A multi-criteria decision making approach in feature selection for enhancing text categorization

S. Doan\textsuperscript{1} & S. Horiguchi\textsuperscript{2}
\textsuperscript{1}Graduate School of Information Science, Japan Advance Institute of Science and Technology, Japan
\textsuperscript{2}Graduate School of Information Science, Tohoku University, Japan

Abstract

This paper considers the problem of feature selection in text categorization. Previous works in feature selection often used a filter model in which features, after ranked by a measure, are selected based on a given threshold. In this paper, we present a novel approach to feature selection based on multi-criteria decision making of each feature. Instead of only one criterion, multi-criteria of a feature are used; and a procedure based on each threshold of the criterion is proposed. This framework seems to be suitable for text data and can be applied to feature selection in text categorization. Experimental results on Reuters-21578 benchmark data show that our approach has a promising scheme and enhances the performance of a text categorization system.

1 Introduction

Feature selection is an interesting issue recently in machine learning as well as data mining communities [1, 2, 3]. Up to now, there has been two most common approaches: the filter and the wrapper [4, 5]. Both approaches use prior knowledge as the heuristics indicator for selecting the optimal feature subset. The filter approach uses measurements of features as the criterion. In [6], Huang listed four measurements of features belonging to the filtering approach, that is information, distance, dependence and consistency measures. Based on a measurement, the optimal subset of features is chosen by filtering the “noise” or “irrelevant” features. Apart from the filter model, the wrapper approach based on the criterion of
classifier accuracy [5]. The subset of features is chosen based on the accuracy of a classifier, the optimal subset is the subset that achieves the highest accuracy of the classifier. In practice, the wrapper method is relatively difficult to implement, especially with a large amount of data. Instead, the filtering approach is usually chosen because it is easy for implementing and independent of classifiers. This paper only considers the feature selection problem in the filter approach.

Intuitively, the measurement of a feature should characterize the importance of each feature within the entire set. Some measurements include document frequency, information gain, mutual information, cross entropy, Gini index and Chi statistics [1, 2] were proposed and applied efficiently to filter approach. However, previous researches using filter approach use only one measurement for selecting optimal feature subset.

Text data, however, has characteristics of linguistics such as semantics, syntax, thesaurus, etc. Therefore, features in text data (keywords, terms, index terms) has a relationship to others. Using one measurement in selecting features is not suitable. In addition, many criteria can be applied to evaluate the importance of features in text data. Of the research investigating the feature selection in text data up to now, almost used only one measurement, for instance, entropy, mutual information, cross entropy to eliminate “noise” data [1, 2, 7, 8]. Because of the arguments discussed above, it is reasonable for using multi-criteria for text data in feature selection problem. Some results of the using multi-criteria of each feature were proposed and reported by Doan and Horiguchi [9, 10].

In this paper, we propose a novel approach to the feature selection problem in text categorization by selecting features based on multi-criteria decision making of features. One feature has several term weights corresponding to each criterion, our proposed approach is to choose subsets based on each criterion and then combine them all.

The paper is organized as follows. Section 2 presents a novel framework for feature selection problem base on multi-criteria decision making of features. The proposed framework is applied in text categorization in Section 3. Experimental results in Section 4 show the performance improvements of our approach. Discussions and conclusions are drawn in Section 5.

2 A novel framework for feature selection

In this Section, we firstly address the problem of feature selection and give the general framework for feature selection based on each threshold of each criterion. Next, we discuss something about how to choose the threshold for each criterion.

2.1 The EFS procedure

The feature selection problem in text categorization can be stated as follows: Given a set \( \mathbf{X} \) consisting of \( n \) features \( x_1, x_2, \ldots, x_n \), the problem in feature selection is to choose the optimal subset \( S \) of \( \mathbf{X} \) (\( ||S|| \ll ||\mathbf{X}|| \)) with highest effectiveness for the system.
Procedure EFS($X$ - original feature set, $S$-optimal feature set, $\tau_1, \ldots, \tau_t$ - threshold values)

for $i = 1$ to $t$ loop
    $S_i \leftarrow \emptyset$;
    Step 1. Ranking all features based on criterion $\theta_i$;
    Step 2. Choose the first features based on $\tau_i$;
    Step 3. Return $S_i$;
end loop;
$S \leftarrow S_1 \cup S_2 \cup \ldots \cup S_t$;
Return $S$

Figure 1: The EFS procedure for selecting the optimal feature subset.

