai, A survey of text classification algorithms, in: Mining text data, Springer, 2012, pp. 163–222.
[2] E. Agirre, O.L. de Lacalle, On robustness and domain adaptation using SVD for word sense disambiguation, in: COLING-08: Proceedings of the 22nd International Conference on Computational Linguistics, Association for Computational Linguistics, 2008, pp. 17–24.
[3] G. Andrew, A hybrid markov/semi-markov conditional random field for sequence segmentation, in: EMNLP 06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2006, pp. 465–472.
[4] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, C.D. Spyropoulos, An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages, in: SIGIR-00: Proceedings of the 23rd Conference on Research and Development in Information Retrieval, ACM, 2000, pp. 160–167.
[5] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, in: NIPS-07: Advances in Neural Information Processing Systems, MIT Press, 2007, pp. 137–144.
[6] S. Bickel, Ecml-pkdd discovery challenge 2006 overview, in: Proceedings of ECML-PKDD Discovery Challenge, 2006, pp. 1–9.
[7] G. Bouchard, B. Triggs, The trade-off between generative and discriminative classifiers, in: IASC 04: 16th Symposium of IASC, Proceedings in Computational Statistics, 2004, pp. 721–728.
[8] P. Raghavan, H. Schutze, C.D. Manning, An Introduction to Information Retrieval, Cambridge University Press, England, 2009.
[9] D. Cai, C.J. Van Rijsbergen, Learning semantic relatedness from term discrimination information, Expert Syst. with Appl. 36 (2) (2009) 1860–1875.
[10] V.R. Carvalho, W. Cohen, Single-pass online learning: performance, voting schemes and online feature selection, in: KDD 2006: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 548–553.
[11] Y.M. Chung, J.Y. Lee, A corpus-based approach to comparative evaluation of statistical term association measures, J. Am. Soc. Inf. Sci. Technol. 52 (4) (2001) 283–296. John Wiley and Sons, Inc.
[12] C. Cortes, M. Mohri, Auc optimization vs. error rate minimization, Adv. Neural Inf. Process. Syst. 16 (2004) 313–320.
[13] I. Dagan, Y. Karov, D. Roth, Mistake driven learning in text categorization, in: EMNLP-97: Proceedings of 2nd Conference on Empirical Methods in Natural Language Processing, 1997.
[14] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, M.W. Mahoney, Feature selection methods for text classification, in: KDD-07: Proceedings of 13th International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 230–239.
[15] O. Dekel, O. Shamir, Multiclass-multilabel classification with more classes than examples, in: International Conference on Artificial Intelligence and Statistics, 2010, pp. 137–144.
[16] S.J. Delany, P. Cunningham, L. Coyle, An assessment of case-based reasoning for spam filtering, Artificial Intelligence Review 24 (3–4) (2005) 359–378.
[17] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[18] G. Druck, C. Pal, A. McCallum, X. Zhu, Semi-supervised classification with hybrid generative/discriminative methods, in: KDD-07: Proceedings of 13th Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 280–289.
[19] G. Forman, An extensive empirical study of feature selection metrics for text classification, The Journal of Machine Learning Research (2003) 1289–1305.
[20] G. Forman, I. Cohen, Learning from little: comparison of classifiers given little training, in: Knowledge Discovery in Databases: PKDD 2004, Springer, 2004, pp. 161–172.
[21] D.H. Fusilier, M. Montes-y Gómez, P. Rosso, R.G. Cabrera, Detection of opinion spam with character n-grams, in: Computational Linguistics and Intelligent Text Processing, Springer, 2015, pp. 285–294.
[22] V. Gupta, G. Lehal, A survey of text mining techniques and applications, J. Emerging Technol. Web Intell. 1 (1) (2009) 60–76.
[23] G. Guyatt, A.D. Oxman, S. Sultan, J. Brozek, P. Glasziou, P. Alonso-Coello, D. Atkins, R. Kunz, V. Montori, R. Jaeschke, et al., Grade guidelines: 11. making an overall rating of confidence in effect estimates for a single outcome and for all outcomes, J. Clinical Epidemiol. 66 (2) (2013) 151–157.
[24] W. Hämäläïnen, StatApriori: an efficient algorithm for searching statistically significant association rules, Knowl. Inf. Syst. 23 (2010) 373–399.Springer London.
[25] M.T. Hassan, A. Karim, Clustering and understanding documents via discrimination information maximization, in: PAKDD 12: Proceedings of the 16th Asia Pacific Conference on knowledge discovery and data mining, 2012, pp. 566–577.
[26] M.T. Hassan, A. Karim, J.-B. Kim, M. Jeon, Cdim: Document clustering by discrimination information maximization, Inf. Sci. 316 (2015) 87–106.
[27] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, Pattern Anal. Mach. Intell. IEEE Trans. 16 (1) (1994) 66–75.
[28] D.A. Hsieh, C.F. Manski, D. McFadden, Estimation of response probabilities from augmented retrospective observations, J. Am. Stat. Assoc. 80 (391) (1985) 651–662.
[29] D. Isa, L.H. Lee, V.P. Kallimani, R. RajKumar, Text document preprocessing with the bayes formula for classification using the support vector machine, IEEE Trans. Knowl. Data Eng. 20 (2008) 1264–1272.
[30] T.S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: NIPS 98: Advances in Neural Information Processing Systems, 1998, pp. 487–493.
[31] R.A. Jacobs, Methods for combining experts' probability assessments, Neural Comput. 7 (1995) 867–888.
[32] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: ECML-98: Proceedings on 10th European Conference on Machine Learning, 1998, pp. 137–142.
[33] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, MIT-Press, 1999.
[34] T. Joachims, Training linear svms in linear time, in: KDD-06 Proceedings of the 12th International Conference on Knowledge Discovery and Datamining, 2006, pp. 217–226.
[35] K.N. Junejo, A. Karim, PSSF: a novel statistical approach for personalized service-side spam filtering, in: WI-07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2007, pp. 228–234.
[36] K.N. Junejo, A. Karim, A robust discriminative term weighting based linear discriminant method for text classification, in: ICDM-08: Proceedings of 8th International Conference on Data Mining, 2008, pp. 323–332.
[37] C. Kang, J. Tian, A hybrid generative/discriminative bayesian classifier, in: FLAIRS-06: Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2006, pp. 562–567.
[38] B.M. Kelm, C. Pal, A. McCallum, Combining generative and discriminative methods for pixel classification with multi-conditional learning, in: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 2, IEEE, 2006, pp. 828–832.

