academicJournals

Vol. 8(37), pp. 1818-1828, 4 October, 2013
DOI 10.5897/SRE10.1214
ISSN 1992-2248 © 2013 Academic Journals
http://www.academicjournals.org/SRE

Scientific Research and Essays

Full Length Research Paper

A hybrid approach for text categorization by using x^2 statistic, principal component analysis and particle swarm optimization

Harun Uğuz

Department of Computer Engineering, Selçuk University, Konya, Turkey.

Accepted 6 July, 2012

Today, the number of text documents in digital form is progressively increasing and text categorization becomes the key technology of dealing with organizing text data. A major problem of text categorization is a huge-scale number of features. Most of those are useless, irrelevant or redundant for text categorization. Therefore, these features can decrease the classification performance. In order to eliminate this deficiency, feature selection is often used in text categorization for the purpose of reducing the dimensionality of the feature space and improving the performance of text categorization. In this study, in order to improve the performance of text categorization, a hybrid approach is suggested based on x^2 statistic, particle swarm optimization (PSO) and principal component analysis (PCA). In this context, initially, each term within the document is ranked depending on their importance for the classification using x^2 statistic method and, particle swarm optimization (PSO) and principal component analysis (PCA) feature selection and feature extraction methods are applied separately on the terms of which importance are ranked in decreasing order and dimension reduction is carried out. In this way, during the text categorization, less importance terms are ignored, feature selection and feature extraction methods are applied on the highest importance terms, and cost of computational time and complexity to be occurred in the course of the application are reduced. To evaluate the effectiveness of purposed model, experiments were conducted using K-nearest neighbor (KNN) and C4.5 decision tree algorithm on Reuters-21578 and Classic3 datasets collection for text categorization. The experimental evaluation showed that the proposed model was effective for text categorization.

Key words: Text categorization, feature selection, particle swarm optimization, principal component analysis, x^2 statistic.

INTRODUCTION

Text categorization is widely used for organizing the documents in the digital form. Due to the increasing number of documents in digital form, the automated text categorization has become more promising in the last ten years.

Text categorization is defined as assigning new documents to a set of pre-defined categories based on the classification patterns (Sebastiani, 1999). One of the major problem of text categorization is the high dimensionality of the feature space due to a large number of terms. This problem may cause the computational complexity of machine learning methods used for text categorization to increase, and may bring about inefficiency and low accuracy results due to redundant or irrelevant terms in the feature space (Zifeng et al., 2007; Li et al., 2008). For the solution of this problem, feature

E-mail: harun_uguz@hotmail.com Tel: +90 332 223 19 26. Fax: +90 332 241 06 35.

extraction and feature selection techniques can be used.

Feature extraction is a process that extracts a set of new features from the original features into a distinct feature space (Wyse et al., 1980). Some feature extraction methods have been successfully used in text categorization such as principal component analysis (PCA) (Selamat and Omatu, 2004; Lam and Lee, 1999), latent semantic indexing (Sun et al., 2004), clustering methods (Slonim and Tishby, 2000), etc. Among too many methods which are used for feature extraction, PCA has attracted a lot of attention. PCA (Jolliffe, 1986) is a statistical technique for dimensionality reduction which aims at minimizing the loss in variance in the original data. It can be viewed as a domain independent technique for feature extraction, which is applicable to a wide variety of data (Selamat and Omatu, 2004).

Feature selection is a process that select a subset from the original feature set according to some criteria of importance of features (Liu et al., 2005). Feature selection is to remove redundant and irrelevant features from the feature space, and the selected feature set should contain sufficient and reliable information about the original feature set (Forman, 2003). Consequently, feature selection should both reduce the high dimensionality of the feature space, and also provide a better understanding of the features, in order to improve the classification result (Li et al., 2009).

Although, there are too many methods for feature selection, particle swarm optimization (PSO) has a lot of attention because of its easy implementation, its simplicity in coding and its ability to solve efficiently (Marinakis, 2009). PSO (Kennedy and Eberhart, 1995) was originally developed to solve real-value optimization problems.

However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and levels of variables (Chuang, 2011). Therefore, Kennedy and Eberhart (1997) presented binary version of particle swarm optimization (PSO) algorithm for discrete combinatorial optimization problem. İnary particle swarm optimization (BPSO) approach has recently been applied successfully to solving feature selection problems (Chuang et al., 2008, 2011; Zhou et al., 2006).

Feature ranking is a kind of feature selection process which ranks the features based on their relevancies and importance with respect to the problem (Hong et al., 2008). Therefore, feature ranking can be viewed as a kind of flexible feature selection approach.

In the literature, there are many feature ranking methods such as information gain (IG), x^2 statistic, reliefF etc. Among these methods, x^2 statistic is one of the most effective measure among the measures, which is based on the experiments reported so far (Zheng et al., 2003; Yang and Pedersen, 1997).

In this study, in order to reduce high dimensionality of feature space composing of a large number of terms, and remove redundant or irrelevant features from feature space, a hybrid approach was suggested for text categorization. According to suggested approach, initially, each term in the text is ranked depending on their importance for the classification in decreasing order using x^2 statistic method. So, terms of high importance take place in the first ranks and terms of less importance take place in the following ranks and BPSO method selected for feature selection and PCA method selected for feature extraction are applied separately on the terms of thehighest importance in accordance with x^2 statistic methods and dimension reduction is carried out. In this way, during the text categorization, terms of less importance are ignored, feature selection and feature extraction methods are applied on the terms of highest importance, and cost of computational time and complexity to be occurred in the course of the application are reduced.