To solve this problem, our basic idea is to filter features based on a procedure of multi-criteria ranking for terms (terms in text categorization are equivalent to features). Each feature, according to a criterion, will be weighted with a term weight; thus, with $t$ criteria, we will have $t$ ways of ordering features as follows. The feature selection problem can be stated mathematically as follows.

Choose a proper subset of $X$, given a set of criteria $\theta_1, \ldots, \theta_t$, within which each criterion determines a ranking of $X$. Formally, we have:

$$\text{Criterion } \theta_1 : x_{\sigma_1(1)} \preceq_{\theta_1} \ldots \preceq_{\theta_1} x_{\sigma_1(n)}$$

$$\cdots \cdots \cdots$$

$$\text{Criterion } \theta_t : x_{\sigma_t(1)} \preceq_{\theta_t} \ldots \preceq_{\theta_t} x_{\sigma_t(n)}$$

where $\sigma_i$ is a permutation of the set $\{1, \ldots, n\}$, and $\preceq_{\theta_i}$ is the order relation based on criterion $\theta_i$.

After ranked $X$ according to a multiple criteria as above, for each criterion $\theta_i$, we select a subset $S_i$ of $X$ based on a threshold $\tau_i$. Then the set of selected terms is defined by

$$S = \bigcup_{i=1} S_i$$

Algorithmically, the process of term selection is depicted as in Figure 1.

2.2 How to choose the threshold for each criterion

There is a question raising in the EFS procedure: How to choose the subsets of features based on threshold values ? Intuitively, suppose that for each criterion $\theta_i$, the optimal subset is $S_{opt_i}$, the final subset should be obtained by combination of optimal subsets $S_{opt_i}$.

$$S_{opt} = S_{opt_1} \cup \ldots \cup S_{opt_n}$$
Finding the optimal subset for each criterion is also very difficult problem. In addition, it depends on the data type and characteristics of the problem under consideration. Thus, the threshold values should be approximated by experiments.

Among the criteria of feature in the problem under consideration, there must be a principal or primary criterion regards to the problem, denoted \( \theta_p \). It should be reasonable to choose the optimal subset \( S_{opt_p} \) as the “core” set firstly and then other subsets are added increment ally to the core set. Finally the optimal subset obtained could be approximated as

\[
S_{opt} = S_1 \cup \ldots \cup S_{opt_p} \cup \ldots \cup S_n. \tag{3}
\]

### 3 Application to text categorization

#### 3.1 Measurements of text data

Text categorization is the problem of assigning a document into one or more given classes. It consists of two main steps: pre-processing and classifier building and the general framework is depicted as in Figure 2. Pre-processing includes tasks such as feature extraction, feature selection and document representation. After pre-processing, a document will be represented as a vector of features in vector space model [11] or a “bag-of-words” in probabilistic model.

Given a set of categories \( \mathbf{C} = \{c_i\}_{i=1}^{m} \), features in text categorization usually are selected by one of the following measurements.

1. Document frequency criterion: Features are selected by their frequencies in document, with a threshold.
2. Class-based criterion: Select features based on their frequency in a class.
3. Information gain measure: The information gain of term $x$ is given by [1, 3]:

$$IG(x) = -\sum_{i=1}^{m} P(c_i) \log P(c_i) + \sum_{i=1}^{m} P(c_i) \log(c_i|x) + P(x) \sum_{i=1}^{m} P(c_i|x) \log(c_i|x).$$

(4)

4. Mutual information measure: Mutual information of term $t$ is given by [1, 3]:

$$MI(x) = \sum_{i=1}^{m} \log \frac{P(x \land c_i)}{P(x).P(c_i)}.$$  

(5)

There are other measures for feature selection, for example, chi-square and odd-ratio ...[1, 2, 3]

3.2 The naive Bayes classifier

After the pre-processing step, a document is represented by features which are inputs for the second text categorization step, classifier building. Several existing machine learning techniques can be applied, for example, decision tree, neural network, perception, naive Bayes algorithm, genetic algorithms, etc [3, 12]. Among them, naive Bayes is one of the most common used in text categorization and is viewed as the baseline method [3, 12]. For the purpose of comparison, this paper considers naive Bayes as the baseline classifier.

The naive Bayes algorithm can be briefly described as follows.

