Applied Soft Computing Journal 86 (2020) 105836

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods

Gang Kou [a], Pei Yang [a], Yi Peng [b,*], Feng Xiao [a], Yang Chen [a], Fawaz E. Alsaadi [c]

[a] School of Business Administration, Southwestern University of Finance and Economics, Chengdu 611130, China
[b] School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 610054, China
[c] Department of information Technology, Faculty of Computing and IT, King Abdulaziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 21 February 2019
Received in revised form 2 August 2019
Accepted 8 October 2019
Available online 25 October 2019

Keywords:
Feature selection
Text classification
MCDM
Small sample dataset

ABSTRACT

The evaluation of feature selection methods for text classification with small sample datasets must consider classification performance, stability, and efficiency. It is, thus, a multiple criteria decision-making (MCDM) problem. Yet there has been few research in feature selection evaluation using MCDM methods which considering multiple criteria. Therefore, we use MCDM-based methods for evaluating feature selection methods for text classification with small sample datasets. An experimental study is designed to compare five MCDM methods to validate the proposed approach with 10 feature selection methods, nine evaluation measures for binary classification, seven evaluation measures for multi-class classification, and three classifiers with 10 small datasets. Based on the ranked results of the five MCDM methods, we make recommendations concerning feature selection methods. The results demonstrate the effectiveness of the used MCDM-based method in evaluating feature selection methods.

1. Introduction

The increasing amount of digitized text from sources such as web pages, emails, blogs, digital libraries, social media, online advertisements, corporate documents, and product reviews improves the value of text classification [1–3]. Text classification using supervised machine learning [4,5] places texts into predefined classes based on the content (e.g., positive or negative, spam or not spam, one topic or another, and helpful or not helpful).

Preprocessing, feature selection, text representation, and text classification comprise four stages in a fundamental text classification scheme. Feature selection is an important component in machine learning [6] and a necessary step for text classification [7]. It also reduces computational complexity, improves classification performance, and avoids the overfitting problem. Hence, researchers in many fields have given significant attention to feature selection [8] and have proposed various feature selection methods for text classification.

Text classification tasks now face the problem of small samples with a small number of labeled samples and high dimensionality. The growth of the number of labeled texts is less than that of unlabeled texts because labeling texts requires human involvement in the text classification problem. Consequently, text classification problems due to a lack of labeled texts are increasing. Furthermore, text classification tasks are often high dimensional [9].

Small samples and high-dimensional datasets introduce three problems into the feature selection process. First, feature selection is not stable with small samples and high dimensionality [10]. Second, feature selection consumes more time with high dimensionality. Third, classification performance may not be good enough using a specific feature selection method. Therefore, multiple factors should be considered to select an appropriate feature selection method for classifying texts with small samples for text classification. A feature selection method with good classification performance may not necessarily have good stability and efficiency. So, the evaluation of feature selection methods for classifying small texts for text classification must consider multiple measures. We can model the evaluation as a multiple criteria decision-making (MCDM) problem [11].

The classification performance can be evaluated by three kinds of measures [2] which reflect different aspects of classification performance and are irreplaceable with each other. Classification performance is the most common metric for evaluating feature selection methods. However, most studies of feature selection methods in text classification only apply single measure. The stability and efficiency of the feature selection methods have received little attention, and works that address both classification performance and stability [9,12] have not considered these criteria together to use a compromise method to evaluate feature selection methods for text classification of small samples.

* Corresponding author.
E-mail address: pengyi@uestc.edu.cn (Y. Peng).

https://doi.org/10.1016/j.asoc.2019.105836

To solve these problems, we use MCDM-based approaches to evaluate feature selection methods for text classification of small sample datasets. To show the effectiveness of the MCDM methods, our experimental study evaluated 10 feature selection methods. Among the five MCDM methods, the most appropriate method is determined. We compare the evaluation results among three classifiers, five MCDM methods, and 10 text classification datasets. Comparing with previous studies, our research approach is different in the following aspects: (1) In feature selection methods for text classification, this is the first attempt to evaluate feature selection method using MCDM; (2) In the selection of evaluation criteria, the criteria we selected can reflect different aspects of the feature selection method and have a low linear relationship; (3) This is also the first comparative analysis to identify suitable MCDM method for evaluating feature selection method in text classification with small sample datasets.

We organize our paper as follows: In Section 2, we introduce related work. In Section 3, we describe the research approach, feature selection methods, evaluation measures, and MCDM methods. In Section 4, we present details of our experimental study. In Section 5, we provide our conclusions and future research direction.

2. Related work

We review works related to our topic in this section. In this paper, we consider feature selection methods for text classification, evaluation measures, and MCDM methods. Although there is little research on the evaluation of feature selection methods for text classification, we examine evaluation methods from other fields for text classification.

2.1. Feature selection methods for text classification

Feature selection in text classification tasks is defined as a process that seeks the minimal size of relevant text features to optimize text classification error. Feature selection methods are usually categorized as filters, wrappers, or embedded methods [3].

Wrappers methods select subsets using an evaluation function and a search strategy. Regardless of the search strategy, the number of possible results will increase geometrically as the number of features increases. The process of embedded methods are embedded in the training of classifiers [13]. Wrappers and embedded methods require much more computation time than filters and may work only with a specific classifier [3]. Thus, filters are the most common feature selection method for classifying text.

Filters are either global or local. Global methods assign a single score to a feature regardless of the number of classes while local methods assign several scores, as every feature in every class has a score [14]. Global methods typically calculate the score for every feature and then choose the top-N features as the feature set, where N is usually determined empirically. Local methods are similar but require converting a feature's multiple scores to a single score before choosing the top-N features. Some commonly used global feature selection methods are the document frequency (DF) [15], information gain (IG) [16], Gini Index (GI) algorithms [17], and distinguishing feature selector (DFS) [18]. Local methods include the Chi-square (CHI) [15] and odds ratio (OR) algorithms [3].

In our work, we evaluate those frequently-used feature selection methods, including six global and four local feature selection methods.

2.2. Evaluation measures of feature selection methods

Reliable evaluation measures are needed to rank performance, but researchers have paid little attention to this area. The most common measure of feature selection quality is classification performance [7,17,19–22]. However, stability is another important measure, as are others.

2.2.1. Classification performance measures

A classification algorithm is a kind of machine learning algorithm. They can be evaluated by empirical assessment, theory, or both, but we focus on empirical evaluation as the most commonly used approach [23].

Frequently-used empirical evaluation measures include accuracy, F1 score, precision, recall, specificity, area under the curve (AUC), mean absolute error (MAE), and mean squared error (MSE).

Ferri et al. [24] divided empirical evaluation measures into three classes based on a threshold and a qualitative understanding of error; a probabilistic understanding of error, and how well the model ranks the examples. The first classification includes F1 score, precision, recall, specificity, accuracy, macro average F1 score, and micro average F1 score. The second includes MAE and MSE. The third includes AUC.

Existing studies focus on the relationship between different measures, especially accuracy and AUC. Cortes [25] showed that "algorithms designed to minimize the error rate may not lead to the best possible AUC values". Rosset [26] showed that AUC yielded better accuracy results than the accuracy measure itself on a validation dataset. Davis and Goadrich [27] found that methods perform well in the receiver operating characteristic (ROC) space only if they perform well in the precision–recall space. Ferri et al. [24] analyzed the linear and rank correlations among 18 measures. From [25–27], we know that the linear relationship between AUC and accuracy is small and is used to measure different aspects of classification performance. Thus, accuracy and AUC are irreplaceable in evaluating classification performance. Besides, we can find that the linearity of the evaluation measures among different classes is small, but bigger within the same class in [24]. This confirms that we need to select representative evaluation measures from three different classes rather than in the same class.

Most feature selection method studies for text classification evaluate their method using measures in the first class [10,14,21,28–32]. However, measures in different classes evaluate different things [24]. Using only one kind of measure will not accurately reflect the quality of a feature selection method. Thus, we select several representative classification performance measures of the three classes to evaluate feature selection methods comprehensively.

2.2.2. Stability measures

Stability is defined as the robustness of a feature subset generated from different training sets from the same distribution [33]. When the changes in a feature subset generated by a feature selection method are small, the method is considered stable. Unstable feature selection performance leads to degraded performance in the final classifier due to failure to identify the most relevant features [9].

Robustness can be evaluated by different measures. These measures to evaluate stability of feature selection methods can be distinguished based on the following [9]:

Feature-focused versus subset-focused—the former evaluates a feature selection method by considering all feature subsets together, while the latter calculates similarities of features in each pair of two subsets. These two methods can provide complementary information, and no one is better than the other [10].

However, the similarities between each pair of two subsets is more reflective of the difference between the subsets than the overall comparison.

Selection-registering versus selection-exclusion-registering—the former considers just the selected features while the latter also considers other features. Because for high-dimensional datasets, there may be a lot of features that do not contribute to the performance of the classification. It is not necessary to consider these features when evaluating stability.

Subset-size-biased versus subset-size-unbiased—the upper and lower bounds of the former change with the number of features selected while the latter do not.

All of those studies that proposed new feature selection methods in text classifications did not consider the stability as evaluation measure [10,14,21,28–32], so we take stability into account in our work. On the selection of stability indicators, on the one hand, there is a high linear relationship between stability indicators, selecting one indicator is enough. On the other hand, due to the high-dimension of text classification a measure that are subset-focused, selection-registering, and subset-size-biased, should be selected

2.3. MCDM methods in algorithm evaluation

MCDM helps decision-makers solve evaluation problems involving multiple measures, especially when they conflict [34–36]. It is an important technique in operations research (OR) [37] and addresses three main types of decision problems: ranking, sorting, and choice [38]. Significant amounts of research into MCDM techniques and applications is published every year [39–41]. The evaluation and selection of MCDM algorithms is an important research field [11,24,42,43]. Over the past 40 years, almost 70 MCDM methods have been explored [44]. We focus our overview on the use of MCDM methods for evaluating algorithms.