To evaluate the effectiveness of dimension reduction methods, experiments were conducted on Reuters-21578 and Classic3 datasets collection via C4.5 decision tree and KNN classifiers for text categorization. The experimental results showed that the proposed model was able to achieve high categorization effectiveness as measured by precision, recall and F-measure.

The remainder of this paper is organized as follows. Subsequently, the study presents a brief overview of materials and methods. The effectiveness of the pur-posed method and experimental results for categorization of text document is demonstrated thereafter, and finally, the paper is concluded.

MATERIALS AND METHODS

Figure 1 shows the the parts of proposed text catogarization structure. These parts are explained in the following subsections:

Datasets

We used two datasets, which have been widely used by the researchers in the information retrieval area. They are called the Reuters-21578 dataset and Classic3 dataset.

Reuters-21578 dataset

There are some publicly datasets that can be used as test collections for text categorization. The most widely used is the Reuters collection (Lewis, 1997); is a set of economic news published by Reuters in 1987. This collection includes 21,578 documents that are organized in 135 categories. In this experiment, 6 most frequently categories including minimum 500 terms are selected. There are 8158 documents belonging to chosen categories. The distributions of the number of documents in 6 most frequently categories are shown in Table 1. According to Table 1, the distribution of documents into the categories is unbalanced. Maximum and minimum categories occupy 45.88 and 6.13% of dataset respectively.

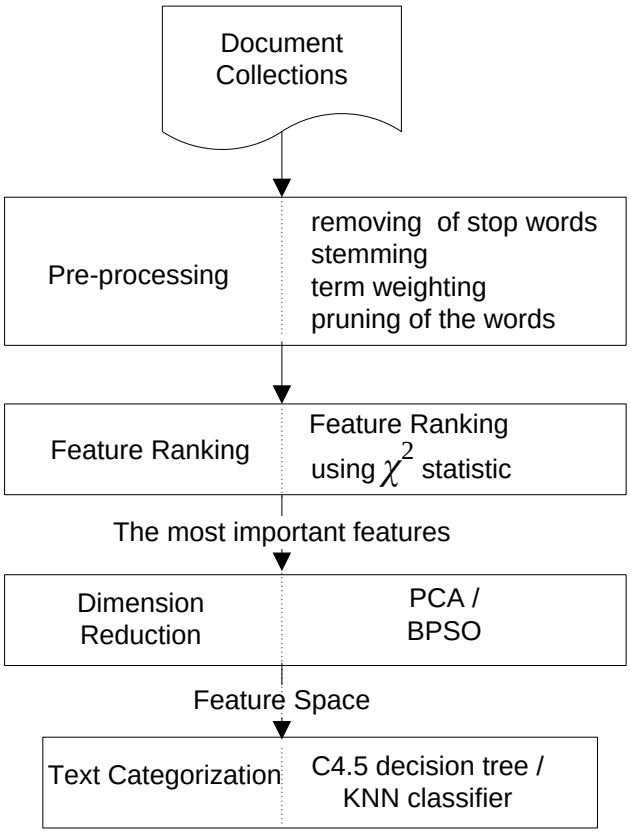

Figure 1. Purposed text categorization structure.

Table 1. Distributions of the 6 most frequently categories for Reuters-21578 dataset.

Category name	Number of document
Earn	3743
Acquisition	2179
Money-fx	633
Crude	561
Grain	542
Trade	500

Classic3 dataset

We implemented the second experiments on the Classic3 dataset, a document collection from the SMART project at Cornell University (ftp.cs.cornell.edu/pub/smart). Classic3 dataset is frequently used to evaluate performance of text categorization algorithms because it contains a known number of fairly well-separated groups. It contains three categories which are 1398 CRANFIELD documents from aeronautical system papers, 1033 MEDLINE documents from medical papers, and 1460 CISI documents from information retrieval papers. The distribution of documents into the categories is balanced since all the categories are represented equally well in the dataset.

Pre-processing

The stages of pre-processing is applied in fallowing four steps:

Removing of stop-words

Words such as a conjunction, pronoun in a text document which does not concern the concept are called as stop-words. This process involves removing the most frequent word that exists in a text document such as 'a', 'an', 'the' etc... Removing these words will save spaces and increase classification performance because stop-words exist nearly in all of the text documents. In the study,

stop words were removed in accordance with the existing stop word list (www.unine.ch/Info/clef/) with 571 words.

Stemming

Stemming is a process of extracting the root form of the word. Thereby, terms of same root which seem as a different word due to the affixes can be determined. For example, the words ‘‘computer,” ‘‘computing,” ‘‘computation,” and ‘‘computes” have similar meaning with the “comput” root. Porter’s stemming algorithm (Porter, 1980) is used for stemming.

Term weighting

After we obtain a set of terms in a document, it is necessary to represent them numerically for text categorization. Term weighting is applied to set a level of contribution of a term to a document (Lertnattee and Theeramunkong, 2004). Thereby, each document can be written in a vector form depending on the terms they contain. This document vector will generally be in the following format:

$$d = \{w_1,..., w_i,..., w_{|T|}\} \quad (1)$$

where w_i, is the weight of the term with number i in the d document, T is the term set, and $|T|$ is the cardinality of T.

