Knowledge-Based Systems 36 (2012) 226–235

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

A novel probabilistic feature selection method for text classification

Alper Kursat Uysal *, Serkan Gunal

Department of Computer Engineering, Anadolu University, Eskisehir, Turkiye

ARTICLE INFO

Article history:
Received 28 December 2011
Received in revised form 14 June 2012
Accepted 14 June 2012
Available online 9 July 2012

Keywords:
Feature selection
Filter
Pattern recognition
Text classification
Dimension reduction

ABSTRACT

High dimensionality of the feature space is one of the most important concerns in text classification problems due to processing time and accuracy considerations. Selection of distinctive features is therefore essential for text classification. This study proposes a novel filter based probabilistic feature selection method, namely distinguishing feature selector (DFS), for text classification. The proposed method is compared with well-known filter approaches including chi square, information gain, Gini index and deviation from Poisson distribution. The comparison is carried out for different datasets, classification algorithms, and success measures. Experimental results explicitly indicate that DFS offers a competitive performance with respect to the abovementioned approaches in terms of classification accuracy, dimension reduction rate and processing time.

1. Introduction

With rapid advance of internet technologies, the amount of electronic documents has drastically increased worldwide. As a consequence, text classification, which is also known as text categorization, has gained importance in hierarchical organization of these documents. The fundamental goal of the text classification is to classify texts of interest into appropriate classes [14]. Typical text classification framework consists of a feature extraction mechanism that extracts numerical information from raw text documents, and a classifier that carries out the classification process using a prior knowledge of labeled data. Text classification has been successfully deployed to various domains such as topic detection [5], spam e-mail filtering [16,19], author identification [9,11], and web page classification [2,7,33].

Majority of text classification studies utilizes bag-of-words technique [21] to represent a document such that the order of terms within the document is ignored but frequencies of the terms are considered. Each distinct term in a document collection therefore constitutes an individual feature. Hence, a document is represented by a multi-dimensional feature vector where each dimension corresponds to a weighted value (i.e., TF-IDF [28]) of the regarding term within the document collection. Since the features are originated from distinct terms, even moderate numbers of documents in a text collection would result in hundreds or even thousands of features. One of the most important issues in text classification is therefore dealing with high dimensionality of the feature space. Excessive numbers of features not only increase computational time but also degrade classification accuracy. As a consequence, feature selection plays a critical role in text classification problems to speed up the computation as well as to improve the accuracy.

Feature selection techniques broadly fall into three categories: filters, wrappers, and embedded methods. Filters assess feature relevancies using various scoring frameworks that are independent from a learning model, or classifier, and select top-N features attaining the highest scores [18]. Filter techniques are computationally fast; however, they usually do not take feature dependencies into consideration. On the other hand, wrappers evaluate features using a specific learning model and search algorithm [17,24]. Wrapper techniques consider feature dependencies, provide interaction between feature subset search and choice of the learning model, but are computationally expensive with respect to the filters. Embedded methods integrate feature selection into classifier training phase; therefore, these methods are specific to the utilized learning model just like the wrappers. Nevertheless, they are computationally less intensive than the wrappers [18,35].

In text classification studies, though there are some hybrid approaches combining the filters and wrappers [14,38], commonly preferred feature selection methods are the filters thanks to their relatively low processing time. Term strength [41], odds ratio [31], document frequency [42], mutual information [27], chi-square [8], information gain [26], improved Gini index [36], measure of deviation from Poisson distribution [32], a support vector machine based feature selection algorithm [39], ambiguity measure [29], class discriminating measure [6] and binomial hypothesis testing

* Corresponding author.
E-mail addresses: akuysal@anadolu.edu.tr (A.K. Uysal), serkangunal@anadolu.edu.tr (S. Gunal).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2012.06.005

[40] are just some examples to the filter methods. Combinations of the features, which are selected by different filter methods, are also considered, and their contributions to the classification accuracy under varying conditions are investigated in [14].

In spite of numerous approaches in the literature, feature selection is still an ongoing research topic. Researchers are still looking for new techniques to select distinctive features so that the classification accuracy can be improved and the processing time can be reduced as well. For that purpose, this paper proposes a novel filter based probabilistic feature selection method, namely distinguishing feature selector (DFS), particularly for text classification. DFS selects distinctive features while eliminating uninformative ones considering certain requirements on term characteristics. DFS is compared with successful filter approaches including chi square, information gain, Gini index and deviation from Poisson distribution. The comparison was carried out for different classification algorithms, datasets, and success measures with distinct characteristics so that effectiveness of DFS can be observed under different conditions. Results of the experimental analysis revealed that DFS offers a competitive performance with respect to the above-mentioned approaches in terms of classification accuracy, dimension reduction rate, and processing time.

Rest of the paper is organized as follows: feature selection approaches that are compared with DFS are briefly described in Section 2. Section 3 introduces DFS method. Section 4 explains the classifiers used in the experiments. Section 5 presents the experimental study and results, which are related to similarity, accuracy, dimension reduction rate, and timing analysis, for each dataset, classifier, and success measure. Finally, some concluding remarks are given in Section 6.

2. Existing feature selection methods

As it is pointed out in previous section, there is a mass amount of filter based techniques for the selection of distinctive features in text classification. Among all those techniques, chi square, information gain, Gini index, and deviation from Poisson distribution have been proven to be much more effective [32,36,42]. Therefore, efficacy of DFS was assessed against these four successful approaches. Mathematical backgrounds of these approaches are provided in the following subsections.

2.1. Chi-square (CHI2)

One of the most popular feature selection approaches is CHI2. In statistics, the CHI2 test is used to examine independence of two events. The events, X and Y, are assumed to be independent if

$$p(XY) = p(X)p(Y). \tag{1}$$

In text feature selection, these two events correspond to occurrence of particular term and class, respectively. CHI2 information can be computed using

$$\mathrm{CHI2}(t,C) = \sum_{t\in\{0,1\}}\sum_{C\in\{0,1\}} \frac{(N_{t,C} - E_{t,C})^2}{E_{t,C}}, \tag{2}$$

where N is the observed frequency and E is the expected frequency for each state of term t and class C [28]. CHI2 is a measure of how much expected counts E and observed counts N deviate from each other. A high value of CHI2 indicates that the hypothesis of independence is not correct. If the two events are dependent, then the occurrence of the term makes the occurrence of the class more likely. Consequently, the regarding term is relevant as a feature. CHI2 score of a term is calculated for individual classes. This score can be globalized over all classes in two ways. The first way is to compute the weighted average score for all classes while the second way is to choose the maximum score among all classes. In this paper, the former approach is preferred to globalize CHI2 value for all classes as in

$$\mathrm{CHI2}(t) = \sum_{i=1}^{M} P(C_i) \cdot \mathrm{CHI2}(t,C_i), \tag{3}$$

where $P(C_i)$ is the class probability and $\mathrm{CHI2}(t,C_i)$ is the class specific CHI2 score of term t.

