



Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-TOPSIS classification approach



Xiaoqian Zhu^{a,b}, Jianping Li^{a,*}, Dengsheng Wu^a, Haiyan Wang^c, Changzhi Liang^{a,b}

^a Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, China

^b University of Chinese Academy of Sciences, Beijing 100190, China

^c Jiangsu Province Institute of Quality & Safety Engineering, Nanjing 210046, China

ARTICLE INFO

Article history:

Received 3 October 2012

Received in revised form 21 June 2013

Accepted 2 August 2013

Available online 14 August 2013

Keywords:

Credit scoring

TOPSIS

Credit classification

Credit risk

Support vector machine

Bank risk evaluation

ABSTRACT

Accuracy, complexity and interpretability are very important in credit classification. However, most approaches cannot perform well in all the three aspects simultaneously. The objective of this study is to put forward a classification approach named C-TOPSIS that can balance the three aspects well. C-TOPSIS is based on the rationale of TOPSIS (Technique for Order Preference by Similarity to Ideal Solution). TOPSIS is famous for reliable evaluation results and quick computing process and it is easy to understand and use. However, it is a ranking approach and three challenges have to be faced for modifying TOPSIS into a classification approach. C-TOPSIS works out three strategies to overcome the challenges and retains the advantages of TOPSIS. So C-TOPSIS is deduced to have reliable classification results, high computational efficiency and ease of use and understanding. Our findings in the experiment verify the advantages of C-TOPSIS. In comparison with 7 popular approaches on 2 widely used UCI credit datasets, C-TOPSIS ranks 2nd in accuracy, 1st in complexity and is in 1st rank in interpretability. Only C-TOPSIS ranks among the top 3 in all the three aspects, which verifies that C-TOPSIS can balance accuracy, complexity and interpretability well.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Credit decision making has long attracted a great deal of attention from both academic researchers and practitioners [31]. Moreover, the rapid growth of the credit industry and the subprime crisis in 2007 has pushed this issue to a new climax [44]. Credit classification is one of the key analytical techniques in credit decision making. Credit classification approach is used to categorize applicants as either approved or rejected, on the basis of applicants' information such as income, bank balance, profession, family background and educational background.

Classification accuracy, computation complexity and results interpretability are generally accepted as the three main aspects in the evaluation of a credit classification approach [12]. Firstly, acceptable classification accuracy is the basic and crucial requirement since even a little improvement in accuracy may translate into significant savings [43]. Many approaches have been studied to improve the accuracy of credit classification [19,40]. Next, low computational complexity is also a very important requirement in practice [14]. The objective of a credit classification approach is reduction of the cost of credit decision making. Excessive time

consumption is unworthy and can affect the profit of the firm significantly. Approaches that involve less computing time are more efficient and thus generate more profit for the banks or firms [13]. Lastly, good interpretability is another requirement that cannot be ignored. Inadequate interpretability of an approach can be a major drawback and causes a reluctance to use the approach. It goes even further: when credit has been denied to a customer, the Equal Credit Opportunity Act of the US requires that the financial institution provide specific reasons why the application is rejected. Indefinite and vague reasons for denial are illegal [23]. Therefore, only the approach that performs well in all the three aspects simultaneously is a satisfactory credit classification approach.

All kinds of mathematical approaches such as discriminant analysis, logistic regression, k-nearest neighbor classifier, decision tree, neural network and support vector machine (SVM) have been employed to credit classification up to now [7]. Most of the credit classification approaches can be categorized into two techniques. The former is traditional statistical technique which includes discriminant analysis, logistic regression, k-nearest neighbor classifier, decision tree and so on. The latter is artificial intelligence (AI) technique which includes neural network, support vector machine and so on [5,41].

* Corresponding author. Tel.: +86 10 59358805.

E-mail address: ljp@casipm.ac.cn (J. Li).

