A Comprehensive Filter Feature Selection for Improving Document Classification

Le Nguyen Hoai Nam
School of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
lnhnam@fit.hcmus.edu.vn

Ho Bao Quoc
School of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
hbquoc@fit.hcmus.edu.vn

Abstract

High dimension of bag-of-words vectors poses a serious challenge from sparse data, overfitting, irrelevant features to document classification. Filter feature selection is one of effective methods for dimensionality reduction by removing irrelevant features from feature set. This paper focuses on two main problems of filter feature selection which are the feature score computation and the imbalance in the feature selection performance between categories. We propose a novel filter feature selection method, named ExFCFS, to comprehensively resolve these problems. We experiment on related filter feature selection methods with two benchmark datasets - Reuters-21578 dataset and Ohsumed dataset. The experimental results show the effectiveness of our solutions in terms of both Micro-F1 measure and Macro-F1 measure.

Keywords— bag-of-words vector, filter feature selection, document classification

1 Introduction

Document classification is to assign documents to predefined categories based on their text contents (Sebastiani 2002). It is a useful tool for managing the organization of a large set of documents. In the document classification, a bag-of-words vector is usually used for presenting a document (Yang et al. 2012), (Joachims 1996). Concretely, a document is shown in the form of a vector in which each term appearing in the document is considered as a feature.

However, with a large set of documents, the dimension of a bag-of-words vector can reach thousands (Fragoudis et al. 2005), (Yang et al. 2012). Therefore, it poses a serious challenge from sparse data, overfitting, irrelevant features to document classification (Fragoudis et al. 2005), (Sebastiani 2002). In (Bellman 1961), the author referred it to as "the curse of dimensionality". Thus, dimensionality reduction is a major research area.

The aim of dimensionality reduction is to decrease the number of features without degrading the performance of the system (Sebastiani 2002). An efficient approach for dimension reduction is Feature Selection (FS) (Yang and Pedersen 1997). Feature selection eliminates irrelevant features to select a good subset of the original feature set. A strong point of FS is that the interpretation of the important features in the original set is not altered in dimensionality reduction process.

Two main types of FS are wrapper methods (Bermejo et al. 2014) and filter methods (Yang and Pedersen 1997). Wrapper methods select a subset of features which is the most suitable with a specific classification algorithm. Conversely, filter methods do not depend on any classification algorithms. It relies on a function for evaluating the importance of a feature in the classification process. A subset of features is selected by simply ranking the value of every feature on the evaluation function. Therefore, it is commonly used in document classification (Fragoudis et al. 2005), (Yang et al. 2012).

29th Pacific Asia Conference on Language, Information and Computation pages 169 - 177
Shanghai, China, October 30 - November 1, 2015

In this paper, we focus on filter feature selection methods. Table 1 shows their general structure.

```
Input: Bag-of-words vectors; $L$: the number of selected features.
Output: $S_L$: a subset of features with predefined size $L$
Step 1: For each term $t_k$ ($k = 1 \ldots |T|$)
Step 2:     For each category $C_i$ ($i = 1 \ldots |C|$)
Step 3:         Compute the importance of term $t_k$ for the prediction of category $C_i$: $catScore(t_k, C_i)$.
Step 4:     End for
Step 5:     Compute global score of term $t_k$ for the prediction of all categories from $catScores$ of term $t_k$: $globalScore(t_k)$.
Step 6 : End for
Step 7: Select L terms from top L highest $globalScores$: $S_L$.
```

Table 1: The general structure of filter FS methods

Specifically, filter feature selection methods compute the importance of term t_k for the prediction of category C_i, noted by $catScore(t_k, C_i)$. Then, the importance of term t_k for the prediction of all categories, noted by $globalScore(t_k)$, is calculated by using the average or maximum value of category-specific scores of term t_k over the different categories (Yang and Pedersen 1997). The terms from top highest $globalScore$ are selected to the final set. Next, we present main methods for computing $catScore(t_k, C_i)$ as following:

1.1 Information Gain

The basic idea of Information Gain (Quinlan 1986) is to measure predictable bits of category value if we know in advance the occurrence of a term. With IG, the score of term t_k with respect to a specific category C_i is as following:

$$catScore_{IG}(t_k, C_i) = \sum_{C \in \{C_i, \overline{C_i}\}} \sum_{t \in \{t_k, \overline{t_k}\}} P(t, C) \log \frac{P(t,C)}{P(t).P(C)}$$

where $P(t, C)$ is the probability of a document belonging to category C and containing term t; $P(\mathrm{C})$ is the probability of a document belonging to category C; $P(t)$ is the probability of a document containing term t. However, it is impossible to determine a set of features whose IG is maximal. It is NP problem (Yan et al. 2005). Therefore, IG formula is applied for each feature and the final set consists of the features from the top global scores. With this greedy characteristic, Information Gain is a non-optimal method (Yan et al. 2005).