Rokach [45] suggested that algorithm selection can be considered as an MCDM problem, but few studies have been made in this field. One proposal applied three MCDM methods to evaluate classification algorithms for financial risk prediction [46]. But it does not analyze which MCDM method is more suitable. To solve disagreements in the evaluation of multi-class classification using different MCDM methods, Peng et al. [42] proposed an effective fusion method. Similarly, an approach based on Spearman's rank correlation coefficient was proposed to solve the conflict among MCDM methods when evaluating classification algorithms [47]. Although in [42] and [47], these methods solve the conflicts of the evaluation results of different MCDM methods in algorithm evaluation, it is usually more appropriate to rely on one result of a suitable method when facing with an MCDM problem than to integrate the results of multiple MCDM methods. Another study evaluated clustering algorithms for use with three MCDM methods in financial risk analysis [11]. Nevertheless, it does not integrate the evaluation results of the three methods, nor does it analyze which method is more suitable for evaluating the clustering algorithm. None of the four studies [11,42,46,47] involved the evaluation of feature selection methods in text classification.

Only one study has used MCDM to evaluate feature selection methods. Raman et al. [43] proposed an TOPSIS-based method to evaluate feature selection methods for a network traffic dataset. They evaluated ten feature selection methods according to nine measures, with seven of them based on classification performance. However, their work was not in the field of text classification, and, as most measures addressed classification performance, they did not consider stability and efficiency, making their evaluation one-sided.

In order to solve these problems, we have selected multiple MCDM methods for evaluating feature selection methods in text classification, and conducted a comparative analysis of several methods to determine the most appropriate method.

Table 1
The categories of the 10 methods.

Categories	FS methods
Global	DF, IG, GI, DFS, ECE, and CDM
Local	CHI, OR, MI, and WLLR

3. Research methodology

In this paper, we utilize an MCDM-based approach to evaluate feature selection methods for text classification. For our empirical study, we chose 10 feature selection methods, 9 binary classification measures, 7 multi-class classification measures, and 5 MCDM methods to validate each evaluation approach (see Fig. 1). In this section, we provide details of the proposed approach, the feature selection methods, the performance measures, and the MCDM methods.

3.1. MCDM-based approach for evaluation of feature selection methods

As mentioned in Section 1, multiple measures, including classification performance, stability, and efficiency measures, are required to evaluate feature selection methods. No feature selection method performs best on all of the measures. Therefore, selecting a method requires trade-offs that can be modeled as an MCDM problem. We propose a hybrid evaluation process for feature selection methods that combines the three kinds of measures as shown in Fig. 2.

The first step applies the ten feature selection methods to ten datasets to calculate the various measures, storing them as matrices. The second step ranks the feature selection methods using five MCDM algorithms according to the change in the number of features for each dataset. According to the rankings obtained using these MCDM methods, the last step recommends a feature selection method.

3.2. Feature selection methods and measures

We evaluate 10 well-known filter methods for feature selection, chosen for their performance variations among the criteria. The following paragraphs present the different methods and the measures for binary and multi-class classification.

3.2.1. Feature selection methods

The chosen filters for the evaluation are: document frequency (DF), information gain (IG), Gini index (GI), distinguishing feature selector (DFS), expected cross-entropy (ECE), class discriminating measure (CDM), Chi-squared (CHI), odds ratio (OR), mutual information (MI), and weighted log likelihood ratio (WLLR). The categories of these methods are summarized in Table 1.

We use the following notations in our presentation. For a given dataset of input texts, N represents the number of documents. C_i represents the ith class. M is the number of classes. $P(C_i)$ is the proportion of documents in class C_i relative to the whole document set. $P(t)$ and $P(\bar{t})$ are the proportion of documents in which term t is present or absent, respectively. $P(C_i|t)$ and $P(C_i|\bar{t})$ are the proportions of documents in class C_i in the documents in which the term is present or absent. $P(t|C_i)$ and $P\left(t|\overline{C}_i\right)$ represent the proportion of documents in which term t is present in the documents in class C_i and not in class C_i. a_i represents the number of documents that contain term t in class C_i, b_i represents the number of documents that do not contain term t in class C_i, c_i represents the number of documents that contain term t but belong to all classes except class C_i, and d_i represents the number

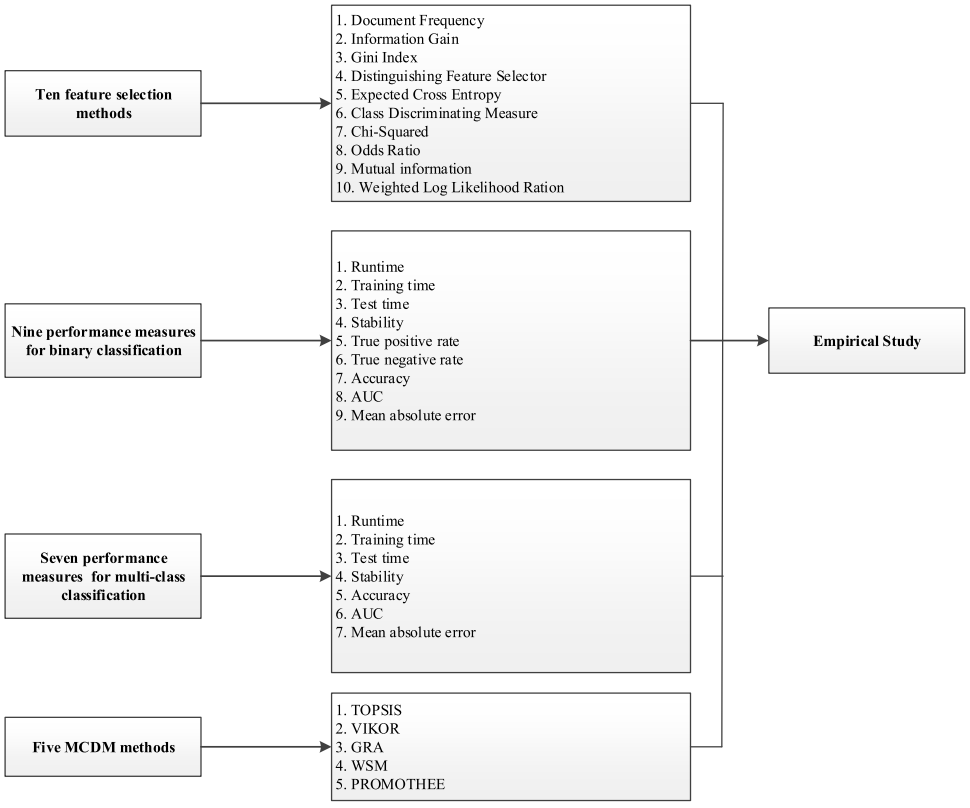

Fig. 1. Feature selection methods, performance measures, and MCDM methods used in this study.

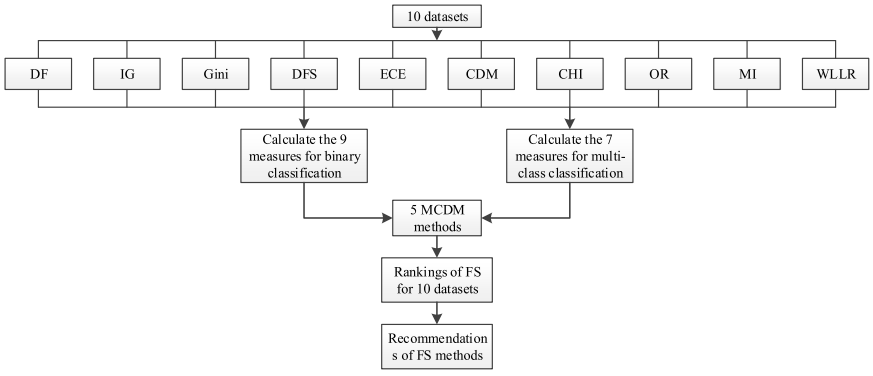

Fig. 2. Evaluation process for feature selection methods.

of documents that do not contain term t but belong to all classes except class C_i.

(1) Document frequency

DF is a simple and effective unsupervised feature-selection method [13] that scores features according to the number of appearances in the document [15]. That is, more frequent features are more important. DF is calculated by counting the number of documents in which a feature occurs. An obvious disadvantage is that some high-frequency terms that are not helpful for classification, such as stop words, will be counted as features.

(2) Information gain

IG is a supervised method designed to determine a term's contribution according to a ratio calculated by counting its presence or absence in a document set [29]. The exact calculation is:

$$IG(t) = -\sum_{i=1}^{M} P(C_i)\log(P(C_i)) + P(t)\sum_{i=1}^{M} P(C_i|t)\log(P(C_i|t)) + P(\bar{t})\sum_{i=1}^{M} P(C_i|\bar{t})\log(P(C_i|\bar{t})). \quad (1)$$

(3) Gini index

GI was originally used in decision-tree algorithms, but Shang et al. [9] proposed an improved GI for feature selection within text. It is a supervised method with a simpler calculation than IG:

$$GI(t) = \sum_{i=1}^{M} P(t|C_i)^2 P(C_i|t)^2 \quad (2)$$

(4) Distinguishing feature selector

Proposed by Uysal and Gunal [18], DFS is a successful, relatively new feature-selection method for text classification. It is a supervised method intended to pick up the most distinctive

features and remove uninformative ones. It is calculated as:

$$\mathrm{DFS}(t) = \sum_{i=1}^{M} \frac{P(C_i|t)}{P\left(\bar{t}|C_i\right) + P\left(t|\bar{C}_i\right) + 1} \tag{3}$$

(5) Expected cross-entropy

ECE is a supervised method that considers the presence of a term t and ignores its absence [48]. ECE can be calculated as:

$$\mathrm{ECE}(t) = P(t) \sum_{i=1}^{M} P(C_i|t) \log\left(\frac{P(C_i|t)}{P(C_i)}\right) \tag{4}$$

(6) Class discriminating measure

CDM is a global feature selection method derived from the odds ratio proposed by Chen et al. [48]. It is calculated as:

$$\mathrm{CDM}(t) = \sum_{i=1}^{M} \left| \log \frac{P(t|C_i)}{P\left(t|\bar{C}_i\right)} \right| \tag{5}$$

(7) Chi-squared

CHI is a supervised, one-sided feature selection method that calculates the correlation of term t with class C_i [15]. CHI is calculated as:

$$\chi^2(t, C_i) = \frac{N \times (a_i d_i - b_i c_i)^2}{(a_i + b_i) \times (a_i + c_i) \times (b_i + d_i) \times (c_i + d_i)} \tag{6}$$

To calculate the global CHI of term t, we use $\chi^2(t) = \max_i \chi^2(t, C_i)$ to represent the global feature weight of term t in this paper.

