OCFS: Optimal Orthogonal Centroid Feature Selection for Text Categorization

Jun Yan[1], Ning Liu[2], Benyu Zhang[3], Shuicheng Yan[4],
Zheng Chen[3], Qiansheng Cheng[1], Weiguo Fan[5], Wei-Ying Ma[3]

[1]LMAM, Department of Information Science, School of Mathematical Science, Peking University, Beijing, 100871
yanjun@math.pku.edu.cn, qcheng@pku.edu.cn

[2]Department of Mathematics, Tsinghua University, Beijing, 100871, P. R. China
Liun01@mails.tsinghua.edu.cn

[3]Microsoft Research Asia, 49 Zhichun Road, Beijing, 100080, P. R. China
{byzhang, zhengc, wyma}@microsoft.com

[4]Department of Computer Science, Chinese University of Hong Kong, Hong Kong
scyan@ie.cuhk.edu.hk

[5]Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, USA
wfan@vt.edu

ABSTRACT

Text categorization is an important research area in many Information Retrieval (IR) applications. To save the storage space and computation time in text categorization, efficient and effective algorithms for reducing the data before analysis are highly desired. Traditional techniques for this purpose can generally be classified into *feature extraction* and *feature selection*. Because of efficiency, the latter is more suitable for text data such as web documents. However, many popular feature selection techniques such as Information Gain (IG) and χ^2 -test (CHI) are all greedy in nature and thus may not be optimal according to some criterion. Moreover, the performance of these greedy methods may be deteriorated when the reserved data dimension is extremely low. In this paper, we propose an efficient optimal feature selection algorithm by optimizing the objective function of Orthogonal Centroid (OC) subspace learning algorithm in a discrete solution space, called *Orthogonal Centroid Feature Selection* (OCFS). Experiments on 20 Newsgroups (20NG), Reuters Corpus Volume 1 (RCV1) and Open Directory Project (ODP) data show that OCFS is consistently better than IG and CHI with smaller computation time especially when the reduced dimension is extremely small.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—text processing.

General Terms

Algorithms, Performance.

Keywords

Feature Selection (FS), Feature Extraction (FE).

1. INTRODUCTION

Many information retrieval problems [1, 21] such as filtering, routing or searching for relevant information benefit from the text categorization research. For instance, building a news directory needs one to identify a modest number of training examples to train a classifier, and then classify the unknown data to populate the directory. However, with the explosive growth of the web data set, algorithms that can improve the classification efficiency while maintaining accuracy are highly desired [3, 23]. Dimension Reduction techniques have attracted much attention recently since effective dimension reduction make the learning task such as categorization more efficient and save more storage space [25]. Moreover, the lower dimension the reduced data is, the faster IR systems should be. The complexity of many learning algorithms [3] increase nonlinearly with increased data dimension. Thus efficient algorithms that can reduce the original data into a small dimensional space effectively are highly desired.

Dimension reduction techniques can generally be classified into Feature Extraction (FE) approaches [14] and Feature Selection (FS) approaches [13, 25]. The traditional FE algorithms reduce the dimension of data by linear algebra transformations (such as Principal Component Analysis (PCA) [10], Linear Discriminant Analysis (LDA) [17] and Maximum Margin Criterion (MMC) [6, 24], etc.) or nonlinear transformations (Locally Linear Embedding (LLE) [22], ISOMAP [8], etc.). On the other hand, FS algorithms reduce the dimension of data by select features from the original vectors directly. Though the FE algorithms have been proved to be very effective for dimension reduction, the high dimension of data sets in the text domain often fails many FE algorithms due to their high computational cost. Thus FS algorithms are more popular for real life text data dimension reduction problems.

In contrast to FE approaches, FS techniques aim to remove non-informative features according to corpus statistics. Many novel FS approaches, such as PCA based algorithm [16], Margin based algorithm [20], SVM-based algorithm [2], etc. were proposed in the past decades. In the text domain, the most popular used FS algorithms are still the traditional ones such as Information Gain (IG), χ^2 -test (CHI), Document Frequency (DF) and Mutual

SIGIR'05, August 15–19, 2005, Salvador, Brazil.
Copyright 2005 ACM 1-59593-034-5/05/0008...$5.00.

This work is done at Microsoft Research Asia.

Information (MI) [18, 25], etc. Information gain measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. Given a corpus of training text, we compute the information gain of each term, and then remove those features whose information gain was less than some pre-determined threshold. The computation of CHI, DF, and MI are similar to that of IG. The differences are the approaches to rank features. However, MI are not comparable with IG, DF, and CHI on text categorization [25]. We use two of them, IG and CHI, as our baselines in this paper.

Though IG and CHI are popular in text categorization, they are greedy in nature and thus their solutions are not optimal according to some criterion. In this paper, we propose a novel feature selection algorithm based on the Orthogonal Centroid algorithm [7, 15]. We call this algorithm as *Orthogonal Centroid Feature Selection* (OCFS). The main advantages of this algorithm are: 1), it is optimal according to the objective function implied by the original Orthogonal Centroid algorithm and thus it can get superior performance in sparse data sets of an extremely small dimension; 2), it is more efficient than the popular IG and CHI; 3), and it is very easy to be implemented with simple theoretical background. Experiments on 20 Newsgroups data, Reuters Corpus Volume 1 and ODP data show the efficiency and effectiveness of our proposed approach.