2.2. Information Gain (IG)

IG measures how much information the presence or absence of a term contributes to make the correct classification decision on any class [13]. IG reaches its maximum value if a term is an ideal indicator for class association, that is, if the term is present in a document if and only if the document belongs to the respective class. IG for term t can be obtained using

$$\begin{aligned}\mathrm{IG}(t) = &-\sum_{i=1}^{M} P(C_i)\log P(C_i) + P(t)\sum_{i=1}^{M} P(C_i|t)\log P(C_i|t) \\ &+ P(\bar{t})\sum_{i=1}^{M} P(C_i|\bar{t})\log P(C_i|\bar{t}),\end{aligned} \tag{4}$$

where M is the number of classes, $P(C_i)$ is the probability of class C_i, $P(t)$ and $P(\bar{t})$ are the probabilities of presence and absence of term t, $P(C_i|t)$ and $P(C_i|\bar{t})$ are the conditional probabilities of class C_i given presence and absence of term t, respectively.

2.3. Gini Index (GI)

GI is another feature selection method which is an improved version of the method originally used to find the best split of attributes in decision trees [36]. It has simpler computation than the other methods in general [32]. Its formulation is given as

$$\mathrm{GI}(t) = \sum_{i=1}^{M} P(t|C_i)^2 P(C_i|t)^2, \tag{5}$$

where $P(t|C_i)$ is the probability of term t given presence of class C_i, $P(C_i|t)$ is the probability of class C_i given presence of term t, respectively.

2.4. Deviation from Poisson distribution (DP)

DP is derived from Poisson distribution which is also applied to information retrieval for selecting effective query words and this metric is adapted to feature selection problem to construct a new metric [32]. The degree of deviation from the Poisson distribution is used as a measure of effectiveness. If a feature fits into Poisson distribution, the result of this metric would be smaller and this indicates that the feature is independent from the given class. Conversely, the feature would be more discriminative if the result of the metric is greater. This method can be formulated as

$$\begin{aligned}&\mathrm{DP}(t,C) = \frac{(a-\hat{a})^2}{\hat{a}} + \frac{(b-\hat{b})^2}{\hat{b}} + \frac{(c-\hat{c})^2}{\hat{c}} + \frac{(d-\hat{d})^2}{\hat{d}} \\ &\hat{a} = n(C)\{1-\exp(-\lambda)\} \\ &\hat{b} = n(C)\exp(-\lambda) \\ &\hat{c} = n(\bar{C})\{1-\exp(-\lambda)\} \\ &\hat{d} = n(\bar{C})\exp(-\lambda) \\ &\lambda = \frac{F}{N},\end{aligned} \tag{6}$$

where F is the total frequency of term t in all documents, N is the number of documents in the training set, $n(C)$ and $n(\bar{C})$ are the

numbers of documents belonging to class C and not belonging to class C, λ is the expected frequency of the term t in a document, respectively. The quantities a and b represents the number of documents containing and not containing term t in documents of class C. While the quantity c represents the number of documents containing term t and not belonging to class C, the quantity d represents the number of documents with absence of term t and class C at the same time. Furthermore, the quantities $\hat{a}, \hat{b}, \hat{c}, \hat{d}$ are predicted values for a, b, c, d respectively. In order to globalize class specific scores over the entire collection, the weighted average scoring [32] is used as given below.

$$\mathrm{DP}(t) = \sum_{i=1}^{M} P(C_i) \cdot \mathrm{DP}(t, C_i). \tag{7}$$

3. Distinguishing feature selector

An ideal filter based feature selection method should assign high scores to distinctive features while assigning lower scores to irrelevant ones. In case of text classification, each distinct term corresponds to a feature. Then, ranking of terms should be carried out considering the following requirements:

1. A term, which frequently occurs in a single class and does not occur in the other classes, is distinctive; therefore, it must be assigned a high score.
2. A term, which rarely occurs in a single class and does not occur in the other classes, is irrelevant; therefore, it must be assigned a low score.
3. A term, which frequently occurs in all classes, is irrelevant; therefore, it must be assigned a low score.
4. A term, which occurs in some of the classes, is relatively distinctive; therefore, it must be assigned a relatively high score.

Based on the first and second requirements, an initial scoring framework is constituted as

$$\sum_{i=1}^{M} \frac{P(C_i|t)}{P(\bar{t}|C_i) + 1}, \tag{8}$$

where M is the number of classes, $P(C_i|t)$ is the conditional probability of class C_i given presence of term t and $P(\bar{t}|C_i)$ is the conditional probability of absence of term t given class C_i, respectively. It is obvious from this formulation that a term occurring in all documents of a class and not occurring in the other classes will be assigned 1.0 as the top score. Moreover, features rarely occurring in a single class while not occurring in the other classes would get lower scores. However, this formulation does not satisfy the third requirement because the features occurring in every document of all classes are invalidly assigned 1.0 as well. In order to resolve this issue, the formulation is extended to

$$\mathrm{DFS}(t) = \sum_{i=1}^{M} \frac{P(C_i|t)}{P(\bar{t}|C_i) + P(t|\overline{C}_i) + 1}, \tag{9}$$

where $P(t|\overline{C}_i)$ is the conditional probability of term t given the classes other than C_i. Since addition of $P(t|\overline{C}_i)$ to the denominator decreases scores of the terms occurring in all classes, the third requirement is also satisfied. Considering the entire formulation, the fourth and last requirement is satisfied as well. The formulation provides global discriminatory powers of the features over the entire text collection rather than class specific scores. It is obvious from this scoring scheme that DFS assigns scores to the features between 0.5 and 1.0 according to their significance. In other words, the most discriminative terms have an importance score that is close to 1.0 while the least discriminative terms are assigned an importance score that converges to 0.5. Once the discriminatory powers of all terms in a given collection are attained, top-N terms can be selected just as in the case of the other filter techniques.

A sample collection is provided in Table 1 to illustrate how DFS works.

$$\begin{aligned}\mathrm{DFS}(\text{'cat'}) &= \frac{(2/6)}{(0/2+4/4+1)} + \frac{(2/6)}{(0/2+4/4+1)} + \frac{(2/6)}{(0/2+4/4+1)} \\ &= 0.5000,\end{aligned}$$

$$\begin{aligned}\mathrm{DFS}(\text{'dog'}) &= \frac{(1/2)}{(1/2+1/4+1)} + \frac{(1/2)}{(1/2+1/4+1)} + \frac{(0/2)}{(2/2+2/4+1)} \\ &= 0.5714,\end{aligned}$$

$$\begin{aligned}\mathrm{DFS}(\text{'mouse'}) &= \frac{(0/3)}{(2/2+3/4+1)} + \frac{(2/3)}{(0/2+1/4+1)} \\ &\quad + \frac{(1/3)}{(1/2+2/4+1)} \\ &= 0.7000,\end{aligned}$$

$$\begin{aligned}\mathrm{DFS}(\text{'fish'}) &= \frac{(0/2)}{(2/2+2/4+1)} + \frac{(0/2)}{(2/2+2/4+1)} + \frac{(2/2)}{(0/2+0/4+1)} \\ &= 1.0000.\end{aligned}$$

In this sample scenario, maximum score is assigned to 'fish' that occurs in all documents of just a single class, namely $C3$. The successor is determined as 'mouse' due to its occurrence in all documents of class $C2$ and just a single document of $C3$. The term 'dog' is selected as the third informative feature since it appears once in both class $C1$ and $C2$ out of three classes. Finally, the least significant term is determined as 'cat' due to its occurrence in all documents of all three classes. Here, 'fish' and 'cat' represent two extreme cases in terms of discrimination. While 'fish' is present in all documents of just a single class, 'cat' is present in all documents of the collection. Therefore, 'fish' is assigned an importance score of 1.0, which is the highest possible DFS score, whereas 'cat' is assigned an importance score of 0.5, which is the lowest possible DFS score. In summary, DFS sensibly orders the terms based on their contributions to class discrimination as 'fish', 'mouse', 'dog', and 'cat'.