(8) Odds ratio

OR is a supervised and one-sided method obtained by calculating membership and non-membership in a specific class with its numerator and denominator, respectively [22]. It is calculated as:

$$\mathrm{OR}(t, C_i) = \log\left(\frac{P(t|C_i)(1 - P\left(t|\bar{C}_i\right))}{(1 - P(t|C_i))P\left(t|\bar{C}_i\right)}\right) \tag{7}$$

To calculate the global OR of term t, we use $\mathrm{OR}(t) = \max_i \mathrm{OR}(t, C_i)$ to represent the global feature weight of term t.

(9) Mutual information

MI is a supervised and one-sided method that represents the correlation between classes and features. It is calculated as:

$$\mathrm{MI}(t, C_i) = \log \frac{P(t|C_i)}{P(t)} \tag{8}$$

To calculate the global MI of term t, we use $\mathrm{MI}(t) = \max_i \mathrm{MI}(t, C_i)$ to represent the global feature weight of term t.

(10) Weighted log likelihood ratio

WLLR is proposed by Nigam et al. [49]. It is a supervised and one-sided method and calculated by:

$$\mathrm{WLLR}(t, C_i) = P(t|C_i) \log \frac{P(t|C_i)}{P\left(t|\bar{C}_i\right)} \tag{9}$$

To calculate the global WLLR of term t, we use $\mathrm{WLLR}(t) = \max_i \mathrm{WLLR}(t, C_i)$ to represent the global feature weight of term t.

3.2.2. Performance measures

To obtain the hybrid evaluation, we have selected nine performance measures for binary classification: runtime, training time, test time, stability, true positive rate (TPR), true negative rate (TNR), accuracy, area under the ROC curve (AUC), and mean absolute error (MAE). We selected seven measures for multi-class classification: runtime, training time, test time, stability, accuracy, AUC, and MAE. We describe these 10 measures below. We use runtime, training time, and test time to evaluate the efficiency of each method, and we use stability to evaluate the robustness. We use the other measures to evaluate the performance of the classifiers used by different feature-selection methods.

(1) Accuracy

Accuracy is a classic and frequently-used measure for evaluating classifier performance in text applications [3]. Accuracy represents the proportion of documents that are correctly classified in the document set:

$$\mathrm{Accuracy} = \frac{TN + TP}{TP + FP + TN + FN} \tag{10}$$

Accuracy is a benefit criterion, with higher values being desirable.

(2) Area under the curve

AUC is a classic measure to evaluate classification performance, defined as the area under the ROC curve. AUC is a benefit criterion.

(3) True positive rate and true negative rate

TPR and TNR are both benefit criteria. *TPR* is the proportion of correctly classified positive instances of all positive instances. It is calculated as:

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

In formula (11), *TP* represents the number of correctly classified positive instances, and *FN* represents the number of positive instances misclassified as negative. The sum of *TP* and *FN* is the number of positive instances.

Similar to *TPR*, *TNR* is calculated as:

$$TNR = \frac{TN}{TN + FP} \tag{12}$$

In formula (12), *TN* represents the number of negative instances that are correctly classified, and *FP* represents the number of negative instances that are misclassified as positive. The sum of *TN* and *FP* is the number of negative instances.

(4) Mean absolute error

MAE measures how much the predictions deviate from the true probability. It is calculated as:

$$\mathrm{MAE} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} |f(i, j) - P(i, j)|}{M \times N} \tag{13}$$

In formula (13), each $f(i, j)$ is 1 if the ith instance is in class j and 0 otherwise; $P(i, j)$ represents the probability that the ith instance belongs to class j. $P(i, j)$ is calculated by classifier with the classification result $\hat{C}_i = argmax_j P(i, j)$. MAE is a cost criterion, with increasing values being undesirable.

(5) Stability

Stability measures the robustness of a feature-selection method. Robustness requires that the selected features remain stable when the training set changes. The stability is calculated as:

$$\mathrm{Stability}(S) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} I_J(S_i, S_j) \tag{14}$$

In formula (14), $I_J\left(S_i, S_j\right) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$ is the Jaccard index [50]. The latter measures the similarity between S_i and S_j, which are two feature sets selected by one feature-selection method from two datasets.

There are many other methods to calculate stability. We choose this similarity-based method for several reasons. First, as we have said before, we need a measure that are subset-focused, selection-registering, and subset-size-biased. Second, for text-based input like ours, the Jaccard index's use of the intersection and union of two word sets intuitively reflects the differences and similarities between two feature sets. Stability is a benefit criterion.

(6) Runtime, training time, and test time

The runtime is the execution time required for feature selection. Runtime includes the time required to calculate feature weights, to sort features-by-feature weight, and to choose the top-N features. Training time is the time needed to train the classifier. Test time is the time required for classifier testing after training the classifier.

Classifiers using different feature selection methods may differ in running time, training time, and test time, so we choose these three times to reflect efficiency from several perspectives. These times are cost criteria.

Although we use multiple classification performance measures including TPR, TNR, accuracy, AUC, and MAE, they evaluate different things. TPR and TNR rate classification performance for a single class, while the others evaluate overall performance. In addition, good TPR performance does not imply good TNR performance. The linear and rank correlations are low among accuracy, AUC, and MAE [24].

3.3. MCDM methods

Various MCDM methods have been proposed over the years. To avoid giving preference for any one method and to obtain more representative evaluation results, we choose five MCDM methods: TOPSIS, VIKOR, GRA, Weighted sum method (WSM) and PROMOTHEE.

3.3.1. Technique for order preference by similarity to ideal solution (TOPSIS)

TOPSIS is a relatively early MCDM method proposed by Hwang and Yoon [51], which calculates the distances between a real solution and the ideal and negative ideal solutions. The shorter the distance between the real solution and the ideal solution, or the longer the distance between the real solution and the negative ideal solution, the better the real solution. We use the following TOPSIS process [11]:

Step 1: Calculate the normalized criteria values:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{J} x_{ij}^2}} \tag{15}$$

In formula (15), $j = 1, 2, \ldots, J$, and $i = 1, 2, \ldots, n$ respectively represent the indices of feature selection methods and criteria. The performance of the jth feature selection method and ith criterion is x_{ij}.

Step 2: Calculate the weighted criteria values:

$$v_{ij} = w_i r_{ij} \tag{16}$$

In formula (16), w_i represents the weight of the ith criterion.

Step 3: Find the ideal solution S^+ and the negative ideal solution S^- as follows:

$$S^+ = \left\{v_1^+, \ldots, v_n^+\right\} = (\max_j v_{ij} | i \in I'), (\min_j v_{ij} | i \in I'') \tag{17}$$

$$S^- = \left\{v_1^-, \ldots, v_n^-\right\} = (\min_j v_{ij} | i \in I'), (\max_j v_{ij} | i \in I'') \tag{18}$$

In formula (17) and (18), I′ and I″ represent the benefit criteria and cost criteria, respectively. The higher the benefit criteria or the lower the cost criteria, the better the solution. Stability, TPR, TNR, accuracy and AUC are the benefit criteria, while runtime, training time, and test time are cost criteria.

Step 4: Calculate the Euclidean distance between the real and ideal solutions, and between the real solution and negative ideal solution, as follows:

$$D_j^+ = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^+)^2} \tag{19}$$

$$D_j^- = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^-)^2} \tag{20}$$

Step 5: Calculate R_j^+:

$$R_j^+ = \frac{D_j^-}{D_j^+ + D_j^-} \tag{21}$$

Step 6: Rank feature selection methods by maximizing R_j^+.

3.3.2. VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR)

VIKOR is a well-known MCDM method proposed by Opricovic et al. [52,53]. VIKOR relies on an aggregating function that represents closeness to the ideal [17]. We use the following VIKOR procedure as adopted by Kou et al. [11].

Step 1: Find the best and worst values for all criteria:

$$f_i^* = \begin{Bmatrix} \max_j f_{ij}, \text{for benefit criteria} \\ \min_j f_{ij}, \text{for cost criteria} \end{Bmatrix} \tag{22}$$

$$f_i^- = \begin{Bmatrix} \min_j f_{ij}, \text{for benefit criteria} \\ \max_j f_{ij}, \text{for cost criteria} \end{Bmatrix} \tag{23}$$

In formula (22) and (23), $i = 1, \ldots$ n and $j = 1, \ldots, J$ represent the indices of criteria and alternatives, respectively, and f_{ij} is the value of the ith criterion for the jth alternative.

Step 2: Calculate S_j and R_j, which are used in subsequent steps:

$$S_j = \sum_{i=1}^{n} w_i \left(f_i^* - f_{ij}\right) \left(f_i^* - f_i^-\right) \tag{24}$$

$$R_j = \max_i [w_i \left(f_i^* - f_{ij}\right) \left(f_i^* - f_i^-\right)] \tag{25}$$

In formula (24) and (25), w_i is the weight of the ith criterion.

Step 3: Calculate Q_j:

$$Q_j = \frac{v(S_j - S^*)}{S^- - S^*} + \frac{(1-v)(R_j - R^*)}{R^- - R^*} \tag{26}$$

In formula (26), $S^* = \min_j S_j$, $S^- = \max_j S_j$, $R^* = \min_j R_j$, $R^- = \max_j R_j$, and v is the strategy weight of the majority of criteria,. We set v to 0.5.

Step 4: Generate rankings of the alternatives in decreasing order according to S, R, and Q.

Step 5: Propose the alternative a', which is ranked best by Q, as a compromise solution if the following conditions are satisfied:

(1) $Q\left(a''\right) - Q\left(a'\right) \geq \frac{1}{J-1}$;

(2) Alternative a' is ranked the best by S or R; where a'' is the second-ranked alternative by Q. If condition 2 alone is not satisfied, then alternatives a' and a'' are proposed as compromise solutions. If condition 1 is not satisfied, alternatives a', a'',…, a^M are proposed as compromise solutions, where a^M is the Mth ranked alternative Q and the last alternatives that makes the relation $Q\left(a''\right) - Q\left(a'\right) < \frac{1}{J-1}$.