1.2 Chi-square

Similar to IG, Chi-square (Yang and Pedersen 1997) (CHI) is a greedy algorithm. It measures the independence of category value and feature value. The formula of CHI is as following:

$$catScore_{CHI}(t_k, C_i) = \frac{n.P(t_k)^2.(P(C_i|t_k) - P(C_i))^2}{P(t_k).(1 - P(t_k)).P(C_i)(1 - P(C_i))}$$

Where n is the number of documents, $|C|$ is the number of categories, $P(t_k)$ is the probability of a document containing term t_k , $P(C_i)$ is the probability of a document belonging to category C_i, $P(C_i|t_k)$ is the conditional probabilities of a document belonging to category C_i given that it contains term t_k.

1.3 Frequency-based approach

This approach only focuses on the term-category frequency matrix for computing $catScore(t_k, C_i)$ as Document Frequency (DF) (Yang and Pedersen 1997), DIA association factor (DIF) (Sebastiani 2002), Comprehensively Measure Feature Selection (CMFS) (Yang et al. 2012). In CMFS, term t_k is important in the prediction of category C_i if term t_k largely appears in category C_i and the frequency of term t_k in the training set focuses much on category C_i. Therefore, $catScore_{CMFS}(t_k, C_i)$ is computed as following:

$$catScore_{CMFS}(t_k, C_i) = P(t_k|C_i).P(C_i|t_k) \qquad (1)$$

Where $P(t_k|C_i)$ is the conditional probabilities of term t_k given that it occurred in category C_i ; $P(C_i|t_k)$ is the conditional probabilities of category C_i given the occurrence of term t_k. In $catScore_{CMFS}(t_k, C_i)$, $P(t_k|C_i)$ presents a intra-category condition for the frequency of terms in category C_i , while $P(C_i|t_k)$ indicates a inter-category condition related to the frequency of term t_k not only in category C_i but also in various categories.

1.4 Cluster-based approach

This approach aims at selecting a subset of features in order to optimize objective functions for clustering where each cluster is corresponding to a predefined document category. Orthogonal Centroid Feature Selection (OCFS) is a wellknown method of this approach (Yan et al. 2005). It optimizes the separation of categories (clusters) in the filter FS process. It is implemented into $global\,Score$ of a term as following:

$$globalScore_{OCFS}(t_k) = \sum_{i=1}^{|C|} \frac{n_{C_i}}{n}\left(m^{(t_k)} - m_{C_i}^{(t_k)}\right)^2 \quad (2)$$

Where n is the number of documents in training set; m is the mean vector of all documents in training set; n_{C_i} is the number of documents in category C_i; m_{C_i} is the mean vector of all documents in C_i; $m^{(t_k)}$ denotes the feature value of term t_k in global centroid vector m; $m_{C_i}^{(t_k)}$ denotes the feature value of term t_k in category centroid vector m_{C_i}

According to (Yang and Pedersen 1997), a way for computing the global score of term t_k for the category prediction, $globalScore(t_k)$, is the average of the category-specific scores of term t_k over the different categories as following:

$$globalScore(t_k) = \sum_{i=1}^{|C|} P(C_i)catScore(t_k, C_i) \quad (3)$$

From Eq. (2) and Eq. (3), $catScore_{OCFS}\ (t_k, C_i\ )$ can be presented as following:

$$catScore_{OCFS}\ (t_k, C_i\ ) = \left(m^{(t_k)} - m_{C_i}^{(t_k)}\right)^2 \quad (4)$$

2 Approach

In this section, we analyze two filter feature selection approaches which are the frequency-based approach and the cluster-based approach. Our aim is to point out their weak points and strong points to propose a filter feature selection method for improving the performance of document classification.

For the frequency-based approach, $catScore_{CMFS}(t_k, C_i)$ is a comprehensive combination of the frequency-based intra-category condition, which is $P(t_k|C_i)$, and the frequency-based inter-category condition, which is $P(C_i|t_k)$. Regarding the frequency-based inter-category condition, $P(C_i|t_k)$ is rewritten according to conditional probability theory as following:

$$P(C_i|t_k) = \frac{tf(t_k, C_i) + 1}{tf(t_k) + |C|}$$

Where $tf(t_k, C_i)$ is the frequency of term t_k in category C_i; $tf(t_k)$ is the frequency of term t_k in the training set; $|C|$ is the number of categories. For $P(C_i|t_k)$, the greatness of the proportion of the frequency of term t_k in category C_i to the frequency of term t_k in the other categories is utilized to present the contribution of term t_k for discriminating category C_i from the other categories. However, this is not really perfect because a term t_k almost never showed in category C_i but often appearing in the other categories is still useful for classifying a document into category C_i.