The sample collection and the related results are provided to show briefly how DFS method works. Actual performance of DFS on various benchmark datasets with distinct characteristics is thoroughly assessed in the experimental work.

4. Classification algorithms

Since DFS is a filter based technique, it does not depend on the learning model. Therefore, three different classification algorithms were employed to investigate contributions of the selected features to the classification accuracy. The first classifier is Decision Tree (DT), which is a non-linear classifier [37]. The second one is linear support vector machine (SVM) classifier [22]. The third

Table 1
Sample collection.

Document name	Content	Class
Doc 1	cat	C1
Doc 2	cat dog	C1
Doc 3	cat dog mouse	C2
Doc 4	cat mouse	C2
Doc 5	cat fish	C3
Doc 6	cat fish Mouse	C3

and last classifier is a neural network (NN) classifier [4]. All those classification methods have been commonly used for text classification research in the literature and proven to be significantly successful [10,14,23,25,43].

4.1. DT classifier

Decision, or classification, trees are multistage decision systems in which classes are consecutively rejected until an accepted class is reached [37]. For this purpose, feature space is split into unique regions corresponding to the classes. The most commonly used type of decision trees is binary classification tree that splits the feature space into two parts sequentially by comparing feature values with a specific threshold. Thus, an unknown feature vector is assigned to a class via a sequence of Yes/No decisions along a path of nodes of a decision tree. One has to consider splitting criterion, stop-splitting rule, and class assignment rule in design of a classification tree.

The fundamental aim of splitting feature space is to generate subsets that are more class homogeneous compared to former subsets. In other words, the splitting criterion at any node is to obtain the split providing the highest decrease in node impurity. Entropy is one of the widely used information to define impurity, and can be computed as

$$I(t) = -\sum_{i=1}^{M} P(C_i|t)\log_2 P(C_i|t), \tag{10}$$

where $P(C_i|t)$ denotes the probability that a vector in the subset X_t, associated with a node t, belongs to class C, $i = 1, 2, \ldots, M$. Assume now that performing a split, N_{tY} points are sent into "Yes" node (X_{tY}) and N_{tN} into "No" node (X_{tN}). The decrease in node impurity is then defined as

$$\Delta I(t) = I(t) - \frac{N_{tY}}{N_t} I(t_{\mathrm{YES}}) - \frac{N_{tN}}{N_t} I(t_{\mathrm{NO}}), \tag{11}$$

where $I(t_{\mathrm{YES}})$, $I(t_{\mathrm{NO}})$ are the impurities of the t_{YES} and t_{NO} nodes, respectively. If the highest decrease in node impurity is less than a certain threshold or a single class is obtained following a split, then splitting process is stopped. Once a node is declared to be terminal or leaf, then a class assignment is made. A commonly used assignment method is the majority rule that assigns a leaf to a class to which the majority of the vectors in the corresponding subset belong.

4.2. SVM classifier

SVM is one of the most effective classification algorithms in the literature. SVM algorithm has both linear and non-linear versions. In this study, linear version of SVM is employed. The essential point of SVM classifier is the notion of the margin [22,37]. Classifiers utilize hyperplanes to separate classes. Every hyperplane is characterized by its direction (w) and its exact position in space (w_0). Thus, a linear classifier can be simply defined as

$$w^T x + w_0 = 0. \tag{12}$$

Then, the region between the hyperplanes $w^T x + w_0 = 1$ and $w^T x + w_0 = -1$, which separates two classes, is called as the margin. Width of the margin is equal to $2/\|w\|$. Achieving the maximum possible margin is the underlying idea of SVM algorithm. Maximization of the margin requires minimization of

$$J(w, w_0, \varepsilon) = \frac{1}{2}\|w\|^2 + K\sum_{i=1}^{N} \varepsilon_i, \tag{13}$$

which is subject to

$$\begin{aligned} &w^T x_i + w_0 \geqslant 1 - \varepsilon_i, \quad \text{if } x_i \in c_1 \\ &w^T x_i + w_0 \leqslant -1 + \varepsilon_i, \quad \text{if } x_i \in c_2 \\ &\varepsilon_i \geqslant 0. \end{aligned} \tag{14}$$

In (13), K is a user defined constant, and ε is the margin error. Margin error occurs if data belonging to one class is on the wrong side of the hyperplane. Minimizing the cost is therefore a trade-off issue between a large margin and a small number of margin errors. Solution of this optimization problem is obtained as

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i, \tag{15}$$

which is the weighted average of the training features. Here, λ_i is a Lagrange multiplier of the optimization task, and y_i is a class label. Values of λ's are nonzero for all the points lying inside the margin and on the correct side of the classifier. These points are known as support vectors, and the resulting classifier as the support vector machine.

In case of multi-class classification problems, one of two common approaches, namely one-against-all and one-against-one, can be preferred to adopt two-class classification to multi-class case [20].

4.3. NN classifier

One of the widely used application fields of neural networks are pattern recognition problems [12]. While some neural networks such as perceptron is known to be successful for linear classification problems, multi-layer neural networks can solve both linear and non-linear classification problems. A neural network consists of neurons which are very simple processing elements and connected to each other with weighted links. Multi-layer neural networks consist of input, output and hidden layer(s). While one hidden layer is sufficient for many cases, using two hidden layers may increase performance in some situations [12]. A simple multi-layer feed-forward neural network is shown in Fig. 1, where n represents the dimension of input vector and m represents the number of outputs.

Back-propagation is one of the most popular training methods for multi-layer feed forward neural networks. Training with back-propagation has three stages given as below:

i. The feed-forward of input training pattern.

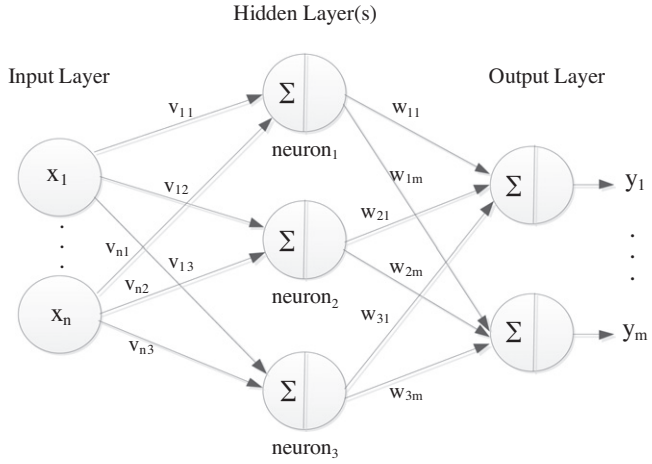

Fig. 1. A simple multi-layer feed-forward neural network.

ii. The back-propagation of error.
iii. The adjustment of weights.