Statistical techniques are firstly employed to credit decision making. Linear discriminant analysis is the first approach that is applied to credit classification by Durand [9] who shows that the approach can produce good predictions of credit repayment. Afterwards, researches on proposing mathematic credit scoring approaches spring up, which demonstrate that discriminant analysis is a mediocre credit scoring approach, always performing neither outstandingly nor poorly among numerous approaches in many ways [21,22,27,35]. Decision tree is initially applied to credit scoring by Mehta [24]. The classification process of decision tree is very clear and its computational efficiency is relatively high, nevertheless its accuracy is not satisfactory [35]. k-Nearest neighbor, a nonparametric approach, is explored for credit scoring by Chatterjee and Barcun [4]. k-Nearest neighbor approach is characterized by its intuitive simplicity. Despite this merit, it has not been widely adopted in the credit scoring industry and one reason for this is its perceived computational demand [12]. Logistic regression is firstly employed to credit scoring by Wiginton [37] and is one of the most commonly used approaches in developing scorecard [7,11]. It gives superior results than discriminant analysis, however, neither approach is good enough [37]. Generally speaking, statistical approaches have the attractive feature of simplicity, but their accuracies are not sufficiently good to be cost effective for credit classification.

Later in the 1990s, with the development of information technology, the new era of AI approaches in credit decision making begins. Neural network approach has been successfully tested in many real-world data sets. Nevertheless, it is essentially black box and cannot permit the reasons why this approach has reached its decision. Besides, it takes much longer to train compared with other statistical approaches [12]. SVM, proposed by [6], has attracted wide attention because of its excellent accuracy performance. However, its parameter selection is time-consuming and its classification process is a black box that is not interpretable [8,11]. After that some revised SVM, such as least square support vector machine (LSSVM), is proposed [29] and introduced to credit classification. As a revised SVM, LSSVM performs better in computing time, however, its interpretability is still a problem. Besides, inductive learning, artificial neural networks, genetic algorithms, artificial immune system and other AI approaches also come into use [34]. Generally, the AI approach performs satisfactorily in accuracy, but it is time-consuming and its results are not interpretable in terms of original input variables. Above literature review shows that some approaches perform well in one or two aspects at most while bad in the remaining aspects. All in all, nearly none of the approaches can balance accuracy, complexity and interpretability.

The technique for order preference by similarity to ideal solution (TOPSIS), firstly developed by Hwang and Yong in 1981 [16], is a classical multi-criteria decision-making (MCDM) approach widely used in evaluation studies [28]. The evaluation results of TOPSIS are considered reliable. Besides, since only linear computation is involved, the computing process of TOPSIS is very time-saving. Last but not least, the evaluation results of TOPSIS are very easy to understand and explain. Since TOPSIS has the above three advantages, it is perceived that TOPSIS might be suitable for credit classification. However, TOPSIS is a ranking approach rather than classification approach, attributes of TOPSIS are distinguished intuitively and weights are always set subjectively by experts. The three challenges make it difficult to modify TOPSIS into a classification approach.

The objective of this paper is to propose a novel classification approach based on TOPSIS by overcoming the above-mentioned three challenges. The proposed approach retains the advantages of TOPSIS and is deduced to perform well in terms of accuracy, complexity and interpretability simultaneously. To the best of our knowledge, this study is the first paper to propose a systematic

classification approach based on the rationale of TOPSIS. The proposed approach is named C-TOPSIS and the capital letter C in C-TOPSIS in the acronym for classification.

This paper is organized as follows. Section 2 presents a review of TOPSIS in credit decision making. Section 3 introduces the proposed C-TOPSIS approach. Section 4 compares the accuracy, complexity and interpretability of C-TOPSIS with 7 popular approaches on two UCI credit datasets. Section 5 provides the conclusion and possible future directions.

2. TOPSIS in credit decision making

The ranking process of TOPSIS is similar to the credit scoring process. In TOPSIS, according to attributes, every alternative attains a score called closeness coefficient. Alternatives are ranked according to the closeness coefficient, the larger the closeness coefficient is, the more preferred the alternative is. In credit scoring, according to the attributes such as income, age and profession, the credit scoring model can output a credit score for every applicant. The larger the credit score is, the better the expected credit of the applicant is [30]. Moreover, TOPSIS is known for reliable evaluation results, quick computing process, and ease of use and understanding. So it is conceived that TOPSIS can probably be modified into a competitive credit classification approach which can balance accuracy, complexity and interpretability.

However, three features of TOPSIS make it difficult to modify it into a classification approach, that is, attributes are distinguished intuitively, weights are always set subjectively by experts and TOPSIS is a ranking approach rather than classification approach. Intuitive attribute distinguishing and subjective weight setting are prone to be affected by experience, knowledge and so on, and are not helpful for improving classification accuracy. In addition, after attaining the credit score, threshold is needed to classify the applicants. Therefore, three corresponding challenges are faced, that is, how to distinguish attributes objectively, how to set weights properly and how to determine the threshold reasonably.