Given $m$ classes $C = (c_1, c_2, \ldots, c_m)$, with a document $d'$, our problem is to build a classifier $\sigma$ that can assign the document $d'$ to a class.

The naive Bayes algorithm is based on a probabilistic model, in which each document can be represented as a bag-of-words. This means that words existing in documents can be chosen to represent them. Without loss of generality, suppose a document $d'$ consisting of terms $x_1, x_2, \ldots, x_n$. The naive Bayes algorithm calculates the probability of a class belonging to each document with the assumption of independent variables (attributes). The formulation based on the Bayes theorem and the Maximum A Posterior (MAP) principle is given by:

$$P(c_i|d') \propto P(d'|c_i)P(c_i) = P((x_1, \ldots, x_n)|c_i)P(c_i) = \prod_{j=1}^{n} P(x_j|c_i)P(c_i).$$

(6)

Thus, the class of document $d'$ is calculated by the following formula,

$$\sigma(d') = \arg\max_{i \in [1...m]} P(c_i|d').$$

(7)
Table 1: Details of top 10 categories of Reuters-21578 data set.

<table>
<thead>
<tr>
<th>Category</th>
<th>#training docs</th>
<th>#testing docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earn</td>
<td>2,877</td>
<td>1,083</td>
</tr>
<tr>
<td>Acq</td>
<td>1,650</td>
<td>719</td>
</tr>
<tr>
<td>Money-fx</td>
<td>538</td>
<td>179</td>
</tr>
<tr>
<td>Grain</td>
<td>433</td>
<td>149</td>
</tr>
<tr>
<td>Crude</td>
<td>389</td>
<td>189</td>
</tr>
<tr>
<td>Trade</td>
<td>368</td>
<td>117</td>
</tr>
<tr>
<td>Interest</td>
<td>347</td>
<td>131</td>
</tr>
<tr>
<td>Ship</td>
<td>197</td>
<td>89</td>
</tr>
<tr>
<td>Wheat</td>
<td>212</td>
<td>71</td>
</tr>
<tr>
<td>Corn</td>
<td>181</td>
<td>56</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>7,769</strong></td>
<td><strong>3,019</strong></td>
</tr>
</tbody>
</table>

4 Experimental results

4.1 Real-world data set

To examine our proposed method, we used a standard text data set Reuters-21578 [13]. It is a data set collected by Lewis; originally in SGML format and up to now it has been viewed as the standard data for text categorization research. There are various versions of Reuters, of which Reuters-21578 is the most common used [1, 3]. The top 10 categories were chosen for implementation; they are described in Table 1.

Reuters-21578 data set is preprocessed by removing common words such as the, a, an, etc. in the stop list, words are stemmed by the Porter algorithm. After preprocessing, the number of vocabulary is 19,791 words.

In our experiments, we chose two standard methods in feature selection, all terms (that is method containing all terms in vocabulary) and feature selection based on mutual information measure. For easily understanding later, we called the first case all term method and the second case the baseline method.

In the baseline method, mutual information measure is considered as the most common measure used in feature selection [1, 14]. Thus, we take it as the principal criterion $\theta_p$ and $S_{opt_p}$ is taken by a threshold of mutual information measure. The most threshold using this measure is $\approx 1/10$ vocabulary [7, 14]. Thus we chose the number of vocabulary was 2,000.

To compare our method with the baseline method and all term method, we used two criteria, the mutual information and class-based frequency. The principal criterion here was mutual information measurement. A threshold for the principal
Table 2: The contingency table for a category.

| Category set $C = \{c_1, c_2, \ldots, c_{|C|}\}$ | Human assign YES $A = \sum_{i=1}^{\mid C \mid} a_i$ | Human assign NO $B = \sum_{i=1}^{\mid C \mid} b_i$ |
|-----------------------------------------------|----------------------------------|----------------------------------|
| Classifier predict YES                       |                                 |                                  |
| Classifier predict NO                        | $C = \sum_{i=1}^{\mid C \mid} c_i$ | $D = \sum_{i=1}^{\mid C \mid} d_i$ |

criterion was $\tau_1 = 2,000$ and two thresholds for class-based frequency measure were selected, $\tau_2 = 100$ and $\tau_2 = 200$, respectively. For convenience, we called the first case in our proposed method the EFS-100 and the second the EFS-200. The number of terms in the EFS-100 and EFS-200 are 2,314 and 2,619 terms respectively.