[39] K. Lang, Newsweeder: Learning to filter netnews, in: ICML-95: Proceedings of the Twelfth International Conference on Machine Learning, 2006, pp. 331–339.

[40] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1188–1196.

[41] H. Li, J. Li, L. Wong, M. Feng, Y.P. Tan, Relative risk and odds ratio: a data mining perspective, in: PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2005, pp. 368–377.

[42] J. Li, G. Liu, L. Wong, Mining statistically important equivalence classes and delta-discriminative emerging patterns, in: KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 430–439.

[43] L. Li, X. Jin, S.J. Pan, J.-T. Sun, Multi-domain active learning for text classification, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2012, pp. 1086–1094.

[44] J. Liu, J.-Q. Song, Y.-L. Huang, A generative/discriminative hybrid model: Bayes perceptron classifier, in: Machine Learning and Cybernetics, 2007 International Conference on, 5, IEEE, 2007, pp. 2767–2772.

[45] T. Liu, Z. Chen, B. Zhang, W.-y. Ma, G. Wu, Improving text classification using local latent semantic indexing, in: Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on, IEEE, 2004, pp. 162–169.

[46] D.G. Luenberger, Linear and Nonlinear Programming, second ed., Reading, Mass: Addison-Wesley, 1984.

[47] H. Malik, D. Fradkin, F. Moerchen, Single pass text classification by direct feature weighting, Knowledge and Information Systems 28 (1) (2011) 79–98.

[48] C.D. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[49] A. McCallum., K. Nigam, A comparison of event models for naive bayes text classification, in: In AAAI 98: Workshop on Learning for Text Categorization, 1998, pp. 41–48.

[50] A. McCallum, C. Pal, G. Druck, X. Wang, Multi-conditional learning: generative/discriminative training for clustering and classification, in: AAAI-06: Proceedings of the 21st National Conference on Artificial Intelligence, vol.21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 433.