Therefore, an inter-category condition in $catScore(t_k, C_i)$ is presented more clearly under the view point of clustering. Concretely, this is the deviation of the representative of term t_k in category/cluster C_i, which is the centroid value of term t_k in C_i ($m_{C_i}^{(t_k)}$), to the representative of term t_k in the training set, which is the centroid value of term t_k in the training set ($m^{(t_k)}$) as shown in $catScore_{OCFS}\ (t_k, C_i\ )$. In the other hand, $catScore_{OCFS}\ (t_k, C_i\ )$ presents such a good inter-category condition but does not mention any conditions of term t_k for intra-category C_i. Therefore, according to the conclusion of CMFS (Yang et al. 2012), this is not good for a filter FS process.

Based on this observation, we propose a novel filter feature selection approach for the combination of the cluster-based inter-category condition, which is $catScore_{OCFS}\ (t_k, C_i\ )$ as Eq. (4), and the frequency-based intra-category condition, which is the first part of Eq. (1). The formula of FCFS is as following:

$$catScore_{FCFS}(t_k, C_i) = P(t_k|C_i).\left(m^{(t_k)} - m_{C_i}^{(t_k)}\right)^2 = \frac{tf(t_k, C_i) + 1}{tf(t, C_i) + |T|}.\left(m^{(t_k)} - m_{C_i}^{(t_k)}\right)^2$$

Where $tf(t, C_i)$ is the sum of the frequency of all terms in category C_i; $|T|$ is the number of terms in the bag-of-words vector.

Furthermore, FCFS does not consider the imbalance in the classification performance between categories after the filter feature selection process. This problem is caused by two factors.

Firstly, classification algorithms tend to focus on categories containing more training documents than the others. This is a big challenge of data mining field. Secondly, the computation of $catScore_{FCFS}(t_k, C_i)$ does not mention the separation degree of the category C_i from the others. Concretely, if the separation degree of category C_n is greater than that of category C_m from the other categories, presented terms of category C_n obviously have higher score compared with those of category C_m. Therefore, after the term score ranking, there are a large number of terms supporting category C_n to be selected into the final set, while it does not contain enough terms for classifying category C_m.

To solve this problem, we propose an Extended version of FCFS, named ExFCFS, with aim of strengthening the score of a term with respect to rare categories and poor separation categories, and weakening the score of a term with respect to abundant categories and great separation categories. Therefore, in ExFCFS, we modify $catScore_{FCFS}(t_k, C_i)$ in inverse proportion to the number of training document of category C_i (n_{C_i}) and the separation degree of category C_i from the other categories as following:

$$catScore_{ExFCFS}(t_k, C_i) = \frac{\frac{tf(t_k, C_i) + 1}{tf(t, C_i) + |T|} \cdot \left(m^{(t_k)} - m_{C_i}^{(t_k)}\right)^2}{n_{C_i} . catSep(C_i)}$$

Where $catSep(C_i)$ is the separation degree of category C_i from the other categories. According to (Friedman et al. 2001), (Chakraborti et al. 2007), (Howland and Park 2004), under the view point of clustering where each cluster is considered as a predefined document category, $catSep(C_i)$ is computed using the "within-cluster" (W) and "between-cluster" (B) factor of cluster (category) C_i as following:

$$catSep(C_i) = \frac{B(C_i)}{W(C_i)} = \frac{\left\|m_{C_i} - \mathrm{m}\right\|^2}{\frac{\sum_{j \in C_i} \left\|d_j - m_{C_i}\right\|^2}{n_{C_i}}}$$

To compute the importance of a term globally, the maximum value of the category-specific term scores of a term over the different categories is particularly useful according to (Aggawal and Zhai 2012):

$$globalScore(t_k) = \max_{i=1\ldots|C|} catScore(t_k, C_i) \quad (5)$$

Therefore, in this paper, we apply Eq. (5) for computing the global score of ExFCFS as following:

$$globalSocre_{ExFCFS}(t_k) = \max_{i=1\ldots|C|} \left\{ \frac{\frac{tf(t_k, C_i) + 1}{tf(t, C_i) + |T|} \cdot \left(m^{(t_k)} - m_{C_i}^{(t_k)}\right)^2}{n_{C_i} . catSep(C_i)} \right\}$$