To obtain the term vector of T, the *tfidf* is generally used as its weight scheme. Accordingly, let the term frequency tf_i be the number of occurrence of t_i in document and let the document frequency df_i be the number of the document in which t_i term is seen at least once. Inverse document frequency idf_i is calculated as shown in the Equation 2 using df_i (Salton and Buckley, 1988).

$$idf_i = \log\left(\frac{|D|}{df_i}\right) \quad (2)$$

where $|D|$ is the number of all documents in training set and w_i is calculated in accordance with Equation 3.

$$w_i = tf_i.idf_i \quad (3)$$

Pruning of the words

The pruning process basically filters less frequent features in a document collection. Term vector acquired following the term weighting is very high-dimensional and sparse. Also, it is seen that a number of element of term vector is “0”. Therefore, we use pruning in order to reach a smaller but more discriminative feature set. To this end, prune the words which appear less than two times in the documents.

Feature ranking with x^2 statistic

x^2 statistic is one of the popular approaches employed as a term importance criterion in the text document data (Zheng et al., 2003; Yang and Pedersen, 1997). In text categorization, x^2 statistic is often used to measure the degree of dependency between a term and a specific category. x^2 statistic of term t in category c is defined as the following equation (Yang and Pedersen, 1997):

$$\chi^2(t,c) = \frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \quad (4)$$

where A is the number of times t and c co-occur, B is the number of times t occurs without c, C is the number of times c occurs without t, D is the number of times neither c nor t occurs, and N is the total number of documents.

For each term t, x^2 statistic was computed for every category. The maximum score was taken as the x^2 statistic for term t as fallow:

$$\chi^2(t,c) = \max_{i=1}^{m}\{\chi^2(t,c_i)\} \quad (5)$$

where m denotes the number of categories.

In this study, before the dimension reduction, each term within the text is ranked depending on their importance for the classification in decreasing order using x^2 statistic method. Thereby, in the process of text categorization, terms of less importance are ignored; dimension reduction methods are applied on the terms of highest importance.

Dimension reduction methods

At the end of the pre-processing step, terms of highest importance in documents are acquired through x^2 statistic method. In this manner, even though the number of terms in document reduces, the main problem fort the text categorization is the high dimensionality of the feature space. Therefore, so as to reduce the feature space dimension and computational complexity of machine learning algorithms to be used in the text categorization, increase the performances thereof, PCA and BPSO dimension reduction methods are applied. The aim of these methods is to minimize information loss while maximizing reduction in dimensionality.

Principal component analysis (PCA)

PCA is a statistical technique, being used for extracting information from multi-variety dataset. This process is performed via having principal components of original variables with linear combinations identified. While the original dataset with the maximum variability is represented with first principal component, the dataset from the remaining with the maximum variability is represented with second principal component. The process goes on consecutively as such, with the dataset from the remaining with the maximum variability being represented with the next principal component. Therefore, PCA is a technique, being used for producing the lower-dimensional version of the original dataset (Zhang, 2007). Details of PCA can be reached from Jolliffe (1986).

The most significant stage in the application of PCA is the determination of the number of principal component. The p number of principal components to be chosen among the all principal components should be the principal components to represent the data at their very best. In this study, cumulative percentage of variance criteria has been applied in determining the number of principal components, for its simplicity, and eligible performance (Valle et al., 1999).

According to this criterion, principal components, with their cumulative percentage of variance is higher than a prescribed threshold value, are being chosen. Although a sensible threshold is

very often in the range 70 to 90%, it can sometimes be higher or lower depending on the practical details of a particular dataset. However, it should be noticed that some authors point out that there is no ideal solution to the problem of dimensionality in a PCA (Jolliffe, 1986). Therefore, the choice of threshold is often selected heuristically (Warne et al., 2004). In this study, threshold value was specified as 75% in all application performed via PCA on both datasets.

Binary particle swarm optimization

Particle swarm optimization (PSO) is an optimization technique based on the idea of swarm intelligence in biological populations. PSO is firstly developed by Kennedy and Eberhart (1995). In PSO algorithm, each individual as a particle represents a potential solution in the search space. Each particle adjusts its position according to fitness value evaluated by the fitness function to be optimized. To discover the optimal solution, each particle is updated according to the Equation 6 and 7 (Kennedy and Eberhart, 1995) by following two parameters called *pbest* and *gbest* at each iteration. The value *pbest* is a local fitness value, while the value *gbest* constitutes a global fitness value.

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 rand_1(pbest_{i,j} - x_{i,j}(t)) + c_2 rand_2(gbest_{i,j} - x_{i,j}(t)) \quad (6)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t) \quad (7)$$

where *w* is the inertia weight, *i* is the index of particle, *j* is the index of position in particle. Velocitys $v_{i,j}(t+1)$ and $v_{i,j}(t)$ are those of the updated and before being updated particles, respectively. $rand_1$ and $rand_2$ are the random numbers interval [0,1]. c_1 and c_2 are the acceleration numbers. $x_{i,j}(t+1)$ and $x_{i,j}(t)$ are those of the updated particle position and before being updated particle position, respectively.

PSO algorithm is originally developed to solve real-value optimization problems. However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and levels of variables (Chuang et al., 2011). Therefore, Kennedy and Eberhart (1997) presented binary version of PSO algorithm for discrete combinatorial optimization problem. In the BPSO algorithm, the position of every particle is limited to a range of [0 to 1]. The sigmoid function is applied to normalize all real valued velocities to the range [0 to 1], as shown in Equation 8.

$$S(v_{i,j}(t+1)) = \frac{1}{1+e^{-v_{i,j}(t+1)}} \quad (8)$$

In BPSO, positions of particles are updated using Equation 9.

$$x_{i,j}(t+1) = \begin{cases} 1 & if\ rand >= S(v_{i,j}(t+1)), \\ 0 & if\ rand\ < S(v_{i,j}(t+1)), \end{cases} \quad (9)$$

where *rand* is a random number between [0 to 1].