[51] A.K. McCallum, Mallet: a machine learning language toolkit, 2002. http://mallet.cs.umass.edu.

[52] G. McDonald, C. Macdonald, I. Ounis, Using part-of-speech n-grams for sensitive-text classification, in: Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ACM, 2015, pp. 381–384.

[53] V. Metsis, I. Androutsopoulos, G. Paliouras, Spam filtering with naive bayes – which naive bayes? in: CEAS-06: Proceedings of 3rd Conference on Email and Anti-Spam, 2006, pp. 27–28.

[54] D. Miao, Q. Duan, H. Zhang, N. Jiao, Rough set based hybrid algorithm for text classification, Expert Syst. Appl. 36 (5) (2009) 9168–9174.

[55] S.M. Mohammad, S. Kiritchenko, X. Zhu, Nrc-canada: building the state-of-the-art in sentiment analysis of tweets, in: Second Joint Conference on Lexical and Computational Semantics ($*$ SEM), vol.2, 2013, pp. 321–327.

[56] J.A. Nasir, I. Varlamis, A. Karim, G. Tsatsaronis, Semantic smoothing for text clustering, Knowl. Based Syst. 54 (2013) 216–229.

[57] K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, in: IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999, pp. 61–67.

[58] N.C. Oza, K. Tumer, Classifier ensembles: Select real-world applications, Inf. Fusion 9 (1) (2008) 4–20.

[59] V. Kumar, P.N. Tan, J. Srivastava, Selecting the right objective measure for association analysis, Inf. Syst. 29 (4) (2004) 293–313.

[60] R. Raina, Y. Shen, A.Y. Ng, Classification with hybrid generative/discriminative models, NIPS 03: Advances in Neural Information Processing Systems, 2003.

[61] C. Salperwyck, V. Lemaire, Learning with few examples: An empirical study on leading classifiers, in: Neural Networks (IJCNN), The 2011 International Joint Conference on, IEEE, 2011, pp. 1010–1019.

[62] K.-M. Schneider, A comparison of event models for naive bayes anti-spam e-mail filtering, in: In EACL 03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2003, pp. 307–314.

[63] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surveys 34 (1) (2002) 1–47.

[64] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández, Syntactic n-grams as machine learning features for natural language processing, Expert Syst. Appl. 41 (3) (2014) 853–860.

[65] J. Suzuki, A. Fujino, H. Isozaki, Semi-supervised structured output learning based on a hybrid generative and discriminative approach, in: (ACL 07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007, pp. 791–800.

[66] A. Tariq, A. Karim, F. Gomez, H. Foroosh, Exploiting topical perceptions over multi-lingual text for hashtag suggestion on twitter, in: The Twenty-Sixth International FLAIRS Conference, 2013.

[67] M.J. Thun, B.D. Carter, D. Feskanich, N.D. Freedman, R. Prentice, A.D. Lopez, P. Hartge, S.M. Gapstur, 50-year trends in smoking-related mortality in the united states, N. Engl. J. Med. 368 (4) (2013) 351–364.

[68] P.D. Turney, Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews, in: ACL '02: Proceedings of the 40th Annual Meating of the Association for Computational Linguistics, 2002, pp. 417–424.

[69] C.H. Wan, L.H. Lee, R. Rajkumar, D. Isa, A hybrid text classification approach with low dependency on parameter by integrating k-nearest neighbor and support vector machine, Expert Syst. Appl. 39 (15) (2012) 11880–11888.

[70] P. Wang, C. Domeniconi, Building semantic kernels for text classification using wikipedia, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 713–721.

[71] X. Wu, Z. Zheng, R. Srihari, Feature selection for text categorization on imbalanced data, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 80–89.

[72] L. Zhang, J. Zhu, T. Yao, An evaluation of statistical spam filtering techniques, ACM Trans. Asian Lang. Inf. Process. (TALIP) 3 (4) (2004) 243–269.

[73] Z. Zhen, X. Zeng, H. Wang, L. Han, A global evaluation criterion for feature selection in text categorization using kullback-leibler divergence, in: SoCPaR 11: International Conference of Soft Computing and Pattern Recognition, IEEE, 2011, pp. 440–445.