For the first stage, the net inputs to neurons need to be calculated and some summation and multiplication operations are performed as shown in (16). This formula simulates the calculation of the net input for $neuron_1$ in Fig. 1.

$$y\_neuron_1 = v_{11} \cdot x_1 + v_{21} \cdot x_2 + \ldots + v_{n1} \cdot x_n. \quad (16)$$

After calculation of the net inputs, transfer functions are used to compute each neurons output from the net inputs. Some examples of the transfer functions are linear, logarithmic sigmoid and tangent sigmoid transfer functions. There exist many variants of back-propagation training such as Levenberg-Marquardt, gradient descent and gradient descent with momentum and adaptive learning rate. Following the first stage, the second and third stages are carried out respectively. These operations are repeated until a predefined stopping criterion is achieved. The stopping criteria can be minimum error goal and/or maximum iteration count. The first stage consisting of straightforward calculations is repeated in order to execute the testing phase of neural network.

For this paper, the parameters of the neural network are given as follows: Gradient descent with momentum and adaptive learning rate back-propagation algorithm was selected for learning. A neural network with two hidden layers having 20 neurons in each layer was employed. Transfer functions were tangent sigmoid for the hidden layers, and linear for the output layer. The minimum error goal was set as 0.01, and the maximum iteration was determined as 1000 epoch. In the experimental work, all the reported results belonging to NN classifier are average of five runs.

5. Experimental work

In this section, an in-depth investigation is carried out to compare DFS against the abovementioned feature selection methods in terms of feature similarity, classification accuracy, dimension reduction rate, and processing time. For this purpose, four different datasets with varying characteristics and two different success measures were utilized to observe effectiveness of DFS method under different circumstances. In the following subsections, the utilized datasets and success measures are described briefly. Then, similarity, accuracy, dimension reduction, and timing analysis are presented. It should also be noted that stop-word removal and stemming [34] were carried out as the two pre-processing steps.

5.1. Datasets

Characteristic of the dataset is one of the most important factors to assess efficiency of a feature selection method. Consequently, in this study, four distinct datasets with varying characteristics were used for the assessment. The first dataset consists of the top-10 classes of the celebrated Reuters-21578 ModApte split [3]. The second dataset contains ten classes of another popular text collection, namely 20 Newsgroups [3]. The third dataset is a short message service (SMS) message collection introduced in [1]. The final dataset is a spam e-mail collection, namely Enron1, which is one of the six datasets used in [30]. The detailed information regarding those datasets is provided in Tables 2–5. It is obvious from these tables that Reuters, SMS and Enron1 datasets are imbalanced, that is, numbers of documents in each class are quite different. On the contrary, Newsgroups dataset is a balanced one with equal number of documents per class. Furthermore, Reuters and Newsgroups datasets are multi-class whereas SMS and Enron1 are good examples to the binary class datasets.

Table 2
Reuters dataset.

No.	Class label	Training samples	Testing samples
1	Earn	2877	1087
2	Acq	1650	719
3	Money-fx	538	179
4	Grain	433	149
5	Crude	389	189
6	Trade	369	117
7	Interest	347	131
8	Ship	197	89
9	Wheat	212	71
10	Corn	181	56

Table 3
Newsgroups dataset.

No.	Class label	Training samples	Testing samples
1	Alt.atheism	500	500
2	Comp.graphics	500	500
3	Comp.os.ms-windows.misc	500	500
4	Comp.sys.ibm.pc.hardware	500	500
5	Comp.sys.mac.hardware	500	500
6	Comp.windows.x	500	500
7	Misc.forsale	500	500
8	Rec.autos	500	500
9	Rec.motorcycles	500	500
10	Rec.sport.baseball	500	500

Table 4
SMS dataset.

No.	Class label	Training samples	Testing samples
1	Legitimate	1436	3391
2	Spam	238	509

Table 5
Enron1 dataset.

No.	Class label	Training samples	Testing samples
1	Legitimate	2448	1224
2	Spam	1000	500

5.2. Success measures

The two success measures employed in this study are well known F1 measures, namely Macro-F1 and Micro-F1 [14,28].

In macro-averaging, F-measure is computed for each class within the dataset and then the average over all classes is obtained. In this way, equal weight is assigned to each class regardless of the class frequency. Computation of Macro-F1 can be formulated as

$$\text{Macro-F1} = \frac{\sum_{k=1}^{C} F_k}{C}, F_k = \frac{2 \cdot p_k \cdot r_k}{p_k + r_k}, \quad (17)$$

where pair of (p_k, r_k) corresponds to precision and recall values of class k, respectively.

On the other hand, in micro-averaging, F-measure is computed globally without class discrimination. Hence, all classification decisions in the entire dataset are considered. In case that the classes in a collection are biased, large classes would dominate small ones in micro-averaging. Computation of Micro-F1 can be formulated as

$$\text{Micro-F1} = \frac{2 \cdot p \cdot r}{p + r}, \quad (18)$$

where pair of (p, r) corresponds to precision and recall values, respectively, over all the classification decisions within the entire dataset not individual classes.

Table 6
Top-10 features in (a) Reuters, (b) Newsgroups, (c) SMS and (d) Enron1 dataset.

No.	1	2	3	4	5	6	7	8	9	10
(a)										
CHI2	cts	net	shr	qtr	rev	loss	**acquir**	**profit**	**note**	**dividend**
IG	cts	net	wheat	bank	shr	qtr	tonn	**export**	trade	agricultur
GI	cts	net	shr	wheat	oil	barrel	qtr	**march**	rev	**crude**
DP	**mln**	**dlr**	cts	loss	net	bank	**pct**	**billion**	trade	**share**
DFS	cts	wheat	net	oil	shr	tonn	**corn**	barrel	qtr	agricultur
(b)										
CHI2	basebal	forsal	auto	atheism	motorcycl	rec	comp	sport	hardwar	**sys**
IG	comp	window	rec	**car**	**dod**	**misc**	**sale**	**refer**	**apr**	hardwar
GI	atheism	basebal	motorcycl	forsal	auto	sport	os	mac	ms	**graphic**
DP	window	**path**	**id**	**newsgroup**	**date**	**messag**	**subject**	**organ**	**line**	**cantaloup**
DFS	atheism	basebal	motorcycl	forsal	auto	sport	os	mac	ms	hardwar
(c)										
CHI2	txt	call	free	mobil	servic	text	award	box	**stop**	contact
IG	call	txt	free	mobil	www	text	claim	servic	award	ur
GI	call	txt	free	www	claim	mobil	prize	text	ur	servic
DP	txt	call	free	mobil	servic	text	award	box	contact	**urgent**
DFS	txt	free	www	claim	mobil	call	prize	**guarante**	**uk**	servic
(d)										
CHI2	http	cc	enron	gas	ect	pm	meter	forward	hpl	**www**
IG	cc	gas	ect	pm	meter	http	corp	**volum**	**attach**	forward
GI	subject	enron	cc	hpl	gas	forward	ect	daren	hou	pm
DP	ect	hou	enron	meter	**deal**	subject	gas	pm	cc	corp
DFS	enron	cc	hpl	gas	ect	daren	hou	pm	forward	meter