For the feature selection, the final set consists of the terms from the top L highest global term scores where L is a predefined size of the selected feature set. The detail of ExFCFS is presented in Table 2.

```
Input: Bag-of-words vectors; $L$: the number of selected features
Output: $S_L$: the subset of features with the predefined size $L$
Step 1: For each category $C_i$ ($i = 1 \ldots |C|$)
    Step 2: Compute the sum of term frequency of all terms in category $C_i$: $tf(t, C_i)$.
    Step 3: Compute the centroid vector of all documents in category $C_i$: $m_{C_i}$.
Step 4: End for
Step 5: Compute the centroid vector of all documents: $m$.
Step 6: For each term $t_k$ ($k = 1 \ldots |T|$)
    Step 7: Get the value of term $t_k$ in global centroid vector $m$: $m^{(t_k)}$.
    Step 8: For each category $C_i$ ($i = 1 \ldots |C|$)
        Step 9: Get the value of term $t_k$ in category centroid vector $m_{C_i}$: $m_{C_i}^{(t_k)}$.
        Step 10: Compute the frequency of term $t_k$ in category $C_i$: $tf(t_k, C_i)$.
        Step 11: Get the number of training documents in category $C_i$: $n_{C_i}$.
        Step 12: Compute the score of term $t_k$ with category $C_i$ from $tf(t, C_i)$, $tf(t_k, C_i)$, $m^{(t_k)}$, $m_{C_i}^{(t_k)}$, $n_{C_i}$, $m_{C_i}$, $m$: $catScore_{ExFCFS}(t_k, C_i)$.
        Step 13: Compute the maximum of $catScore_{ExFCFS}(t_k, C_i)$: $globalScore_{ExFCFS}(t_k)$
    Step 14 : End for
Step 15: End for
Step 16: Select $L$ terms from the top $L$ highest $globalScore_{ExFCFS}$: $S_L$
```

Table 2: The description of ExFCFS

3 Experiment

3.1 Experimental steps

In the experiment, we compare the performance of the proposed filter FS method with that of related filter feature selection methods as CMFS (Yang et al. 2012), OCFS (Yan et al. 2005), IG (Quinlan 1986), CHI (Yang and Pedersen 1997). The experimental steps are as following:

- For preprocessing, stop words are removed by using a set of 659 stop words. The stemming process is executed with Porter Stemming algorithm (Porter 1997). For text representation, we use TF-IDF of every term as well as bag-of-words technique.
- The training bag-of-words vectors are reduced by a filter FS method. Then, they are used for building a leaning model using SVM classifier by SMO (Platt 1999) with default setting of WEKA tool (Hall et al. 2009).
- The testing bag-of-words vectors are created only based on the selected terms from the filter feature selection process. The classification system is evaluated on these bag-of-words vectors.

3.2 Dataset

In this paper, we use two benchmark datasets for evaluating the performance of filter feature selection methods. The first dataset is the top-10 categories of Reuters-21578 ModApte's split (Asuncion and Newman 2007). They consist of stories collected from the Reuters news. The second dataset is top-10 categories of medical abstracts of year 1991 from U.S National Library of Medicine, named Ohsumed collection. A standard training and testing split of Ohsumed collection is Joachim's split (Joachims 1998). The detailed description of these datasets is presented in Table 3-4.

3.3 Measure

Two standard measures for evaluating the performance for multi categories classification are Macro-F1 and Micro-F1 (Sebastiani 2002). Macro-F1 measure considers all categories equally including rare categories (Tascı and Güngör 2013). Concretely, Macro-F1 is computed as following:

$$P_{macro} = \frac{\sum_{i=0}^{|C|} P_i}{|C|} \quad R_{macro} = \frac{\sum_{i=0}^{|C|} R_i}{|C|}$$

$$F1_{macro} = \frac{2R_{macro}P_{macro}}{R_{macro} + P_{macro}}$$

Where P_i and R_i are precision and recall measure on category C_i, $|C|$ is the number of categories. Contrary to Macro-F1, Micro-F1 measure ignores the category discrimination. The Micro-F1 measure is computed globally as following:

$$P_{micro} = \frac{\sum_{i=0}^{|C|} TP_i}{\sum_{i=0}^{|C|}(TP_i + FP_i)} \quad R_{micro} = \frac{\sum_{i=0}^{|C|} TP_i}{\sum_{i=0}^{|C|}(TP_i + FN_i)}$$

$$F1_{micro} = \frac{2R_{micro}P_{micro}}{R_{micro} + P_{micro}}$$

To explicitly compare the performance of filter feature selection methods, (Gunal & Edizkan 2008) relies on the above measures to propose dimension reduction rate as following:

$$S = \frac{1}{k}\sum_{i=1}^{k} \frac{Dim_N}{Dim_i} R_i \qquad (10)$$

where k is the number of tests in the experiment, Dim_i is the number of selected features in i^{th} test, R_i is the accuracy measure in i^{th} test, and Dim_N is the maximum feature size which is tested.

