Applied Intelligence (2019) 49:1597–1619
https://doi.org/10.1007/s10489-018-1349-1

Soft voting technique to improve the performance of global filter based feature selection in text corpus

Deepak Agnihotri[1] · **Kesari Verma**[1] · **Priyanka Tripathi**[1] · **Bikesh Kumar Singh**[2]

Published online: 21 November 2018

Abstract

In text classification, the Global Filter-based Feature Selection Scheme (GFSS) selects the top-N ranked words as features. It discards the low ranked features from some classes either partially or completely. The low rank is usually due to varying occurrence of the words (terms) in the classes. The Latent Semantic Analysis (LSA) can be used to address this issue as it eliminates the redundant terms. It assigns an equal rank to the terms that represent similar concepts or meanings, e.g. four terms "carcinoma", "sarcoma", "melanoma", and "cancer" represent a similar concept, i.e. "cancer". Thus, any selected term by the algorithms from these four terms doesn't affect the classifier performance. However, it does not guarantee that the selection of top-N LSA ranked terms by GFSS are the representative terms of each class. An Improved Global Feature Selection Scheme (IGFSS) solves this issue by selecting an equal number of representative terms from all the classes. However, it has two issues, first, it assigns the class label and membership of each term on the basis of an individual vote of the Odds Ratio (OR) method thereby limiting the decision making capability. Second, the ratio of selected terms is determined empirically by the IGFSS and a common ratio is applied to all the classes to assign the positive and negative membership of the terms. However, the ratio of positive and negative nature terms varies from one class to another and it may be very less for one class, whereas high for other classes. Thus, one common negative features ratio used by the IGFSS affects those classes of a dataset in which there is an imbalance between positive and negative nature words. To address these issues of IGFSS, a new Soft Voting Technique (SVT) is proposed to improve the performance of GFSS. There are two main contributions in this paper: (i) The weighted average score (Soft Vote) of three methods, viz. OR, Correlation Coefficient (CC), and GSS Coefficients (GSS) improves the numerical discrimination of words to identify there positive and negative membership to a class. (ii) A mathematical expression is incorporated in the IGFSS that computes a varying ratio of positive and negative memberships of the terms for each class. The membership is based on the occurrence of the terms in the classes. The proposed SVT is evaluated using four standard classifiers applied on five bench-marked datasets. The experimental results based on Macro_F1 and Micro_F1 measures show that SVT achieves a significant improvement in the performance of classifiers in comparison of standard methods.

Keywords Feature selection · Text classification · Term frequency · Text analysis · Text mining

1 Introduction

Accurate and timely information is the basic need for effective decision making. Wondrous growth in the e-corpus of various fields (e.g. Business, Biomedical, Engineering, News, etc.) [30, 47] demands for an intelligent Decision Support System (DSS) that helps in an Automated Text Document Classification (ATDC) process [10, 11]. In this context, a model is built by observing the occurrence of words in the training set documents with known class labels. The trained model has the capability to predict the class labels of test documents with maximum accuracy [16, 42].

The prediction completely relies on the contents of the documents. Substantial contents of these documents are stored as text [44, 45]. The word (term) is the smallest constituent of text and play a vital role in the ATDC process [27, 33, 39]. The processing steps followed by the ATDC process are as follows: the first step extracts features from the entire corpus (i.e. generation of tokens from the text contents), after this some less informative words (e.g. stop

✉ Deepak Agnihotri
agnihotrideepak@hotmail.com

Extended author information available on the last page of the article.

words, punctuation marks, white spaces) are eliminated in the second step, and in third step lemmatization/stemming is performed in the remaining terms of the corpus. Finally, the resultant terms are used to build a vocabulary of the entire corpus [4]. The resultant terms of this vocabulary are represented by vectors, where the frequency of each term in the documents represents a vector [34]. The collection of term vectors in a matrix form is called a vector space. In this vector space, each individual term constitutes one dimension. For a typical document collection, there may be millions of terms, hence ATDC requires to cater a large number of dimensions which makes the classification process cumbersome [19, 30]. In literature, feature selection techniques are used to select the most relevant features by eliminating the less important features. Feature selection increases the performance as well as the speed of the classifier and is considered as an important step in the classification process.

The Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer [34] is applied to normalize the weight of the terms, but the TF-IDF vectors tend to be of high dimension since they have one component for every term in the vocabulary. The terms which represent similar concepts or with similar meanings are treated as individual words in the corpus and enormously increase the size of TF-IDF vectors.

A linear combination of terms defined using Latent Semantic Analysis (LSA), identifies the relationships among terms.[1] The LSA creates a vector representation of a document that helps to compare documents based on similarity. It assigns an equal weight to the terms representing similar concepts or meanings.[2] E.g., the four words "carcinoma", "sarcoma", "melanoma", and "cancer" represent a similar concept, i.e. "cancer". Thus, an equal weight is assigned by LSA to these words and they contribute equally to the resulting LSA component and any selected word by the algorithms doesn't affect the classifier performance [12]. Further, the LSA processed vector space has a considerable number of features that are not relevant to the text of the specific class. In order to improve the scalability of ATDC, an effective feature selection technique is needed to reduce the feature set.

Many feature subset selection methods have been proposed and studied in machine learning paradigm. It can be broadly divided into three categories: Filter, Wrapper, and Embedded [6, 32, 44]. The filter methods compute the score of a feature using an evaluation function. It is independent of any classification algorithm and determined using the mutual correlation of the data. In contrast, the wrappers and embedded methods require a frequent classifier interaction in their flow to estimate the value of a given subset. The requirement of a classifier interaction may increase running time and force the feature selection method to work according to a specific learning model. Thus, filter-based methods are preferred more in comparison to wrappers and embedded methods [32, 44]. The Global Filter-based Feature Selection Scheme (GFSS) assigns a score to each feature and the topmost, N features are selected using this score, where N is an empirically determined number [23, 44]. The Filter-based methods are further subdivided into One Sided Local Filter-based Feature Selection Scheme (OLFSS) and Global filter-based feature selection scheme (GFSS).

In the OLFSS, the local class-based score for each feature is computed and used as a final score. The GFSS follows a global policy and converts the multiple local scores into a global score to compute the final score of the features. The local and global scores can be directly used in the feature ranking. The features are sorted in descending order and the top-N features are included in the Final Feature Set (FSS). The Information Gain (IG), Gini Index (GI), Distinguishing Feature Selector (DFS) and Gain Ratio (GR) etc. are known as the methods of GFSS, whereas Mutual Information (MI), Odds Ratio (OR), GSS coefficients (GSS), and Correlation Coefficient (CC) as a method of OLFSS [44]. The selected discriminating features of FFS are used by the classifiers in the final step of the ATDC process. A meta-heuristic technique can also be used to search for a configuration that produces a highly effective text classifier. This model selection procedure is commonly named in the literature as hyper-parameter optimization [43].

Although, the GFSS method improves the performance of the classifiers, but it has some limitations. The GFSS is suitable for the balanced dataset where each class contains an equal number of documents along with a sufficient number of terms. In the case of an unbalanced dataset, having a large number of classes with variable distribution of terms, affects the performance of GFSS. The GFSS eliminates the informative features of the class either partially or completely from the topmost, N features. Most of the studies in the literature are focused on providing some improvements to specific feature selection methods rather than providing a new generic scheme.

Uysal [44] extended the work of [19] and proposed a solution named an Improved Global Feature Selection Scheme (IGFSS). IGFSS selects an equal number of representative features from each class in the final feature set. There are mainly two issues with the IGFSS, first, it assigns the class label of each feature with positive or negative

[1] http://blog.josephwilk.net/projects/latent-semantic-analysis-in-python.html

[2] http://mccormickml.com/2016/03/25/lsa-for-text-classification-tutorial/

membership by considering an individual vote of the Odds Ratio (OR) method [34].

An individual method may have some weakness such as the OR method assigns a positive score to a term for a class if the occurrence of the term is more in that class, otherwise, a negative score is assigned. However, the numerical difference between the OR score of the terms for positive and negative membership is very less. Thus, it affects the process of the class label and membership assignment. In this paper, the membership of terms is referred to as nature of terms, i.e. positive or negative membership of a term means positive or negative nature of the terms. Second, the ratio of negative nature features is determined empirically by the IGFSS and a common negative features ratio is applied to all the classes to select the positive and negative nature features. It affects those classes of a dataset which have more positive features than negative or vice-versa [4].

To address these two issues, a new technique named Soft Voting Technique (SVT) is proposed. It is based on the presumption that the ensemble votes of several methods give better results than an individual vote and determines the most appropriate class label of the features. This technique can be useful for a set of equally well-performing methods to balance out their individual weaknesses. The flow of SVT is similar to IGFSS, but in an improved way and based on the key points of IGFSS we have given a generic solution for the filter based feature selection methods. There are two main contributions in this paper:

1. The SVT uses the weighted average score (Soft Vote) of three methods, viz. OR, Correlation Coefficient (CC), and GSS Coefficients (GSS) to predict the class labels of terms and it computes a more balanced score of a word than OR which improves numerical discrimination of positive and negative nature words.
2. A mathematical expression is incorporated in IGFSS that computes a varying ratio of positive and negative nature terms for each class which is based on the occurrence of the terms in the classes.

The proposed SVT is evaluated using four standard classifiers, viz. Linear Support Vector Machine (LSVM), Softmax regression (SOFT MAX), Stochastic Gradient Descent Classifier (SGDC), and RIDGE. The classifiers are applied to five benchmarked text data sets, viz. Webkb, Classic4, Reuters10, Trec2004 and Ohsumed10. The experimental results of SVT, which is based on Macro_F1 and Micro_F1 are compared with classical information science methods and IGFSS.

The rest of the paper is organized as follows. In Section 2, a brief overview of the state-of-the-art methods and related works are discussed. Section 3 presents the details of the proposed SVT. The experimental setup and performance evaluation measures are discussed in Section 4. Section 5 present the experimental results and discussions. Finally, the paper concludes in Section 6.

2 Related works

Substantial works have been carried out in the area of filter-based feature selection. The most common methods, viz. Mutual Information (MI), Information Gain (IG), Distinguishing Feature Selector (DFS), Gini Index (GI), Gain Ratio (GR), and Odds Ratio (OR) are briefly described as follows:

Mutual information (MI) concept [18, 48, 49] is carried out from information theory to measure the dependencies between random variables and used to measure the information contained by a term t_i. It is strongly influenced by the marginal probabilities of the terms. It assigns higher weight to rare terms than common and sparse terms. Therefore, the weights of the terms are not comparable for the terms with widely differing frequencies. The final score (i.e. MI) of term t_i is the maximum class-based score as shown in (1). The brief preliminary notations are shown in the Table 1.

$$MI(t_i) = \max_{j=1}^{j=r} \left[\log \left(\frac{p(t_i, C_j)}{p(t_i) \times p(C_j)} \right) \right] \tag{1}$$

Information Gain (IG) [18, 44, 45, 48, 49] assigns higher weight to common terms distributed in many categories than rare terms. The IG is also known as average Mutual Information. (see (2)).

$$\begin{aligned} IG(t_i) = \; & p(t_i) \times \sum_{j=1}^{j=r} p(C_j|t_i) \times \log p(C_j|t_i) \\ & + p(\bar{t_i}) \times \sum_{j=1}^{j=r} p(C_j|\bar{t_i}) \times \log p(C_j|\bar{t_i}) \\ & - \sum_{j=1}^{j=r} p(C_j) \times \log p(C_j) \end{aligned} \tag{2}$$

Gini Index (GI) is a global feature selection method for text classification which can be defined as an improved version of an attribute selection algorithm used in decision tree construction (see (3)) [44].

$$GI(t_i) = \sum_{j=1}^{r} p\left(t_i|C_j\right)^2 \; p\left(C_j|t_i\right)^2 \tag{3}$$

Distinguishing Feature Selector (DFS) [44, 45] is an improvement of Mutual Information by reducing the effect of marginal probabilities of the terms by normalizing the

Table 1 Preliminary Notations [5, 7]