The Orthogonal Centroid (OC) algorithm is a traditional FE algorithm by QR matrix decomposition. It has been proved to be very effective on text data [15]. However, the main drawback of it is the high computational complexity of QR matrix decomposition. This makes the OC algorithm not suitable for real life large scale web document categorization problems. In this paper, we first define the objective function implied by the orthogonal centroid computation. Then we give our OCFS algorithm by optimize the objective function in a discrete solution space.

This rest of the paper is organized as follows. In section 2, we give the mathematical notations used in this paper and our problem definition. In section 3, we introduce some related work on text data feature selection. In section 4, we describe our proposed OCFS algorithm. In section 5, the experimental results on large scale data sets are given. Section 6 concludes our paper.

2. NOTATIONS AND PROBLEM DEFINININITION

In this paper, a corpus of documents are mathematically represented by a $d \times n$ term by document matrix $X \in R^{d \times n}$, which is generated by the traditional TFIDF indexing in Vector Space Model (VSM) [5], where *n* is the number of documents, and *d* is the number of features (terms). Each document is denoted by a column vector x_i, $i = 1,2,\cdots,n$, and the k^{th} entry of x_i is denoted by x_i^k, $k = 1,2,\cdots,d$. X^T is used to denote the transpose of matrix *X*. Assume that these feature vectors are belonging to *c* different classes and the class size of the j^{th} class is n_j. Using c_j to represent class *j*, $j = 1,2,\cdots,c$, the mean vector of the j^{th} class is $m_j = (1/n_j)\sum_{x_i \in c_j} x_i$. The mean vector of all these documents is $m = (1/n)\sum_{i=1}^{n} x_i = (1/n)\sum_{j=1}^{c} n_j m_j$.

The dimension reduction problem could be defined as the finding of a function $f : R^d \to R^p$, where *p* is the dimension of data after dimension reduction (*p*<<*d*), so that a document $x_i \in R^d$ is transformed into $y_i = f(x_i) \in R^p$. From the FE point of view, the dimension reduction problem aims to find an optimal transformation matrix $W \in R^{d \times p}$ according to some criterion such that $y_i = f(x_i) = W^T x_i \in R^p$, $i = 1,2,\cdots,n$ are the *p*-dimensional representation of original data. The solution space $R^{d \times p}$ is continuous and is consisted of all the real $d \times p$ matrices.

On the other hand, from FS point of view, the purpose of dimension reduction is to find a subset of features indexed by k_l, $l = 1,2,\cdots,p$ such that the low dimensional representation of original data x_i is denoted by $y_i = f(x_i) = (x_i^{k_1}, x_i^{k_2}, \cdots, x_i^{k_p})^T$. Notice that FS could be formulated under the same model with FE to make the selection optimal according to some criterion. In other words, the goal of the FS problem is to find an optimal transformation matrix $\tilde{W} \in R^{d \times p}$ according to some criterion subject to the constraint that $\tilde{W} = \{\tilde{w}_i^k\}$ is a binary matrix whose entries equal to zero or one and each column of $\tilde{W}$ has one and only one non-zero element. Then the low dimensional representation of original data is $y_i = f(x_i) = \tilde{W}^T x_i = (x_i^{k_1}, x_i^{k_2}, \cdots, x_i^{k_p})^T$. The solution space of the FS problem is discrete and consists of all matrices $\tilde{W} \in R^{d \times p}$ that satisfy the constraint given above. We define this space as $H^{d \times p}$.

Following the notations and discussions above, we define the optimal feature selection problem for text data categorization as:

> *given a set of labeled training documents X, learn a transformation matrix* $\tilde{W} \in H^{d \times p}$ *such that* $\tilde{W}$ *is optimal according to some criterion* $J(\tilde{W})$ *in space* $H^{d \times p}$.

Then we can transform the unlabeled *d*-dimensional data into a low *p*-dimensional space by applying $y_i = \tilde{W}^T x_i$ and classify these unlabeled data in the *p*-dimensional space.

3. RELATED WORK

An essential technique to improve the efficiency and effectiveness of the web document categorization problem is dimension reduction. Among them, feature selection approaches have attracted much attention due to their efficiency. In this paper, we involve two popular used feature selection algorithms, Information Gain (IG) and χ^2-test (CHI), which have been proved to be effective [18, 25] in the text domain as our baselines. We next give a brief introduction on IG and CHI in this section.

3.1 Information Gain

Following the notations above, information gain of a selected group of terms $T = (t^{k_1}, t^{k_2}, \cdots, t^{k_p})$ could be calculated by:

$$IG(T) = -\sum_{j=1}^{c} P_r(c_j)\log P_r(c_j) + P_r(T)\sum_{j=1}^{c} P_r(c_j \mid T)\log P_r(c_j \mid T)$$

$$+ P_r(\tilde{T})\sum_{j=1}^{c} P_r(c_j \mid \tilde{T})\log P_r(c_j \mid \tilde{T})$$