Category	Train Docs	Test Docs
C01	423	506
C04	1163	1467
C06	588	632
C08	473	600
C10	621	941
C12	491	548
C14	1249	1301
C20	525	695
C21	546	717
C23	1799	777
The number of features in bag-of-words vector: 17756		

Table 3: The description of Ohsumed dataset

Category	Training Docs	Testing Docs
Corn	181	56
Wheat	212	71
Ship	197	89
Trade	369	117
Interest	347	131
Grain	433	149
money-fx	538	179
Crude	389	189
Acq	1650	719
Earn	2877	1087
The number of features in bag-of-words vector: 16684		

Table 4: The description of Reuters-21578dataset

3.4 Experimental Result and Discussion

Table 5-8 show the experimental results of the filter feature selection methods in our study. It can be noted from these tables as following:

- In terms of Macro-F1, the best filter selection methods are FCFS and ExFCFS. In comparison between them, ExFCFS products better result than FCFS.
- Regarding Micro-F1, ExFCFS attains the most favourable result. FCFS is often superior to IG, CHI, OCFS, CMFS, but at the large number of selected features, their differences are rather small.

An exact explanation for the goodness of FCFS and ExFCFS is the effective combination of the clustered-based inter-category condition and frequency-based intra-category condition in the computation of their term score. This lends support to the theory of CMFS (Yang et al. 2012).

To observe detailed performance of filter feature selection methods, we present F1-measure of each category with CMFS, IG, FCFS, and ExFCFS at 60 features in Fig. 1-2. Specifically, FCFS and ExFCFS show the effectiveness with rare categories as “Ship, Trade, Grain, Interest, Money-Fx, Crude” of Reuters-21578 dataset and “C01, C06, C08, C10, C12, C20, C21” of Ohsumed dataset in comparison with IG and CMFS. This occurs due to the reason that in case of IG, CMFS, the score of a term with respect to a category is based on the greatness of the frequency of a term in the entire category, while the frequency of a term in rare categories is very low. Conversely, FCFS and ExFCFS only use the centroid value of a term in every category and in the training set for term score computation. Therefore, they preliminarily improve the feature selection performance of rare categories.

Next, we consider the correlation between performance of FCFS and ExFCFS. ExFCFS is actually an extended version of FCFS for radically overcoming the imbalance of classification performance between categories after filter feature selection process. As analyzed in this paper, this problem is directly caused by the imbalance of the number of training documents between categories and the imbalance of the separation degree between categories. Therefore, in ExFCFS, we adjust FCFS score of a term with respect to a category in inverse proportion to these factors in order to improve the classification performance of rare categories and poor separation categories after filter feature selection process. Especially, both of these two factors are occurred in Reuters-21578 datset and Ohsumed dataset. Under these properties of two experimental datasets, the performance of ExFCFS is superior to that of FCFS. This accounts for the effectiveness of our adjustments in ExFCFS formula.

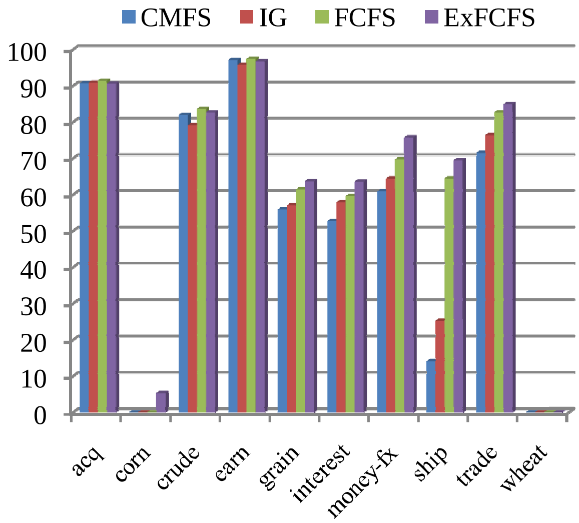


Fig. 1: F1-measure of CMFS, IG, FCFS, and ExFCFS on Reuter dataset at 60 features