Notations	Value	Meaning
a	$count(t_i, C_j)$	Document count of word t_i in the class C_j
b	$count(t_i, \bar{C}_j)$	Document count of word t_i in other classes $\bar{C}_j$
c	$count(\bar{t}_i, C_j)$	Document count of other words $\bar{t}_i$ in the class C_j
d	$count(\bar{t}_i, \bar{C}_j)$	Document count of other words $\bar{t}_i$ in other classes $\bar{C}_j$
N	$(a+b+c+d)$	Total number of documents containing the word t_i in all the classes
$p(t_i)$	$(a+b)/N$	The probability of word t_i
$p(\bar{t}_i)$	$(c+d)/N$	The probability of other words $\bar{t}_i$
$p(C_j)$	$(a+c)/N$	The probability of class C_j
$p(\bar{C}_j)$	$(b+d)/N$	The probability of other classes $\bar{C}_j$
$p(t_i, C_j)$	a/N	The probability of word t_i for being in class C_j
$p(t_i, \bar{C}_j)$	b/N	The probability of other words $\bar{t}_i$ for being in class C_j
$p(\bar{t}_i, C_j)$	c/N	The probability of word t_i for being in other classes $\bar{C}_j$
$p(\bar{t}_i, \bar{C}_j)$	d/N	The probability of other words $\bar{t}_i$ for being in other classes $\bar{C}_j$
$p(t_i \mid C_j)$	$a/(a+c)$	The probability of word t_i when class C_j is present
$p(\bar{t}_i \mid C_j)$	$c/(a+c)$	The probability of word t_i when other classes $\bar{C}_j$ are present
$p(t_i \mid \bar{C}_j)$	$b/(b+d)$	The probability of other words $\bar{t}_i$ when class C_j is present
$p(\bar{t}_i \mid \bar{C}_j)$	$d/(b+d)$	The probability of other words $\bar{t}_i$ when other classes $\bar{C}_j$ are present
$p(C_j \mid t_i)$	$a/(a+b)$	The probability of class C_j when word t_i is present
$p(\bar{C}_j \mid t_i)$	$b/(a+b)$	The probability of other classes $\bar{C}_j$ when word t_i is present
$p(C_j \mid \bar{t}_i)$	$c/(c+d)$	The probability of class C_j when other words $\bar{t}_i$ are present
$p(\bar{C}_j \mid \bar{t}_i)$	$d/(c+d)$	The probability of other classes $\bar{C}_j$ when other words $\bar{t}_i$ are present

weight of the terms. It gives weight of the term in a range of [0,1] defined by (4).

$$DFS(t_i) = \sum_{j=1}^{r} \left[\frac{p\left(C_j \mid t_i\right)}{p\left(\overline{t_i} \mid C_j\right) + p\left(t_i \mid \overline{C}_j\right) + 1} \right] \tag{4}$$

Gain Ratio (GR) is proposed in information science to reduce the effect of the most common terms and marginal probabilities of the terms by normalizing their weights, obtained using IG [24] (see (5)).

$$GR(t_i) = \sum_{j=1}^{j=r} \frac{IG(t_i)}{-p\left(C_j\right) \times log\left(p\left(C_j\right)\right)} \tag{5}$$

Odds Ratio (OR) reflects the odds of the word occurring in the positive class normalized by that of the negative class. It has been used for relevance ranking in information retrieval [18, 28, 34, 44, 46] (see (6)).

$$OR\left(t_i, C_j\right) = \frac{p(t_i \mid C_j)(1 - p(t_i \mid \overline{C}_j))}{1 - p(t_i \mid C_j) \times p(t_i \mid \overline{C}_j)} \tag{6}$$

Correlation Coefficient $CC(t_i, C_j)$ of a word t_i with a category C_j is a variant of the χ^2 metric, where $CC^2 = \chi^2 \times CC$ can be viewed as a "one-sided" chi-square metric. The positive values correspond to features indicative of membership, while negative values indicate non-membership. The greater (smaller) the positive (negative) values are, the stronger the terms will be to indicate the membership (non-membership) [36, 39, 50].

$$CC\left(t_i, C_j\right) = \frac{\sqrt{N} \times \left[p(t_i, C_j) \times p(\bar{t}_i, \overline{C}_j) - p(t_i, \overline{C}_j) \times p(\bar{t}_i, C_j)\right]}{\sqrt{p(t_i) \times p(\bar{t}_i) \times p(C_j) \times p(\overline{C}_j)}} \tag{7}$$

GSS Coefficient (GSS) is another simplified variant of the χ^2 statistics proposed by [20]. Similar to CC, the positive values correspond to features indicative of membership, while negative values indicate non-membership [50].

$$GSS\left(t_i, C_j\right) = p(t_i, C_j) \times p(\bar{t}_i, \overline{C}_j) - p(t_i, \overline{C}_j) \times p(\bar{t}_i, C_j) \tag{8}$$

Uysal [44] proposed an ensemble method named as Improved Global Feature Selection Scheme (IGFSS) which provides a generic solution for the GFSS. The IGFSS has merged the power of local and global feature selection

methods. It is an ensemble of OR with any one method of GFSS at a time. The OR is used to assign the class label as well as membership value to the features. It computes the negative value of a feature for the class, if the presence of that feature is very less or none in that class. Similarly, a positive value of a feature for the class, if it occurs most frequently in the class. Further, the IGFSS uses the maximum absolute score of the feature for a class to assign the class label and the sign of the maximum value is used to find out the membership of the feature.

Algorithm 1 Algorithm for Improved Global Feature Selection Scheme (IGFSS) [6, 44].

Input: A set D of documents $D = [d_1, d_2, ..., d_L]$, where $L > 0$ is the total number of documents in the corpus such that each g^{th} document $d_g \in C_j$. Where C_j is a j^{th} class of $C = [C_1, C_2, ...C_j]$, where, $0 < j <= r$ *(r is the total number of classes)*. The feature selection schemes, FSS = {MI, IG, GI, DFS, and GR} and the One Sided Local Feature Selection Scheme, i.e. OLFSS={OR}. In this Algorithm 1 the subscript notations i and j are used to represent the terms and class respectively.

Output: A final feature set FFS of most discriminating features ($t[k] \subset t[m] \subset t[p]$), used as a vocabulary.

Method:

1: Split the corpus D into training and test set, $D = D_{train} + D_{test}$, where D_{train} is the training, and D_{test} is the test corpus.

2: Call function PREPROCESSING(D)

3: Apply TF-IDF Vectorizer on D_{train} and D_{test}

4: Generate the Vocabulary (V) of TF-IDF Vectorized terms of the D_{train}, $V = \{t_1, t_2, \ldots, t_i\}\ \forall i = 1, 2, .., m$

5: Apply the methods of FSS, i.e. MI (1), IG (2), GI (3), DFS (4), GR (5) to compute the score of the terms t_i, $\forall i = 1, 2.., m, \forall j = 1, 2, .., k, .., r$

$$FSS\_Score(t_i) = FSS\left(t_i, C_j\right) \tag{9}$$

6: Arrange the Global feature set (GFS) in descending order based on their Score using (9).

7: Use the OR method to compute the local score and to find out the class label of each term $t_i \subset GFS$

$$Class\_Label(t_i) = C_k,\ ifmax\left(\left|OLFSS\left(t_i, C_j\right)\right|\right) \times is\ for\ class\ C_k, \quad \forall i = 1, 2.., m, \forall j = 1, 2, .., k, .., r \tag{10}$$

8: Compute the positive or negative membership of each term t_i towards the obtained Class_Label C_k

$$Membership\left(t_i, C_k\right) = \begin{cases} pos, if\ (OLFSS\left(t_i, C_k\right)) \geq 0 \\ neg, if\ (OLFSS\left(t_i, C_k\right)) < 0 \end{cases} \tag{11}$$

9: Compute total count of positive and negative features in each class C_j $if\ Class\_Label(t_i) = C_j$ **AND** $Membership\left(t_i, C_j\right) = pos$

$$count_{pos}\left(C_j\right) = \sum_{i=1}^{m} Class\_Label(t_i) \tag{12}$$

else

$$count_{neg}\left(C_j\right) = \sum_{i=1}^{m} Class\_Label(t_i) \tag{13}$$

end if

10: Let the length of the final feature set (FFS) is N

11: Compute the negative features ratio (nfr) and positive features ratio (pfr)

$$nfr = \max_{j=1,2,..r} \frac{count_{neg}(C_j)}{count_{pos}(C_j) + count_{neg}(C_j)},\ where 0 \leq nfr \geq 1 \tag{14}$$

$$pfr = 1 - nfr \tag{15}$$

12: Compute an equal split criteria to select an equal number of features from each class C_j

$$Equal_{Split}\left(C_j\right) = \frac{N}{Total\ number\ of\ categories} \tag{16}$$

13: Compute the selection criteria of positive and negative features in each class C_j

$$Equal\_Pos_{Split}\left(C_j\right) = Equal_{Split}\left(C_j\right) \times pfr \tag{17}$$

$$Equal\_Neg_{Split}\left(C_j\right) = Equal_{Split}\left(C_j\right) \times nfr \tag{18}$$

14: Select the top-N number of features using (16)–(18) and keep them into FFS.

15: Compute selection criteria to add the new features **if** $length(FFS) < N$ **then**

$$N_l = (N - length(FFS)) \tag{19}$$

end if

16: Add the top-N_l number of new features into FFS, based on their scores obtained using (9).

The summarized steps of IGFSS, presented in Algorithm 1 is as follows: (i) Step 5 computes a global score of each term using methods of GFSS (i.e. MI, IG, GI, DFS, and GR), (ii) Step 6 sort the terms based on their computed global score, (iii) Step 7 determines the class label of features, (iv) Step 8 computes positive and negative membership of features, (v) Step 9 computes positive and negative features count for each class, (vi) Steps 10-14 determine the selection criterion based on positive and negative features ratio, and (vii) Steps 14-16 select an equal number of, the most informative features from each class by applying all the above steps.

In IGFSS algorithm, in the worst case, all features need to be traversed once and some of them may be traversed two times while constructing the candidate feature set. Let L be the total number of documents, r as the total number of classes, p as the total number of terms, and m as the number of terms obtained after removal of less informative terms, viz. stop words, punctuation marks, and white spaces. Let N be the number of IGFSS weighted terms that are selected as the most informative terms based on the length of the final feature set.

The values of n, r, m and N are much smaller compared to p, because the total number of terms p *is* in millions, and others are in the hundreds or thousands. Thus, the overall time complexity of the Algorithm 1 is computed as $\Theta(p)$. In the special cases, if the number of documents is in millions and there are fewer terms and classes in comparison to the documents. The resultant number of terms is also in millions because they are extracted by the combination of these documents into a corpus. Thus, the time complexity of the Algorithm 1 will be Θ(n).

3 Proposed Soft Voting Technique (SVT)

Having reviewed the related studies which helped in the ATDC process, it is found that in most of the reported techniques the class label of features is determined using a single criteria [6, 44]. However, the ensemble techniques are less explored in Text Mining to decide the class label of the features. Using single criteria for feature selection has shown the limited capability in knowledge discovery and decision making systems [41]. Therefore, we introduce a new Soft Voting Technique (SVT) to determine the most appropriate class labels of the features. The IGFSS improved the GFSS by using an individual vote of Odds Ratio (OR) method to define the class label of each feature. We extend this state-of-the-art approach by providing a more generic solution for all filter-based feature selection method. The SVT is based on the presumption that the ensemble votes of several methods can yield better results than an individual vote. This technique can be useful for a set of equally well-performing model in order to balance out their individual weaknesses. The SVT uses the weighted average score (Soft Vote) of three methods, viz. Odds Ratio (OR), Correlation Coefficient (CC), and GSS Coefficients (GSS) to predict the class label of features.

The central idea behind use of weighted average score (i.e. Soft Vote) of these three methods is as follows: (i) all three methods, i.e. OR, CC, and GSS assign a positive score to a term for a class if the occurrence of a term is more in that class, otherwise a negative score is assigned, (ii) the numerical difference between the OR score of positive and negative nature terms is very less, (iii) however, the numerical range of scores assigned to the terms by OR is higher than CC and GSS methods, (iv) thus, the resultant sum of OR, CC, and GSS is not much different from OR score, but their weighted average score is a more balanced score than OR, and (v) it helps in discrimination of positive and negative nature of terms for a class due to the computation of balanced numerical difference by the SVT.

To address the second issue of IGFSS, i.e instead of using one common negative features ratio for each class a mathematical expression is incorporated in IGFSS which computes a varying ratio of positive and negative features for each class. The resulting ratio solves the problem of imbalance between positive and negative nature of terms in each class and select the most appropriate positive and negative nature terms from each class.

3.1 Explanation by synthetic data

The concept of SVT is now explained using a synthetic dataset shown in the Table 2. Assume that, we have to select the top 6 words from this data. The OLFSS and weighted average scores of the features of this dataset is shown in Table 3. Further, the class labels as well as membership assigned to these features using SVT are shown in Table 4. The SVT and IGFSS both select two features from each class as shown in Table 5.