In process of BPSO, for the reduction of feature space dimension selected via x^2 statistic method, each particle was coded in the form of binary vector in a manner to compose of cells as the number of feature in each feature space. If the value of the cell, which is coded in binary system is “1”, it means that the corresponding feature is selected, or contrary, if the value of cell is “0”, it means that the corresponding feature is not selected.

In order to cut down cost by reducing the computing time, the population size is set to 20 particles, and iteration count (termination criterion) is fixed at 300. In our experiments, the acceleration parameters are set to c_1=c_2 = 2, and inertia weight is set to 0.48 as in Chuang (2011). It is also important to mention that each optimization process is replicated 30 times owing to the use of stochastic search algorithms.

After the initialization of BPSO parameters, each particle is evaluated by the fitness function to be optimized. In this study, the fitness value of each particle was calculated using the average value of F-measure (Equation 15) which was obtained C4.5 or KNN classifier for text categorization.

Text categorization methods

In this study, C4.5 decision tree and KNN classifier methods are used for text categorization. Brief descriptions of these methods are discussed subsequently.

KNN classifier

The KNN (Cover and Hart, 1967) algorithm is a well-known instance-based approach that has been widely applied to text categorization due to its simplicity and accuracy (Yang, 1997; Lam and Han, 2003).

To categorize an unknown document, the KNN classifier ranks the document’s neighbors among the training documents, and use the class labels of *k* most similarity neighbors. Similarity between two documents may be measured by the Euclidean distance, cosine measure, and etc. The similarity score of each nearest neighbor document to the test document is used as the weight of the classes of the neighbor document. If a specific category is shared by more than one of the *K*-nearest neighbors, then the sum of the similarity scores of those neighbors is obtained from the weight of that particular shared category (Mitra et al., 2007). Detailed procedure of KNN can be referred to (Cover and Hart, 1967).

At the phase with classification by means of KNN, the most important parameter affecting classification is *k* nearest neighbor number. Usually, the optimal value of *k* is empirically determined. In our study, *k* value was determined so that it would give the least classification error (*k*=3 was determined). In addition, at the phase of finding the *k* nearest neighborhood, Euclidean distance was used as distance metric.

C4.5 decision tree classifier

Decision tree is a well-known machine learning approach to automatic induction of classification trees based on training data (Quinlan, 1986). In a typical decision tree training algorithm, there are usually two phases. The first phase is tree growing where a tree is built by greedily splitting each tree node. Since the tree can overfit the training data, a second phase overfitted branches of the tree are removed (Damerau et al., 2004). C4.5 is univariate decision tree algorithm. At each node, only one attribute of instances are used for decision making. Details of C4.5 can be reached from (Fuhr and Buckley, 1991).

In our application by using C4.5 decision tree algorithms, in the pruning phase, the post-pruning method was used to decide when to stop expanding a decision tree. The confidence factor is used for

pruning the tree. In our study, the confidence factor assigned is 0.25. The pruned tree consisted of 4 leaves and 8 nodes.

Evaluation of the performance

The F-measure, precision and recall is usually employed to evaluate the accuracy of the text categorization results. These measures were used to evaluate the accuracy of the result of the KNN and C4.5 classifiers for text categorization. The F-measure is a harmonic combination of the precision and recall values used in information retrieval (Rijsbergen, 1979). Precision is the proportion of the correctly proposed documents to the proposed documents, while recall is the proportion of the correctly proposed documents to the test data that have to be proposed (Li et al., 2009).

In this study, F-measure, precision and recall was not separately calculated for each category; average values of such measures were used. Precision P_i and recall R_i of category i are defined in Equation (10) and (11), respectively.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{10}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{11}$$

where TP_i, FP_i and FN_i represent the number of true positives, false positives, and false negatives, respectively. Then, the average precision (P) and recall (R) measures are calculated as Equation (12) and (13), respectively.

$$P = \frac{\sum_{i=1}^{N} d_i . P_i}{\sum_{i=1}^{N} d_i} \tag{12}$$

$$R = \frac{\sum_{i=1}^{N} d_i . R_i}{\sum_{i=1}^{N} d_i} \tag{13}$$

where d_i is the number of documents category i contains. N is the number of categories.

F-measure F_i of category i is defined in Equation (14).

$$F_i = \frac{2 . P_i . R_i}{P_i + R_i} \tag{14}$$

Then, the average F-measure (F) is calculated as Equation (15).

$$F = \frac{\sum_{i=1}^{N} d_i . F_i}{\sum_{i=1}^{N} d_i} \tag{15}$$

where d_i is the number of documents category i contains. N is the number of categories.

RESULTS AND DİSCUSSİON

Experiments were conducted for text categorization on two different datasets to examine the performance of purposed method, dimension reduction and classifier techniques. Pre-processing, dimension reduction and classification processes were implemented by the Matlab software package. *10 fold* cross validation procedure was preferred classification stages. All experiments have been run on a machine with 2.8 GHz CPU, 4 GB of RAM, 500 GB HDD space, and Windows 7 operation system.