where *t* is used to denote a unique term, $IG(T)$ is the information gain of a term group, $P_r(c_j)$ is the probability of class c_j, $P_r(T)$ is the probability of term group T and $P_r(c_i | T)$ is the corresponding conditional probability. Following the problem definition in section 2, we define $J(\tilde{W}) = IG(T)$. In other words, IG aims to find an optimal $\tilde{W} \in H^{d \times p}$ so that each document is represented by p terms $T = (t^{k_1}, t^{k_2}, \cdots, t^{k_p})$ after the projection $y_i = \tilde{W}^T x_i$, then these p terms could maximize $J(\tilde{W}) = IG(T)$. However, in practice this is a NP problem and a greedy approach is typically used. Given a corpus of training text, we compute the information gain of each term by:

$$IG(t) = -\sum_{j=1}^{c} P_r(c_j) \log P_r(c_j) + P_r(t) \sum_{j=1}^{c} P_r(c_j | t) \log P_r(c_j | t)$$
$$+ P_r(\tilde{t}) \sum_{j=1}^{c} P_r(c_j | \tilde{t}) \log P_r(c_j | \tilde{t})$$

Then we remove those features whose information gain is less than some predetermined threshold. Obviously, the greedy IG is not optimal according to $J(\tilde{W}) = IG(T)$.

3.2 χ^2 -Test

CHI is also aiming at maximizing a criterion $J(\tilde{W})$. We ignore the details in this paper to save space. It is also a greedy algorithm to save the computation cost and thus is not optimal either. To a given term *t* and a category c_j, suppose *A* is the number of times *t* and c_j co-occur, *B* is the number of times the *t* occurs without c_j, *C* is the number of times c_j occurs without t, *D* is the number of times neither c_j nor *t* occurs. The χ^2 statistics is:

$$\chi^2(t, c_j) = \frac{n(AD - CB)^2}{(A+C)(B+D)(A+B)(C+D)} .$$

We can compute the χ^2 statistics between each unique term and each category in a training corpus, and then combine the category specific scores of each term into:

$$\chi^2(t) = \sum_{j=1}^{c} P_r(c_j) \chi^2(t, c_j) .$$

The computational complexity of IG and CHI are very similar. The main computation time of them are spent on the evaluation of conditional probability and the computation of $\chi^2(t, c_j)$ respectively with complexity *O(cd).*

4. ORTHOGONAL CENTROID FEATURE SELECTION

The Orthogonal Centroid Feature Selection (OCFS) selects features optimally according to the objective function implied by the Orthogonal Centroid algorithm, which is the foundation of our approach. Thus in this section we introduce the original OC algorithm firstly. After that, we transform the OC algorithm into an optimization problem by giving the objective function implied by the OC algorithm. After that, we optimize this objective function in the discrete solution space $H^{d \times p}$ and give our proposed OCFS algorithm. At the end of this section, we analyze the complexity and the choice of optimal dimension by OCFS.

4.1 Orthogonal Centroid Algorithm

Orthogonal Centroid (OC) algorithm is a recent proposed supervised feature extraction algorithm which utilizes orthogonal transformation on centroid [7, 15]. It has been proved very effective for classification problems on text data [7, 15] and is based on the Vector Space Computation in linear algebra by *QR* matrix decomposition [9]. The OC algorithm also aims to find the transformation matrix $W \in R^{d \times p}$ that maps each column $x_i \in R^d$ of $X \in R^{d \times n}$ to a vector $y_i \in R^p$. However, the time and space cost of *QR* decomposition can not meet the requirements of web documents since the scale of web documents are growing rapidly nowadays. To address this issue, we formulate the OC algorithm as a feature selection problem and find the solution in a corresponding discrete space. Theorem 1 introduces the objective function implied by OC. And then we propose the OCFS algorithm to optimize this objective function.

Theorem 1 The solution of Orthogonal Centroid algorithm equals to the solution of the following optimization problem,

$$\arg\max J(W) = \arg\max trace(W^T S_b W) \text{, subject to } W^T W = I .$$

where $S_b = \sum_{j=1}^{c} \frac{n_j}{n}(m_j - m)(m_j - m)^T$, which is partial objective function of LDA [17], is called inter-class scatter matrix.

The detailed proof of this theorem could be found in [7, 15]. This criterion defined by inter-class scatter matrix intuitively aims at making the data of different classes as far as possible in the transformed low dimensional space through the optimal projection matrix *W.* Based on the feature selection problem defined in section 2, we use the criterion $J(W)$ implied by the OC algorithm to derive our optimal feature selection algorithm by optimize $J(W)$ in $H^{d \times p}$ in the next subsection.

4.2 Optimization and Algorithm Summary

According to the discussion in Section 2, the feature selection problem according to criterion $J(\tilde{W})$ is an optimization problem:

$$\arg\max J(\tilde{W}) = \arg\max trace(\tilde{W}^T S_b \tilde{W}) \text{ subject to } \tilde{W} \in H^{d \times p} .$$

Suppose $K = \{k_i,\ 1 \le k_i \le d, i = 1, 2, \cdots, p\}$ is a group of indices of features. Since $\tilde{W}$ belongs to space $H^{d \times p}$, it must be a binary matrix with its elements of zero or one, and there are one and only one non-zero element in each column. Following this constraint, let $\tilde{W} = \{\tilde{w}_i^k\}$ and let:

$$\tilde{w}_i^k = \begin{cases} 1 & k = k_i \\ 0 & otherwise \end{cases}, \qquad (1)$$