It can be observed from Table 3 that all three methods, i.e. OR, CC, and GSS assign a positive score to a term for a class if the occurrence of term is more in that class, otherwise a

Table 2 Synthetic data

Document	Contents	Class Label
D1	deer hagfish shark rays goat	C1
D2	deer hagfish monkey shark toad	C1
D3	deer goat shark rays cow	C1
D4	leopard goat lizard turtle tiger	C2
D5	goat snack lizard turtle	C2
D6	snack lizard turtle tiger toad	C2
D7	lizard turtle snack	C2
D8	deer leopard emu penguin parrot ostrich	C3
D9	deer ostrich emu penguin	C3
D10	deer bird ostrich emu penguin tiger	C3
D11	bird lizard emu penguin ostrich	C3
D12	deer emu penguin ostrich peacock cow	C3

Table 3 OLFSS Score on Synthetic data

Features	Global IG Score	Local Score of C1			Mean	Local Score of C2			Mean	Local Score of C3			Mean
		OR	GSS	CC		OR	GSS	CC		OR	GSS	CC	
"bird"	0.331	−5.408	−0.025	−0.447	−1.960	−5.525	−0.016	−0.632	−2.058	5.994	0.016	1.183	2.398
"cow"	0.256	1.376	0.025	0.447	0.616	−5.525	−0.016	−0.632	−2.058	0.403	0.002	0.169	0.191
"deer"	0.589	6.322	0.062	0.845	2.410	−6.775	−0.055	−1.673	−2.834	1.665	0.015	0.831	0.837
"emu"	1.011	−6.322	−0.062	−0.845	−2.410	−6.439	−0.039	−1.195	−2.558	6.909	0.040	2.236	3.062
"goat"	0.422	1.934	0.049	0.707	0.897	1.093	0.016	0.500	0.536	−6.350	−0.023	−1.336	−2.570
"hagfish"	0.635	6.504	0.074	1.342	2.640	−5.525	−0.016	−0.632	−2.058	−5.658	−0.011	−0.845	−2.172
"leopard"	0.210	−5.408	−0.025	−0.447	−1.960	0.841	0.008	0.316	0.388	0.403	0.002	0.169	0.191
"lizard"	0.748	−6.322	−0.062	−0.845	−2.410	6.775	0.055	1.673	2.834	−1.665	−0.015	−0.831	−0.837
"ostrich"	1.011	−6.322	−0.062	−0.845	−2.410	−6.439	−0.039	−1.195	−2.558	6.909	0.040	2.236	3.062
"penguin"	1.011	−6.322	−0.062	−0.845	−2.410	−6.439	−0.039	−1.195	−2.558	6.909	0.040	2.236	3.062
"ray"	0.635	6.504	0.074	1.342	2.640	−5.525	−0.016	−0.632	−2.058	−5.658	−0.011	−0.845	−2.172
"shark"	1.016	6.909	0.111	1.732	2.917	−5.930	−0.023	−0.816	−2.257	−6.063	−0.017	−1.091	−2.390
"snack"	0.692	−5.812	−0.037	−0.577	−2.142	6.621	0.047	1.633	2.767	−6.063	−0.017	−1.091	−2.390
"tiger"	0.311	−5.812	−0.037	−0.577	−2.142	1.932	0.023	0.816	0.924	−0.467	−0.003	−0.218	−0.230
"toad"	0.282	1.376	0.025	0.447	0.616	0.841	0.008	0.316	0.388	−5.658	−0.011	−0.845	−2.172
"turtle"	1.014	−6.099	−0.049	−0.707	−2.285	6.909	0.063	2.000	2.990	−6.350	−0.023	−1.336	−2.570

negative score is assigned. However, there is less numerical difference among OR scores of positive and negative nature terms for the classes. Also, the numerical range of scores assigned to the terms by the OR is higher than CC and GSS methods. Thus, the resultant sum of OR, CC, and GSS does not have much difference from the OR score, but their weighted average score is a more balanced score than OR. It helps in discrimination of positive and negative nature of terms for a class due to balanced numerical difference among scores of terms. The selection process of SVT, IGFSS, and GFSS based on IG is explained as follows:

The SVT follows the structure of IGFSS as shown in the Algorithm 2 to select the final feature set. The entire process for assignment of class labels to the features and the flow

Table 4 Class Label assignment on Synthetic data using SVT

Features	IG Score	C1 Mean	C2 Mean	C3 Mean	Class Label	Membership
"bird"	0.331	−1.960	−2.058	**2.398**	"C3"	'positive'
"cow"	0.256	0.616	**−2.058**	0.191	"C2"	'negative'
"deer"	0.589	2.410	**−2.834**	0.837	"C2"	'negative'
"emu"	1.011	−2.410	−2.558	**3.062**	"C3"	'positive'
"goat"	0.422	0.897	0.536	**−2.570**	"C3"	'negative'
"hagfish"	0.635	**2.640**	−2.058	−2.172	"C1"	'positive'
"leopard"	0.210	**−1.960**	0.388	0.191	"C1"	'negative'
"lizard"	0.748	−2.410	**2.834**	−0.837	"C2"	'positive'
"ostrich"	1.011	−2.410	−2.558	**3.062**	"C3"	'positive'
"penguin"	1.011	−2.410	−2.558	**3.062**	"C3	'positive'
"ray"	0.635	**2.640**	−2.058	−2.172	"C1"	'positive'
"shark"	1.016	**2.917**	−2.257	−2.390	"C1"	'positive'
"snack"	0.693	−2.142	**2.767**	−2.390	"C2"	'positive'
"tiger"	0.311	**−2.142**	0.924	−0.230	"C1"	'negative'
"toad"	0.283	0.616	0.388	**−2.172**	"C3"	'negative'
"turtle"	1.014	−2.285	**2.990**	−2.570	"C2"	'positive'

Table 5 Selected words from Synthetic data

Class→	C1		C2		C3	
	Positive	Negative	Positive	Negative	Positive	Negative
Total words count	3	2	3	2	4	2
Selected top 6 features count (SVT)	1	1	1	1	1	1
Selected top 6 features count (IGFSS)	0	2	0	2	0	2

of SVT is shown in the Algorithm 2. In SVT, (25)–(26) are used to compute the negative features ratio (nfr) and positive features ratio (pfr) for all three classes as follows:

1. $nfr[C1] = 2/(3+2) = 2/5 = 0.4$, $C1 = 1 - nfr = 1 - 0.4 = 0.6$
2. $nfr[C2] = 2/(3+2) = 2/5 = 0.4$, $C2 = 0.6$
3. $nfr[C3] = 2/(3+2) = 2/5 = 0.4$, $C3 = 0.6$

Therefore, using (27) the equal split criteria if the length of Final Feature Set (FFS) is 6 $Equal_{Split} = 6/3 = 2$. Further, using (28) the selected positive words count in class $C1 = Equal_{Split} \times nfr[C1] = 2 \times 0.6 = 1.2 \approx 1$, in class $C2 = 2 \times 0.6 = 1.2 \approx 1$, and in class $C3 = 2 \times 0.6 = 1.2 \approx 1$. Similarly, using (29) the selected negative words count in class $C1 = 2 \times 0.4 = 0.8 = 1$, in class $C2 = 2 \times 0.4 = 0.8 = 1$, and in class $C3 = 2 \times 0.4 = 0.8 = 1$.

Whereas, the IGFSS chooses a common nfr from the set of nfrs in the range of 0 to 1 that is based on the experimental evaluation, e. g. if we choose $nfr = 0.8$, then $pfr = 1 - 0.8 = 0.2$. The selected positive words count in class $C1 = 2 \times 0.2 = 0.4 \approx 0$, in class $C2 = 2 \times 0.2 = 0.4 \approx 0$, and in class $C3 = 2 \times 0.2 = 0.4 \approx 0$. Similarly, the selected negative words count in class $C1 = 2 \times 0.8 = 1.6 = 2$, selected negative words count in class $C2 = 2 \times 0.8 = 1.6 = 2$, and selected negative words count in class $C3 = 2 \times 0.8 = 1.6 = 2$. Thus, it can be observed from above discussions that SVT has used varying nfrs and pfrs instead of a common nfr and pfr which is chosen by the IGFSS. The SVT follows distribution of positive and negative nature terms in the classes while computing the nfr and pfr values. The distribution of words in the three classes C1, C2, and C3 using SVT and IGFSS are as follows:

1. C1- **pos:**'shark', 'hagfish', 'ray', **neg:** 'leopard', 'tiger', C2-pos:'turtle', 'lizard', 'snack', neg: 'cow', 'deer'.
2. C3- **pos:** 'penguin', 'ostrich', 'emu', 'bird', **neg:** 'goat', 'toad'.
3. IG selected top 6 features→ 'shark', 'turtle', 'emu', 'penguin', 'ostrich', 'lizard' (C1=1, C2=2, C3=3).
4. IGFSS selected top 6 features→**C1:** 'leopard', 'tiger', **C2:** 'cow', 'deer', and **C3:** 'goat', 'toad'.
5. SVT selected top 6 features→ **C1:** 'shark', 'tiger', **C2:** 'turtle', 'deer', **C3:** 'emu', 'goat'.

It has been observed from the results on synthetic data that the IGFSS has used one negative features ratio for all the classes. Whereas, the SVT has selected negative and positive features from each class based on distribution of features in the classes as negative or positive.

The negative nature of features has similar importance as the positive nature of features to discriminate the class label of a document. E.g., the two features 'leopard' and 'tiger' has been selected as negative features for class C1 by the IGFSS method, their presence in a document ensures that this document cannot be classified as class C1. These two features are present in the documents of class C2 and C3, but absent in the documents of class C1. Therefore, the two features 'leopard' and 'tiger' became negative features for class C1. The similar situation occurs for other negative features of this dataset. The GFSS based on IG has selected top 6 features, i.e. from class C1=1, C2=2, and C3=3.

As shown in Table 5, there are a less number of features from class C1 than C3 and C2. This issue has been resolved by choosing an equal number of features from each class. The assignment of the class labels of the features using IGFSS depends upon an individual vote of the OR method, whereas, the SVT has used the soft voting technique by computing the weighted average score of the features. Thus, SVT reduces the bias in feature selection towards single criteria by an ensemble of three methods method, viz. OR, GSS, and CC.

The stop words (e.g. "this", "that", "those", etc.), punctuation marks, white spaces, links, email addresses, numbers, etc. are less informative to decide the class label of the documents in ATDC. They are removed from the corpus in the pre-processing steps [1–3, 22, 44] with function PREPROCESSING(D) as follows:

function Preprocessing(D){

1. $T = [t_1, t_2, ..., t_p] \leftarrow Tokenizer(D)$ // Tokenization
2. T = stopWordsRemoval(T)
3. T = punctuationMarksRemoval(T)
4. T = whiteSpaceRemoval(T)
5. T = lemmatize(T) // convert the word to its root form (e.g. went, gone–> go)
6. $T \leftarrow [t_1, t_2, ..., t_m]$ // Where m < p
7. **return** (T) }

Algorithm 2 Algorithm for Improved Global Feature Selection Scheme (IGFSS) using Soft Voting Technique.

Input: A set D of documents $D = [d_1, d_2, ..., d_L]$, where $L > 0$ is the total number of documents in the corpus such that each gth document $d_g \in C_j$. Where C_j is a jth class of $C = [C_1, C_2, ...Cj]$, where, $0 < j <= r$ *(r is the total number of classes)*. The feature selection schemes, FSS = {MI, IG, GI, DFS, and GR} and the One Sided Local Feature Selection Scheme, i.e. OLFSS={OR,CC,GSS}. In this Algorithm 2 the subscript notations i and j are used to represent the terms and class respectively.

Output: A final feature set FFS of most discriminating features ($t[k] \subset t[m] \subset t[p]$), used as a vocabulary.

Method:

1: Split the corpus D into training and test set, $D = D_{train} + D_{test}$, where D_{train} is the training, and D_{test} is the test corpus.

2: call function PREPROCESSING (D)

3: Apply TF-IDF Vectorizer on D_{train} and D_{test}

4: Generate the Vocabulary (V) of TF-IDF Vectorized terms of the D_{train}, $V = \{t_1, t_2, \ldots, t_i\}\ \forall i = 1, 2, .., m$

5: Apply the methods of FSS, i.e. MI (1), IG (2), GI (3), DFS (4), GR (5) to compute the score of the terms t_i, $\forall i = 1, 2.., m, \forall j = 1, 2, .., k, .., r$

$$FSS\_Score(t_i) = FSS\left(t_i, C_j\right) \tag{20}$$

6: Arrange the Global feature set (GFS) in descending order based on their Score using (20).

7: Use the weighted average score of OR, CC, GSS methods to compute the local score and to find out the class label of each term $t_i \subset GFS$.

$$OLFSS_{(t_i,C_j)} = \sum_{j=1}^{j=r} \frac{M_k(t_i,C_j)}{3}, \quad Where\ M_k = [OR, CC, GSS],\ and\ k = 1, 2, 3 \tag{21}$$

8: Compute the positive or negative membership of each term t_i towards the obtained Class_Label C_k

$$Membership\left(t_i, C_k\right) = \begin{cases} pos, if\ (OLFSS\,(t_i, C_k)) \geq 0 \\ neg, if\ (OLFSS\,(t_i, C_k)) < 0 \end{cases} \tag{22}$$

9: Compute total count of positive and negative features in each class C_j ***if*** $Class\_Label(t_i) = C_j$ **AND** $Membership\left(t_i, C_j\right) = pos$

$$count_{pos}\left(C_j\right) = \sum_{i=1}^{m} Class\_Label(t_i) \tag{23}$$

else

$$count_{neg}\left(C_j\right) = \sum_{i=1}^{m} Class\_Label(t_i) \tag{24}$$

end if

10: Let the length of the final feature set (FFS) is N

11: Compute the negative features ratio (nfr) and positive features ratio (pfr) for each class C_j and store them into a set of negative features ratio (nfrs) and positive features ratio (pfrs) defined as, $nfrs = [x : x \subset nfr[j]]$, $pfrs = [x : x \subset (1 - nfr[j])]$

$$nfr[j] = \frac{count_{neg}(C_j)}{count_{pos}(C_j) + count_{neg}(C_j)}, \quad where 0 \leq nfr[j] \geq 1 \tag{25}$$

$$pfr[j] = 1 - nfr[j] \tag{26}$$

12: Compute an equal split criteria to select an equal number of features from each class C_j

$$Equal_{Split}\left(C_j\right) = \frac{N}{Total\ number\ of\ categories} \tag{27}$$

13: Compute the selection criteria of positive and negative features in each class C_j

$$Equal\_Pos_{Split}\left(C_j\right) = Equal_{Split}\left(C_j\right) \times pfr[j], \quad where\ pfr[j] \subset pfrs \tag{28}$$

$$Equal\_Neg_{Split}\left(C_j\right) = Equal_{Split}\left(C_j\right) \times nfr[j], \quad where\ nfr[j] \subset nfrs \tag{29}$$

14: Select the top-N number of features using (28)–(30) and keep them into FFS.