5.3. Term similarity analysis

Profile of the features that are selected by a feature selection method is one of the good indicators to effectiveness of that method. If distinctive features are assigned high scores by a feature selection method, the classification accuracy obtained by those features will most likely be higher. On the contrary, if irrelevant features are assigned high scores by a feature selection method, the accuracy obtained by those features would be degraded. For this purpose, similarities and dissimilarities of the features that are selected by DFS were first compared against the other selection techniques. Initially, top-10 terms selected by each method are presented in Table 6. The terms that are specific to an individual selection method are indicated in bold. One can note from the table that DFS selects similar as well as dissimilar terms in each dataset with respect to the other methods. As an example, in Reuters dataset, nine out of ten terms selected by DFS were also selected by the other methods. However, the remaining one term, namely "corn", is specific to DFS. Considering that "corn" has an occurrence rate of 73% in class-10 and much lower occurrence rate in the other classes, this term can be regarded as a discriminative feature. Therefore, presence of this term in the top-10 list is quite meaningful.

To observe the generalized behavior of DFS, similarities and dissimilarities of top-100 and top-500 features selected by DFS were analyzed for each dataset. Results of this analysis are presented in Figs. 2–5. For instance, in Reuters dataset, 71% of top-500 features, which are selected by DFS, are common with the ones selected by CHI2 whereas the remaining 29% of the features are specific to DFS method. In general, while DFS selected particular amount of similar features to the ones selected by CHI2 in balanced dataset (Newsgroups), and by GI in imbalanced datasets (Reuters, SMS, and Enron1), it also selected completely distinct features with varying quantities in each dataset.

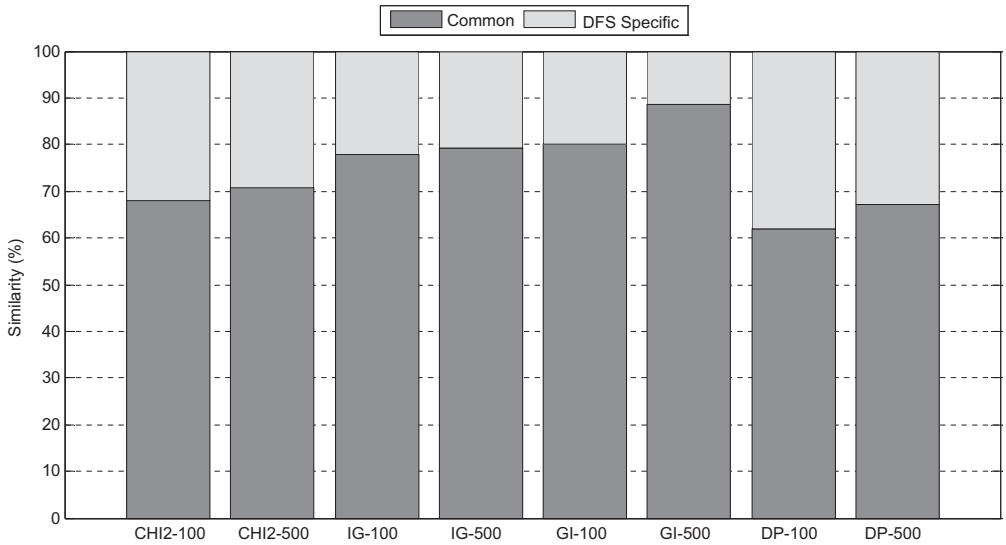

Fig. 2. Reuters: similarity of the features selected by DFS against the other methods.

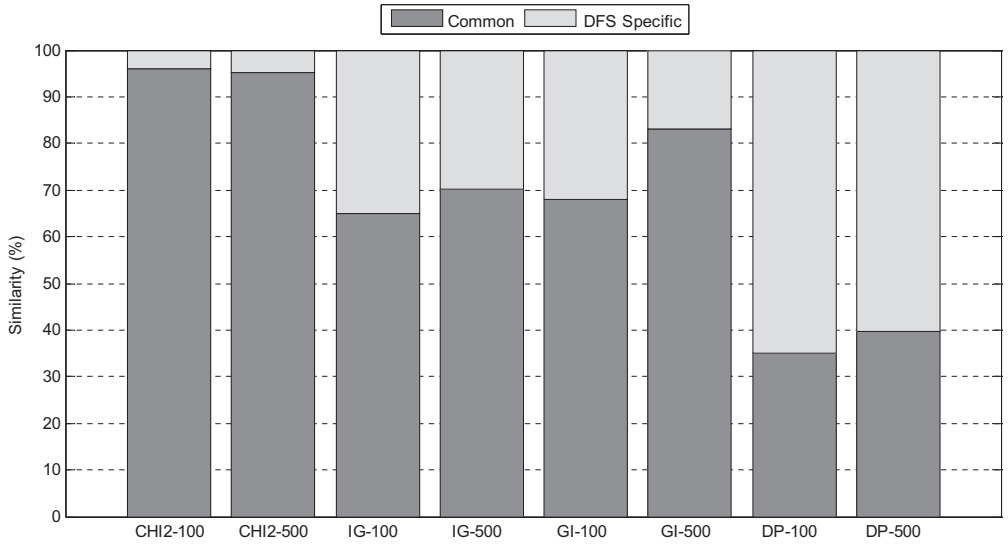

Fig. 3. Newsgroups: similarity of the features selected by DFS against the other methods.

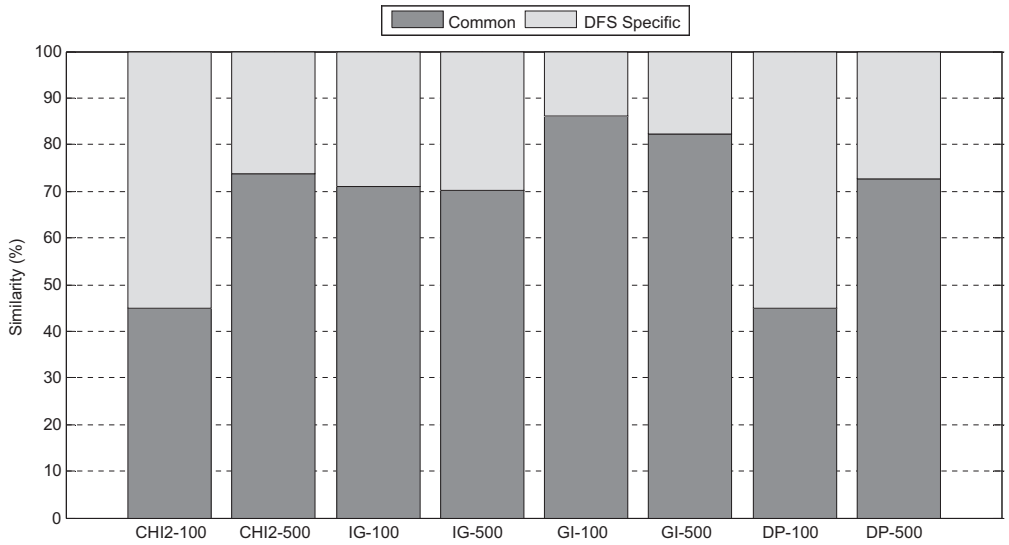

Fig. 4. SMS: similarity of the features selected by DFS against the other methods.