Because of the existence of the three challenges, only a few researchers have associated TOPSIS with credit evaluation or classification [3]. Wu and Olson [38] develop a TOPSIS classifier and use it to evaluate financial performance of Canadian companies. In this TOPSIS classifier, the attributes are distinguished subjectively, weights are calculated by least square regression and threshold is set according to the rank of the last good observation in training data. In this research, threshold is set relatively reasonably, however, whether using the coefficients of least square regression as attributes' weights is reasonable is open to doubt. iC and Yurdakul [17] use TOPSIS to evaluate the credibility of manufacturing firms in Turkey. Fuzzy numbers are used to set weights for attributes, and then TOPSIS is applied to evaluate the performance of industries and firms. In this research, fuzzy numbers are set by experts, so weights are also set subjectively. Attributes distinguishing and threshold determination are not involved. Li et al. [20] hybridize TOPSIS with case-based reasoning for business failure prediction. Firms are classified according to their distances from positive ideal observation and negative ideal observation. In this research, attributes distinguishing and weight setting are not discussed.

In summary, only if attributes distinguishing, weight setting and threshold determination are all settled, can TOPSIS be successfully modified into a systematic classification approach. However, none of the above research works can settle them simultaneously. In this study, logistic regression is used to distinguish attributes, a novel but effective method is proposed to set weights and a reasonable way is employed to determine threshold for classification. Details are presented in Section 3. To the best of our knowledge,

this study is the first paper to propose a systematic classification approach based on the rationale of TOPSIS.

3. The proposed C-TOPSIS approach

In this section, we present the novel C-TOPSIS classification approach. Fig. 1 shows how TOPSIS is modified into C-TOPSIS in this study. In TOPSIS, attributes are distinguished intuitively, weights are always set subjectively by experts and TOPSIS is a ranking approach rather than a classification approach. So three challenges are faced in modifying TOPSIS into C-TOPSIS, i.e. how to distinguish attributes objectively? How to set weights properly? How to set threshold reasonably? In this study, three corresponding strategies are worked out to overcome the three challenges. Logistic regression is used to distinguish attributes, a novel method is proposed to set weights and a reasonable method is employed to determine the threshold. After that, TOPSIS is modified into C-TOPSIS systematically.

In order to introduce C-TOPSIS logically, first, TOPSIS is explained and then the challenges and the corresponding strategies for modifying TOPSIS into C-TOPSIS are presented. Lastly, the concept and detailed steps of C-TOPSIS are given.

3.1. TOPSIS

TOPSIS is always used to rank alternatives according to some attributes and is known for the fewest rank reversals. It is based on the concept that the most preferred alternative should have the shortest distance from the positive ideal solution (PIS) and the largest distance from the negative ideal solution (NIS). The PIS presents the best solution that maximizes the benefit attributes and minimizes the cost attributes, whereas the NIS presents the opposite, i.e. worst solution that minimizes the benefit attributes and maximizes the cost attributes. Alternatives are ranked according to the closeness coefficient defined as the ratio of the distance to NIS and the sum of distance to NIS and PIS. The closer an alternative is to PIS and farther to NIS, the larger the closeness coefficient is, and vice versa. The steps of TOPSIS are as follows [18].

Assume that there are n alternatives to be evaluated and each alternative has m evaluation attributes. Let $X = [x_{ij}]_{n \times m}$ denote the decision matrix, where x_{ij} is the j th attribute value of the i th alternative. Let w_j denote the weight of the j th attribute.

Step 1: Normalize the decision matrix

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (1)$$

where r_{ij} represents the normalized value of j th attribute of the i th alternative.

Step 2: Weight the normalized decision matrix

$$v_{ij} = w_j r_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (2)$$

where v_{ij} represents the weighted normalized value of j th attribute of the i th alternative. Weights are always set subjectively by experts.

Step 3: Determine PIS and NIS

$$A^+ = \{v_j^+, j = 1, \dots, m\} \quad (3)$$

$$A^- = \{v_j^-, j = 1, \dots, m\} \quad (4)$$

where A^+ and A^- represent the PIS and the NIS, respectively. $v_j^+ = \max\{v_{ij}, i = 1, \dots, n\}$ if the j th attribute is a benefit attribute, otherwise $v_j^+ = \min\{v_{ij}, i = 1, \dots, n\}$. $v_j^- = \min\{v_{ij}, i = 1, \dots, n\}$ if the j th attribute is a benefit attribute, otherwise $v_j^- = \max\{v_{ij}, i = 1, \dots, n\}$. PIS and NIS are the weighted normalized best and worst alternatives.