15: Compute selection criteria to add the new features **if** $length(FFS) < N$ **then**

$$N_l = (N - length(FFS)) \tag{30}$$

end if

16: Add the top-N_l number of new features into FFS, based on their scores obtained using (20).

The steps of SVT can be summarized as follows,

1. The SVT computes the weighted average score (Soft Vote) of the three methods, i.e. OR, CC, and GSS to find out the final score of the features. Further, it is used to determine the class label of the features. In this regard, it computes the negative value of a feature for the class, if the presence of that feature is very less or none in that class. Similarly, a positive value of a feature for the class, if it occurs most frequently in the class. Further, the SVT uses the maximum absolute score of the feature for a class to assign the class label and the sign of the maximum value to find out the membership of the feature.

2. The features are sorted in descending order using the scores obtained using any one method of GFSS at a time. Further, the negative and positive features ratios are derived using a mathematical model as shown in (26)–(30).

4 Experimental setup and performance evaluation

In order to evaluate the performance of SVT over IGFSS and various methods of GFSS (MI, IG, GI, DFS, and GR), all the experiments have been carried out on a machine with Intel core i7, 8GB RAM, 1.8 GHz Processor in UBUNTU 16.04 64-bit OS. The process of document classification-tokenization, preprocessing of the words of the corpus (D), feature extraction ($t[m] \subset t[p]$), feature selection ($t[k] \subset t[m]$), classification, and performance analysis are performed in Python 2.7 with nltk, scipy, numpy, ipython notebook, scikitlearn, matplotlib etc. packages.[3] To speed up the computing process and resolve the memory related issues the entire corpus is sliced into multiple arrays of each class, in spite of loading entire corpus into a single array. The number of features selected for analysis is in the range of 300, 400, and 500. The statistical tests have been performed using Java and KEEL software tool to evaluate the performance of the proposed SVT method with other compared methods using LSVM, SOFT MAX, SGDC, and RIDGE classifiers. The average rankings of the compared methods are also computed using the Java and KEEL software tool.[4]

4.1 Data set

In this study, five distinct standard text datasets (viz. Reuters10, Ohsumed10, Webkb, Classic4, and Trec2004) with varying characteristics were used for the assessment of the proposed technique (see Table 6). The brief description of these datasets is as follows: The *Reuters10 dataset* consists of top-10 classes of the Reuters-21578 dataset.[5,6] The *Ohsumed10 dataset* [35, 37] is a subset containing most frequent 10 categories of original Ohsumed23 dataset. The Ohsumed10 is highly dense, unbalanced and challenging dataset. The *Webkb dataset*[7] consists of four classes [14]. In the Webkb dataset, the "student" class has the most samples, whereas the "project" class has the least samples. The *Classic4 dataset*[8] and its class distribution is nearly homogeneous among four classes [29]. In Classic4 dataset the most of the samples are from class "cacm", whereas, class "med" has least number of samples. The *Trec2004 dataset* [13] is the original subset of MEDLINE for the TREC 2004 Genomics Track9. The Trec2004 dataset consists of 10 years of completed citations from the database includes from 1994 to 2003. The full text articles are extracted from the Pubmed database[9] in the form of XML file. These articles are based on four categories: mouse tumor biology (tumor), embryologic mouse gene expression (expression), mouse gene ontology (GO), and alleles of mutant mouse phenotypes (allele). The above four categories of documents are searched and saved in xml files. Subsequently, pubmed id, title, and abstract are parsed from relevant xml files using the R xml parser.[10]

The training and test documents are already defined in the Reuters10 dataset, whereas for other datasets, viz. Ohsumed10, Webkb, Classic4, and Trec2004 the stratified nested 5-fold cross-validation scheme is used to split the dataset in training and test sets. This cross-validation object is a variation of k-fold that returns stratified folds; each fold contains approximately the same percentage of samples of each target class as the complete set. Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole.

4.2 Classification algorithms

In order to prove the efficacy of the proposed technique four state-of-the-art classifiers viz. Linear Support Vector Machine (LSVM) [27, 33], SOFT MAX classifiers [8], Stochastic Gradient Descent Classifier (SGDC) [9], and RIDGE [38] are employed on text datasets. As the text classification problems tend to be quite high dimensional (many features) and the high dimensional problems are likely to be linearly separable. Therefore, the performance of the linear classifiers is likely, well if SOFT MAX, SGDC, RIDGE or LSVM is used with a linear kernel. However, to get good performance the regularization parameters need to be properly tuned. In the experiments, python scikit-learn[11] are used to classify the documents [17]. A brief description of the methods are as follows:

A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space which can be used for classification, regression or other tasks. The SVM is one of the most successful classifiers for text classification. SVM search for a decision boundary that is

[3] http://nbviewer.ipython.org/gist/rjweiss/7158866

[4] http://www.keel.es/

[5] https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

[6] https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[7] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[8] http://www.dataminingresearch.com/index.php/category/dataset/

[9] http://www.ncbi.nlm.nih.gov/pubmed/?term=mouse+gene+ontology

[10] https://cran.r-project.org/web/packages/XML/XML.pdf

[11] http://scikit-learn.org/stable/

Table 6 Details of the datasets

Dataset	Categories Name	# Class
Reuters10	earn, acq, money-fx, grain, crude, trade, interest, ship, wheat, corn	10
Ohsumed10	C01, C04, C06, C08, C10, C12, C14, C20, C21, C23	10
Webkb	course, faculty, project, student	4
Classic4	cacm, cisi, cran, med	4
Trec2004	allele, expression, GO, tumor	4

maximally far away from any data point. The distance from the decision surface to the closest data point determines the margin of the classifier. The SVM classifier is based on the margin maximization concept [27, 33].

Softmax Regression (synonyms: Multinomial Logistic, Maximum Entropy Classifier, or just Multi-class Logistic Regression) is a generalization of logistic regression that we can use for multi-class classification (under the assumption that the classes are mutually exclusive). In contrast, we use the (standard) Logistic Regression model in binary classification tasks.[12] SVM methods required less variables than Logistic Regression to achieve a better (or equivalent) performance. The sigmoid logistic function is replaced by the softmax function in this classifier [8].

Stochastic Gradient Descent (SGD) is a simple and very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines. The SGD has been successfully applied to the large-scale and sparse machine learning problems often encountered in the text classification and natural language processing [9].

The ridge classifier uses ridge regression with L2 Regularization technique to classify the data. The stochastic average gradient descent solver used in ridge regression classifier to speed up the iterative procedure of the classification. The Ridge regression is simply a type of linear regression that controls the magnitude of the coefficients to mitigate the effect of over-fitting. The major strength of this classifier is that there is no need for feature selection if the model is properly tuned using the regularization parameter [38].

4.3 Performance evaluation measures

In this paper, the benchmarked macro and micro averaged F1 measures [44] are used to evaluate the performance of classifiers. The F-measure (F_β and F_1) can be interpreted as a weighted harmonic mean of the precision and recall. The F_β score weights recall more than precision by a factor of beta. A F_β measure reaches its best value at 1 and its worst score at 0. With $\beta = 1$, F_β and F_1 are equivalent, and the recall and the precision are equally important.[13] The accuracy gives the same weight to all classes and it is not suitable for imbalanced datasets. The Macro_F1 measure computes metrics for each label, and find their unweighted mean and does not consider label imbalance. Whereas Micro_F1 calculate metrics globally by counting the total true positives, false negatives and false positives. In this context, the notions of precision (macro (31) and micro (32)), recall (macro (33) and micro (34)), accuracy (35), F_β (36), Macro_F1 (37), and Micro_F1 (38) measures are as follows:

$$Precision_{macro} = \frac{1}{n(C)} \sum_{C=1}^{C=r} \frac{TP_C}{TP_C + FP_C} \tag{31}$$

$$Precision_{micro} = \frac{\sum_{C=1}^{C=r} TP_C}{\sum_{C=1}^{C=r} TP_C + \sum_{C=1}^{C=r} FP_C} \tag{32}$$

$$Recall_{macro} = \frac{1}{n(C)} \sum_{C=1}^{C=r} \frac{TP_C}{TP_C + FN_C} \tag{33}$$

$$Recall_{micro} = \frac{\sum_{C=1}^{C=r} TP_C}{\sum_{C=1}^{C=r} TP_C + \sum_{C=1}^{C=r} FN_C} \tag{34}$$

$$accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \tag{35}$$

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{36}$$

$$Macro\_F1 = 2 \times \frac{Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{37}$$

$$Micro\_F1 = 2 \times \frac{Precision_{micro} \times Recall_{micro}}{Precision_{micro} + Recall_{micro}} \tag{38}$$

Where $C = 1$ to $C = r$ represent r class labels and $n(C)$ is the count of the total number of classes. The TP is the count of true positives, FP is the count of false positives, FN is the count of false negatives, TN is the count of true negatives. In

[12] http://www.kdnuggets.com/2016/07/softmax-regression-related-logistic-regression.html

[13] http://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics

the multi class and multi label classification task, the notions of precision, recall, and F-measures can be applied to each class label independently.

The Z-test statistics [21] has been used to evaluate the performance of VGFSS over IGFSS and the methods of GFSS. A set of pairwise comparisons can be associated with a set or family of hypotheses. As [15] explained, the test statistics for comparing the i^{th} and j^{th} classifier is,

$$z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}} \tag{39}$$

where R_i and R_j is the average rank computed through the Friedman test for the i^{th} and j^{th} classifier, k is the number of classifiers to be compared and N is the number of data sets used in comparison. The z value is used to find the corresponding probability (p-value) from the table of the normal distribution, which is further compared with an appropriate level of significance α. There are two basic procedures for doing that [31]:

1. **Holm's Procedure** [25]**:** The value of α is adjusted using a step-down method. Let $p_1, ..., p_m$ be the ordered p-values arranged in ascending order and $H_1, ..., H_m$ be the comparable hypotheses. The Holm's procedure rejects the hypothesis from H_1 to $H_{(i-1)}$, if $p_i > \alpha/(m - i + 1)$, where i is the smallest integer.

   The set of all pairwise comparisons builds a group of a logically interrelated hypothesis. If there are three hypotheses of pairwise equality associated with the pairwise comparisons of three classifiers C_i, where $i = 1, 2, 3$. Therefore, the true and false hypothesis of all the comparisons are not possible and if any one of these hypotheses is false, then at least one other must be false. For example, if C_1 is better or worse than C_2, then it is not possible that C_1 has the same performance as C_3 and there cannot be one false and two true hypotheses among these three relations at the same time. Shaffer [40] proposed two procedures to address this issue and used a logical relation between the family of hypotheses to adjust the value of α.
2. **Shaffer's static procedure:** Similar to Holm's step down method, at stage j, instead of rejecting H_i if $p_i \leq \alpha/(m - i + 1)$, reject H_i if $p_i \leq \alpha/t_i$, where t_i is the maximum number of hypotheses which can be true given that any $(i - 1)$ hypotheses are false. **Shaffer's dynamic procedure:** It is an improvement of static procedure and uses the value α/t_i* in place of α/t_i at stage i, where t_i* is the maximum number of hypotheses that could be true, given that the previous hypotheses are false. It is a dynamic procedure since t_i* depends not only on the logical structure of the hypotheses but also on the hypotheses already rejected at step i.

The presented results of this study are based on a static procedure. However, the hypothesis can be examined using other advanced [26] dynamic procedure on the presented experimental results.

5 Results and discussions

In this section, the experimental results of the proposed SVT algorithm and its comparison with classical information science methods and IGFSS are presented. The results are analyzed based on the selected features (viz. 300, 400, and 500) and the performance of the four classifiers. The dispersed distribution of the features in the various classes of all five datasets can be observed from Figs. 1, 2, 3, 4 and 5. The distribution of features in the classes is based on the class labels assigned by the IGFSS and SVT. The classifier results based on Macro_F1 and Micro_F1 measures are shown in the Tables 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16. The maximum performance measures achieved by the algorithms are shown in bold letters in these tables.