Further analysis on the selected features revealed that GI and DP may include some uninformative terms in their top-N lists. For instance, "subject" term occurs in all documents of Newsgroups dataset that makes "subject" uninformative. However, this term was present within the top-50 features selected by GI and DP. Nevertheless, this term and other terms with similar characteristics were not available even in the top-500 features selected by DFS.

5.4. Accuracy analysis

Varying numbers of the features, which are selected by each selection method, were fed into DT, SVM, and NN classifiers. Resulting Micro-F1 and Macro-F1 scores are listed in Table 7–10 for each dataset respectively where the highest scores are indicated in bold. Considering the highest scores, DFS is either superior to all other methods or runner up with just a slight difference. For instance, in Reuters dataset, the features selected by DFS provided both the highest Micro-F1 and Macro-F1 scores using DT classifier. Similarly, in Newsgroups dataset, the highest Micro-F1 and Macro-F1 scores were obtained with SVM and NN classifiers that use the features selected by DFS. As another example, in SMS dataset, both the highest Micro-F1 and Macro-F1 scores were attained by DT and NN classifiers using the features that are selected by DFS, as well. Finally, in Enron1 dataset, both the highest Micro-F1 and Macro-F1 scores were attained by all of the three classifiers using the features that are selected by DFS.

5.5. Dimension reduction analysis

In addition to accuracy, dimension reduction rate is another important aspect of feature selection. Therefore, an analysis for dimension reduction was also carried out during the experiments. To compare the efficiency of DFS in terms of dimension reduction rate together with the accuracy, a scoring scheme [15] that combines these two information was employed. This scheme favors better accuracy at lower dimensions as given in

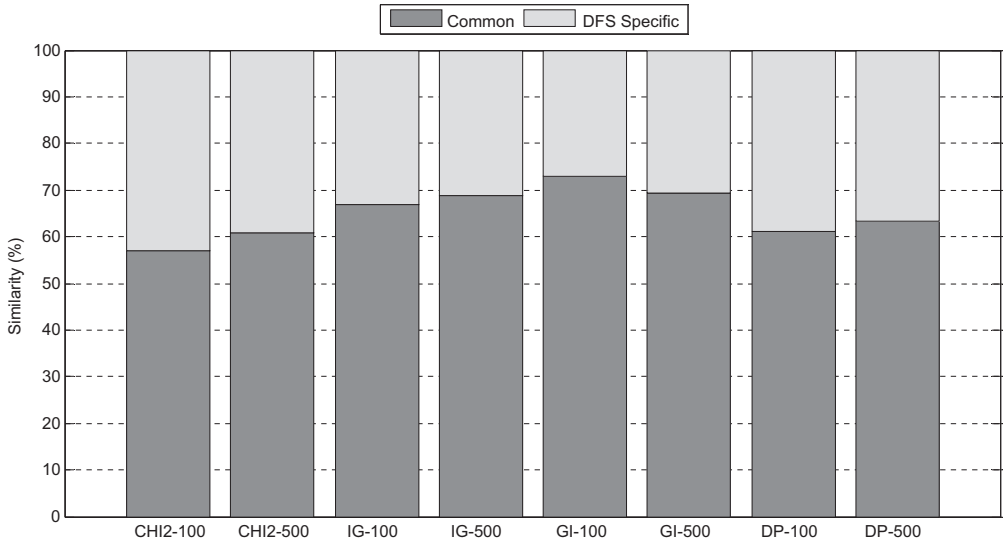

Fig. 5. Enron1: similarity of the features selected by DFS against the other methods.

$$\text{Score} = \frac{1}{k}\sum_{i=1}^{k}\frac{\dim_N}{\dim_i}R_i, \tag{19}$$

where N is the maximum feature size utilized, k is the number of trials, $\dim_i$ is the feature size at the ith trial and R_i is the success rate of the ith trial. The result of dimension reduction analysis using the described scoring scheme is presented in Table 11 where the highest scores are indicated in bold. It is apparent from this table that DFS provides comparable and even better performance with respect to the other methods most of the time.

5.6. Timing analysis

Algorithmic complexities of all feature selection methods considered in this study were computed to be the same. Therefore, the processing time of DFS, rather than its algorithmic complexity, was investigated and compared to the other methods. For this purpose, the computation time of importance score for a single term was considered. The measurements were taken on a computer equipped with Intel Core i7 1.6 GHz processor and 6 GB of RAM. The results of the timing analysis, which are given in Table 12, indicate that DFS is the fastest method among all.

6. Conclusions

In this study, a novel filter based feature selection method, namely DFS, was introduced for text classification research. DFS assesses the contributions of terms to the class discrimination in a probabilistic approach and assigns certain importance scores to them. Using different datasets, classification algorithms and success measures, effectiveness of DFS was investigated and compared against well known filter techniques. The results of a thorough experimental analysis clearly indicate that DFS offers a considerably successful performance in terms of accuracy, dimension reduction rate and processing time. Adaptation of DFS to the other pattern classification problems remains as an interesting future work.

Table 7
Success measures (%) for Reuters dataset using (a) DT, (b) SVM and (c) NN.

Feature size	Micro-F1						Macro-F1					
	10	50	100	200	300	500	10	50	100	200	300	500
(a)												
CHI2	61.97	80.70	82.63	82.63	83.06	82.74	17.36	56.74	57.28	57.09	58.75	58.99
IG	69.61	81.34	83.03	82.89	83.21	82.81	35.32	58.75	58.04	57.64	59.18	59.01
GI	67.74	81.63	81.84	83.21	81.88	82.74	27.61	58.99	57.40	58.74	58.54	58.58
DP	68.35	79.26	81.27	81.70	82.38	82.42	37.13	54.72	59.07	57.39	58.10	58.15
DFS	70.15	82.78	82.63	82.92	83.10	**83.28**	33.91	**61.25**	58.40	58.65	59.22	59.42
(b)												
CHI2	62.04	83.17	85.83	85.76	85.97	85.90	15.11	55.97	60.67	64.05	64.96	64.26
IG	69.72	83.57	85.68	**86.33**	86.01	85.86	34.90	59.66	61.48	**66.55**	65.16	64.92
GI	68.82	83.42	85.79	86.04	85.94	**86.33**	28.38	59.95	62.13	64.62	65.41	65.91
DP	67.17	81.99	85.47	85.76	85.83	86.11	32.36	55.43	61.26	64.48	66.04	65.47
DFS	70.11	84.18	85.72	86.04	85.90	85.79	32.12	61.39	62.55	64.68	65.98	64.93
(c)												
CHI2	61.43	81.66	84.74	85.35	85.57	85.64	15.06	54.65	60.97	63.14	63.09	**64.74**
IG	68.68	81.34	83.04	85.39	85.27	85.80	32.58	57.89	59.56	64.02	62.49	63.16
GI	67.87	81.64	84.28	85.40	85.36	85.57	28.49	59.52	63.42	62.76	62.84	63.83
DP	65.86	79.52	84.03	85.62	85.78	85.68	26.80	52.49	59.93	64.37	63.73	63.51
DFS	69.26	81.07	84.07	85.40	**85.96**	85.92	30.30	60.08	62.09	63.18	64.33	64.31

Table 8
Success measures (%) for Newsgroups dataset using (a) DT, (b) SVM and (c) NN.