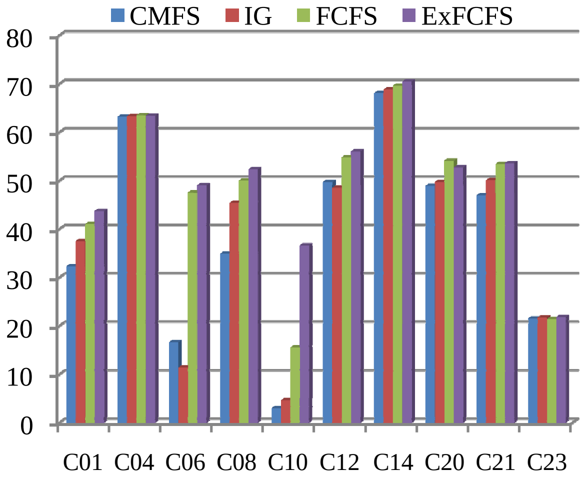
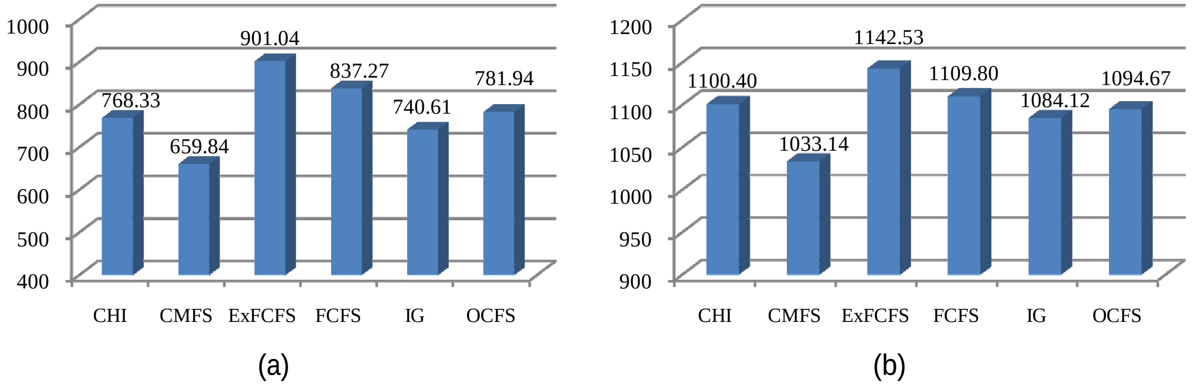


Fig. 2: F1-measure of CMFS, IG, FCFS, and ExFCFS on Ohsumed dataset at 60 features

Table 9 shows the performance of dissimilar terms and similar terms selected by filter FS methods. For the comparison between two FS methods, similar terms are terms selected by both of them, while dissimilar terms are terms selected by only one of them. Clearly, dissimilar terms are the most important for considering two FS methods. The result listed in Table 9 shows that at top-60 selected terms, dissimilar terms of FCFS are superior to those of CHI, IG, CMFS, and OCFS but is inferior to those of ExFCFS. This is one of strong evidences for the superiority of ExFCFS and FCFS over the other methods.

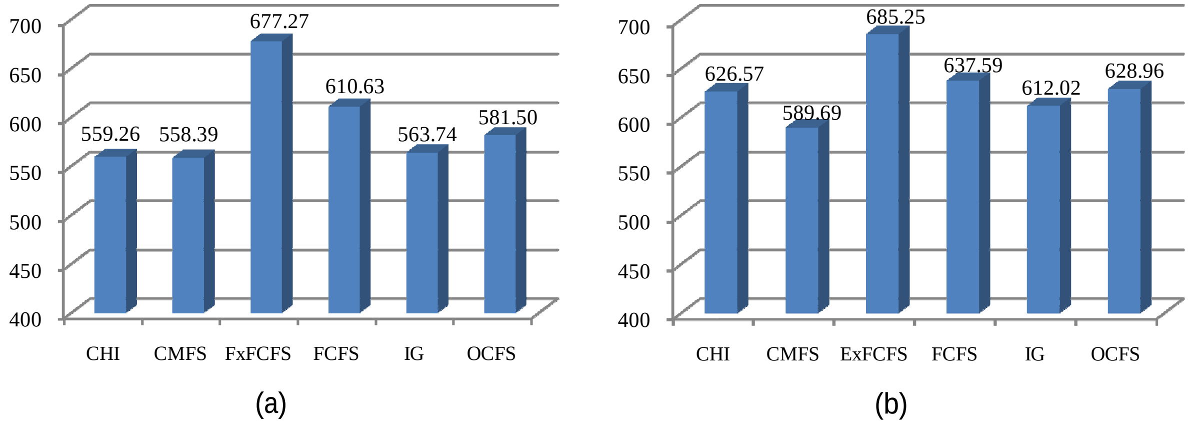
Regarding dimension reduction rate, due to the best Micro-F1 and Macro-F1 results of ExFCFS, it produces better dimension reduction rate than the other methods in all two datasets as shown in Fig. 3-4. FCFS is superior to CHI, IG, CMFS and OCFS at the small number of selected features and they show the competition at the larger number of features. However, based on dimension reduction rate formula presented in Eq. (10), FS methods having better performance at smaller number of selected features are preferred. Therefore, dimension reduction rate of FCFS is better than that of CHI, IG, CMFS, and OCFS as presented in Fig. 3-4.