Then,

$$trace(\tilde{W}^T S_b \tilde{W}) = \sum_{i=1}^{p} \tilde{w}_i^T S_b \tilde{w}_i = \sum_{i=1}^{p} \sum_{j=1}^{c} \frac{n_j}{n} (m_j^{k_i} - m^{k_i})^2 . \quad (2)$$

From (2), we can see that if a set of indices $K = \{k_i,\ 1 \le k_i \le d, i = 1, 2, \cdots, p\}$ can maximize $\sum_{i=1}^{p} \sum_{j=1}^{c} \frac{n_j}{n} (m_j^{k_i} - m^{k_i})^2$, the binary matrix $\tilde{W}$ generated by *K* following (1) should maximize, $J(\tilde{W}) = trace(\tilde{W}^T S_b \tilde{W})$. Then this index set K should be the optimal solution of the feature selection problem according to the criterion $J(\tilde{W})$ subject to $\tilde{W} \in H^{d \times p}$. The problem now is to find an index set K such that $\sum_{j=1}^{c} \sum_{i=1}^{p} \frac{n_j}{n} (m_j^{k_i} - m^{k_i})^2$ is maximized. It can be seen that this could be solved simply by finding the p largest ones from $\sum_{j=1}^{c} \frac{n_j}{n} (m_j^k - m^k)^2, k = 1, 2, \cdots, d$. This motivates us to propose an optimal feature selection algorithm according to $J(\tilde{W})$. The details of the OCFS algorithm are given in Table 1.

From table 1, the selected index set K can define a matrix $\tilde{W}$ by (1). This matrix is the solution of the optimization problem $J(\tilde{W}) = \arg\max trace(\tilde{W}^T S_b \tilde{W})$ in the space $\tilde{W} \in H^{d \times p}$.

Table 1. Orthogonal Centroid Feature Selection

Step 1, compute the centroid m_i *i=1,2,...,c* of each class for training data;

Step 2, compute the centroid m of all training samples;

Step 3, compute feature score $s(i) = \sum_{j=1}^{c} \frac{n_j}{n} (m_j^i - m^i)^2$ for all the features;

Step 4, find the corresponding index set K consisted of the p largest ones in set $S = \{s(i) \mid 1 \le i \le d\}$

4.3 An Illustrating Example

We demonstrate our algorithm with a simple example. The UCI machine learning dataset is a repository of databases, domain theories and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms[2]. We use the IRIS data set of UCI as our sample data to show how our algorithm works. The documentation of this data set is complete, and there are 3 classes, 4 numeric attributes and 150 samples. There are 50 samples in each class. Class 1 is linearly separable from the other two, but the other two are not linearly separable from each other. Without loss of generality and for intuition, we do not split the IRIS into training and testing data in this example. Suppose *P=2*, Following our proposed OCFS:
Step 1, computing the class mean of each class respectively;

$$m_1 = \frac{1}{n_1} \sum_{x_i \in class\ 1} x_i = (5.006, 3.418, 1.464, 0.244)$$

$$m_2 = \frac{1}{n_2} \sum_{x_i \in class\ 2} x_i = (5.936, 2.770, 4.260, 1.326)$$

$$m_3 = \frac{1}{n_3} \sum_{x_i \in class\ 3} x_i = (6.588, 2.974, 5.552, 2.026)$$

Step 2, computing the mean of all the 150 samples;

$$m = \frac{1}{n} \sum_{i=1}^{n} x_i = (5.8433, 3.054, 3.7587, 1.1987)$$

Step 3, computing the feature scores of all the features;

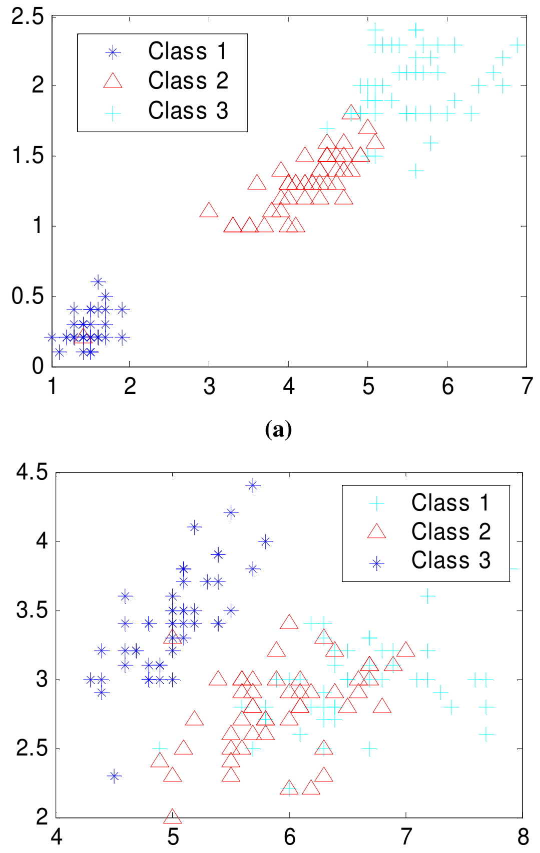
$$s(1) = \sum_{j=1}^{3} \frac{n_j}{n} (m_j^1 - m^1)^2 = 1.2642$$