The average rankings of the algorithms GFSS (MI, IG, GI, DFS, and GR), GFSS + IGFSS, and GFSS + SVT is shown in the Table 18. In this table, the Micro_F1 based average rank of MI, MI+IGFSS, and MI+VGFSS are 2.942, 2.025, and 1.033 respectively. Here, the highest value of average rank of the algorithm means the last performer,

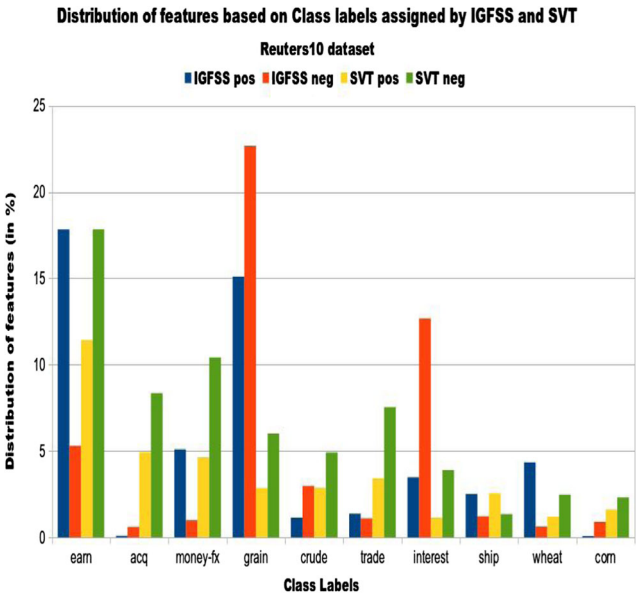

Fig. 1 Reuters10 dataset

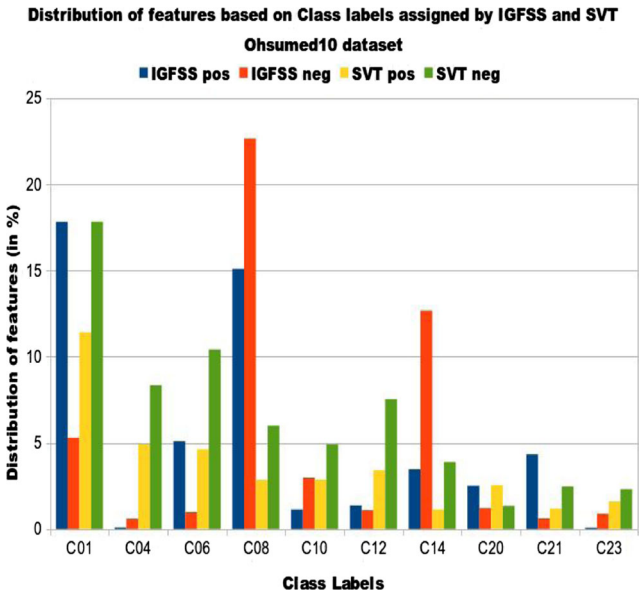

Fig. 2 Ohsumed10 dataset

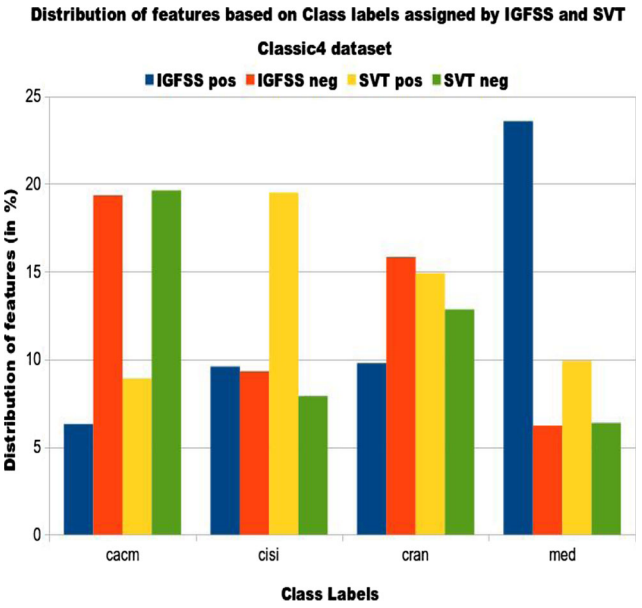

Fig. 4 Classic4 dataset

while a minimum average rank value means top performer. The GFSS+SVT is found to be the top performer in all four classifiers (as shown in bold letters).

5.1 Data and statistical analysis

As can be observed from Figs. 1–5 that due to the selection of two principal components using LSA the total number of features selected using SVT has been reduced (≈ up to 1000 features) in comparison to IGFSS for each dataset. The LSA assigns an equal weight to the words (e.g. "shark", "hagfish", and "rays") representing similar concepts or meanings (category of "fish") and it doesn't matter which word is selected by the algorithm in the final feature set. The IGFSS selects one common negative features ratio (nfr) for each class while the positive and negative features have been selected based on that nfr value. It affects those classes of a dataset which have more positive features than negative or vice-versa. The SVT solves this issue by selecting a variable negative and positive features ratio for each class based on the count of negative and positive features in the classes. Thus, SVT has selected an equal number of features from each class, but a variable number of positive and negative

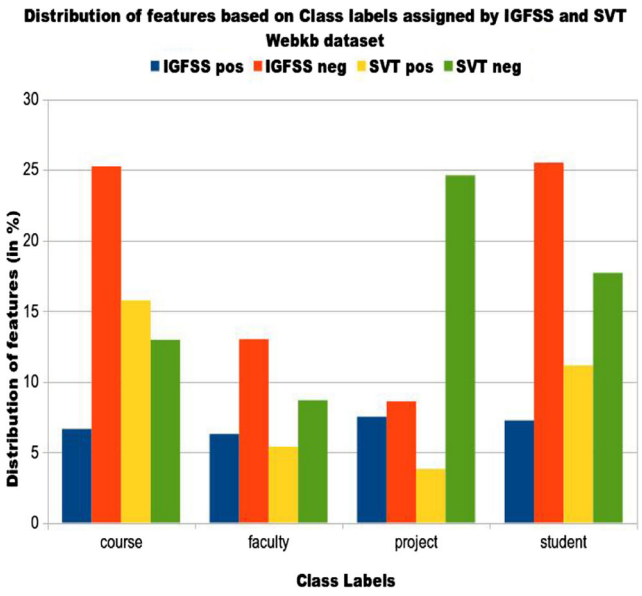

Fig. 3 Webkb dataset

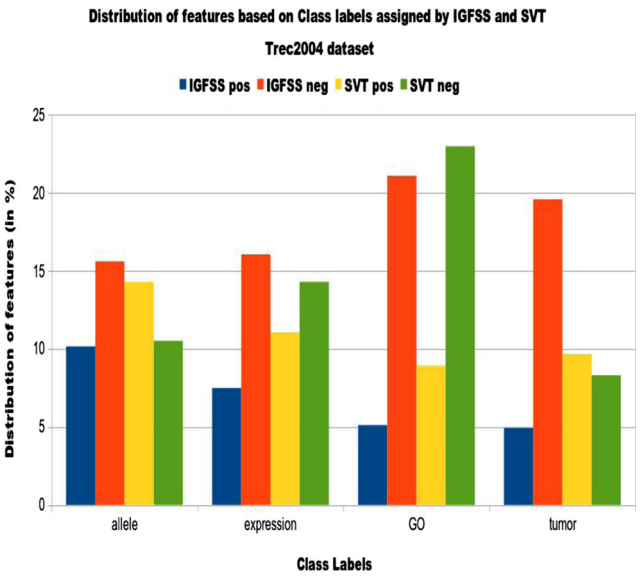

Fig. 5 Trec2004 dataset

Table 7 Macro_F1 measure for Ohsumed10 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	46.66	48.57	**51.63**	48.73	49.81	**51.5**	46.25	47.83	**50.87**	49.8	50.48	**51.63**	49.51	49.81	**51.59**
SOFT MAX (300)	46.11	47.89	**51.15**	48.36	49.04	**50.98**	46.01	47.26	**50.12**	48.82	49.74	**51.15**	49.02	49.04	**51.09**
SGDC (300)	44.26	44.71	**49.06**	47.25	45.91	**49.11**	42.48	43.64	**48.42**	47.78	47.19	**49.06**	48.63	46.52	**49.01**
RIDGE (300)	47.4	48.65	**51.69**	49.02	49.64	**51.51**	47.49	48.34	**51.15**	50.18	50.21	**51.69**	49.88	49.64	**51.55**
LSVM (400)	47.73	49.54	**52.51**	49.28	50.45	**52.1**	47.43	48.76	**51.56**	50.3	51.11	**52.51**	49.6	50.45	**52.15**
SOFT MAX (400)	46.9	48.80	**51.84**	49.28	49.88	**51.43**	46.62	48.09	**50.69**	49.62	50.44	**51.84**	49.41	49.88	**51.48**
SGDC (400)	45.85	46.01	**50.46**	48.49	47.59	**50**	44.52	44.17	**49.05**	49.08	47.83	**50.46**	47.6	47.77	**49.94**
RIDGE (400)	48.77	49.70	**52.53**	50.66	50.52	**52.11**	48.72	49.16	**51.84**	50.9	51	**52.53**	50.83	50.51	**52.21**
LSVM (500)	48	50.05	**52.85**	49.72	51.03	**52.41**	47.61	49.64	**52.01**	50.29	51.8	**52.85**	49.95	51.03	**52.57**
SOFT MAX (500)	47.12	49.34	**52.25**	49.27	50.23	**51.95**	47.05	48.76	**51.15**	49.72	51.11	**52.25**	49.48	50.23	**51.98**
SGDC (500)	46.4	46.54	**51.27**	49.32	48.18	**50.57**	44.13	45.23	**50.17**	48.29	49.19	**51.27**	49.15	48.25	**50.74**
RIDGE (500)	49.5	50.48	**53.16**	50.95	51.25	**52.69**	49.46	49.85	**52.44**	51.17	51.81	**53.16**	51.07	51.25	**52.74**

Table 8 Macro_F1 measure for Reuters10 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	62.93	64.84	**66.09**	64.93	65.06	**66.9**	62.95	63.88	**64.25**	62.49	63.54	**65.05**	64.47	65.97	**66.12**
SOFT MAX (300)	67.48	67.54	**68.56**	67.45	67.96	**68.69**	68.46	68.87	**69**	68.74	68.33	**68.84**	68.84	68.96	**69.88**
SGDC (300)	67.52	67.62	**68.51**	67.79	67.92	**68.9**	67.3	67.49	**68.28**	67.24	68.69	**68.84**	67.15	68.01	**69.58**
RIDGE (300)	67.19	67.82	**69.78**	68.56	68.14	**68.63**	68.87	69.36	**69.04**	67.63	68.71	**68.79**	68.73	68.78	**69.87**
LSVM (400)	63.15	64.8	**65.87**	63.92	64.33	**66.85**	63.66	63.49	**64.26**	63.74	63.76	**64.45**	63.48	65.41	**67.9**
SOFT MAX (400)	67.9	68.33	**67.61**	67.98	68.23	**68.75**	67.73	68.47	**68.62**	68.81	68.86	**68.95**	67.79	68.38	**69.12**
SGDC (400)	67.75	67.84	**68.49**	67.82	67.95	**68.38**	62.18	65.97	**68.3**	67.48	67.52	**68.84**	65.92	68.43	**69.21**
RIDGE (400)	68.29	69.19	**69.64**	68.98	69.51	**69.72**	68.09	69.53	**68.71**	68.27	68.6	**68.9**	67.78	68.79	**68.92**
LSVM (500)	63.01	63.05	**67.48**	62.73	63.23	**67.6**	62.69	63.8	**64.09**	64.72	63.64	**66.89**	62.2	63.09	**67.87**
SOFT MAX (500)	67.76	67.93	**68.48**	68.02	67.24	**68.49**	66.98	67.11	**67.33**	68.44	68.55	**68.82**	68.08	68.16	**69.7**
SGDC (500)	67.59	67.89	**68.77**	65.36	66.57	**67.78**	62.27	67.52	**68.02**	67.97	68.88	**68.9**	67.56	66.21	**68.33**
RIDGE (500)	68.12	68.61	**69.13**	68	68.12	**68.95**	66.98	68.18	**68.38**	68.25	68.45	**68.61**	67.79	67.9	**69.11**