(a)

(b)

Fig. 3. Dimension Reduction Rate on Reuters-21578 dataset: (a) for Macro-F1; (b) for Micro-F1



(a)

(b)

Fig. 4. Dimension Reduction Rate on Ohsumed dataset: (a) for Macro-F1; (b) for Micro-F1

FS	20	60	100	200	400	600	800	1000	1200	1400	1600	1800	2000
CHI	48.88	58.12	62.21	64.91	64.22	65.03	66.24	67.97	65.95	66.53	67.32	66.91	66.67
CMFS	35.82	55.83	61.3	63.44	64.56	65.92	67.53	66.01	65.85	67.78	67.53	66.43	66.46
ExFCFS	**58.89**	**67.08**	**73.83**	**72.57**	**73.35**	**70.72**	**71.53**	**71.15**	**71.75**	**71.64**	**71.86**	**71.67**	**71.18**
FCFS	**53.55**	**62.00**	**70.12**	**71.75**	**71.85**	**67.74**	**68.9**	**68.58**	**68.24**	**70.62**	**69.8**	**69.52**	**69.37**
IG	45.45	58.56	61.23	63.99	64.18	64.6	65.48	66.7	67.39	66.8	67.3	67.39	66.36
OCFS	49.66	60.00	63.02	64.43	67.36	66.65	66.97	67.13	67.86	67.18	66.95	66.88	66.87

Table 5: Macro-F1 result on Reuters-21578 dataset. Bold numbers are the top 2 performances

FS	20	60	100	200	400	600	800	1000	1200	1400	1600	1800	2000
CHI	72.95	81.93	86.13	86.73	87.14	87.51	88.05	**88.23**	87.73	87.69	87.76	87.55	87.48
CMFS	65.14	80.45	84.79	85.8	85.91	86.16	88.2	88.05	87.94	**88.27**	**88.05**	87.59	**87.69**
ExFCFS	**76.48**	**85.74**	**87.15**	**88.23**	**89.23**	**89.94**	**89.05**	**88.94**	**89.20**	**89.05**	**89.23**	**89.09**	**88.76**
FCFS	**73.12**	**84.1**	**87.06**	**87.82**	87.82	87.97	88.07	88.11	87.11	87.23	87	**87.89**	87.61
IG	70.88	81.98	85.89	86.41	87.09	87.87	88.02	87.94	87.98	87.66	87.69	87.8	87.33
OCFS	71.36	83.89	86.11	87.73	**88.30**	**88.05**	**88.23**	88.16	**88.2**	87.94	87.73	87.48	87.51

Table 6: Micro-F1 result on Reuters-21578 dataset. Bold numbers are the top 2 performances

FS	20	60	100	200	400	600	800	1000	1200	1400	1600	1800	2000
CHI	32.59	44.59	49.83	50.71	53.52	54.32	53.82	52.96	52.34	51.59	51.08	50.98	50.37
CMFS	33.72	43.08	47.4	49.69	51.76	51.96	52.65	52.44	52.74	52.5	52.27	51.91	51.66
ExFCFS	**43.93**	**51.66**	**53.33**	**54.33**	**56.40**	**56.75**	**56.15**	**56.51**	**55.97**	**55.02**	**54.79**	**54.36**	**54.29**
FCFS	**37.26**	**49.21**	**50.82**	**51.8**	**54.07**	**54.49**	54.2	**54.13**	53.47	53.28	**52.78**	**52.37**	52.23
IG	33.07	45.34	48.8	51.28	53.43	54.44	53.82	52.98	52.34	51.6	51.1	50.98	50.36
OCFS	34.53	46.68	49.8	51.88	53.77	54.2	**54.54**	54.03	**53.59**	**53.46**	52.65	52.02	**52.3**

Table 7: Macro-F1 result on Ohsumed dataset. Bold numbers are the top 2 performances

FS	20	60	100	200	400	600	800	1000	1200	1400	1600	1800	2000
CHI	39.69	**48.8**	**50.71**	**51.8**	52.88	**54.04**	53.43	52.79	52.44	51.94	51.37	51.45	50.88
CMFS	38.23	43.91	44.9	48.01	50.68	51.56	51.63	52.69	53.15	53.04	52.94	52.57	52.39
ExFCFS	**45.22**	**51.60**	**52.97**	**53.06**	**55.54**	**56.22**	**55.96**	**56.67**	**56.09**	**55.43**	**55.37**	**55.21**	**54.99**
FCFS	**41.35**	47.97	50.51	50.97	**53.2**	53.85	53.87	**54.24**	53.75	**54.76**	53.35	**53.19**	**52.9**
IG	39.79	44.31	48.78	50.65	52.99	**54.15**	53.42	52.81	52.43	51.93	51.39	51.45	50.87
OCFS	40.66	47.24	49.83	50.61	53.02	53.63	**54.39**	54.12	**53.97**	54.08	**53.4**	52.96	52.34