Feature size	Micro-F1						Macro-F1					
	10	50	100	200	300	500	10	50	100	200	300	500
(a)												
CHI2	71.32	97.84	97.86	97.68	97.70	97.78	69.32	97.85	97.87	97.69	97.71	97.79
IG	78.38	97.80	97.78	97.70	97.72	97.62	78.37	97.81	97.79	97.71	97.73	97.63
GI	87.62	97.86	97.88	**97.90**	97.70	97.72	85.03	97.87	97.89	**97.91**	97.71	97.73
DP	22.56	97.62	97.58	97.74	97.74	97.64	15.83	97.63	97.59	97.75	97.75	97.65
DFS	88.10	97.84	97.76	97.76	97.80	97.78	86.04	97.85	97.77	97.77	97.81	97.79
(b)												
CHI2	70.36	97.02	97.20	96.84	96.60	96.22	66.12	97.01	97.19	96.85	96.61	96.23
IG	78.40	97.24	97.14	96.14	95.88	96.32	76.89	97.23	97.14	96.15	95.89	96.32
GI	87.96	97.20	96.84	97.04	96.14	96.28	85.41	97.19	96.83	97.05	96.15	96.28
DP	20.44	96.96	97.12	95.90	95.20	95.50	10.64	96.95	97.12	95.91	95.20	95.50
DFS	88.18	97.06	**97.32**	96.88	96.56	96.18	86.00	97.05	**97.32**	96.88	96.57	96.19
(c)												
CHI2	67.13	96.12	96.42	96.94	96.65	96.44	60.66	96.11	96.40	96.93	96.64	96.42
IG	68.79	94.60	95.75	95.76	96.40	96.16	66.42	94.55	95.74	95.75	96.39	96.15
GI	86.52	95.72	95.99	96.58	96.76	96.29	82.84	95.67	95.98	96.57	96.76	96.28
DP	20.29	95.32	96.61	96.54	96.28	95.87	10.65	95.29	96.60	96.53	96.27	95.85
DFS	86.62	96.00	96.36	**96.98**	96.68	96.42	83.06	95.96	96.35	**96.97**	96.67	96.41

Table 9
Success measures (%) for SMS dataset using (a) DT (b) SVM (c) NN.

Feature size	Micro-F1						Macro-F1					
	10	50	100	200	300	500	10	50	100	200	300	500
(a)												
CHI2	93.97	95.85	96.13	96.03	96.18	96.33	84.50	90.20	91.01	90.55	91.05	91.45
IG	94.67	96.23	96.36	96.18	96.13	96.23	86.89	91.23	91.49	90.91	90.94	91.21
GI	94.69	96.41	96.23	96.21	96.26	96.28	86.78	91.56	91.21	91.19	91.36	91.40
DP	93.80	95.77	96.13	96.05	96.18	96.33	83.89	89.98	91.01	90.72	91.05	91.45
DFS	94.41	**96.49**	96.00	96.23	96.26	96.33	85.81	**91.77**	90.53	91.26	91.33	91.48
(b)												
CHI2	93.05	96.54	96.64	97.05	97.05	97.08	84.03	91.84	92.07	93.11	93.14	93.17
IG	94.08	96.74	97.23	97.13	96.87	97.31	85.78	92.23	93.56	93.20	92.57	93.68
GI	94.00	96.82	97.26	97.33	97.15	97.41	85.56	92.51	93.67	93.83	93.37	**94.00**
DP	93.33	96.67	96.82	97.18	97.15	96.95	83.83	92.18	92.54	93.42	93.35	92.84
DFS	94.18	96.95	96.90	97.18	97.05	**97.44**	85.92	92.98	92.85	93.43	93.09	93.94
(c)												
CHI2	94.19	96.37	96.70	97.33	97.19	97.15	85.56	91.38	92.19	93.77	93.37	93.21
IG	94.50	96.65	97.15	97.08	97.02	97.12	86.57	92.09	93.36	93.10	92.91	93.10
GI	94.50	96.82	97.23	97.32	97.28	97.26	86.62	92.53	93.51	93.69	93.59	93.47
DP	94.11	96.48	96.60	97.31	97.17	97.28	85.30	91.63	91.90	93.70	93.31	93.57
DFS	94.55	97.00	97.15	97.17	96.94	**97.39**	86.47	93.02	93.36	93.40	92.74	**93.82**

Table 10
Success measures (%) for Enron1 dataset using (a) DT (b) SVM (c) NN.

Feature size	Micro-F1						Macro-F1					
	10	50	100	200	300	500	10	50	100	200	300	500
(a)												
CHI2	81.15	91.13	91.01	90.31	90.26	91.88	79.89	89.62	89.50	88.49	88.42	90.39
IG	78.48	89.62	90.55	90.08	90.43	89.50	76.90	87.88	88.86	88.16	88.52	87.51
GI	80.97	90.08	90.84	91.30	91.13	91.07	79.70	88.32	88.93	89.53	89.39	89.33
DP	74.88	91.13	90.14	89.15	89.56	89.56	73.91	89.72	88.43	87.05	87.58	87.54
DFS	83.64	90.31	90.02	91.24	**92.00**	91.65	82.27	88.60	88.16	89.54	**90.42**	90.02
(b)												
CHI2	79.52	89.79	90.95	90.49	90.43	92.75	78.28	88.23	89.60	88.90	88.79	91.38
IG	78.89	88.34	91.18	92.00	91.59	92.58	77.34	86.58	89.80	90.61	90.07	91.25
GI	80.63	89.79	89.50	91.36	91.71	91.76	79.34	88.20	87.67	89.86	90.04	90.09
DP	75.58	90.08	91.18	91.47	92.34	90.89	74.20	88.57	89.80	89.95	90.91	89.21
DFS	83.30	89.97	89.85	92.63	**93.21**	93.10	81.90	88.37	88.17	91.39	**91.96**	91.70
(c)												
CHI2	79.91	90.75	91.37	91.37	91.25	91.96	78.66	89.38	90.05	89.90	89.78	90.44
IG	79.18	89.93	91.16	91.97	91.59	92.31	77.53	88.38	89.74	90.56	90.083	90.82
GI	80.92	90.91	91.33	92.16	92.24	93.75	79.58	89.30	89.65	90.72	90.74	92.48
DP	75.87	90.99	91.85	91.89	92.38	92.04	74.42	89.60	90.51	90.42	90.97	90.55
DFS	83.41	91.04	91.42	92.63	93.48	**94.35**	81.99	89.59	89.94	91.34	92.27	**93.19**

Table 11
Performance scores in (a) Reuters (b) Newsgroups (c) SMS (d) Enron1 dataset.