Table 9 Macro_F1 measure for Webkb dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	89.52	90.67	**91.67**	89.04	92.06	**92.53**	88.27	91.49	**91.62**	89.07	92.23	**92.49**	89.52	91.89	**92.11**
SOFT MAX (300)	90	91.08	**92.34**	90.31	91.84	**93.45**	89.12	91.53	**91.34**	90.68	92.19	**93.28**	90.6	92.04	**92.65**
SGDC (300)	90.21	91.85	**92.11**	90.57	91.66	**92.36**	90.97	91.99	**91.68**	90.38	91.73	**93.42**	89.91	91.77	**91.46**
RIDGE (300)	90.67	91.39	**91.37**	90.97	90.99	**92.34**	90.38	91.27	**91.85**	90.28	91.71	**93.22**	89.51	91.2	**91.62**
LSVM (400)	91.1	91.73	**91.87**	90.86	92.32	**92.87**	89.64	91.53	**91.78**	91.4	92.11	**93.3**	90.1	92.47	**91.42**
SOFT MAX (400)	91.78	92.06	**92.71**	91.13	92	**93.17**	90.38	91.91	**92.57**	91.84	92.15	**93.08**	91.62	92.04	**92.68**
SGDC (400)	91.29	92.04	**92.89**	91.93	91.84	**93.22**	90.7	91.43	**91.71**	91.65	91.62	**92.6**	90.25	91.91	**92.21**
RIDGE (400)	91.56	91.79	**92.67**	91.64	91.59	**93.15**	91.88	91.33	**92.87**	92.08	92.57	**93.46**	90.56	91.82	**91.62**
LSVM (500)	91.07	92.61	**93.17**	90.42	92.62	**92.16**	89.64	91.78	**91.78**	91.21	92.19	**93.18**	91.67	92.54	**93.84**
SOFT MAX (500)	91.98	92.28	**93.23**	91.49	92.13	**93.03**	91.96	91.94	**93.03**	91.94	92.2	**93.94**	91.94	92.29	**93.02**
SGDC (500)	91.8	92.68	**93.2**	91.61	92.83	**92.35**	91.71	91.94	**93.17**	91.6	92.16	**93.29**	92.91	92.47	**93.13**
RIDGE (500)	91.25	92.2	**93.13**	92.26	92.41	**93.49**	91.57	91.64	**93.06**	91.4	92.09	**93.23**	91.42	92.41	**93.37**

Table 10 Macro_F1 measure for Classic4 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	93.37	94.86	**94.92**	93.41	94.86	**95.11**	92.72	94.83	**95.37**	93.2	94.95	**95.23**	93.37	94.69	**95.08**
SOFT MAX (300)	94.95	94.96	**95.02**	94.7	94.96	**95.51**	94.11	94.93	**95.29**	94.99	95	**95.62**	94.95	94.98	**95.13**
SGDC (300)	94.89	94.5	**95.32**	94.07	94.46	**95.21**	94	93.97	**94.76**	94.14	94.54	**94.3**	94.98	94.82	**95.25**
RIDGE (300)	91.71	94.68	**94.96**	91.39	94.68	**95.12**	91.16	94.37	**95.63**	91.83	94.78	**95.92**	91.53	94.58	**95.13**
LSVM (400)	93.98	95.21	**95.58**	94.01	95.2	**95.3**	93.85	94.13	**95.1**	94.26	95.19	**95.69**	93.98	95.27	**95.97**
SOFT MAX (400)	94.84	95.3	**95.68**	94.5	95.29	**95.66**	94.2	94.34	**95.68**	94.36	95.23	**95.49**	94.84	95.51	**95.88**
SGDC (400)	95.11	95.65	**95.81**	94.43	94.55	**95.81**	93.9	94.6	**95.51**	94.97	94.79	**95.44**	94.57	94.67	**96.12**
RIDGE (400)	92.9	94.99	**95.43**	92.62	94.95	**95.03**	91.85	94.83	**95.15**	92.79	94.85	**95.11**	93.02	95.08	**95.53**
LSVM (500)	94.81	95.46	**95.83**	94.13	95.46	**95.74**	94.25	95.44	**95.7**	94.7	95.56	**95.87**	94.81	95.55	**96.07**
SOFT MAX (500)	94.96	95.51	**96.19**	94.77	95.51	**95.94**	94.92	95.66	**95.95**	94.95	95.63	**96.11**	94.96	95.44	**96.15**
SGDC (500)	94.25	94.91	**95.81**	94.68	94.49	**96.15**	94.79	94.66	**95.94**	94.8	94.88	**95.76**	94.6	94.85	**96.03**
RIDGE (500)	93.22	95.12	**96.03**	93.11	95.11	**95.66**	93.14	95.16	**95.17**	93.44	94.92	**95.33**	93.22	95.04	**95.74**

Table 11 Macro_F1 measure for Trec2004 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	85.24	87.19	**87.75**	84.53	86.82	**87.95**	82.43	85.62	**86.93**	84.37	86.89	**87.79**	85.24	86.92	**87.78**
SOFT MAX (300)	85.61	86.6	**87.28**	85.18	86.37	**86.9**	84.94	85.81	**86.55**	85.05	86.58	**87.65**	86.61	86.66	**87.28**
SGDC (300)	86.1	87.43	**88.15**	86.41	87.84	**87.9**	84.93	86.52	**86.74**	86.61	87.66	**88.51**	86.75	87.73	**88.28**
RIDGE (300)	87.34	87.42	**88.78**	87.39	87.53	**89.09**	87.51	87.23	**87.82**	87.16	87.45	**88.05**	87.43	87.82	**89.21**
LSVM (400)	84.27	87.49	**87.69**	83.48	86.99	**87.3**	82.96	86.06	**87.79**	84.21	87.19	**87.65**	84.27	86.79	**87.42**
SOFT MAX (400)	86.22	86.74	**87.89**	86.21	86.8	**86.43**	84.52	85.95	**86.36**	86.8	86.71	**87.14**	87.22	87.73	**87.89**
SGDC (400)	86.69	87.85	**88.1**	85.26	87.61	**87.69**	83.71	86.66	**87.14**	86.34	87.41	**87.8**	86.73	87.64	**88.62**
RIDGE (400)	87.97	88.1	**89.14**	87.97	88.04	**88.69**	87.32	87.37	**89.22**	87.63	88.13	**88.55**	87.07	87.97	**88.99**
LSVM (500)	83.58	87.39	**86.57**	83.78	87.03	**87.02**	83.52	85.79	**86.92**	84.38	87.13	**87.67**	83.58	87.24	**87.92**
SOFT MAX (500)	86.76	86.94	**87.5**	86.04	87.09	**87.76**	85.03	85.96	**86.26**	86.58	86.96	**87.19**	86.76	87.03	**87.85**
SGDC (500)	85.07	86.04	**87.15**	87.18	87.27	**87.75**	83.82	85.82	**86.81**	85.26	87.53	**87.75**	87	87.84	**88.17**
RIDGE (500)	87.46	87.77	**88.99**	87.95	87.98	**88.46**	87.43	87.54	**87.9**	87.6	87.77	**88.59**	87.7	88.14	**88.69**

Table 12 Micro_F1 measure for Ohsumed10 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	47.39	49.13	**51.35**	49.09	50.12	**51.23**	47.24	48.36	**50.45**	49.91	50.55	**51.35**	49.61	50.12	**51.22**
SOFT MAX (300)	47.02	48.66	**51.07**	48.94	49.46	**50.96**	46.97	48.08	**49.99**	49.09	49.96	**51.07**	49.17	49.46	**50.99**
SGDC (300)	45.19	45.94	**50.22**	48.33	47.21	**50.25**	44.24	44.87	**49.68**	49.06	48.29	**50.22**	49.15	47.73	**50.09**
RIDGE (300)	48.81	49.71	**51.92**	50.2	50.42	**51.76**	48.93	49.4	**51.31**	50.83	50.81	**51.92**	50.58	50.42	**51.72**
LSVM (400)	48.47	49.96	**52.34**	49.75	50.74	**51.91**	48.16	49.05	**51.13**	50.36	51.13	**52.34**	49.82	50.74	**51.86**
SOFT MAX (400)	47.78	49.33	**51.85**	49.86	50.25	**51.46**	47.39	48.7	**50.58**	49.89	50.6	**51.85**	49.85	50.25	**51.4**
SGDC (400)	47.12	47.23	**51.47**	49.22	48.5	**51.1**	45.55	45.47	**50.19**	49.94	48.6	**51.47**	48.81	48.55	**50.95**
RIDGE (400)	49.9	50.49	**52.83**	51.48	51.3	**52.41**	50.13	49.98	**51.96**	51.54	51.49	**52.83**	51.58	51.3	**52.4**
LSVM (500)	48.81	50.48	**52.64**	50.18	51.29	**52.24**	48.32	49.95	**51.65**	50.39	51.75	**52.64**	50.23	51.29	**52.31**
SOFT MAX (500)	48.01	49.86	**52.13**	49.59	50.64	**51.93**	47.72	49.29	**50.98**	50.02	51.1	**52.13**	49.73	50.63	**51.82**
SGDC (500)	47.28	47.44	**52.12**	48.99	48.95	**51.58**	44.8	46.25	**51.01**	49.22	49.92	**52.12**	49.1	49.2	**51.65**
RIDGE (500)	50.72	51.26	**53.38**	51.75	51.86	**53**	50.5	50.66	**52.51**	51.79	52.16	**53.38**	51.72	51.88	**52.98**

Table 13 Micro_F1 measure of Reuters10 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	86.15	86.94	**87.05**	86.44	86.83	**87.2**	85.93	87.37	**87.67**	85.32	87.15	**87.69**	86.4	86.98	**87.37**
SOFT MAX (300)	86.33	87.01	**87.28**	86.4	87.26	**87.49**	86.47	87.08	**87.73**	85.4	87.12	**87.58**	86.54	87.26	**87.65**
SGDC (300)	86.37	86.37	**87.44**	86.04	86.72	**86.37**	85.47	86.76	**87.8**	84.54	86.47	**87.51**	85.32	86.44	**87.11**
RIDGE (300)	86.11	86.15	**87.15**	86.15	86.26	**85.93**	86.15	86.54	**87.61**	84.43	86.37	**87.58**	85.97	86.29	**87.68**
LSVM (400)	86.29	87.19	**87.3**	86.22	87.19	**87.41**	86.26	87.15	**87.44**	86.19	87.37	**87.51**	86.15	87.15	**87.54**
SOFT MAX (400)	86.51	87.48	**87.44**	86.65	87.48	**87.63**	86.54	87.26	**87.44**	86.58	87.44	**87.51**	86.44	87.55	**87.9**
SGDC (400)	85.4	87.05	**87.87**	85.68	86.83	**86.72**	85.9	86.9	**87.19**	86.15	86.72	**87.41**	85.29	86.83	**87.54**
RIDGE (400)	86.11	87.01	**87.9**	85.83	87.26	**86.76**	85.76	87.01	**87.64**	85.68	86.65	**87.12**	85.76	86.76	**87.51**
LSVM (500)	85.86	86.94	**87.23**	86.11	87.15	**87.3**	85.79	87.41	**87.61**	86.29	87.26	**87.26**	85.97	87.01	**87.37**
SOFT MAX (500)	86.33	87.23	**87.73**	86.26	87.12	**87.59**	86.08	87.23	**87.26**	86.54	87.08	**87.44**	86.44	87.15	**87.76**
SGDC (500)	85.61	86.51	**87.08**	85.07	87.12	**87.52**	85.61	86.87	**87.05**	85.97	86.72	**87.12**	85.86	86.9	**87.69**
RIDGE (500)	85.5	86.62	**87.3**	85.68	86.54	**87.12**	85.22	86.44	**87.01**	86.01	86.37	**87.4**	85.76	86.44	**87.91**

Table 14 Micro_F1 measure of Webkb dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	90.00	91.28	**92.29**	90.52	91.76	**92.89**	89.71	92.40	**92.48**	90.29	92.93	**92.86**	90.00	92.62	**92.57**
SOFT MAX (300)	90.62	91.95	**92.76**	91.24	91.76	**93.81**	90.95	92.31	**92.48**	91.43	92.93	**93.43**	91.33	92.86	**93.14**
SGDC (300)	91.10	91.99	**92.71**	91.67	91.62	**92.29**	91.81	92.78	**92.86**	91.38	92.67	**93.55**	91.57	92.66	**92.00**
RIDGE (300)	91.48	91.40	**92.10**	91.76	91.16	**92.76**	91.52	92.12	**92.86**	91.86	92.64	**93.46**	91.29	92.28	**92.19**
LSVM (400)	91.19	91.86	**92.57**	91.10	92.90	**93.33**	90.14	92.43	**92.86**	91.90	92.88	**93.45**	91.19	93.05	**92.29**
SOFT MAX (400)	91.52	92.66	**93.33**	91.71	92.66	**93.81**	91.33	92.59	**93.52**	92.81	92.93	**93.52**	91.43	92.67	**93.24**
SGDC (400)	92.05	92.90	**92.97**	91.75	92.69	**93.48**	91.00	92.43	**92.76**	92.05	92.55	**93.24**	91.86	92.74	**92.86**
RIDGE (400)	91.98	92.40	**92.89**	92.48	92.55	**93.86**	92.16	92.33	**93.62**	92.95	92.98	**93.63**	91.38	92.71	**92.38**
LSVM (500)	91.38	92.78	**93.29**	91.71	93.28	**92.67**	90.33	92.57	**92.67**	92.48	92.96	**93.33**	92.38	93.28	**93.86**
SOFT MAX (500)	92.48	93.00	**93.81**	92.29	92.83	**93.62**	92.00	92.64	**93.81**	92.67	93.02	**94.29**	92.48	92.97	**93.62**
SGDC (500)	92.67	92.96	**93.06**	92.86	93.57	**92.95**	92.16	92.74	**93.20**	92.57	93.02	**93.14**	92.92	92.68	**93.19**
RIDGE (500)	92.19	93.05	**93.16**	92.65	93.19	**93.55**	92.38	92.55	**93.33**	92.43	92.97	**93.26**	92.29	92.74	**93.54**