$$s(2) = \sum_{j=1}^{3} \frac{n_j}{n} (m_j^2 - m^2)^2 = 0.21955$$

$$s(3) = \sum_{j=1}^{3} \frac{n_j}{n} (m_j^3 - m^3)^2 = 8.7329$$

$$s(4) = \sum_{j=1}^{3} \frac{n_j}{n} (m_j^4 - m^4)^2 = 1.1621$$

Step 4, selecting the features corresponding to the indices of the 2 largest ones among $S = \{s(i) \mid 1 \le i \le 4\}$. Then represent the original data with these 2 features. It is obvious that we should preserve the third and the first features here.
The OCFS aims at finding out a group of features from all the features of original data such that this group of features could maximize the $J(\tilde{W}) = trace(\tilde{W}^T S_b \tilde{W})$ in space $\tilde{W} \in H^{d \times p}$.



(a)

(b)

Figure 1. 2-dimensional visualization of IRIS: (a), features picked by OCFS; (b), the two features left-out by OCFS.

Intuitively, OCFS aims at finding out a subset of features that can make the sum of distance between all the class means maximized in the selected subspace. Step 1 is used to compute all the class means and then use the class means to represent different classes. Step 2 is used to calculate the mean of all the samples and then the sum of distance between all the class means can be computed

[2] http://www.ics.uci.edu/~mlearn/MLRepository.html

by computing the distance between each class mean and the mean of all samples. In step 3, the score of a feature is in fact the weighted sum of distance among all the class means along the direction of this feature. Step 4 is used to select the directions with maximum sum of distance. Our theoretical analysis above could prove that the features selected this way are optimal according to our proposed criterion. Figure 1 shows the 2-dimensional visualization of IRIS by select different 2 features.

The first picture is the IRIS in the 2-dimensional space whose coordinates are selected by OCFS. The second one is the 2-dimensional visualization of IRIS whose coordinates are the left-out two features of OCFS. It can be seen that in the subspace selected by OCFS, the three classes are easier to be separated than in the other subspace.

4.4 Algorithm Analysis

4.4.1 *Complexity and Robustness*

The main computation cost of OCFS is spent on the calculation of each feature score. The algorithm tells us that its time complexity is *O(cd)* which is the same as its counterparts: IG and CHI. However OCFS only need to compute the simple square function instead of some functional computation such as logarithm of IG. Thus though the time complexity are the same, OCFS should be much more efficient than IG and CHI. Experiments tell us that OCFS can process a dataset with about half time in contrast to IG and CHI. OCFS is also robust since OCFS focuses only on the mean of each class and all samples (see table 1). That means that a little amount of mislabeled data could not affect the final solution, i.e. the robustness is determined by the algorithm itself.

4.4.2 *The Number of Selected Features*

A question regarding OCFS is how to determine the optimal number of features to be selected. We use the energy function approach to solve this problem just like what the Principal Component Analysis [10] has done to select the subspace dimension. Without loss of generality, suppose the feature score of all the d features are: $s(k_1) \geq s(k_2) \geq \cdots \geq s(k_d)$, the energy function is defined as:

$$E(p) = \frac{\sum_{j=1}^{p} s(k_j)}{\sum_{i=1}^{d} s(i)}$$

Giving a threshold such as T=80%, i.e. the proportion of energy to be preserved after the feature selection procedure, we can get the optimal number of features $p*$ by:

$$p* = \arg\min E(p) \text{ subject to } E(p) \geq T \text{ .}$$

Note that the larger the threshold T is, the more features will be selected and vice visa. This paper is not focusing on the optimal subspace dimension. We only compare the performance of different approaches on a given extremely low dimensional space.

5. EXPERIMENTS

In this section, we conduct our experiments on three real large scale text data sets to show the performance of OCFS. We first describe the experiments setup, then give the experimental results, and finally discuss the results.

5.1 Experiments Setup

5.1.1 *Datasets*

To demonstrate the efficacy of OCFS, we performed experiments on three data sets: 20 Newsgroups [11], Reuters Corpus Volume 1 (RCV1) [12], and Open Directory Project (ODP)[3].

- 20 Newsgroups.

The 20 Newsgroups data consists of Usenet articles Lang collected from 20 different newsgroups. “Over a period of time 1000 articles were taken from each of the newsgroups, which make an overall number of 20000 documents in this collection. Except for a small fraction of the articles, each document belongs to exactly one newsgroup.”[4] In this paper, we select the five classes of computer science: (1), comp.graphics, (2), comp.os.mswindows.misc, (3), comp.sys.ibm.pc.hardware, (4), comp.sys.mac.hardware, and (5), comp.windows.x as our data set. 1000 samples for each class. The dimension of data is 131,072 by the TFIDF indexing. We use the "bydate" version of data whose training and testing data are split previously by the data provider. [1]

- Reuters Corpus Volume 1

Reuters Corpus Volume 1 (RCV1) data set which contains over 800,000 documents and the data dimension is about 500,000. We choose the data samples with the highest four topic codes (CCAT, ECAT, GCAT, and MCAT) in the “Topic Codes” hierarchy, which contains 789,670 documents. Then we split them into 5 equal-sized subsets, and each time 4 of them are used as the training set and the remaining ones are left as the test set. The experimental results reported in this paper are the average of the five runs. Moreover, we use this dataset as a single label problem, i.e. we only keep the first label if a sample is multi-labeled.