Experiments are executed in SunOS 5.8 operating system, Perl, sed, awk, C programming languages and libbow library [15].

4.2 Performance measures

Conventional performance measures in text categorization are precision/recall, $F_1$ measure and break-even point ($BEP$). They based on two basic measures precision and recall.

Mathematically, they are expressed through the contingency table as in Table 2, for each category $c_i$, the precision $P_i$ and the recall $R_i$ are defined as follows,

$$P_i = \frac{a_i}{a_i + b_i} \text{ and } R_i = \frac{a_i}{a_i + c_i}. \quad (8)$$

Macro-averaging performances of the system are given by:

$$\text{macro-}P = \sum_{i=1}^{k} \frac{P_i}{k} \text{ and } \text{macro-}R = \sum_{i=1}^{k} \frac{R_i}{k}. \quad (9)$$

Microaveraging performances are calculated by,

$$\text{micro-}P = \sum_{i=1}^{k} \frac{a_i}{\sum_{i=1}^{k} (a_i + b_i)}, \quad (10)$$

$$\text{micro-}R = \sum_{i=1}^{k} \frac{a_i}{\sum_{i=1}^{k} (a_i + c_i)}. \quad (11)$$

$F_1$ is defined as

$$F_1 = 2PR/(P + R). \quad (12)$$
Table 3: \(BEP\) performance measures of Reuters-21578.

<table>
<thead>
<tr>
<th>Category</th>
<th>all terms</th>
<th>baseline</th>
<th>EFS-100</th>
<th>EFS-200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earn</td>
<td>97.65</td>
<td>97.47</td>
<td>97.43</td>
<td>97.38</td>
</tr>
<tr>
<td>Acq</td>
<td>96.45</td>
<td>96.04</td>
<td>96.60</td>
<td>96.66</td>
</tr>
<tr>
<td>Money-fx</td>
<td>76.54</td>
<td>75.98</td>
<td>76.54</td>
<td>76.19</td>
</tr>
<tr>
<td>Grain</td>
<td>50.34</td>
<td>49.49</td>
<td>51.04</td>
<td>51.50</td>
</tr>
<tr>
<td>Crude</td>
<td>80.00</td>
<td>78.09</td>
<td>78.51</td>
<td>78.51</td>
</tr>
<tr>
<td>Trade</td>
<td>79.15</td>
<td>84.62</td>
<td>84.12</td>
<td>84.12</td>
</tr>
<tr>
<td>Interest</td>
<td>72.52</td>
<td>68.96</td>
<td>70.23</td>
<td>70.23</td>
</tr>
<tr>
<td>Ship</td>
<td>62.92</td>
<td>60.00</td>
<td>59.55</td>
<td>59.55</td>
</tr>
<tr>
<td>Wheat</td>
<td>31.76</td>
<td>40.85</td>
<td>41.13</td>
<td>39.72</td>
</tr>
<tr>
<td>Corn</td>
<td>33.93</td>
<td>35.40</td>
<td>37.50</td>
<td>37.50</td>
</tr>
<tr>
<td>macro ave</td>
<td>68.13</td>
<td>68.69</td>
<td>69.26</td>
<td>69.14</td>
</tr>
<tr>
<td>micro ave</td>
<td>72.31</td>
<td>74.54</td>
<td>74.55</td>
<td>74.55</td>
</tr>
</tbody>
</table>

\(BEP\) measure is the point that \(P = R\), this point is often calculated by taking the average of \(P\) and \(R\). The macro- and micro-\(F_1\) and \(BEP\) are calculated by replacing \(P, R\) with the corresponding macro and micro of \(P, R\).

Macro-averaging and microaveraging of \(F_1\) and \(BEP\) are treated as the measures to compare performances of text categorization systems.

4.3 Experimental results

Tables 3 and 4 shows the results of \(BEP\) and \(F_1\). Results indicated that both two proposed methods the EFS-100 and the EFS-200 had always higher performances than the baseline and the all term methods.

The macroaveraging \(BEP\) for the EFS-100 is 69.26% vs. 68.13% when using the all term method and 68.69% when using the baseline method. In case of the EFS-200, the macroaveraging \(BEP\) is 69.14%; it is higher than both baseline and the all term methods but lower than the EFS-100. The microaveraging \(BEP\) for both proposed methods is the same (74.55%). It is also not different from that for the baseline method (74.54%) but higher than the all term method (72.74%).