Step 4: Calculate the distance from each alternative to PIS and NIS

$$S_i^+ = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^+)^2}, \quad i = 1, \dots, n \quad (5)$$

$$S_i^- = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^-)^2}, \quad i = 1, \dots, n \quad (6)$$

where S_i^+ and S_i^- represent the distance between the i th alternative and the PIS and the distance between the i th alternative and the NIS, respectively.

Step 5: Calculate the closeness coefficient

$$C_i = \frac{S_i^-}{S_i^- + S_i^+}, \quad i = 1, \dots, n \quad (7)$$

where C_i is the closeness coefficient that denotes the degree of closeness of the i th alternative to PIS and NIS.

The above are the whole steps of TOPSIS. Alternatives are ranked by closeness coefficient. The larger the closeness coefficient is, the more preferred the alternative is.

3.2. Strategies for modifying TOPSIS into C-TOPSIS

TOPSIS is a ranking approach. Nevertheless, the ranking process of TOPSIS is similar to the credit scoring process and TOPSIS is well known for reliable evaluation results, a quick computing process and ease of use and understanding. So it may be a good idea to modify TOPSIS into a credit classification approach as follows. Firstly, according to the attributes such as income, debt and profession, TOPSIS can give every applicant a credit score. Then, a reasonable threshold is determined to classify the applicants into two classes. Applicants with high credit scores above the threshold

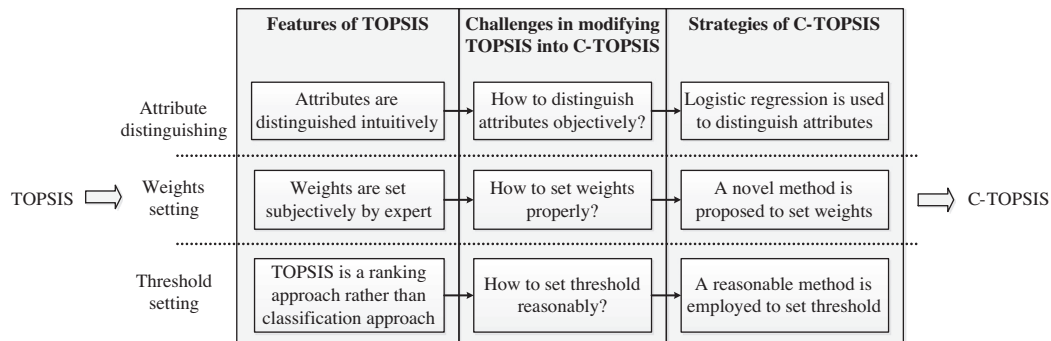


Fig. 1. The process of modifying TOPSIS into C-TOPSIS.

are classified as approved applicants while those with low credit scores below the threshold are classified as the rejected applicants.

Generally speaking, if rational benefit and cost attributes distinguishing, objective weights setting and reasonable threshold determination problems are overcome, TOPSIS can be successfully transformed into a classification approach. Moreover, the classification approach will retain the advantages of TOPSIS and can balance accuracy, complexity and interpretability well.

For the sake of simplicity, applicants who attain high credit scores and will be approved are called good applicants while those who attain low credit scores and will be rejected are called bad applicants in the following text.

3.2.1. Benefit and cost attributes distinguishing

Attributes need to be distinguished as benefit or cost attributes in TOPSIS. Benefit attributes are those which have positive effects on credit score, such as income. It is generally accepted that the higher the income is, the better the repayment capability, so the higher the credit score should be. On the contrary, cost attributes are those which have negative effects on credit score, such as debt. It is generally accepted that the more debt an applicant owes, the worse the repayment capability is, so the lower the credit score of the applicant should be. Proper attribute distinguishing is the premise of using TOPSIS. In many other studies, attributes are distinguished easily by experts, however, in credit scoring, some attributes cannot be distinguished subjectively. Besides, titles of the attribute are always changed into meaningless symbols for the sake of confidentiality of the data. So a proper objective attribute distinguishing approach is urgently needed in credit classification.

In this paper, logistic regression is used to select and distinguish attributes. Specifically, class is set as dependent variable and the other attributes are set as independent variables. After logistic regression, attributes with non-significant coefficients are omitted and attributes with significant positive or negative coefficients are regarded as benefit attributes or cost attributes, respectively.