- Open Directory Project

Open Directory Project (ODP) consists of web documents crawled from the Internet. In this paper, we use the first layer ODP and only consider those documents in English and ignore all other non-English documents thus involve 13 classes: Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Science, Shopping, Society, Sports.

5.1.2 *Baseline Algorithms*

There are lots of feature selection algorithms for data preprocessing of classification problems. Among them, Information Gain (IG) and χ^2 -test (CHI) are dominant in the area of text categorization since they have been proved to be very effective and efficient. Moreover, they are two of the most widely used dimension reduction algorithms for real web document categorization problems. Thus in this paper, we select the IG and CHI as our baseline algorithms.

IG and CHI are the state-of-the-art text feature selection approaches. In this paper, we applied them on all training data to generate 10, 100, 1000 and 10000 dimensional spaces. The original IG and CHI selects a given number of features for each class and can not select a given number of features globally; To control the global number of features selected by IG and CHI, in

[3] http://rdf.dmoz.org/

[4] http://people.csail.mit.edu/u/j/jrennie/public_html/20Newsgroups

this paper we select the given number of features by compute their average score (weighted summation [4]) in different classes and select the largest ones to meet the given number.

5.1.3 Performance Measurement

Precision, Recall and F1 are the most widely used performance measurements for text categorization problems nowadays. Precision could be computed by the number of correctly categorized data over the number of all testing data. Recall could be computed by the number of correctly categorized data over the number of all the assigned data. F1 is a common measure in text categorization that combines recall and precision. In this paper, we use Micro F1 measure as our effectiveness measurement which combines recall and precision into a single score according to the following formula:

$$\text{Micro } F1 = \frac{2P \times R}{P+R},$$

where P is the Precision and R is the Recall. In the figures of this section, we use F1 to denote Micro F1. The efficiency is evaluated by the real CPU runtime. We ignore the I/O time and record only the time of the feature selection procedure.

5.1.4 Key Steps of Experiments

We apply the OCFS on all the training data to select 10, 100, 1000 and 10000 features to compare the effectiveness with baselines. In all our experiments, we use a single computer with Pentium(R) 4 2.80GHz CPU, and 2GB of RAM, to conduct the experiments. The experiment consists of the following steps:

- Apply the feature selection algorithm on a specific size of the training data to select a group of features with determined numbers;
- Transform all the training data into the low dimensional space;
- Train the SVM classifier by SMO [19] (linear kernel is used and the parameters used are all defaulted ones);
- Transform all the testing data to the selected low dimensional space;
- Evaluate the classification performance, using Micro F1, on the transformed testing data;
- Rerun this procedure on different training and testing data and record the average Micro F1 of all the algorithms involved.

5.2 Experimental Results

All the experimental results are shown in this section. Besides the Micro F1 and CPU runtime, we also give the number of overlap among the features selected by different feature selection approaches. Moreover some selected terms by IG, CHI and OCFS are given for intuition.

5.2.1 20NG

The classification performance on 20NG data is summarized in Figure 2. The x-axis is the number of selected features and the y-axis is the Micro F1. From this figure, we can infer that the OCFS is constantly better than its counterpart selected by IG and CHI. In other words, OCFS algorithm can achieve better performance for text categorization than the widely used traditional algorithms.

The time spent by each algorithm in feature selection for classification is reported in Figure 3. We can see that the feature selection time of OCFS spent is much less than the others. And then we can draw the conclusion that, OCFS has better performance with the traditional feature selection algorithms especially in extremely low dimension space and it is more efficient. For instance the improvements by Micro F1 are about 0.1 and 0.05 respectively in contrast to CHI and IG.

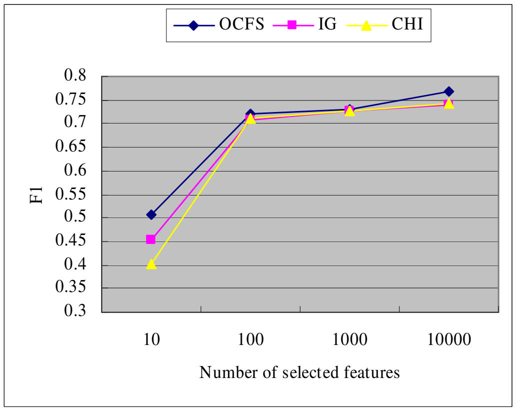


Figure 2. Micro F1 of classification on 20 Newsgroups data dimension reduced by OCFS, IG and CHI

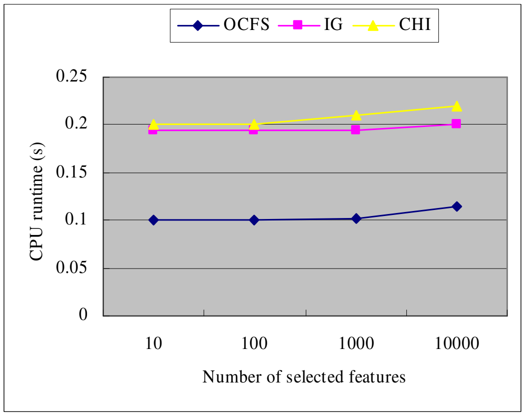


Figure 3. CPU runtime on 20 Newsgroups data dimension reduced by OCFS, IG and CHI