Results on Reuters-21578 dataset

Pre-processing

Pre-processing process was performed in 4 stages. The first step consists of removing the stop words, since they are useless for the classification. In the study, stop words were removed in accordance with the existing stop word list with 571 words (www.unine.ch/Info/clef/). After removing stopwords, the dataset contained 10764 unique words. The second step, the Porter algorithm (Porter, 1980) was used for stemming. The third step, the document vectors were built with *tfidf* weighting scheme. The fourth step, in order to reduce the size of the term set, we discarded terms which appear in less than 2 documents and the total number of terms extracted finally was 7542. Thereby, a document-term matrix was acquired in the dimension of 8158 × 7542 at the end of pre-processing.

Feature ranking, dimension reduction and text categorization with C4.5 and KNN classifiers on Reuters-21578 dataset

In this study, C4.5 decision tree and KNN classifier methods which are frequently used for text categorization due to simplicity and accuracy were used. These methods were separately applied in the classification of datasets of which dimension acquired at the end of the BPSO and PCA application was reduced. The reason why classifier is used is to compare the performances of the both methods in the text categorization.

Initially, KNN and C4.5 decision tree classifiers were applied on whole of the document-term feature space (without dimension reduction) with the dimension of 8158 × 7542 acquired at the end of the pre-processing for the purpose of testing efficiency of dimension reduction stage in text categorization. The experimental results with KNN and C4.5 decision tree clasifier are summarized in Table 2. The results in terms of precision, recall and F-measure are the averaged values calculated across all *10-fold* cross validation experiments. As seen in Table 2, in applications made without using any dimension reduction

Table 2. The performance (average value of precision, recall and F-measure) of KNN and C45. Decision tree classifier on Reuters-21578 dataset.

Classifier	Number of features	Precision (%)	Recall (%)	F-measure (%)
KNN	7542	73.36	95.59	83.02
C4.5	7542	84.64	89.23	86.88

Table 3. The performance (average value of precision, recall and F-measure) of KNN and C4.5 Decision tree classifier with x^2 statistic on Reuters-21578 dataset.

Percentage of feature	KNN				C4.5 decision tree			
	Number of Features	Precision (%)	Recall (%)	F-measure (%)	Number of features	Precision (%)	Recall (%)	F-measure (%)
1	75	96.09	95.89	95.99	75	94.76	94.60	94.68
2	151	97.07	96.47	96.77	151	95.74	94.90	95.32
3	226	96.63	96.66	96.65	226	96.02	95.48	95.75
4	302	97.24	97.06	97.15	302	95.66	95.30	95.48
5	377	95.86	97.73	96.79	377	95.73	95.32	95.53
6	453	95.83	97.62	96.72	453	95.67	95.59	95.63
7	528	95.26	97.68	96.45	528	95.84	95.43	95.64
8	603	93.92	97.81	95.83	603	95.87	95.40	95.63
9	679	93.10	97.65	95.32	679	95.28	95.46	95.37
10	754	92.20	97.57	94.81	754	95.87	95.46	95.66

method, highest accuracy is obtained when C4.5 classifier are used.

After that, feature ranking was applied via x^2 statistic method in order to reduce high dimension of the feature space. In this phase, the effects of individual feature ranking operation by x^2 statistic method, on classifier performance were examined. Accordingly, features were ranked in decreasing order (in means of importance) for classification by feature ranking performed by x^2 statistic. 1 to 10% of features ranked by x^2 statistic were separately classified from C4.5 and KNN classifiers. Table 3 shows the classification performances at the end of feature ranking operation performed by x^2 statistic.

According to Table 3, highest accuracy with KNN classifier is obtained when 4% of the ranked features are used. In addition, highest accuracy with C4.5 classifier is obtained when 3% of the ranked features are used. When the classifier performances compared, the KNN algorithm shows higher performance than C4.5 decision tree algorithm. If Table 3 is compared with Table 2, we can see that highest accuracies are obtained at the end of feature ranking operations made by x^2 statistics. Furthermore, it is seen that using ranking features (1 to 10%) via x^2 statistics in stead of all features contributed to the classifier performances in an affirmative manner.

Finally, the effects of x^2 statistic–BPSO and x^2 statistic–PCA based hybrid methods on classifier performances were examined. Accordingly, dimension reduction process was applied separately by BPSO and PCA to the 1 to 10% of features ranked according to importance for classification by x^2 statistic method.

Table 4 shows the classification performances at the end of feature ranking and feature selection operation performed by hybrid x^2 statistic–BPSO method. According to Table 4, highest accuracy is obtained when 8 and 7% of the ranked features for KNN and C4.5 classifier are used, respectively. When analyzing Tables 3 and 4, although fewer features are selected via hybrid x^2 statistic–BPSO method, precision, recall and F-measure values are higher only in comparison to feature selection carried out via x^2 statistic method. Moreover, when Tables 2 to 4 is examined, it can be observed that highest accuracy with least number of features is obtained by the proposed hybrid x^2 statistic–BPSO method.

Table 5 shows the classification performances at the end of feature ranking and feature extraction operation performed by hybrid x^2 statistic–PCA method. According to Table 5, highest accuracy is obtained when 7% of the ranked features for KNN and C4.5 classifier are used. Similar to hybrid x^2 statistic–BPSO method, although fewer features are selected via hybrid x^2 statistic–PCA method, precision, recall and F-measure values are higher only in comparison to feature selection carried out via x^2 statistic method. When Tables 2 to 5 is examined, it can be observed that hybrid x^2 statistic–BPSO shows

Table 4. The performance (average value of precision, recall and F-measure) of KNN and C4.5 Decision tree classifier with hybrid x^2 statistic–BPSO method on Reuters-21578 dataset.