	SVM		DT		NN	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
(a)						
CHI2	801	325	791	337	792	322
IG	866	**498**	856	491	851	**472**
GI	858	444	840	427	845	444
DP	842	469	840	**500**	825	416
DFS	**870**	478	**863**	484	**856**	459
(b)						
CHI2	912	877	923	906	883	829
IG	979	967	982	982	893	874
GI	1059	1038	1059	1037	1044	1013
DP	495	414	516	460	491	411
DFS	**1061**	**1043**	**1063**	**1046**	**1045**	**1016**
(c)						
CHI2	1100	1010	1106	1009	1110	1023
IG	1110	1027	1112	**1031**	1113	1033
GI	1110	1026	**1113**	1030	1114	**1035**
DP	1103	1010	1104	1003	1109	1021
DFS	**1111**	**1029**	1110	1022	**1114**	1034
(d)						
CHI2	966	951	982	966	972	957
IG	960	942	956	937	965	946
GI	975	959	979	963	982	965
DP	935	918	928	914	940	923
DFS	**999**	**982**	**1001**	**984**	**1003**	**987**

Table 12
Timing analysis.

	CHI2	IG	GI	DP	DFS
Computation time (s)	0.0632	0.0693	0.0371	0.0797	0.0343

References

[1] T.A. Almeida, J.M.G. Hidalgo, A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, in: Proceedings of the 11th ACM Symposium on Document, Engineering, 2011, pp. 259–262.
[2] I. Anagnostopoulos, C. Anagnostopoulos, V. Loumos, E. Kayafas, Classifying web pages employing a probabilistic neural network, IEE Proceedings – Software 151 (3) (2004) 139–150.
[3] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, University of California, Department of Information and Computer Science, Irvine, CA, 2007.
[4] C.M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag Inc., New York, 2006.
[5] D.B. Bracewell, J. Yan, F. Ren, S. Kuroiwa, Category classification and topic discovery of Japanese and English news articles, Electronic Notes in Theoretical Computer Science 225 (2009) 51–65.
[6] J. Chen, H. Huang, S. Tian, Y. Qu, Feature selection for text classification with Naive Bayes, Expert Systems with Applications 36 (3) (2009) 5432–5435.
[7] R.-C. Chen, C.-H. Hsieh, Web page classification based on a support vector machine using a weighted vote schema, Expert Systems with Applications 31 (2) (2006) 427–435.
[8] Y.-T. Chen, M.C. Chen, Using chi-square statistics to measure similarities for text categorization, Expert Systems with Applications 38 (4) (2011) 3085–3090.
[9] N. Cheng, R. Chandramouli, K.P. Subbalakshmi, Author gender identification from text, Digital Investigation 8 (1) (2011) 78–88.
[10] H. Drucker, D. Wu, V. Vapnik, Support vector machines for spam categorization, IEEE Transactions on Neural Networks 10 (5) (1999).
[11] S. Efstathios, Author identification: using text sampling to handle the class imbalance problem, Information Processing and Management 44 (2) (2008) 790–799.
[12] L. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice-Hall, Inc., 1994.
[13] G. Forman, An extensive empirical study of feature selection metrics for text classification, Journal of Machine Learning Research 3 (2003) 1289–1305.
[14] S. Gunal, Hybrid feature selection for text classification, Turkish Journal of Electrical Engineering and Computer Sciences (in press), doi: 10.3906/elk-1101-1064.
[15] S. Gunal, R. Edizkan, Subspace based feature selection for pattern recognition, Information Sciences 178 (19) (2008) 3716–3726.
[16] S. Gunal, S. Ergin, M.B. Gulmezoglu, O.N. Gerek, On feature extraction for spam e-mail detection, Lecture Notes in Computer Science 4105 (2006) 635–642.
[17] S. Gunal, O.N. Gerek, D.G. Ece, R. Edizkan, The search for optimal feature set in power quality event classification, Expert Systems with Applications 36 (7) (2009) 10266–10273.
[18] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[19] T.S. Guzella, W.M. Caminhas, A review of machine learning approaches to spam filtering, Expert Systems with Applications 36 (7) (2009) 10206–10222.
[20] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.
[21] T. Joachims, A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 143–151.
[22] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of the 10th European Conference on Machine Learning, 1998, pp. 137–142.
[23] D.E. Johnson, F.J. Oles, T. Zhang, T. Goetz, A decision-tree-based symbolic rule induction system for text categorization, IBM Systems Journal 41 (3) (2002) 428–437.
[24] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.
[25] M.A. Kumar, M. Gopal, A comparison study on multiple binary-class SVM methods for unilabel text categorization, Pattern Recognition Letters 31 (11) (2010) 1437–1444.
[26] C. Lee, G.G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Information Processing and Management 42 (1) (2006) 155–165.
[27] H. Liu, J. Sun, L. Liu, H. Zhang, Feature selection with dynamic mutual information, Pattern Recognition 42 (7) (2009) 1330–1339.
[28] C.D. Manning, P. Raghavan, H. Schtze, Introduction to Information Retrieval, Cambridge University Press, New York, USA, 2008.
[29] S.S.R. Mengle, N. Goharian, Ambiguity measure feature-selection algorithm, Journal of the American Society for Information Science and Technology 60 (5) (2009) 1037–1050.
[30] V. Metsis, I. Androutsopoulos, G. Paliouras, Spam filtering with naive Bayes – which naive Bayes? in: Proceedings of the 3rd Conference on Email and Anti-Spam, 2006.
[31] D. Mladenic, M. Grobelnik, Feature selection on hierarchy of web documents, Decision Support Systems 35 (1) (2003) 45–87.
[32] H. Ogura, H. Amano, M. Kondo, Feature selection with a measure of deviations from Poisson in text categorization, Decision Support Systems 36 (3) (2009) 6826–6832.
[33] S.A. Ozel, A web page classification system based on a genetic algorithm using tagged-terms as features, Expert Systems with Applications 38 (4) (2011) 3407–3415.
[34] M.F. Porter, An algorithm for suffix stripping, Program 14 (3) (1980) 130–137.
[35] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[36] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Systems with Applications 33 (1) (2007) 1–5.
[37] S. Theodoridis, K. Koutroumbas, Pattern Recognition, fourth ed., Academic Press, 2008.
[38] H. Uguz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowledge-Based Systems 24 (7) (2011) 1024–1032.
[39] K. Wu, B.L. Lu, M. Uchiyama, H. Isahara, A probabilistic approach to feature selection for multi-class text categorization, Lecture Notes in Computer Science 4491 (2007) 1310–1317.
[40] J. Yang, Y. Liu, Z. Liu, X. Zhu, X. Zhang, A new feature selection algorithm based on binomial hypothesis testing for spam filtering, Knowledge-Based Systems 24 (6) (2011) 904–914.
[41] Y. Yang, Noise reduction in a statistical approach to text categorization, in: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95), 1995, pp. 256–263.
[42] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 412–420.
[43] B. Yu, D.-h. Zhu, Combining neural networks and semantic feature space for email classification, Knowledge-Based Systems 22 (5) (2009) 376–381.