3.3.3. Grey relational analysis (GRA)

GRA incorporates grey theory and is another suitable method to select a best alternative [47]. We implement the procedure of Kuo et al. [12] as follows:

Step 1: Generate the grey relation. For the benefit criteria, this is:

$$y_{ij} = \frac{x_{ij} - \min\left\{x_{ij}, j = 1, 2, \ldots, J\right\}}{\text{Max}\left\{x_{ij}, j = 1, 2, \ldots, J\right\} - \text{Min}\left\{x_{ij}, j = 1, 2, \ldots, J\right\}} \tag{27}$$

For the cost criteria, this is:

$$y_{ij} = \frac{\max\left\{x_{ij}, j = 1, 2, \ldots, J\right\} - x_{ij}}{\max\left\{x_{ij}, j = 1, 2, \ldots, J\right\} - \min\left\{x_{ij}, j = 1, 2, \ldots, J\right\}} \tag{28}$$

For the-closer-the-desired-value-the-better, this is:

$$y_{ij} = 1 - \frac{\left|x_{ij} - x_{ij}^{*}\right| x_{ij} - \min\left\{x_{ij}, j = 1, 2, \ldots, J\right\}}{\max\left\{\max\left\{x_{ij}, j = 1, 2, \ldots, J\right\} - x_{ij}^{*}, x_{ij}^{*} - \min\left\{x_{ij}, j = 1, 2, \ldots, J\right\}\right\}} \tag{29}$$

In all three cases, $i = 1, \ldots$ n and $j = 1, \ldots, J$ represent the indices of criteria and alternatives, respectively.

Step 2: Define the reference sequence:

This is similar to the ideal solution and defined as $X_0 = \{x_{10}, x_{20}, \ldots, x_{n0}\} = 1, 1, \ldots, 1$. The closer a sequence is to the reference sequence, the better the sequence.

Step 3: Calculate the grey relational coefficient:

$$\gamma\left(x_{i0}, x_{ij}\right) = \frac{V_{\min} + \varsigma V_{\max}}{V_{ij} + \varsigma V_{\max}} \tag{30}$$

In formula (30), $V_{ij} = \left|y_{i0} - y_{ij}\right|$, $V_{\min} = \text{Min}\{V_{ij}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, J\}$, $V_{\max} = \text{Max}\{V_{ij}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, J\}$, and ς is the distinguishing coefficient with $\varsigma \in [0, 1]$. ς expands or compresses the range of the grey relational coefficient. We set it to 0.5 in this study.

Step 4: Calculate the grey relational grade:

$$\Gamma\left(Y_0, Y_j\right) = \sum_{i=1}^{n} w_i \gamma\left(y_{i0}, y_{ij}\right) \tag{31}$$

In formula (31), $Y_j = y_{1j}, y_{2j}, \ldots, y_{nj}$ represents the jth alternative, and w_i is the weight of the ith criterion.

Step 5: Rank feature selection methods by maximizing $\Gamma\left(X_0, X_j\right)$.

3.3.4. Weighted sum method (WSM)

The WSM is the simplest and most straightforward MCDM method. We perform it as follows.

Step 1: Calculate the scores of the benefit and cost criteria:

$$A_j^{benefit} = \sum_{i \in \text{benefit criteria}} w_i x_{ij} \tag{32}$$

$$A_j^{cost} = \sum_{i \in cost \text{ criteria}} w_i x_{ij} \tag{33}$$

In formula (32) and (33), $i = 1, \ldots$ n and $j = 1, \ldots, J$ represent the indices of criteria and alternatives, respectively, and x_{ij} is the value of the ith criterion for the jth alternative.

Step 2: Calculate the WSM scores:

$$A_j^{WSM-scores} = A_j^{benefit} - A_j^{cost} \tag{34}$$

Step 3: Rank the feature selection methods by maximizing $A_j^{WSM-scores}$.

We applied the normalization methods from TOPSIS and GRA (step 1 in both algorithms) prior to calculating the WSM scores. WSM-N1 and WSM-N2 refer to the WSM results with TOPSIS and GRA normalization, respectively.

3.3.5. Preference ranking organization method for enrichment of evaluations (PROMETHEE)

PROMETHEE is an MCDM method based on pairwise comparisons. We use PROMETHEE II in our work and calculate it as follows.

Step 1: Transform all criteria into benefit criteria. For "native" benefit criteria, this is:

$$y_{ij} = x_{ij} \tag{35}$$

For cost criteria, this is:

$$y_{ij} = \max_j x_{ij} - x_{ij} \tag{36}$$

In formula (35) and (36), $i = 1, \ldots$ n and $j = 1, \ldots, J$ represent the indices of criteria and alternatives, respectively, and x_{ij} is the value of the ith criterion for the jth alternative.

Step 2: Select the appropriate preference function and parameters. In our work, we use the linear preference function

$$p_i\left(a, b\right) = \begin{cases} 0, if a_i - b_i \le s_1 \\ 1, if a_i - b_i > s_2 \\ \frac{a_i - b_i - s_1}{s_2 - s_1}, otherwise \end{cases} \tag{37}$$

In formula (37), $a, b \in A$ represent two alternatives, and a_i and b_i are the values of the ith criterion for a and b. We choose different parameters for different criteria. Because relatively small differences in classification performance result in significant differences in the final measures, we set $s_1 = 0.01$ and $s_2 = 0.1$ for accuracy, AUC, MAE, TPR and TNR. For stability, runtime, training time and test time, we set $s_{1i} = \min_j x_{ij}$ and $s_{2i} = \max_j x_{ij}$, which are similar to the GRA normalization. We have tested other values of s_1 and s_2 for classification performance measures ($s_1 \in [0.005, 0.05]$ and $s_2 \in [0.05, 0.2]$) but we observed no significant changes in the evaluation of feature selection methods using PROMOTHEE. Because small changes have little impact on the results, we show the evaluation results when $s_1 = 0.01$ and $s_2 = 0.1$.

Step 3: Define and calculate aggregated preference indices for each pair of alternatives:

$$\begin{cases} \pi\left(a, b\right) = \sum_{i=1}^{n} p_i\left(a, b\right) w_i \\ \pi\left(b, a\right) = \sum_{i=1}^{n} p_i\left(b, a\right) w_i \end{cases} \tag{38}$$

Step 4: Define and calculate the positive and the negative outranking flows as follows:

The positive outranking flow is:

$$\phi^{+}\left(a\right) = \frac{1}{n-1} \sum_{x \in A, x \ne a} \pi(a, x) \tag{39}$$

The negative outranking flow is:

$$\phi^{-}\left(a\right) = \frac{1}{n-1} \sum_{x \in A, x \ne a} \pi(x, a) \tag{40}$$

Step 5: Compute the net outranking flow for each alternative:

$$\phi\left(\text{a}\right) = \phi^{+}\left(a\right) - \phi^{-}\left(a\right) \tag{41}$$

Step 6: Rank feature selection methods by ϕ.

4. Experiment

In this section, we present our experiment and results of our validation of our used MCDM-based evaluation method. First, we briefly describe the ten text classification datasets, then we describe the experimental process, and finally, we present results and discussion.

4.1. Text classification datasets

Our experiment used 10 text classification datasets with the properties as shown in Table 2. The number of features in each dataset was always much larger than the number of samples. Thus, all datasets have a small number of samples and a large number of dimensions as described in the opening of our paper. The following paragraphs elaborate further on each dataset:

Table 2
Properties of the 10 experimental datasets.

Datasets	No. of classes	No. of samples	No. of features
Pang & Lee dataset 1 (PL1)	2	1,000	3,890
Pang & Lee dataset 2 (PL2)	2	1,000	3,873
Imdb dataset (I)	2	1,000	15,779
Farm-ads dataset (F)	2	1,000	18,554
Spam dataset (S)	2	1,000	2,402
20-newsgroup dataset 1 (20NG1)	20	1,000	26,558
20-newsgroup dataset 2 (20NG2)	20	1,000	27,973
Cade dataset 1 (C1)	12	1,000	17,764
Cade dataset 2 (C2)	12	1,000	16,783
Reuter8 dataset (R)	8	1,000	7,100

4.1.1. Pang & Lee dataset

The Pang & Lee dataset is a classic, widely used sentiment analysis dataset collected by Pang and Lee [54]. There are two classes in the dataset, which has 5331 positive and 5331 negative sentences. Because the full-unmodified dataset has a large number of samples, it does not meet our requirement of a dataset with a small number of samples. To solve this problem, we created two derivative subsets by randomly choosing 1000 samples from the whole. We named the derivative sets as Pang & Lee dataset 1 and Pang & Lee dataset 2.

4.1.2. Imdb dataset

The Imdb dataset is a sentiment analysis dataset collected by Maas et al. [55]. This dataset has 12,500 positive and 12,500 negative reviews. Similar to the Pang & Lee dataset, we randomly choose 1000 samples from the whole dataset to create our test dataset.

4.1.3. Farm-ads dataset

The farm-ads dataset was collected by Mesterharm and Pazzani [56]. The binary labels of this dataset indicate whether the content owner approves of the ad. There are 2210 "approved" samples and 1933 "not approved" samples in the full dataset. Our test dataset consisted of 1000 randomly chosen samples from the whole.

4.1.4. Spam dataset

The spam dataset is a collection of SMS labeled messages from mobile phones, with 4841 ham (i.e., legitimate) and 733 spam messages. This dataset was collected by Almeida and Yamakami [57]. As with the others, we randomly chose 1000 samples from the whole for our test.

4.1.5. 20-newsgroup dataset

The 20-newsgroups dataset is a classical multi-classification dataset for text classification collected by Joachims [58]. It consists of 20 classes, each one representing one Usenet group. There are close to 1000 instances in each class. Twice, we randomly chose 1000 samples to create 20-newsgroups dataset 1 and 20-newsgroups dataset 2.

4.1.6. Cade dataset

The documents in the Cade dataset[1] correspond to a subset of web pages extracted from the CAD Web Directory, which points to Brazilian web pages classified by human experts. It is a large dataset of 40,983 samples, but it is unbalanced, with the number of instances in each class ranging from 625 to 8473. Twice, we randomly chose 1000 samples to create Cade dataset 1 and Cade dataset 2.

4.1.7. Reuter21578 dataset

Similar to the 20-newsgroup dataset, the Reuter21578 dataset is another benchmark dataset for feature selection collected from the Reuters newswire. We used the Reuter8[1] dataset in our experiments. Reuter8 is also unbalanced, with 8 classes and 7,674 instances. Twice, we randomly chose 1000 samples.