Besides the Micro F1 and CPU runtime, we also give the number of overlap among the features selected by different feature selection approaches in Table 2. Moreover the selected different terms by IG, CHI and OCFS are given in Table 3 while their overlaps are ignored. Each line of table 2 shows the number of overlap features selected by two algorithms in all dimensions involved. It can be seen that most features selected by IG and CHI are the same. On the other hand, about half of the features selected by OCFS are different with its counterparts of IG and CHI. It is very interesting that they have comparable performance while the features used are very different in 100, 1000 and 10000 dimensional spaces. The OCFS is outstanding in 10 dimensional case. There are four features (terms) selected by OCFS, IG and CHI are different in the 10 dimensional space, we list their corresponding terms in table 3 to feed the reader for intuition.

Table 2. Number of feature overlap in different dimension of space

	10d	100d	1000d	10000d
OCFS vs IG	7	62	532	6494
OCFS vs CHI	6	55	509	6530
IG vs CHI	9	85	831	9096

Table 3. Un-overlapped features selected by OCFS, IG and CHI in 10 dimension space

OCFS	drive	file	edu	card
IG	drive	3d1	x11r5	motif
CHI	Lcs	3d1	x11r5	motif

5.2.2 RCV1

The classification performance on RCV1 data is summarized in Figure 4. From this figure, we can infer that the low dimensional space selected by OCFS have better performance than its counterpart selected by the popular used IG and CHI. The efficiency is showed in Figure 5. These indicate that in practice on web scale data, the performance of OCFS is outstanding. Though the efficiency improvement is only about half, to a real large scale problem, save half time is significant improvement.

5.2.3 ODP

The performance of different feature selection algorithms on ODP data are reported in Figure 6. To save space, we do not report the time used since the time of OCFS is still about half of its counterpart of IG and CHI. Since the ODP is very large scale data, too low dimension such as 10 could make most of its sample to be zero vectors no matter which feature selection approach used by our experiments. Thus we ignore 10 on ODP. To show the results in low dimensional space, we add the performance in 200 and 500 dimensional space in these experiment.

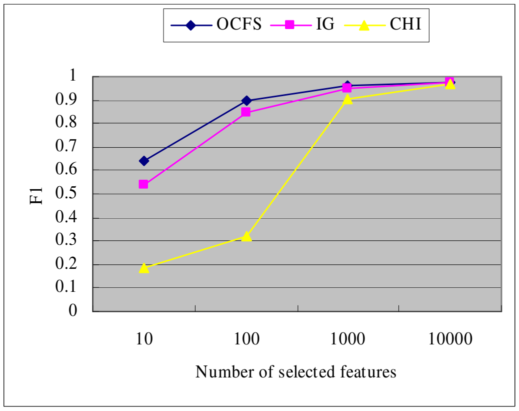


Figure 4. Micro F1 of classification on RCV1 data dimension reduced by OCFS, IG and CHI

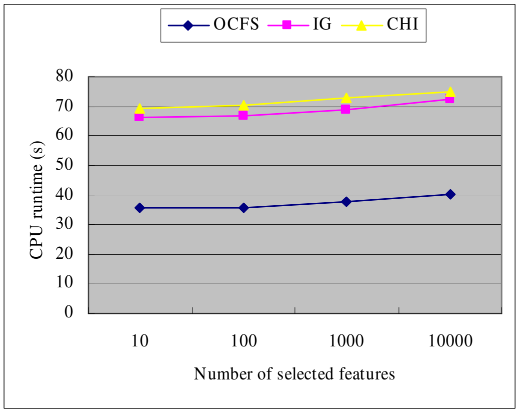


Figure 5. CPU runtime on RCV1, dimension reduced by OCFS, IG and CHI

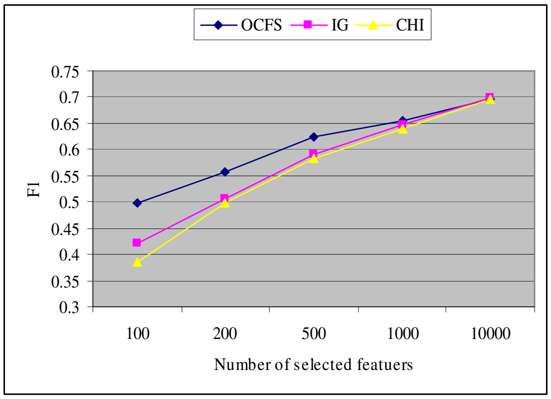


Figure 6. Micro F1 of classification on ODP data dimension reduced by OCFS, IG and CHI