Table 8: Micro-F1 result on Ohsumed dataset. Bold numbers are the top 2 performances

DataSet	Measure	Type	CHI	CMFS	IG	OCFS	ExFCFS	Measure	Type	CHI	CMFS	IG	OCFS	ExFCFS
Reuters	Micro-F1	A	54.64	58.08	52.90	51.31	65.21	Macro-F1	A	31.42	18.92	30.57	18.60	32.47
		B	56.30	63.94	60.28	51.67	62.42		B	32.51	49.62	40.77	38.61	24.18
		C	80.98	78.70	80.93	82.96	83.07		C	57.31	55.27	58.01	59.31	60.51
Ohsumed	Micro-F1	A	19.62	16.69	14.42	10.92	23.93	Macro-F1	A	16.89	11.68	19.72	12.04	22.15
		B	12.26	20.15	19.90	12.98	17.58		B	20.78	23.54	21.37	15.99	18.82
		C	48.70	47.61	45.00	47.05	47.31		C	44.38	44.59	45.77	46.50	47.03

Table 9: Micro-F1 and Macro-F1result of similar terms and dissimilar terms selected by FCFS and the other FS methods at top-60 selected terms. A, B, and C indicate dissimilar terms of the corresponding FS, dissimilar terms of FCFS, and their similar terms respectively.

4 Conclusion

This paper propose a comprehensive filter FS method, named ExFCFS, for computing feature score and overcoming the imbalance of FS performance between categories. In ExFCFS, the feature score with respect to a specific category is the combination of the cluster-based inter-category condition and the frequency–based intra-category condition to exploit the strong point of two related approaches. Then, we adjust this combination in inverse proportion to the number of training document of the category and the separation degree of the category. The experimental results show the effectiveness of our solutions in terms of both Micro-F1 measure and Macro-F1 measure.

References

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In Mining text data (pp. 163-222). Springer US.

Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

Bermejo, P., Gámez, J. A., & Puerta, J. M. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. Knowledge-Based Systems, 55, 140-147.

Bellman, R., (1961). Adaptive control processes: a guided tour (Vol. 4). Princeton: Princeton university press.

Chakraborti, S., Mukras, R., Lothian, R., Wiratunga, N., Watt, S. N., & Harper, D. J. (2007, January). Supervised Latent Semantic Indexing Using Adaptive Sprinkling. In IJCAI (pp. 1582-1587).

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer, Berlin: Springer series in statistics

Fragoudis, D., Meretakis, D., & Likothanassis, S. (2005). Best terms: an efficient feature-selection algorithm for text categorization. Knowledge and Information Systems, 8(1), 16-33.

Gomez, J. C., & Moens, M. F. (2012). PCA document reconstruction for email classification. Computational Statistics & Data Analysis, 56(3), 741-751.

Gunal, S., & Edizkan, R. (2008). Subspace based feature selection for pattern recognition. Information Sciences, 178(19), 3716-3726.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

Howland, P., & Park, H. (2004). Generalizing discriminant analysis using the generalized singular value decomposition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(8), 995-1006.

Joachims, T. (1996). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization (No. CMU-CS-96-118).

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features (pp. 137-142). Springer Berlin Heidelberg.

Liu, H., & Motoda, H. (Eds.). (1998). Feature extraction, construction and selection: A data mining perspective. Springer Science & Business Media.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods—support vector learning, 3.

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys, 34(1), 1-47.

Taşcı, Ş., & Güngör, T. (2013). Comparison of text feature selection policies and using an adaptive framework. Expert Systems with Applications, 40(12), 4871-4886.

Yan, J., Liu, N., Cheng, Q., ... & Ma, W. Y. (2005, August). OCFS: optimal orthogonal centroid feature selection for text categorization. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 122-129). ACM.

Yang, J., Liu, Y., Zhu, X., Liu, Z., & Zhang, X. (2012). A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. Information Processing & Management, 48(4), 741-754.

Yang, J., Liu, Z., Qu, Z., & Wang, J. (2014, June). Feature selection method based on crossed centroid for text categorization. In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on (pp. 1-5). IEEE.

Yang, Y., & Pedersen, J. O. (1997, July). A comparative study on feature selection in text categorization. In ICML (Vol. 97, pp. 412-420)