4.2. Experimental design

To build a set of changing datasets to calculate stability and other measures, we used 5-fold cross-validation. We divided each dataset into 5 equal parts. Each part was used as a test set with the other 4 combined as the training set. Both feature selection and classifier training were based on the training set. We average the runtime, training time, test time, stability, TPR, TNR, accuracy, AUC, and MAE results over the 5 runs. We automated our experiment using the Python programming language as depicted in Fig. 3.

Input: Text classification dataset.

Output: Rankings of feature selection methods.

Step 1: *Dataset preparation.* This step contains includes of stop words and word segmentation. We used the English stop words from the CSDN.[2] We used jieba, a well-known system in China, to perform word segmentation.

Step 2: *Feature selection.* In this step, each word is an alternative feature. First, we calculated the feature weight of each word using the ten feature selection methods. Second, we sorted the words according to their feature weights. Third, we set the thresholdn, which represents the number of features, and selected the Top-n features as the final feature set. In our experiment, we tested all datasets with n set to 100, 200, 500, 1000 and 2000.

Step 3: *Runtime and stability calculation.*

Step 4: *Text representation and classification.* We used the term frequency–inverse document frequency (TF–IDF) method to transform the text to the vector space model and then used SVM [59,60], KNN and NB [61] algorithms for classification. These three classifiers are well-suited to evaluate feature selection performance as all are strongly influenced by the selected features and regularly used for text classification.

Step 5: *Training time, test time, TPR, TNR, accuracy, AUC, and MAE calculation.*

Step 6: *Weight calculation.* We calculated weights using the rank sum weight method [62]:

$$w_i = \frac{n - r_i + 1}{\sum_{k=1}^{n}(n - r_k + 1)}$$

, where r_i is the rank of the ith criterion. We obtained the value of r_i by sending questionnaires to experts and averaging their responses. By means of sending questionnaires to multiple experts,

[1] http://ana.cachopo.org/datasets-for-single-label-text-categorization

[2] http://blog.csdn.net/shijiebei2009/article/details/39696523.

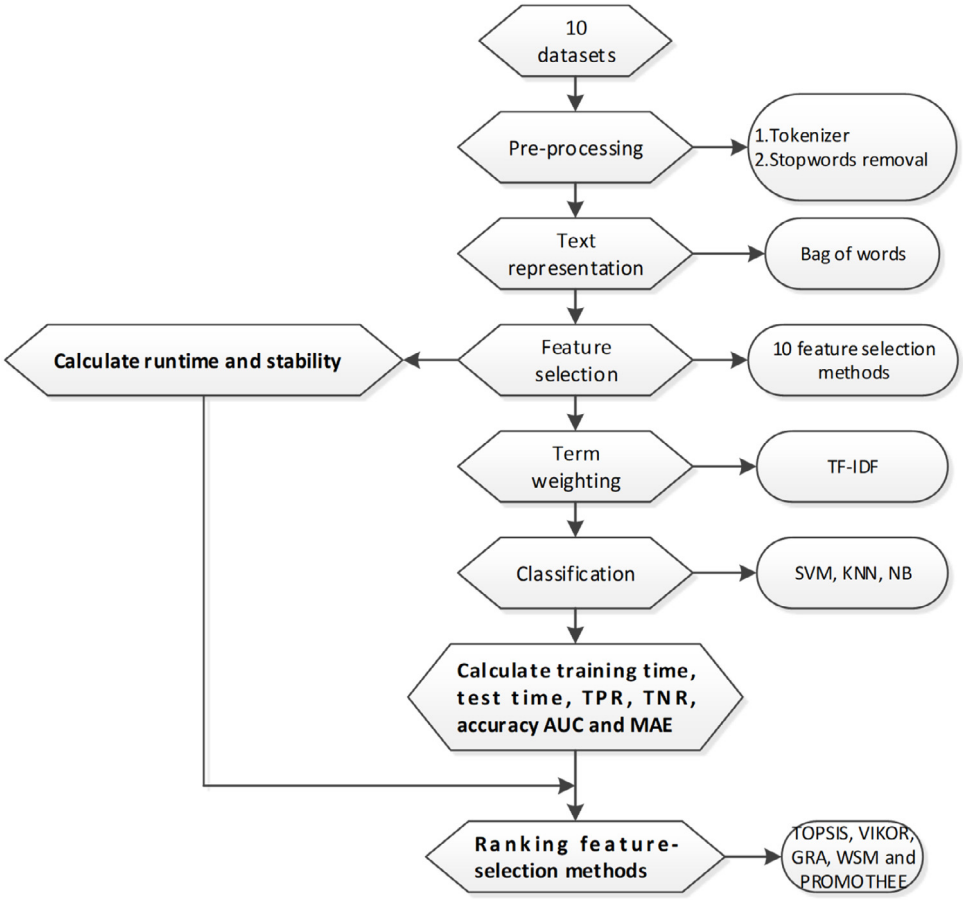

Fig. 3. Experimental process.

we can avoid strategic weight manipulation [63] in MCDM. We chose the RS weight method for two reasons.

First, it is subjective, which is better than objective methods in our work. Criteria weights stabilize with the number of features making it convenient for comparing feature-selection methods with different numbers of features.

Second, this method is linear and changes gradually, which is better than a steeper approach. This method does not give very high weights to the top criteria and thus increases the effect of the other criteria.

Step 7: *Ranking generation.* We generated the rankings of the feature selection methods using TOPSIS, VIKOR, GRA, WSM, and PROMOTHEE.

4.3. Results

The various performance numbers from the 10 feature selection methods with 5 features and 5 classifiers are different would require 150 rows for each dataset. Because that volume of information is not helpful, we present the results of 20-newsgroup dataset 1 and Pang & Lee dataset 1 with 1000 features using an SVM classifier as representative for showing our results. The complete results for each dataset can be found in the appendix. Table 3 summarizes the rankings from experts and different criteria weights. Tables 4 and 5 show the performance results for each measure for the 10 feature selection methods for the two datasets. Tables 6 and 7 show the MCDM rankings of the 10 feature selection methods for 20-newsgroup dataset 1 and Pang & Lee dataset 1, with the accuracy-based rankings included for comparison.

Table 3
The rankings and weights for the different evaluation criteria.

	Binary classification		Multi-class classification	
	Rankings	Weights	Rankings	Weights
Runtime	7	0.0667	5	0.1071
Training time	8	0.0444	6	0.0714
Test time	9	0.0222	7	0.0357
Stability	3	0.1556	3	0.1786
TPR	4	0.1333	Null	Null
TNR	6	0.0889	Null	Null
Accuracy	1	0.2000	1	0.2500
AUC	2	0.1778	2	0.2143
MAE	5	0.1111	4	0.1429

4.3.1. Rankings and weights of criteria

Column 1 in Table 3 shows the measures. Columns 2 and 3 show the rankings and criteria weights for binary classification. Columns 4 and 5 show the rankings and criteria weights for multi-class classification. The most important criterion is classification performance, followed by stability and efficiency. These results are consistent with our predictions. Although the purpose of a feature selection method in the classification process is to improve classification performance and reduce the time required, improving classification performance is always the most important goal of feature selection research. The instability that occurs with datasets with a small number of samples and high dimensionality, which can directly affect classification performance, makes stability the second most important criterion for evaluating feature selection methods.

Table 4
Performance of each measure with each feature selection method on the Pang & Lee dataset 1 with 1000 features using an SVM classifier.

FS method	Accuracy	AUC	TPR	TNR	MAE	Stability	Runtime (s)	Training time (s)	Test time (s)
DF	0.6080	0.5206	0.5684	0.6438	0.4906	0.6354	0.0690	0.1681	0.0056
IG	0.6100	0.5131	0.6884	0.5390	0.4909	0.5089	0.1133	0.1203	0.0030
GI	0.6350	0.4720	0.6484	0.6229	0.5139	0.6249	0.1163	0.1619	0.0050
DFS	0.6050	0.5130	0.6632	0.5524	0.4947	0.5153	0.1127	0.1267	0.0030
ECE	0.6100	0.5207	0.6884	0.5390	0.4858	0.5089	0.1161	0.1219	0.0028
CDM	0.5670	0.4971	0.7221	0.4267	0.5041	0.5120	0.1125	0.1069	0.0022
CHI	0.6030	0.5959	0.6716	0.5410	0.4393	0.5117	0.1125	0.1275	0.0030
OR	0.5670	0.4747	0.7221	0.4267	0.5149	0.5120	0.1139	0.1075	0.0020
MI	0.5280	0.4570	0.0168	0.9905	0.5008	0.5214	0.1205	0.0424	0.0010
WLLR	0.5980	0.5280	0.7516	0.4590	0.4818	0.5106	0.1223	0.1103	0.0026

Table 5
Performance of each measure with each feature selection method on the 20-newsgroup dataset 1 with 1000 features and an SVM classifier.

FS method	Accuracy	AUC	MAE	Stability	Runtime (s)	Training time (s)	Test time (s)
DF	0.8684	0.5150	0.0931	0.8391	0.7772	3.5869	0.0972
IG	0.8861	0.4988	0.0953	0.6256	1.3953	2.2299	0.0602
GI	0.9240	0.5321	0.0921	0.5445	1.3825	1.9351	0.0550
DFS	0.9011	0.4912	0.0966	0.5334	1.4131	0.8780	0.0234
ECE	0.8861	0.5020	0.0942	0.6194	1.4005	1.8297	0.0484
CDM	0.8169	0.4845	0.0969	0.4637	1.4035	0.4968	0.0112
CHI	0.9091	0.4919	0.0965	0.5051	1.4273	0.6479	0.0188
OR	0.4661	0.4922	0.0954	0.4270	1.4197	0.2525	0.0054
MI	0.0700	0.4990	0.0951	0.4891	1.4881	0.0932	0.0038
WLLR	0.9240	0.5012	0.0946	0.4511	1.5163	0.3528	0.0106

Table 6
MCDM rankings of the 10 feature selection methods on the Pang & Lee dataset 1 with 1000 features and an SVM classifier.

FS method	TOPSIS ranking	VIKOR ranking	GRA ranking	WSM_N1 ranking	WSM_N2 ranking	PROMETHEE ranking	Accuracy ranking
DF	7	1	2	2	1	2	4
IG	4	7	6	5	6	7	2
GI	6	3	3	6	3	4	1
DFS	5	4	7	7	7	3	5
ECE	3	5	5	4	4	8	2
CDM	8	8	8	8	8	5	8
CHI	1	2	1	1	2	1	6
OR	9	9	9	9	9	9	8
MI	10	10	10	10	10	10	10
WLLR	2	6	4	3	5	6	7

Table 7
MCDM rankings of the 10 feature selection methods on the 20-newsgroup dataset 1 with 1000 features and an SVM classifier.