Similarly, the macro- and microaveraging of \(F_1\) for both proposed methods are higher than the baseline and the all term methods. The macroaveraging \(F_1\) are 72.83% for the EFS-100 and 72.79 for the EFS-200 respectively, vs. 71.10% for all term method and 72.68% for the baseline method. The microaveraging \(F_1\) are 74.06% for the EFS-100 and 74.03% for the EFS-200 while they are 73.86% and 73.34% for the baseline method and the all term method respectively.
Table 4: $F_1$ performance measures of Reuters-21578.

<table>
<thead>
<tr>
<th>Category</th>
<th>all terms</th>
<th>baseline</th>
<th>EFS-100</th>
<th>EFS-200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earn</td>
<td>98.10</td>
<td>97.91</td>
<td>98.04</td>
<td>98.04</td>
</tr>
<tr>
<td>Acq</td>
<td>96.48</td>
<td>96.21</td>
<td>96.67</td>
<td>96.67</td>
</tr>
<tr>
<td>Money-fx</td>
<td>76.92</td>
<td>75.98</td>
<td>76.54</td>
<td>76.30</td>
</tr>
<tr>
<td>Grain</td>
<td>59.76</td>
<td>54.42</td>
<td>57.47</td>
<td>57.41</td>
</tr>
<tr>
<td>Crude</td>
<td>81.40</td>
<td>79.67</td>
<td>79.43</td>
<td>79.43</td>
</tr>
<tr>
<td>Trade</td>
<td>82.59</td>
<td>85.59</td>
<td>85.60</td>
<td>85.60</td>
</tr>
<tr>
<td>Interest</td>
<td>73.00</td>
<td>73.83</td>
<td>73.38</td>
<td>73.38</td>
</tr>
<tr>
<td>Ship</td>
<td>67.00</td>
<td>67.58</td>
<td>68.75</td>
<td>68.96</td>
</tr>
<tr>
<td>Wheat</td>
<td>39.82</td>
<td>49.24</td>
<td>48.39</td>
<td>47.83</td>
</tr>
<tr>
<td>Corn</td>
<td>35.89</td>
<td>46.40</td>
<td>44.02</td>
<td>44.30</td>
</tr>
<tr>
<td>macro ave</td>
<td>71.10</td>
<td>72.68</td>
<td>72.83</td>
<td>72.79</td>
</tr>
<tr>
<td>micro ave</td>
<td>73.34</td>
<td>73.86</td>
<td>74.06</td>
<td>74.03</td>
</tr>
</tbody>
</table>

In summary, our proposed method outperformed the baseline method and the all term method, especially for macro averaging measures. Furthermore, the results also showed that the EFS-100 has better performance than the EFS-200, it has been suggested that appropriate parameters $\tau_1$ and $\tau_2$ for our proposed method can be tuned for achieving better performance.

In order to compare performances of the system, we compare macroaveraging measures of $BEP$ and macroaveraging of $F_1$. Results shows that with all 10 categories, the proposed method have higher performances than using mutual information criterion and using all terms in both $BEP$ and $F_1$. Macroaveraging of $BEP$ with proposed method is 69.26% vs. 68.13% when using all terms and 68.69% when using mutual information criterion; macroaveraging of $F_1$ with proposed method is 72.83% vs. 71.10% when using all terms and 72.68% when using mutual information criterion.

5 Conclusions

This paper proposed a novel feature selection approach based on the multi-criteria decision making of features in text categorization problem. A general framework for feature selection was proposed and applied to Reuters-21578 data set. Experimental results shows that the proposed method using multi-criteria of features instead of using only one criterion as the baseline methods could enhance the performance of the system, especially for macroaveraging; in compared to methods using only one criterion and whole vocabulary. This results is significant and use-
ful for feature selection in text categorization problem. In summary, the advantages of our proposed as follows:

1. The proposed method reduced the number of terms in the vocabulary and improved the performance of the text categorization system.
2. The proposed method outperformed the performance, including both $F_1$ and $BEP$ measures compared to both the all term method and the baseline method.
3. The proposed method has better performance, especially for macroaveraging measures, compared to the baseline method.

**Acknowledgments**

The authors would like to express appreciation to Ms. Judith Steeh for English correction of this paper.

This work was supported in partly the Grand-in-Aid of Scientific Research, JSPS, Japan.

**References**