Percentage of feature	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	F-measure (%)	Number of features	Precision (%)	Recall (%)	F-measure (%)
1	40	95.26	94.52	94.89	43	94.96	94.92	94.94
2	76	97.38	97.88	97.63	72	96.80	96.98	96.89
3	115	97.47	97.85	97.66	113	96.02	96.24	96.13
4	158	96.85	97.99	97.42	155	96.31	96.04	96.18
5	185	97.72	97.35	97.54	187	96.90	96.26	96.58
6	234	97.74	97.44	97.59	236	97.03	96.42	96.73
7	276	97.95	97.80	97.88	276	96.74	97.12	96.93
8	312	97.90	98.01	97.96	311	96.81	95.97	96.39
9	345	97.78	97.34	97.56	345	96.54	96.12	96.33
10	372	97.52	97.42	97.47	370	96.17	96.82	96.50

Table 5. The performance (average value of precision, recall and F-measure) of KNN and C4.5 decision tree classifier with hybrid x^2 statistic-PCA method on Reuters-21578 dataset.

Percentage of feature	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	F-measure (%)	Number of features	Precision (%)	Recall (%)	F-measure (%)
1	36	94.09	93.99	94.04	36	95.31	94.39	94.85
2	71	96.97	97.36	97.16	71	96.07	96.10	96.09
3	103	96.77	97.57	97.17	103	95.86	95.78	95.82
4	134	96.62	97.86	97.24	134	96.01	95.83	95.92
5	162	97.45	97.03	97.24	162	96.22	95.86	96.04
6	193	97.16	96.95	97.06	193	96.30	95.91	96.10
7	222	97.25	97.30	97.28	222	96.04	96.53	96.28
8	250	97.20	97.25	97.22	250	96.11	95.65	95.88
9	278	97.32	96.87	97.09	278	95.84	95.51	95.68
10	303	97.22	97.09	97.15	303	95.87	96.07	95.97

higher classifier accuracy in comparison to x^2 statistic and hybrid x^2 statistic–PCA method.

As understood from these results, when there are many irrelevant or redundant features in the feature space, performing a feature ranking, feature extraction and feature selection method could remove them. Therefore classifier performance improves. In addition, using x^2 statistic, PCA and BPSO methods as hybrid, improves the classification efficiency and accuracy compared with individual usage of x^2 statistic method.

In classifiers perfomances with dimension reduction methods, the C4.5 decision tree algorithm seems to perform worse than KNN algorithm. However, one of the advantages of the C4.5 decision tree algorithm, is its potential for data exploration purposes. Consequently, it is seen that higher classifier performance is acquired withfewer features through two stage dimension reduction.

Results on Classic3 dataset

Pre-Processing

Similarly, to the application carried out on the Reuters-21578, stop words were removed in accordance with the existing stop word list with 571 words. After removing stopwords, the dataset contained 11398 unique words. The Porter algorithm (Porter, 1980) was used for stemming. Then, the document vectors were built with *tfidf* weighting scheme. In order to reduce the size of the term set, we discarded terms which appear in less than 2 documents and the total number of terms extracted finally is 6679. Thereby, a document-term matrix was acquired

Table 6. The performance (average value of precision, recall and F-measure) of KNN and C45. Decision tree classifier on Classic3 dataset.

Classifier	Number of features	Precision (%)	Recall (%)	F-measure (%)
KNN	6679	60.22	98.99	74.89
C4.5	6679	85.12	89.20	85.19

Table 7. The performance (average value of precision, recall and F-measure) of KNN and C4.5 Decision tree classifier with x^2 statistic on Classic3 dataset.

Percentage of feature	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	F-measure (%)	Number of features	Precision (%)	Recall (%)	F-measure (%)
1	67	90.31	90.71	90.51	67	92.32	90.40	91.35
2	134	88.59	92.23	90.37	134	90.77	91.28	91.02
3	200	89.32	93.56	91.39	200	92.01	92.42	92.22
4	267	88.20	93.05	90.56	267	91.07	92.80	91.93
5	334	88.56	93.37	90.90	334	91.80	92.61	92.20
6	401	88.23	90.46	89.33	401	90.79	92.74	91.75
7	468	88.36	89.70	89.03	468	89.26	91.91	90.57
8	534	84.78	93.62	88.98	534	89.03	92.29	90.63
9	601	81.30	97.47	88.65	601	88.92	92.23	90.54
10	668	81.00	97.79	88.61	668	88.95	92.04	90.47

in the dimension of 3891 × 6679 at the end of pre-processing.

Feature ranking, dimension reduction and text categorization with C4.5 and KNN classifiers on Classic3 dataset

Similarly, to the application carried out on the Reuters-21578 dataset, initially, KNN and C4.5 decision tree classifiers are applied on whole of the document-term feature space. The experimental results with KNN and C4.5 decision tree clasifier are summarized in Table 6. As shown in Table 6, in applications made without using any dimension reduction method, highest accuracy is obtained when C4.5 classifier are used. After that, feature ranking and dimension reduction techniques were applied as individual and hybrid in order to reduce high dimension of the feature space. Success of x^2 statistic, hybrid PCA and BPSO methods in text categorization was separately tested by using KNN and C4.5 decision tree classifier. Table 7 shows the classification performances at the end of feature ranking operation performed by x^2 statistic. As seen in Table 7, highest accuracy with KNN classifier is obtained when 3% of the ranked features are used. Similarly, highest accuracy with C4.5 classifier is obtained when 3% of the ranked features are used. When the classifier performances compared, the C4.5 decision tree algorithm seems to perform better than KNN algorithm. When analyzing Tables 6 and 7, it is seen that using ranking features (1 to 10%) via x^2 statistics instead of all features contributed to the classifier performances in a positive manner.