FS method	TOPSIS ranking	VIKOR ranking	GRA ranking	WSM_N1 ranking	WSM_N2 ranking	PROMETHEE ranking	Accuracy ranking
DF	8	1	2	5	1	2	7
IG	7	4	6	8	5	4	5
GI	6	2	1	7	2	1	1
DFS	1	7	7	3	7	5	4
ECE	3	3	4	6	3	3	6
CDM	5	9	8	4	8	8	8
CHI	2	6	5	2	6	6	3
OR	9	8	9	9	9	10	9
MI	10	10	10	10	10	9	10
WLLR	4	5	3	1	4	7	2

4.3.2. Performance of each criterion

We observe that the results of feature selection methods according to all measures differ, such that no one method achieve best results on all criteria. Thus, the choice of feature selection methods requires trade-offs among multiple criteria. We used MCDM methods to solve this problem.

4.3.3. MCDM rankings for each feature selection method

4.3.3.1. Comparison of the rankings between MCDM based methods and accuracy based method. Tables 6 and 7 show that the accuracy rankings always differ from the MCDM rankings. For example, the GI method ranked first according to accuracy but lower with the MCDM methods. While the accuracy of GI was good, the poor performance of GI according to the AUC, TPR, runtime, training time, and test time measures lowered its ranking. The same trend happened with the other feature selection methods. Therefore, when compared with accuracy-based rankings, MCDM-based rankings offer a more balanced evaluation.

4.3.3.2. Comparison of the rankings among 5 MCDM based methods. Referring again to Tables 5 and 6, rankings between the 5 MCDM-based evaluations conflict with each other. For example, DF was ranked poor by TOPSIS but good by the other MCDM methods in

the test. Thus, we must still determine the most representative MCDM method. In this section, we analyze the differences among the MCDM methods.

(1) Score calculation

The 5 MCDM methods have three different approaches for calculating scores. TOPSIS, VIKOR, and GRA use the general "closeness to the ideal" metric to rank alternatives. WSM used simple additive weighting. PROMOTHEE used pair comparison. In our view, pair comparison is more suitable for our work. If the accuracy, AUC, MAE, TPR, or TNR of a feature selection method is 0.1 higher than another method, the former is obviously superior to the latter. A pair comparison is the only approach that considers this advantage. Thus, we find PROMOTHEE is the best approach for calculating scores.

(2) Normalization

Three kinds of normalization operations have been applied in our tests:, min–max, l2 normalization, and others. TOPSIS and WSM_N1 both perform l2 normalization, $\frac{x_{ij}}{\sqrt{\sum_{j=1}^{J} x_{ij}^2}}$. VIKOR, GRA, and WSM_N2 perform min–max normalization, either $\frac{x_{ij}-\min\{x_{ij},j=1,2,\ldots J\}}{\mathrm{Max}\{x_{ij},j=1,2,\ldots J\}-\mathrm{Min}\{x_{ij},j=1,2,\ldots J\}}$ or $\frac{\max\{x_{ij},j=1,2,\ldots J\}-x_{ij}}{\max\{x_{ij},j=1,2,\ldots J\}-\min\{x_{ij},j=1,2,\ldots J\}}$. PROMOTHEE performs normalization using a preference function.

Min–max normalization increases the differences between two alternative criteria and produces a different result compared with l2 normalization and its calculated distance from an ideal or negative ideal solution. As $\sqrt{\sum_{j=1}^{J} x_{ij}^2} > \max\left\{x_{ij}, j=1,2,\ldots,J\right\} - \min\left\{x_{ij}, j=1,2,\ldots,J\right\}$, there exist $\left|\frac{x_{ij}-x_{il}}{\sqrt{\sum_{j=1}^{J} x_{ij}^2}}\right| < \left|\frac{x_{ij}-x_{il}}{\max\{x_{ij},j=1,2,\ldots J\}-\min\{x_{ij},j=1,2,\ldots J\}}\right|$. This makes the advantage or disadvantage more obvious. For example, from Table 6, the normalized accuracy of DF was 0.3239 using l2 but 0.7477 using min–max, and the normalized accuracy of MI was 0.2812 using l2 but 0 using min–max. DF was only slightly better than MI in accuracy using l2, as $0.3239 - 0.2812 = 0.0427$ but was much better than MI in accuracy using min–max, as $0.7477 - 0 = 0.7477$. With its use of pair comparisons, PROMOTHEE achieved normalization results similar to those obtained with min–max. This explains why TOPSIS ranked DF at the bottom while other MCDM methods rated it much better.

For our purposes, accuracy remains the most important criterion, and we cannot ignore any but the smallest differences in accuracy. Thus, we find normalization with min–max and preference functions better than l2 normalization.

In general, MCDM methods performing pair comparisons or min–max normalization are better than others. Therefore, we find the most suitable MCDM method to be PROMOTHEE.

4.3.3.3. PROMOTHEE rankings for each feature selection method. In view of our finding that PROMOTHEE is the best MCDM method according to our tests, we now examine the ranking results for the various feature selection methods. Figs. 4 through 13 present the PROMOTHEE rankings for the 10 feature selection methods. In all the figures, the X-axis represents datasets and number of features, the Y-axis represents the classifiers, and the Z-axis corresponds to the rankings by PROMOTHEE. Considering that the image is small and the scale is too dense to see clearly, we omitted the scale of feature number. We also provide an Excel spreadsheet to show specific data in the appendix.

(1) DF

Fig. 4 summarizes the DF rankings from PROMOTHEE. A common finding from these results is that the DF method performances varies with the dataset and number of features. However, the overall performance of DF is good according to PROMOTHEE. While DF's accuracy is not good in all situations, its AUC, MAE,

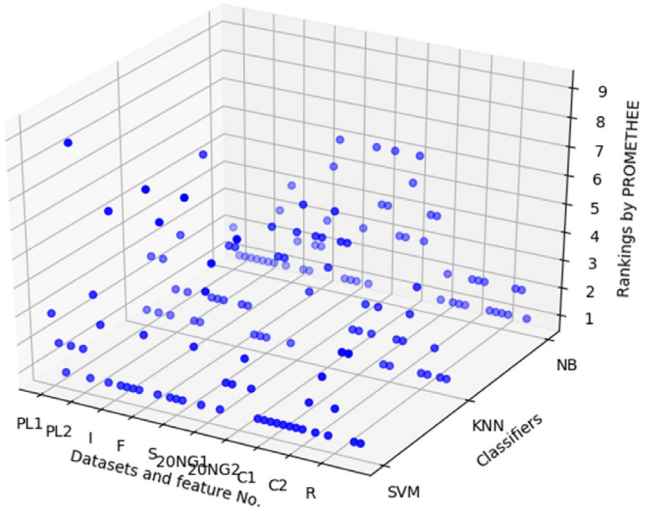

Fig. 4. PROMOTHEE rankings for DF.

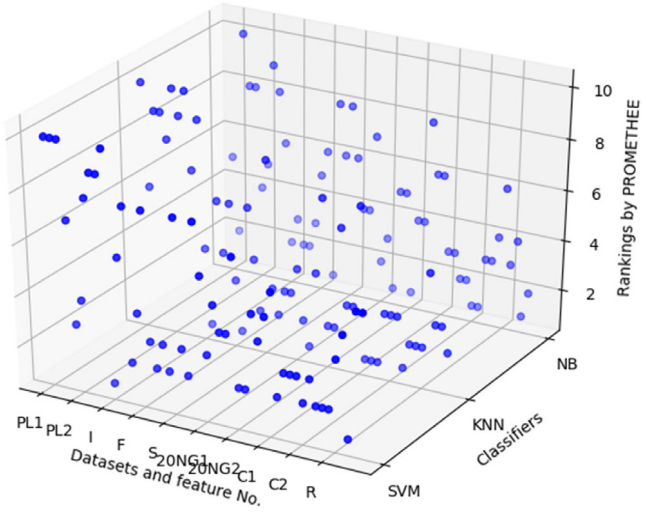

Fig. 5. PROMOTHEE rankings for IG.

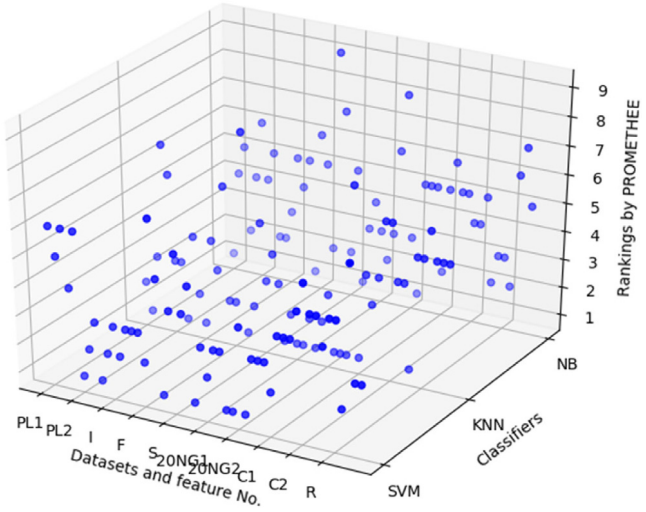

Fig. 6. PROMOTHEE rankings for GI.

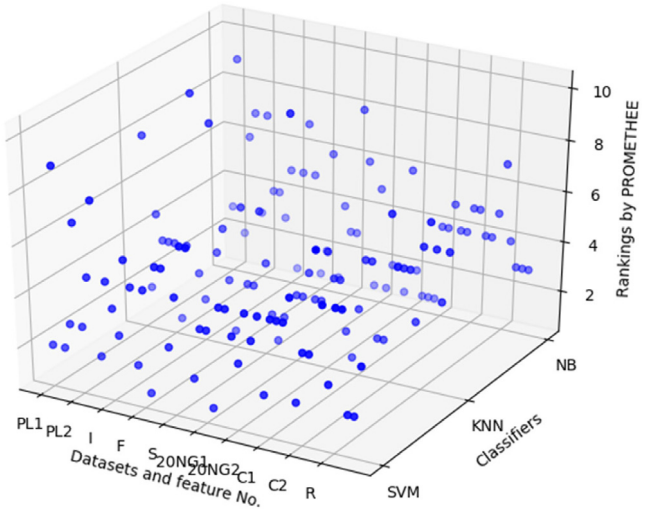

Fig. 7. PROMOTHEE rankings for DFS.