5.3 Discussion of Results

From the experiments we can see that the proposed OCFS is consistently better than IG and CHI especially when the reduced dimension is extremely small for text categorization problems. On the other hand, it is more efficient than the others by using only about half of the time used by baselines to select good features. To very large scale data such as the rapid growth web data, saving about half of the computation time is valuable and exciting. From the dimension by Micro F1 figures (Figure 2, Figure 4, Figure 6) we can draw the conclusion that OCFS can get significant improvements than baselines when the selected subspace dimension is extremely small while get a little better performance when the selected subspace dimension is relative large. This phenomenon occurs due to the reason that when the selected feature dimension is small, the proposed OCFS, which is an optimal feature selection approach, can outperform the greedy ones. With the increasing number of selected features, the saturation of features makes additional features of less value. Then when the number of selected features is large enough, all feature selection algorithms involved can achieve comparable performance no matter they are optimal or greedy. From the tables we can see that our proposed optimal OCFS selects many different features from those by IG and CHI and these different features improved the performance of text categorization.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel efficient and effective feature selection algorithm, Orthogonal Centroid Feature Selection (OCFS), for text categorization. With the growing number of text documents on the Web, many traditional text categorization techniques fail to produce a satisfactory result in handling this scale of data due to their time complexity and storage requirements. The OCFS can help save both data storage space and computation time by feature selection. The OCFS is designed by optimizing the objective function of the effective Orthogonal Centroid algorithm. The main advantages of OCFS are: 1) it is optimal according to the objective function of the Orthogonal Centroid algorithm and thus it can get better performance when the features are not saturated in a extremely small dimensional space; 2) it is more efficient than the popular IG and CHI methods when handling the data at the web scale and has better performance; 3) it is easy to compute.

In the future, we plan to extend our work to unbalanced data through revising the objective function. The other is to use our proposed approach as the first step of feature selection to generate an optimal group of candidate features, and then use other effective techniques such as IG to select features from the candidates to improve the performance.

7. REFERENCES

[1] Belkin, N.J. and Croft, W.B. Retrieval Techniques. *Annual Review of Information Science and Technology*, *220*. 109-145.

[2] Douglas Hardin, Ioannis Tsamardinos and Aliferis, C.F., A theoretical characterization of linear SVM-based feature selection. In *Proceedings of the Twenty-first International Conference on Machine Learning(ICML 04)*, (Banff, Alberta, Canada, 2004).

[3] Duda, R.O., Hart, P.E. and Stork, D.G. *Pattern Classification (2nd ed.)*. WILEY, 2000.

[4] Franca D., Fabrizio S., Supervised term Weighting for Automated Text Categorization. *In proceedings of the 2003 ACM symposium on Applied Computing,* (Florida), 784-788.

[5] Greengrass, E. *Information Retrieval: A Survey*, 30 November 2000.

[6] Haifeng Li, Tao Jiang and Zhang., K., Efficient and Robust Feature Extraction by Maximum Margin Criterion. In *Proceedings of the Advances in Neural Information Processing Systems*, (Vancouver, Canada, 2003), 97 - 104.

[7] Howland, P. and Park, H. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26 (8)*. 995- 1006.

[8] J.B. Tenenbaum, Silva, V.d. and Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*. 2319--2323.

[9] James E. Gentle, J. Chambers, W. Eddy, W. Haerdle, S. Sheather and Tierney, L. *Numerical Linear Algebra for Applications in Statistics.* Springer-Verlag, Berlin, 1998.

[10] Jolliffe, I.T. *Principal Component Analysis*. New York: Spriger Verlag, 1986.

[11] Lang, K. and NewsWeeder, Learning to Filter Netnews. In *Proceedings of the the 12th International Conference on Machine Learning(ICML),*, (1995), 331-339.

[12] Lewis, D., Yang, Y., Rose, T. and Li, F. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research.*

[13] Lewis, D.D., Feature Selection and Feature Extraction for Text Categorization. In *Proceedings of the Speech and Natural Language Workshop*, (1992).

[14] Liu, H. and Motoda, H. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic, Norwell, MA, USA, 1998.

[15] M. Jeon, Park, H. and Rosen, J.B. Dimension Reduction Based on Centroids and Least Squares for Efficient Processing of Text Data, Minneapolis, MN, University of Minnesota.

[16] Malhi, A. and Gao, R.X. PCA-Based Feature Selection Scheme for Machine Defect Classification. *IEEE Transactions on Instrumentation and Measurement*, *53*. 1517-1525.

[17] Martinez, A.M. and Kak, A.C. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*. 228 - 233.

[18] Mitchell, T.M. *Machine Learning*. McGraw Hill, 1997.

[19] Platt, J. Fast training of support vector machines using sequential minimal optimization. In *In Advances in Kernel Methods: support vector learning*, MIT Press, Cambridge, 1999, 185-208.

[20] Ran Gilad-Bachrach, Amir Navot and Tishby, N., Margin based feature selection - theory and algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 04)*, (Banff, Alberta, Canada, 2004).

[21] Ricardo Baeza-Yates and Ribeiro-Neto, B. *Modern Information Retrival*. Addison-Wesley, 1999.

[22] Roweis, S.T. and Saul., L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, *290*. 2323--2326.

[23] Wang, G., Lochovsky, F.H. and Yang, Q., Feature Selection with Conditional Mutual Information MaxMin in Text Categorization. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, (Washington D.C., 2004), 342-349.

[24] Yan, J., Zhang, B., Yan, S., Chen, Z., Fan, W., Xi, W., Yang, Q., Ma, W.-Y. and Cheng, Q., IMMC: Incremental Maximum Margin Criterion. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, (Seattle, WA, USA, 2004), ACM, 725 - 730.

[25] Yang, Y. and Pedersen, J.O., A comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, (1997), 412-420.