Table 8 shows the classification performances at the end of feature ranking and feature selection operation performed by hybrid x^2 statistic–BPSO method. As seen in Table 8, highest accuracy is obtained when 6 and 10% of the ranked features for KNN and C4.5 classifier are used, respectively. As it is evident from Tables 6 to 8, it can be observed that highest accuracy with least number of features is obtained by the proposed hybrid x^2 statistic–BPSO method. In other words, using x^2 statistic and BPSO methods as hybrid, improves the classification efficiency and accuracy compared with individual usage of x^2 statistic method.

Table 9 shows the classification performances at the end of feature ranking and feature extraction operation performed by hybrid x^2 statistic–PCA method. According to Table 9, when proposed x^2 statistic–PCA method is used, higher classification accuracy with least number of features is obtained. These results show that using the x^2 statistic and PCA methods as hybrid improves the classification efficiency and accuracy compared with individual usage of x^2 statistic method. When Tables 6 to 9 is analyzed, it can be observed that hybrid x^2 statistic–BPSO method shows higher classifier accuracy in

Table 8. The performance (average value of precision, recall and F-measure) of KNN and C4.5 Decision tree classifier with hybrid x^2 statistic–BPSO method on Classic3 dataset.

Percentage of feature	KNN				C4.5 Decision Tree			
	Number of features	Precision (%)	Recall (%)	F-measure (%)	Number of features	Precision (%)	Recall (%)	F-measure (%)
1	45	92.48	92.36	92.42	45	92.85	91.97	92.41
2	86	92.66	96.59	94.63	84	94.71	96.10	95.41
3	121	94.23	96.42	95.33	123	93.91	95.11	94.51
4	168	94.76	97.33	96.05	168	96.75	95.63	96.19
5	183	94.33	97.65	95.99	180	96.34	96.98	96.66
6	226	95.28	97.67	96.48	228	96.02	96.97	96.50
7	279	95.15	97.73	96.44	279	96.88	97.21	97.05
8	307	94.15	98.21	96.18	302	95.77	96.91	96.34
9	332	94.01	98.77	96.39	332	96.65	97.43	97.04
10	356	93.81	98.30	96.06	349	96.76	97.53	97.15

Table 9. The performance (average value of precision, recall and F-measure) of KNN and C4.5 Decision tree classifier with hybrid x^2 statistic-PCA method on Classic3 dataset.

Percentage of feature	KNN				C4.5 Decision Tree			
	Number of features	Precision (%)	Recall (%)	F-measure (%)	Number of features	Precision (%)	Recall (%)	F-measure (%)
1	39	91.97	91.85	91.91	39	92.05	91.41	91.73
2	75	91.74	95.39	93.53	75	93.35	94.06	93.71
3	110	93.08	96.08	94.56	110	93.15	94.44	93.79
4	144	94.15	96.59	95.35	144	95.43	94.88	95.15
5	177	94.13	97.28	95.68	177	95.15	96.72	95.93
6	208	94.65	97.22	95.92	208	95.27	96.72	95.99
7	238	94.58	97.09	95.82	238	95.64	96.90	96.27
8	266	93.77	97.98	95.83	266	95.40	96.84	96.11
9	293	93.59	98.61	96.03	293	96.31	97.28	96.79
10	319	93.67	98.10	95.83	319	96.50	97.41	96.95

comparison to x^2 statistic and hybrid x^2 statistic–PCA method.

In classifiers' performances, the C4.5 decision tree algorithm show higher performance than KNN algorithm. Consequently, it is seen that higher classifier performance is acquired with fewer features through hybrid methods.

Conclusion

In this study, a hybrid approach is suggested based on x^2 statistic, BPSO and PCA in order to reduce high dimensionality of feature space composing of a large number of terms, and improve the performance of text categorization. In this context, initially, each term within the document is ranked depending on their importance for the classification using x^2 statistic method. Thus, less importance features are ignored and the highest importance features are selected and then, BPSO and PCA feature selection and feature extraction methods are applied separately on the terms of which importance are ranked in decreasing order and dimension reduction is carried out.

To evaluate the effectiveness of dimension reduction methods on purposed model, experiments are conducted using KNN and C4.5 decision tree algorithm on Reuters-21578 and Classic3 datasets collection for text categorization. As a result of experimental studies, it is seen that using features in reduced via dimension reduction techniques instead of all features contributed to the classifier performances in a positive manner. When there many irrelevant or redundant features in the feature space, performing a feature ranking, feature extraction

and feature selection method could remove them.

Therefore, classifier performance be improve. Also, it is revealed that success of text categorization performed through C4.5 decision tree and KNN algorithms using fewer features selected only via x^2 statistic–BPSO and x^2 statistic–PCA method is higher than the success acquired using features selected via x^2 statistic method. These results show that using the x^2 statistic, BPSO and PCA methods as hybrid improves the classification efficiency and accuracy compared with individual usage of x^2 statistic method. Consequently, the experimental evaluation shows that the proposed model is effective for text categorization.