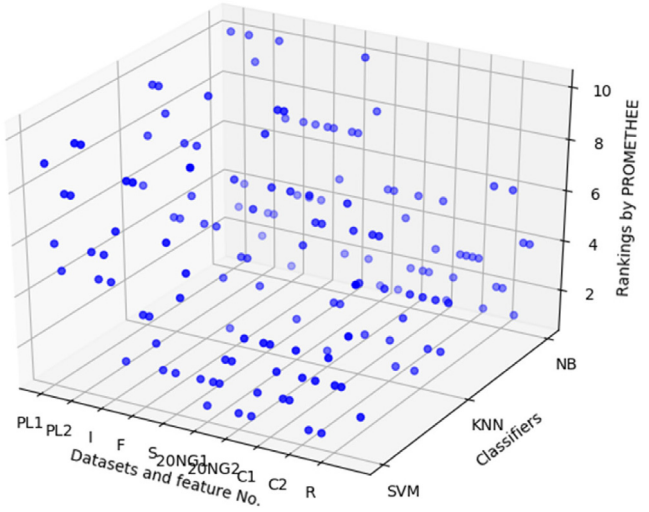

Fig. 8. PROMOTHEE rankings for ECE.

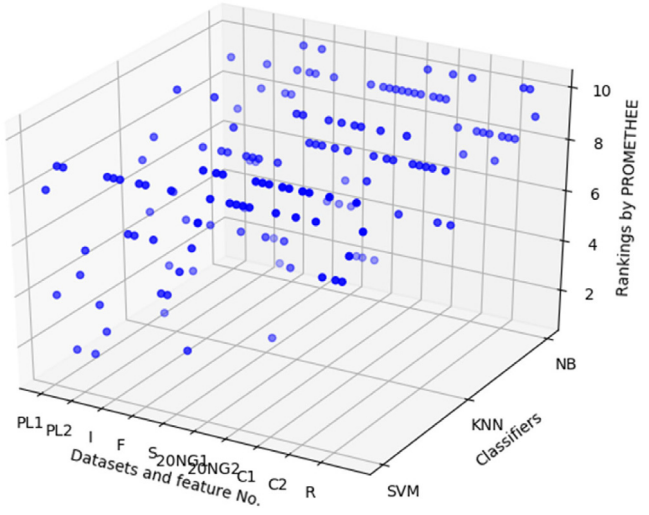

Fig. 9. PROMOTHEE rankings for CDM.

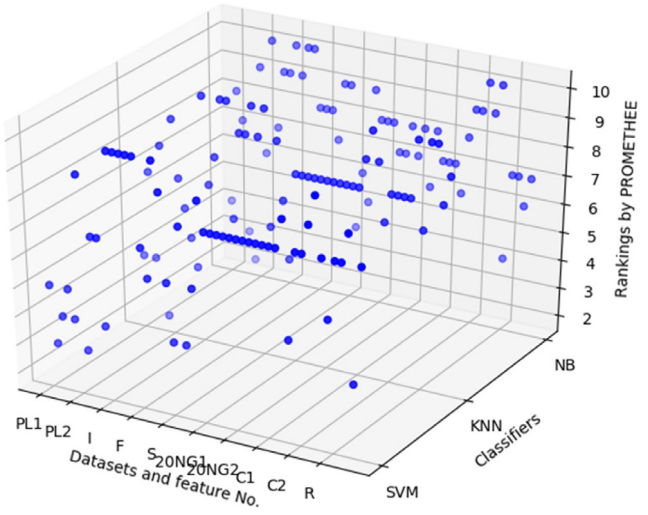

Fig. 10. PROMOTHEE rankings for CHI.

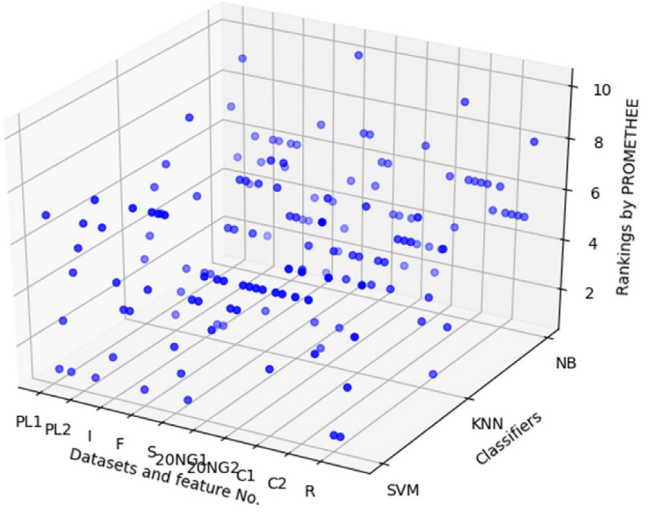

Fig. 11. PROMOTHEE rankings for OR.

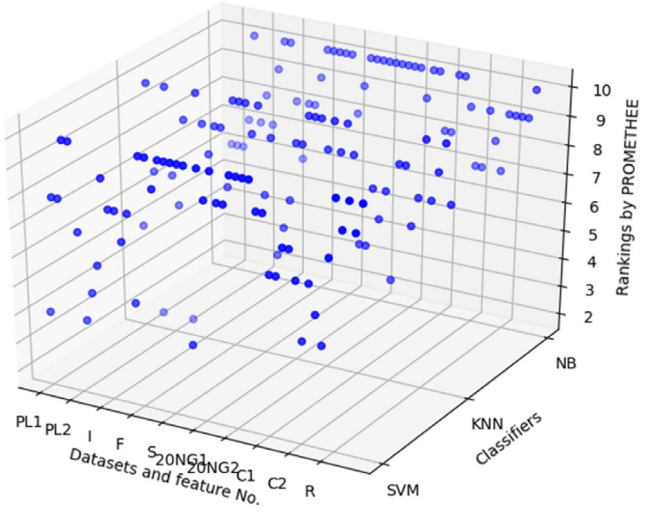

Fig. 12. PROMOTHEE rankings for MI.

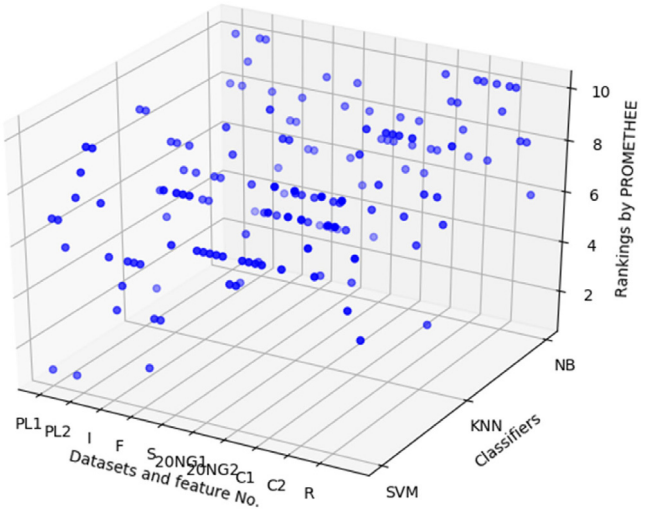

Fig. 13. PROMOTHEE rankings for WLLR.

stability, and runtime performance is consistently good. Thus, DF is a relatively good feature selection method for text classification on datasets with small numbers of samples.

(2) IG

Fig. 5 summarizes the IG rankings according to PROMOTHEE. Its distribution is scattered, indicating unstable performance. IG is not recommended.

(3) GI

Fig. 6 shows the GI rankings from PROMOTHEE, which are in the top five in most cases. Even if GI is not the best feature selection method, it is a good alternative.

(4) DFS

The DFS rankings from PROMOTHEE are summarized in Fig. 7. Being similar to GI's, DFS is also a good alternative.

(5) ECE

Fig. 8 provides the ECE rankings. ECE performs badly with binary classification but good in multi-class classification. ECE is a recommended method for multi-class classification and not recommended otherwise

(6) CDM

Fig. 9 shows CDM rankings, which indicate poor performance in most circumstances. CDM is not recommended.

(7) CHI

The CHI rankings in Fig. 10 show performance similar to CDM, making CHI not recommended.

(8) OR

Fig. 11 shows the OR rankings. While the rankings place OR in the top six methods in most cases, this only qualifies OR as minimally acceptable. It could be considered an alternative under specific circumstances.

(9) MI

The MI rankings in Fig. 12 show poor performance in most cases, making it not recommended.

(10) WLLR

The WLLR rankings in Fig. 13 are similar to MI, making WLLR not recommended also.

In summary, the PROMOTHEE analysis of all methods across all test datasets shows that DF is the most recommended method, with GI, DFS, and OR being acceptable alternatives, and that ECE is suitable for multi-class use but not binary classification. None of the others are recommended on the basis of poor or unstable performance.

5. Conclusions

The problem of small samples and high dimensionality for text classification makes the evaluation of feature selection methods difficult because it involves multiple criteria. A better evaluation method that takes multiple criteria into consideration is needed. To solve this problem, we have used an MCDM-based evaluation method to assess the performance of feature selection methods for text classification on datasets with small numbers of samples. After obtaining features and text classification results from 10 common feature selection methods and three classifiers, the selection methods were evaluated according to classification performance, stability and efficiency. Afterwards, five MCDM methods ranked the feature selection methods by considering all the measures. We validated the effect of the five MCDM methods with an experiment combining 10 feature selection methods, 9 performance criteria for binary classification, 7 performance criteria for multi-class classification, 5 MCDM methods, and 10 text classification datasets.

The results show that no feature selection method achieved the best performance on all criteria regardless of the number of features and the chosen classifier. Thus, it was necessary to use more than one performance measure to evaluate the feature selection methods. From the various results, we have provided our recommendation of feature selection methods, with DF being the overall preferred method.

While our tests find PROMOTHEE to be the MCDM most suited for evaluating classifier performance, there are many other MCDM methods we did not analyze. Furthermore, our experiment tested only 10 datasets. Detailed analyses of other MCDM methods and experiments with more datasets are needed in future research.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105836.