Table 15 Micro_F1 measure of Classic4 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	93.40	94.69	**94.81**	93.40	94.69	**95.02**	92.67	94.67	**95.41**	93.18	94.80	**95.14**	93.40	94.55	**95.16**
SOFT MAX (300)	94.87	94.83	**95.87**	94.64	94.83	**95.48**	94.08	94.80	**95.10**	94.81	94.87	**95.48**	94.87	94.84	**95.21**
SGDC (300)	94.81	94.47	**95.04**	94.14	94.38	**95.34**	94.08	93.91	**94.70**	94.08	94.53	**94.36**	94.93	94.96	**95.63**
RIDGE (300)	91.71	94.53	**95.36**	91.43	94.53	**95.14**	91.15	94.21	**95.57**	91.66	94.60	**95.80**	91.54	94.42	**95.29**
LSVM (400)	93.91	94.97	**95.38**	93.97	94.95	**95.04**	93.80	94.93	**95.59**	94.14	94.97	**95.81**	93.91	95.02	**95.72**
SOFT MAX (400)	94.76	95.07	**95.55**	94.48	95.07	**95.55**	94.02	94.15	**95.43**	94.15	95.00	**95.26**	94.76	95.29	**95.72**
SGDC (400)	95.04	95.52	**95.76**	94.25	94.40	**95.66**	93.80	94.56	**95.26**	94.76	94.69	**95.32**	94.42	94.50	**95.94**
RIDGE (400)	92.78	94.77	**95.32**	92.56	94.73	**95.19**	91.77	94.60	**95.36**	92.73	94.60	**95.76**	92.90	94.84	**95.38**
LSVM (500)	94.70	95.22	**95.97**	94.08	95.22	**95.55**	94.14	95.21	**95.43**	94.48	95.35	**95.72**	94.70	95.31	**95.89**
SOFT MAX (500)	94.76	95.31	**96.00**	94.64	95.31	**95.83**	94.81	95.48	**95.99**	94.81	95.43	**95.94**	94.76	95.24	**96.00**
SGDC (500)	94.02	94.74	**95.60**	94.48	94.32	**96.00**	94.64	94.49	**95.72**	94.53	94.77	**95.60**	94.31	94.71	**95.83**
RIDGE (500)	93.12	94.87	**95.83**	93.01	94.87	**95.38**	93.07	94.93	**95.81**	93.24	94.87	**95.04**	93.12	94.80	**95.55**

Table 16 Micro_F1 measure of Trec2004 dataset

Classifier (features)	MI			IG			GI			DFS			GR		
	MI	IGFSS	SVT	IG	IGFSS	SVT	GI	IGFSS	SVT	DFS	IGFSS	SVT	GR	IGFSS	SVT
LSVM (300)	85.56	87.30	**87.76**	84.85	86.90	**87.96**	82.85	85.67	**86.06**	84.75	86.95	**87.86**	85.56	87.00	**87.76**
SOFT MAX (300)	85.76	86.62	**87.26**	85.36	86.37	**86.86**	85.26	85.87	**86.56**	85.26	86.57	**87.66**	86.76	86.74	**87.36**
SGDC (300)	86.46	87.63	**88.26**	86.86	88.06	**87.96**	85.26	86.73	**87.06**	86.86	87.83	**88.67**	86.96	87.90	**88.37**
RIDGE (300)	87.26	87.63	**88.97**	87.37	87.75	**89.27**	87.46	87.45	**88.06**	87.16	87.65	**88.26**	87.37	88.06	**89.37**
LSVM (400)	84.35	87.50	**87.66**	83.75	87.03	**87.36**	83.35	86.12	**87.86**	84.45	87.15	**87.66**	84.35	86.85	**87.46**
SOFT MAX (400)	86.36	86.65	**87.76**	86.36	86.75	**86.36**	84.85	85.95	**86.36**	86.96	86.95	**87.06**	87.36	87.68	**87.96**
SGDC (400)	86.86	87.98	**88.26**	85.56	87.78	**87.86**	84.15	86.88	**87.36**	86.66	87.55	**87.96**	86.96	87.80	**88.77**
RIDGE (400)	87.97	88.28	**89.27**	87.97	88.23	**88.97**	87.36	87.55	**89.37**	87.67	88.28	**88.77**	87.07	88.16	**89.17**
LSVM (500)	83.85	87.35	**86.66**	84.05	87.03	**87.16**	83.95	85.80	**86.96**	84.55	87.08	**87.86**	83.85	87.23	**87.96**
SOFT MAX (500)	86.56	86.85	**87.46**	86.16	87.05	**87.66**	85.26	85.92	**86.16**	86.76	86.98	**87.16**	86.96	86.98	**87.76**
SGDC (500)	85.56	86.21	**87.36**	87.46	87.48	**87.86**	84.15	86.00	**86.16**	85.56	87.70	**87.96**	87.26	88.01	**88.46**
RIDGE (500)	87.47	87.93	**89.17**	87.97	88.16	**88.67**	87.47	87.53	**88.06**	87.57	87.95	**88.77**	87.77	88.28	**88.87**

Table 17 Holm / Shaffer values

i	algorithms	$z = (R_0 - R_i)/SE$	p	Holm	Shaffer
	Micro_F1 based				
3	MI vs. MI+SVT	10.45	1.43E-25	0.033	0.033
2	MI+IGFSS vs. MI+SVT	5.43	5.6E-8	0.05	0.1
1	MI vs. MI+IGFSS	5.02	5.2E-7	0.1	0.1
3	IG vs. IG+SVT	9.31	1.3E-20	0.033	0.033
2	IG vs. IG+IGFSS	4.8	1.7E-6	0.05	0.1
1	IG+IGFSS vs. IG+SVT	4.52	6.2E-6	0.1	0.1
3	GI vs. GI+SVT	10.5	8.82E-26	0.033	0.033
2	GI+IGFSS vs. GI+SVT	5.93	2.96E-9	0.05	0.1
1	GI vs. GI+IGFSS	4.56	5.01E-6	0.1	0.1
3	DFS vs. DFS+SVT	10.18	2.47E-24	0.033	0.033
2	DFS+IGFSS vs. DFS+SVT	5.57	2.57E-8	0.05	0.1
1	DFS vs. DFS+IGFSS	4.61	4.023E-6	0.1	0.1
3	GR vs. GR+SVT	9.86	6.27E-23	0.033	0.033
2	GR+IGFSS vs. GR+SVT	5.2	1.96E-7	0.05	0.1
1	GR vs. GR+IGFSS	4.67	3.23E-6	0.1	0.1
	Macro_F1 based				
3	MI vs. MI+SVT	10.41	2.31E-25	0.033	0.033
2	MI vs. MI+IGFSS	5.48	4.32E-8	0.05	0.1
1	MI+IGFSS vs. MI+SVT	4.93	8.24E-7	0.1	0.1
3	IG vs. IG+SVT	9.77	1.55E-22	0.033	0.033
2	IG+IGFSS vs. IG+SVT	5.57	2.57E-8	0.05	0.1
1	IG vs. IG+IGFSS	4.2	2.7E-5	0.1	0.1
3	GI vs. GI+SVT	9.91	3.97E-23	0.033	0.033
2	GI+IGFSS vs. GI+SVT	5.3	1.19E-7	0.05	0.1
1	GI vs. GI+IGFSS	4.61	4.03E-6	0.1	0.1
3	DFS vs. DFS+SVT	10.22	1.6E-24	0.033	0.033
2	DFS+IGFSS vs. DFS+SVT	5.93	2.96E-9	0.05	0.1
1	DFS vs. DFS+IGFSS	4.29	1.8E-5	0.1	0.1
3	GR vs. GR+SVT	10.04	1.0E-23	0.033	0.033
2	GR+IGFSS vs. GR+SVT	5.57	2.57E-8	0.05	0.1
1	GR vs. GR+IGFSS	4.47	7.71E-6	0.1	0.1

features (see the Figs. 1–5). For example, in the Reuters10 dataset, there is less percentage of positive features for "acq" class selected by the IGFSS, whereas the SVT selects a balanced percentage of positive and negative features not only for the "acq" class but also for all classes.

Similar cases can be observed for all other datasets, as shown in the Figs. 2–5. However, the changes are not as effective in case of a balanced dataset (e.g. Webkb, Classic4, and Trec2004) in comparison to an unbalanced dataset (e.g. Reuters10, Ohsumed10). It is due to the almost similar distribution of samples as well as terms in the classes of the balanced dataset. The distribution of samples and terms are variable in the classes of an unbalanced dataset, therefore one negative feature ratio, which is determined empirically by the IGFSS, lacks in selecting the most appropriate negative and positive features from all classes. This issue is solved using SVT because it selects positive and negative features using a set of negative features ratio (i.e. $nfrs$) derived using an improved mathematical model. It selects the positive and negative features based on their distribution in the class.

The statistical tests based on Z-test statistics [21] is shown in the Table 17. They illustrate the Holm/Shaffer values of compared methods for four classifiers. The compared methods are shown as algorithms in these tables. There are total 3 hypotheses formed for every five methods (MI, IG, GI, DFS, and GR) and these hypotheses are denoted as i in the tables. The value of α has been selected as 0.05. The Holm's and Shaffer's procedure rejects those hypotheses that have a p-value $\leq$ 0.033. The average ranking of the algorithms has been presented in Table 18 which is prepared by these values.

Table 18 Average rankings of the algorithms

Algorithm	Micro_F1 Rank	Macro_F1 Rank
MI	2.94	2.97
MI+IGFSS	2.03	1.97
MI+SVT	**1.03**	**1.07**
IG	2.86	2.85
IG+IGFSS	1.98	2.08
IG+SVT	**1.16**	**1.07**
GI	2.92	2.9
GI+IGFSS	2.08	2.042
GI+SVT	**1**	**1.08**
DFS	2.9	2.9
DFS+IGFSS	2.06	2.1
DFS+SVT	**1.04**	**1.02**
GR	2.9	2.9
GR+IGFSS	2.03	2.01
GR+SVT	**1.08**	**1.05**

5.2 Discussions

In order to compare the performances, two null hypothesis is assumed, First: "The performance of MI, IG, GI, DFS, and GR methods is equal to IGFSS, and SVT", and Second: "The performance of IGFSS and SVT is equal". In most of the cases, the performance of the MI, IG, GI, DFS, and GR methods are lower than IGFSS and SVT. The performance of the classifiers is significantly improved in comparison to IGFSS when the GFSS based methods are an ensemble with the SVT algorithm. Thus, both the first and second null hypothesis are rejected due to the lower values of α, Holm, and Shaffer from the standard.

The standard MI method assigns higher scores to the low-frequency terms that appear in only one class. It proves that it has the capability to discriminate terms which are present in a specific class, i.e. the terms with a positive nature. However, MI suffers in case of overlapping terms which are identified as negative nature terms in this paper. Thus, the experimental results obtained by MI shows good scores for those datasets that have more positive nature of terms in comparison of negative (e.g. Ohsumed10, trec2005). Other methods, viz. IG, DFS, GI, etc. which assign higher weights to most frequent negative nature terms have performed better in those datasets which have more negative nature terms than positive (e.g. Webkb, Classic4).

From extensive experimental study and statistical analysis, it is found that the performance of all the feature selection methods has been improved by embedding the SVT algorithm. The key points which are the main cause of the success of SVT over IGFSS and other classical methods of GFSS are as follows:

1. **Strength of GFSS:** (a). It selects top-N scored most representative features from all the classes. **Weakness of GFSS:** (a). It discards low scored features from some classes either partially or completely, due to dispersed distribution of the features in the classes.
2. **Strength of IGFSS:** (a). It selects an equal number of most representative features from all the classes. **Weakness of IGFSS:** (a). The OR has its own weakness to assign the adequate class label to the features. If a feature (say, t_o) is present more frequently in all the classes, but absent in any specific class (say C_l) then the presence of this feature in a test sample (say, $D_{test}[i]$) assures that the class label of $D_{test}[i]$ is not class C_l. In this case, as the feature t_o is present in all most all the classes except C_l, the OR method assigns a very high positive value of it to all the classes but a lesser negative value for the class C_l. The IGFSS fails in this situation to assign the most appropriate class label and the membership value of t_o. These types of terms are defined as common negative terms. (b). The negative features ratio is determined empirically, therefore, the selected positive and negative features are not adequate for all the classes.
3. **Strength of SVT:** (a). It uses the ensemble votes of three methods which gives better results than an individual vote and determines the most appropriate class label of the features. (b). The negative features ratio is determined using a mathematical model, therefore, the selected positive and negative features are the most appropriate for all the classes. **Weakness of SVT:** Although, the weighted average score of three methods (i.e. OR, CC, and GSS) balances the weaknesses of OR but not much effective to decide the class label and membership of the common negative features due to their similar numeric scoring nature, e.g. in case of term t_o.