ACKNOWLEDGEMENT

This study has been supported by Scientific Research Project of Selcuk University.

REFERENCES

Cover TM, Hart PE (1967). Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. 13(1):21-27.

Chuang LY, Chang HW, Tu CJ, Yang CH (2008). Improved binary PSO for feature selection using gene expression data. Comp. Biol. Chem. 32:29-38.

Chuang LY, Yang CH, Li JC (2011). Chaotic maps based on binary particle swarm optimization for feature selection. Appl. Soft Comp. 11:239-248.

Damerau FJ, Zhang T, Weiss SM (2004). Nitin Indurkhya Text categorization for a comprehensive time-dependent benchmark. Inf. Proc. Management. 40:209-221.

Forman G (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. The J. Mach. Learn. Res. 3:1289-1305.

Fuhr N, Buckley C (1991). A probabilistic learning approach for document indexing. ACM Trans. on Info. Sys. 9(3):223-248.

Hong Y, Kwong S, Chang Y, Ren Q (2008). Consensus unsupervised feature ranking from multiple views. Patt. Rec. Letters. 29: 595-602.

Jolliffe T (1986). Principal Component Analysis, ACM Computing Surveys. Springer-Verlag, pp.1-47.

Kennedy J, Eberhart RC (1995). Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, 4:1942-1948.

Kennedy J, Eberhart RC (1997). A discrete binary version of the particle swarm algorithm. Int. Conf. Syst. Man Cybernet. 5:4104-4108.

Lam SLY, Lee DL (1999). Feature reduction for neural network based text categorization. In Sixth international conference on database systems for advanced applications (DASFAA'99), 195.

Lam W, Han Y (2003). Automatic textual document categorization based on generalized instance sets and a metamodel. In Proceeding of the IEEE transactions on pattern analysis and machine intelligence, 25(5):628-633.

Lertnattee V, Theeramunkong T (2004). Effect of term distributions on centroid-based text categorization. Info. Sci. J. 158:89-115.

Lewis DD (1997). Reuters-21578 text categorization test collection, distribution 1.0., http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

Li Y, Hsu DF, Chung SM (2009). Combining Multiple Feature Selection Methods for Text Categorization by Using Rank-Score Characteristics. 21st IEEE International Conference on Tools with Artificial Intelligence, pp. 508-517.

Liu L, Kang J, Yu J, Wang Z (2005). A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering. Proceeding ofNLP-KE'O5, pp. 597-601.

Marinakis Y, Marinaki M, Doumpos M, Zopounidis C (2009). Ant colony and particle swarm optimization for financial classification problems. Expet. Syst. Appl. 36:10604-10611.

Mitra V, Wang CJ, Banerjee S (2007). Text classification: A least square support vector machine approach. Appl. Soft Comp. 7:908-914.

Porter MF (1980). An Algorithm for Suffix Stripping, Program (Auto. Lib. Info. Syst.) 14(3):130-137.

Rijsbergen CJV (1979). Information Retrieval, 2nd edition, Buttersworth, London.

Salton G, Buckley C (1988). Term-weighting approaches in automatic text retrieval. Inf. Proc. Management 24(5):513-523.

Sebastiani F (1999). A Tutorial on Automated Text Categorisation, in Proceedings of the ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, 7-35.

Selamat A, Omatu S (2004). Web page feature selection and classification using neural Networks. Inf. Sci. J. 158:69-88.

Slonim N, Tishby N (2000). Document Clustering using, Word Clusters via the Information Bottleneck Method. Proc. SJGIR'00, 208-215.

Sun JT, Chen Z, Zeng HJ, Lu Y, Shi CY, Ma WY (2004). Supervised latent semantic indexing for document categorization. In ICDM. IEEE Pres. Pp. 535-538.

Quinlan JR (1986). Induction of decision trees, J. Mach. Learn. 1(1):81-106.

Valle S, Li W, Qin SJ (1999). Selection of the number of principal components: the variance of the reconstruction error criterion with a comparison to other methods. Ind. Eng. Chem. Res. 38: 4389-4401.

Warne K, Prasad G, Rezvani S, Maguire L (2004). Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion. Eng. Appl. Artif. Intell. 17:871-885.

Wyse N, Dubes R, Jain AK (1980). A Critical Evaluation of Intrinsic Dimensionality Algorithms. Pattern Recognition in Practice, 415-425.

Yang Y (1997). An Evaluation of Statistical Approaches to Text Categorization, IJIRR. 1(1):76-88.

Yang Y, Pedersen JO (1997). A comparative study on feature selection in text categorization. In Proceedings of the 14th international conference on machine learning, pp. 412-420.

Zhang YX (2007). Artificial neural networks based on principal component analysis input selection for clinical pattern recognition analysis. Talanta, 73:68-75.

Zheng Z, Srihari R, Srihari S (2003). A Feature Selection Framework for Text Filtering. Proceedings of the third IEEE international conference on Data Mining, pp. 705-708.

Zhou W, Zhou C, Liu G, Zhu H (2006). Feature selection for microarray data analysis using mutual information and rough set theory, Boston: Springer.

Zifeng C, Baowen X, Weifeng Z, Dawei J, Junling X (2007). CLDA: Feature Selection for Text Categorization Based on Constrained LDA, International Conference on Semantic Computing, pp. 702-712.