Acknowledgment

This research was supported in part by grants from the National Natural Science Foundation of China (#717250013).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2019.105836.

References

[1] L. Gao, S. Zhou, J. Guan, Effectively classifying short texts by structured sparse representation with dictionary filtering, Inform. Sci. 323 (2015) 130–142.
[2] K.N. Junejo, A. Karim, M.T. Hassan, et al., Terms-based discriminative information space for robust text classification, Inform. Sci. 372 (2016) 518–538.
[3] F. Sebastiani, Machine learning in automated text categorization, Acm Comput. Surv. 34 (1) (2001) 1–47.
[4] Y. Kong, M. Owusu-Akomeah, H.A. Antwi, et al., Evaluation of the robusticity of mutual fund performance in ghana using enhanced resilient backpropagation neural network (ERBPNN) and fast adaptive neural network classifier (FANNC), Financial Innov. 5 (1) (2019) 10.
[5] X. Zhong, D. Enke, Predicting the daily return direction of the stock market using hybrid machine learning algorithms, Financial Innov. 5 (1) (2019) 4.
[6] Hanyang Peng, Yong Fan, Feature selection by optimizing a lower bound of conditional mutual information, Inform. Sci. 418–419 (2017) 652–667.

[7] C. Shang, M. Li, S. Feng, et al., Feature selection via maximizing global information gain for text classification, Knowl.-Based Syst. 54 (4) (2013) 298–309.
[8] Z. Zeng, H. Zhang, R. Zhang, et al., A novel feature selection method considering feature interaction, Pattern Recognit. 48 (8) (2015) 2656–2666.
[9] P. Somol, J. Novovicová, Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 1921–1939.
[10] D. Dernoncourt, B. Hanczar, J.D. Zucker, Analysis of feature selection stability on high dimension and small sample data, Comput. Statist. Data Anal. 71 (1) (2014) 681–693.
[11] G. Kou, Y. Peng, G. Wang, Evaluation of clustering algorithms for financial risk analysis using MCDM methods, Inform. Sci. 275 (11) (2014) 1–12.
[12] Y. Kuo, T. Yang, G.W. Huang, The use of grey relational analysis in solving multiple attribute decision-making problems, Comput. Ind. Eng. 55 (1) (2008) 80–93.
[13] J. Yang, Y. Liu, X. Zhu, et al., A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization, Inf. Process. Manage. Int. J. 48 (4) (2012) 741–754.
[14] Şerafettin Taşcı, T. Güngör, Comparison of text feature selection policies and using an adaptive framework, Expert Syst. Appl. 40 (12) (2013) 4871–4886.
[15] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc, 1997, pp. 412–420.
[16] C. Lee, G.G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Inf. Process. Manage. 42 (1) (2006) 155–165.
[17] W. Shang, H. Huang, H. Zhu, et al., A novel feature selection algorithm for text categorization, Expert Syst. Appl. 33 (1) (2007) 1–5.
[18] A.K. Uysal, S. Gunal, A novel probabilistic feature selection method for text classification, Knowl.-Based Syst. 36 (6) (2012) 226–235.
[19] H. Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowl.-Based Syst. 24 (7) (2011) 1024–1032.
[20] A. Rehman, K. Javed, H.A. Babri, et al., Relative discrimination criterion – A novel feature ranking method for text data, Expert Syst. Appl. 42 (7) (2015) 3670–3681.
[21] R.H.W. Pinheiro, G.D.C. Cavalcanti, T.I. Ren, Data-driven global-ranking local feature selection methods for text categorization, Expert Syst. Appl. 42 (4) (2015) 1941–1949.
[22] A.K. Uysal, An improved global feature selection scheme for text classification, Expert Syst. Appl. 43 (C) (2016) 82–92.
[23] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437.
[24] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognit. Lett. 30 (1) (2009) 27–38.
[25] C. Cortes, AUC optimization vs. error rate minimization, Adv. Neural Inf. Process. Syst. (2004) 313–320.
[26] S. Rosset, Model selection via the AUC, in: International Conference on Machine Learning, ACM, 2004, p. 89.
[27] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: ICML '06 : Proceedings of the, International Conference on Machine Learning, ACM Press, New York, Ny, Usa, 2006, pp. 233–240.
[28] K. Javed, S. Maruf, H.A. Babri, A two-stage Markov blanket based feature selection algorithm for text classification, Neurocomputing 157 (2015) 91–104.
[29] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Syst. Appl. 36 (3) (2009) 6843–6853.
[30] R. Neumayer, R. Mayer, K. Nørvåg, Combination of feature selection methods for text categorisation, in: Advances in Information Retrieval, Springer Berlin Heidelberg, 2011, pp. 763–766.
[31] H. Ogura, H. Amano, M. Kondo, Comparison of metrics for feature selection in imbalanced text classification, Expert Syst. Appl. 38 (5) (2011) 4978–4989.
[32] R.H.W. Pinheiro, G.D.C. Cavalcanti, R.F. Correa, et al., A global-ranking local feature selection method for text categorization, Expert Syst. Appl. 39 (17) (2012) 12851–12857.
[33] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: A study on high-dimensional spaces, Knowledge & Information Systems 12 (1) (2007) 95–116.
[34] A. Toloie-Eshlaghy, M. Homayonfar, MCDM methodologies and applications: A literature review from 1999 to 2009, Res.J. Int. Stud. (2011) 86–137.
[35] G. Li, G. Kou, Y. Peng, A group decision making model for integrating heterogeneous information, IEEE Trans. Syst. Man Cybern. Syst. (2016) 1–11.
[36] H. Zhang, G. Kou, Y. Peng, Soft consensus cost models for group decision making and economic interpretations, European J. Oper. Res. 277 (3) (2019) 964–980.
[37] E.K. Zavadskas, Z. Turskis, S. Kildienė, State of art surveys of overviews on MCDM/madm methods, Technol. Econ. Dev. Econ. 20 (1) (2014) 165–179.
[38] G. Kou, C. Lin, A cosine maximization method for the priority vector derivation in AHP, European J. Oper. Res. 235 (1) (2014) 225–232.
[39] G. Kou, D. Ergu, C. Lin, et al., Pairwise comparison matrix in multiple criteria decision making, Technol. Econ. Develop. Econ. 22 (5) (2016) 738–765.
[40] G. Kou, D. Ergu, J. Shang, Enhancing data consistency in decision matrix: Adapting hadamard model to mitigate judgment contradiction, European J. Oper. Res. 236 (1) (2014) 261–271.
[41] Y. Liu, J.W. Bi, Z.P. Fan, Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory, Inf. Fusion 36 (2017) 149–161.
[42] Y. Peng, G. Kou, G. Wang, et al., FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms, Omega 39 (6) (2011) 677–689.
[43] R. Singh, H. Kumar, R.K. Singla, TOPSIS based multi-criteria decision making of feature selection techniques for network traffic dataset, Int. J. Eng. Technol. 5 (6) (2013) 4598–4604.
[44] M.A. Alias, S.Z.M. Hashim, S. Samsudin, Multi crteria decision making and its applications : A literature review, Jurnal Teknologi Maklumat (2008).
[45] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1–2) (2010) 1–39.
[46] Y. Peng, G. Wang, G. Kou, et al., An empirical study of classification algorithm evaluation for financial risk prediction, Appl. Soft Comput. 11 (2) (2011) 2906–2915.
[47] Gang Kou, Yanqun Lu, Yi Peng, et al., Evaluation of classification algorithms using mcdm and rank correlation, Int. J. Inf. Technol. Decis. Mak. 11 (01) (2012) 197–225.
[48] Jingnian Chen, Houkuan Huang, Shengfeng Tian, et al., Feature selection for text classification with Naïve Bayes, Expert Syst. Appl. 36 (3) (2009) 5432–5435.
[49] K. Nigam, A.K. Mccallum, S. Thrun, et al., Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2–3) (2000) 103–134.
[50] Y. Saeys, T. Abeel, Y.V.D. Peer, Robust feature selection using ensemble feature selection techniques, in: Machine Learning and Knowledge Discovery in Databases, European Conference, Ecml/pkdd 2008, Antwerp, Belgium, September (2008) 15-19, Proceedings, DBLP, 2008, pp. 313–325.
[51] C.L. Hwang, K. Yoon, Multiple Attribute Decision Making: Methods and Applications a State-of-the-Art Survey, Springer Science & Business Media, 2012.
[52] S. Opricovic, Multicriteria optimization of civil engineering systems, Fac. Civ. Eng. 2 (1) (1998) 5–21.
[53] S. Opricovic, G. Tzeng, Multicriteria planning of post earthquake sustainable reconstruction, Comput.-Aided Civ. Infrastruct. Eng. 17 (3) (2010) 211–220.
[54] P. Bo, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales (2005), pp. 115–124.
[55] A.L. Maas, R.E. Daly, P.T. Pham, et al., Learning word vectors for sentiment analysis, in: Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 142–150.
[56] C. Mesterharm, Pazzani M.J., Active learning using on-line algorithms, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 850–858.
[57] T.A. Almeida, A. Yamakami, Contributions to the study of SMS spam filtering: New collection and results, in: Proceedings of the 11th ACM Symposium on Document Engineering, ACM, 2011, pp. 259–262.
[58] T. Joachims, A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization, in: International Conference on Machine Learning, 1996.
[59] D. Selvamuthu, V. Kumar, A. Mishra, Indian stock market prediction using artificial neural networks on tick data, Financial Innov. 5 (1) (2019) 16.
[60] Y. Song, H. Wang, M. Zhu, Sustainable strategy for corporate governance based on the sentiment analysis of financial reports with CSR, Financial Innov. 4 (1) (2018) 2.
[61] B. Fazelabdolabadi, A hybrid Bayesian-network proposition for forecasting the crude oil price, Financial Innov. 5 (1) (2019) 30.
[62] W.G. Stillwell, D.A. Seaver, W. Edwards, A comparison of weight approximation techniques in multiattribute utility decision making, Organ. Behav. Hum. Perform. 28 (1) (1981) 62–77.
[63] Yating Liu, Yucheng Dong, Haiming Liang, Francisco Chiclana, Enrique Herrera-Viedma, Multiple attribute strategic weight manipulation with minimum cost in a group decision making context with interval attribute weights information, IEEE Trans. Syst. Man Cybern. Syst. (2018) 1–12.