6 Conclusions

The main contribution of this study is to introduce a new Soft Voting Technique (SVT) for determination of most appropriate class labels of the features. SVT has provided a generic solution for all filter-based global feature selection methods to select the most informative features based on assigned class labels. The proposed SVT technique has used the advantage of ensemble results obtained from the weighted average score (Soft Vote) of three methods, i.e. Odds Ratio (OR), Correlation Coefficient (CC), and GSS Coefficients (GSS) to predict the class labels of the features. This technique can be useful for a set of equally well-performing methods in order to balance out their individual weaknesses. Although, the SVT has selected an equal

number of features from each class similar to IGFSS, but the process for selection of positive and negative features count has followed an improved approach derived using a mathematical model. The use of Latent Semantic Analysis (LSA) at the initial level has reduced the high-dimensional feature space into a smaller one. The constructed final feature set has improved the scalability, efficiency, and accuracy of classifiers in all the five datasets used in this study and proved the efficacy of the proposed Soft Voting Technique. In the future, there is a need to find out some more appropriate methods for the selection of positive and negative features.

References

1. Agnihotri D, Verma K, Tripathi P (2014) Pattern and cluster mining on text data. In: Fourth international conference on communication systems and network technologies. IEEE Computer Society, CSNT, Bhopal, pp 428-432. https://doi.org/10.1109/CSNT.2014.92
2. Agnihotri D, Verma K, Tripathi P (2016) Computing correlative association of terms for automatic classification of text documents. In: Proceedings of the third international symposium on computer vision and the internet, ACM, pp 71–80
3. Agnihotri D, Verma K, Tripathi P (2016b) Computing symmetrical strength of n-grams: a two pass filtering approach in automatic classification of text documents. SPRINGERPLUS 5(942):1–29
4. Agnihotri D, Verma K, Tripathi P (2017) An automatic classification of text documents based on correlative association of words. J Intell Inform Syst. https://doi.org/10.1007/s10844-017-0482-3
5. Agnihotri D, Verma K, Tripathi P (2017) Mutual information using sample variance for text feature selection. In: Proceedings of the 3rd international conference on communication and information processing, ACM, New York, NY, USA, ICCIP '17, pp 39–44. https://doi.org/10.1145/3162957.3163054
6. Agnihotri D, Verma K, Tripathi P (2017) Variable global feature selection scheme for automatic classification of text documents. Expert Syst Appl 81:268–281. https://doi.org/10.1016/j.eswa.2017.03.057. http://www.sciencedirect.com/science/article/pii/S0957417417302208
7. Agnihotri D, Verma K, Tripathi P, Choudhary N (2018) A review of techniques to determine the optimal word score in text classification. In: Perez GM, Tiwari S, Trivedi MC, Mishra KK (eds) Ambient communications and computer systems. Springer, Singapore, pp 497–507
8. Alejandro SD, VAJIA N, Carlos SJ (2012) Comparison between svm and logistic regression: which one is better to discriminate? Revista Colombiana de EstadÃstica 35:223–237. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-17512012000200003&nrm=iso
9. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, Physica-Verlag HD, pp 177–186
10. Caulkins JP, Ding W, Duncan G, Krishnan R, Nyberg E (2006) A method for managing access to web pages: Filtering by statistical classification (fsc) applied to text. Decision Support Syst 42(1):144–161. https://doi.org/10.1016/j.dss.2004.11.015. http://www.sciencedirect.com/science/article/pii/S0167923604002635
11. Chan SW, Chong MW (2004) Unsupervised clustering for nontextual web document classification. Decision Support Syst 37(3):377–396. https://doi.org/10.1016/S0167-9236(03)00035-6. http://www.sciencedirect.com/science/article/pii/S0167923603000356
12. Chen Y, Zhang H, Liu R, Ye Z, Lin J (2018) Experimental explorations on short text topic mining between lda and nmf based schemes. Knowledge-Based Syst. https://doi.org/10.1016/j.knosys.2018.08.011. http://www.sciencedirect.com/science/article/pii/S0950705118304076
13. Cohen AM, Hersh WR (2006) The trec 2004 genomics track categorization task: classifying full text biomedical documents. J Biomed Discov Collab 1(1):4. https://doi.org/10.1186/1747-5333-1-4
14. Craven M, McCallum A, PiPasquo D, Mitchell T, Freitag D (1998) Learning to extract symbolic knowledge from the world wide web. Tech. Rep. No. CMU-CS-98-122, Carnegie-Mellon University Pittsburgh pa School of Computer Science
15. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30. http://dl.acm.org/citation.cfm?id=1248547.1248548
16. Du M, Chen XS (2013) Accelerated k-nearest neighbors algorithm based on principal component analysis for text categorization. J Zhejiang University Sci C 14(6):407–416. https://doi.org/10.1631/jzus.C1200303
17. Fabian P, Gaël V, Alexandre G, Vincent M, Bertrand T, Olivier G, Mathieu B, Peter P, Ron W, Vincent D, Jake V, Alexandre P, David C, Matthieu B, Matthieu P, Édouard D (2011) Scikit-learn: Machine learning in python. J Mach Learn Res 12:2825–2830. http://dl.acm.org/citation.cfm?id=1953048.2078195
18. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
19. Forman G (2004) A pitfall and solution in multi-class feature selection for text classification. In: Proceedings of the twenty-first international conference on machine learning, ACM, New York, NY, USA, ICML '04, pp 38–. https://doi.org/10.1145/1015330.1015356
20. Galavotti L, Sebastiani F, Simi M (2000) Experiments on the use of feature selection and negative evidence in automated text categorization, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 1923. Springer, Berlin, pp 59–68
21. García S, Herrera F (2008) An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. J Mach Learn Res 9:2677–2694
22. García S, Luengo J, Herrera F (2015) Data preprocessing in data mining. Springer, Berlin
23. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182. http://dl.acm.org/citation.cfm?id=944919.944968
24. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier, Amsterdam
25. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70
26. Hommel G, Bernhard G (1999) Bonferroni procedures for logically related hypotheses. J Statist Plann Inference 82:119–128
27. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In European conference on machine learning (pp. 137–142). Springer, Berlin, Heidelberg
28. Kamal N, Kachites MA, Sebastian T, Tom M (2000) Text classification from labeled and unlabeled documents using em. Mach Learn 39(2):103–134. https://doi.org/10.1023/A:1007692713085
29. Kevin B, Moshe L (2013) Uci machine learning repository. http://archiveicsuciedu/ml901

30. Li XM, Ouyang JH, Lu Y (2015) Topic modeling for large-scale text data. Frontiers Inform Technol Electron Eng 16(6):457–465. https://doi.org/10.1631/FITEE.1400352
31. Luengoán J, García S, Francisco H (2009) A study on the use of statistical tests for experimentation with neural networks: analysis of parametric test conditions and non-parametric tests. Expert Syst Appl 36(4):7798–7808
32. Luis T (2005) An Evaluation of Filter and Wrapper Methods for Feature Selection in Categorical Clustering. Springer, Berlin, pp 440–451. https://doi.org/10.1007/11552253_40
33. Manning CD, Raghavan P, Schutze H (2008) Introduction to information retrieval. Cambridge University Press, New York
34. Mladenic D, Grobelnik M (1999) Feature selection for unbalanced class distribution and naive bayes. In: Proceeding of the 16th international conference on machine learning, San Francisco, SF, pp 258–267
35. Moschitti A, Basili R (2004) Ohsumed medical corpus dataset. http://disi.unitn.it/moschitti/corpora.htm
36. Ng HT, Goh WB, Low KL (1997) Feature selection, perceptron learning, and a usability case study for text categorization. SIGIR Forum 31(SI):67–73. https://doi.org/10.1145/278459.258537
37. Nist T (2001) Ohsumed medical corpus dataset. http://trec.nist.gov/data/t9_filtering.html
38. Rohit P, Devansh A, Shuang W, Premkumar N, Pradeep N (2013) Ridge regression based classifiers for large scale class imbalanced datasets. In: Proceedings of the 2013 IEEE workshop on applications of computer vision (WACV), IEEE Computer Society, Washington, DC, USA, WACV '13, pp 267–274. https://doi.org/10.1109/WACV.2013.6475028
39. Sebastiani F (2002) Machine learning in automated text classification. ACM Comput Surv 34(1):1–47
40. Shaffer JP (1986) Modified sequentially rejective multiple test procedures. J Am Stat Assoc 81(395):826–831
41. Singh BK, Verma K, Thoke A, Suri JS (2017) Risk stratification of 2d ultrasound-based breast lesions using hybrid feature selection in machine learning paradigm. Measurement 105:146–157. https://doi.org/10.1016/j.measurement.2017.01.016. http://www.sciencedirect.com/science/article/pii/S026322411730026X
42. Song D, Lau RY, Bruza PD, Wong KF, Chen DY (2007) An intelligent information agent for document title classification and filtering in document-intensive domains. Decision Support Syst 44(1):251–265. https://doi.org/10.1016/j.dss.2007.04.001. http://www.sciencedirect.com/science/article/pii/S0167923607000681
43. Tellez ES, Moctezuma D, Miranda-Jiménez S, Graff M (2018) An automated text categorization framework based on hyperparameter optimization. Knowledge-Based Syst 149:110–123. https://doi.org/10.1016/j.knosys.2018.03.003. http://www.sciencedirect.com/science/article/pii/S0950705118301217
44. Uysal AK (2016) An improved global feature selection scheme for text classification. Expert Syst Appl 43:82–92. https://doi.org/10.1016/j.eswa.2015.08.050. http://www.sciencedirect.com/science/article/pii/S0957417415006077
45. Uysal AK, Gunal S (2012) A novel probabilistic feature selection method for text classification. Knowledge-based Syst, Elsevier 36:226–235
46. Van RCJ (1979) Information retrieval, 2nd edn. Butterworth-Heinemann, Newton
47. Bk W, Yf H, Wx Y, Li X (2012) Short text classification based on strong feature thesaurus. J Zhejiang University Sci C 13(9):649–659. https://doi.org/10.1631/jzus.C1100373
48. Wang D, Zhang H, Liu R, Lv W, Wang D (2014) t-test feature selection approach based on term frequency for text categorization. Pattern Recogn Lett 45:1–10
49. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of the fourteenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '97, pp 412–420. http://dl.acm.org/citation.cfm?id=645526.657137
50. Zheng Z, Srihari R (2003) Optimally combining positive and negative features for text categorization. In: ICML 2003 Workshop, Washington DC, USA

Deepak Agnihotri has received Ph.D. degree in Computer Applications discipline from National Institute of Technology Raipur Chhattisgarh INDIA in January 2018. He has qualified ICAR ARS NET(I)-2018. He has completed Master of Computer Applications degree from Dr. Hari Singh Gaur University Sagar India in year 2007. Earlier, he served National Institute of Technology Raipur Chhattisgarh INDIA as Lecturer from October 2007 to May 2015. He had also worked at National Institute of Technology Raipur, on the project, which has funding support by the Ministry of Electronics and Information Technology, Government of India; Grant number "12(1)/2017-CSRD". He is currently working as a faculty in the Computer Applications Department of National Institute of Technology Raipur Chhattisgarh INDIA. His broad area of research includes machine learning, text mining, jpeg forensics, and data analysis.

Kesari Verma has received Ph.D. degree in Computer Science from Pt. RSU Raipur, India in 2007. She is currently working as Associate Professor in the Department of Computer Applications, National Institute of Technology Raipur, India. She has around 15 years of teaching and research experience. Her research interests include digital image processing and analysis, data mining, pattern classification, bio-metrics, machine learning, etc.

Priyanka Tripathi has received Ph.D. degree in Web Engineering from Maulana Azad National Institute of Technology, Bhopal, India in 2009. She is currently working as Assistant Professor in the Department of Computer Applications, National Institute of Technology Raipur, Chhattisgarh, India. She has around 15 years of teaching and research experience as well as 2 years of industrial experiences. Her research interest includes web engineering, ERP, neural network & fuzzy logic, data mining and software engineering.

Bikesh Kumar Singh has received Ph.D. degree in Biomedical Engineering from National Institute of Technology, Raipur, India in 2015. He is currently working as Assistant Professor in the Department of Biomedical Engineering, National Institute of Technology Raipur, India. He has around 12 years of teaching and research experience. Her research interests include medical image processing and analysis, data mining, pattern classification, biomedical signal processing, machine learning, etc. He has published more than 50 papers in various journals and conferences of repute.

Affiliations

Deepak Agnihotri[1] · Kesari Verma[1] · Priyanka Tripathi[1] · Bikesh Kumar Singh[2]

Kesari Verma
kverma.mca@nitrr.ac.in

Priyanka Tripathi
ptripathi.mca@nitrr.ac.in

Bikesh Kumar Singh
bsingh.bme@nitrr.ac.in

[1] Department of Computer Applications, National Institute of Technology, Raipur, CG, India

[2] Department of Biomedical Engineering, National Institute of Technology, Raipur, CG, India