3.2.2. Weights setting

In TOPSIS, attributes weights are always set subjectively by experts. The weights can reveal the importance of the attributes and what is more, serve the goal of classification. Subjective weights are prone to be affected by human experience, knowledge and so on. More importantly, sometimes they are not conducive for classification. Therefore, proper objective weights helpful for improving classification accuracy are needed.

A novel but effective weight setting method aimed at helping classification as accurate as possible is put forward in this paper. Based on the rationale of TOPSIS, we assume that if good applicants are as close to the best applicant as possible and bad applicants are as close to the worst applicant as possible, the classification will be as accurate as possible. By transforming the concept into a mathematical function, the goal is to minimize the sum of the average distance between weighted good applicants and the best applicant and the average distance between weighted bad applicants and the worst applicant. By solving this goal function, reasonable weights can be obtained.

Fig. 2 shows the possible effect of weighting under a two dimensional space. The boxes denote the bad applicants and circles denote good applicants. The two solid black boxes represent the best and the worst applicant. At first, as the left figure shows, the boundary between good applicants and bad applicants is not so clear. Then, the data is weighted by the proposed method. After weighting, as the right figure shows, good applicants are much closer to the best applicant and bad applicants are much closer to the worst applicant. It is much easier to separate good applicants and bad applicants now. So the proposed weight setting method here is not only objective, but also helpful for classification.

3.2.3. Threshold determination

After credit scores are assigned, a threshold needs to be set to classify the applicants into two groups. Applicants with credit scores higher than the threshold are approved and lower than the threshold are rejected. How to set a reasonable threshold is a problem.

In this paper, an easily intelligible and rational threshold is determined to solve this problem. The better credit the applicant has, the higher the credit score is, so good applicants are assumed to attain higher credit score than bad applicants. Assume there are n_g good applicants and n_b bad applicants in the training data. The n_g th highest credit score is set as the reasonable threshold. After credit scores of testing data are calculated, applicants with credit scores higher than the threshold are classified as good applicants while those with credit scores lower than the threshold are classified as bad applicants.

As per the above description, this study tries to overcome the three challenges so that TOPSIS can be successfully modified into a classification approach named C-TOPSIS.

3.3. C-TOPSIS

3.3.1. Concept of C-TOPSIS

C-TOPSIS is a classification approach based on the rationale of the ranking approach TOPSIS. In credit classification, PIS stands for the best credit applicant characterized by the best attributes value, such as the highest income, the best credit record and the lowest debt. NIS stands for the worst credit applicant characterized by the worst attributes value, such as the lowest income, the worst credit record and the highest debt.

Fig. 3 shows the rationale of C-TOPSIS in a two dimensional space. The solid black boxes represent the best and the worst applicants. The hollow boxes denoted applicants to be evaluated. The best applicant and the worst applicant are the benchmarks for evaluating applicants' credit. The closeness coefficient is called the credit score here, defined as the ratio of the distance to the worst applicant and the sum of distance to the best applicant and the worst applicant. So the closer the applicant is to the best applicant and farther from the worst applicant, the higher the credit score is and vice versa. Then, if the credit score of an applicant is larger than the threshold, it would be predicted as a good applicant. On the contrary, if the credit score is smaller than the threshold, it would be predicted as bad applicant. The rationale of C-TOPSIS in higher dimensional space can be analogized.

3.3.2. Steps of C-TOPSIS

In this section, the specific steps of C-TOPSIS are given. As Fig. 4 shows, the whole process of C-TOPSIS consists of 3 stages and 8 steps in total. In Stage 1 the whole credit data is preprocessed. In Stage 2, the training data is used to calculate attributes weights and threshold. In Stage 3, the testing data is classified as per credit score and threshold.

Give a set of credit data containing n applicants with m evaluation attributes. Let x_{ij} denotes the j th attribute value of the i th applicant. Let $y_i \in \{1, -1\}$ denotes the class of the i th applicant. Let w_j denotes the weight of the j th attribute. The credit data is randomly split into n_{train} training data, containing n_g good applicants and n_b bad applicants, and n_{test} testing data. The whole process of C-TOPSIS for credit classification is as follows.

3.3.2.1. Stage 1: Data preprocess.

Step 1: Distinguish benefit and cost attributes

By setting $Y = [y_i]_{n \times 1}$ as dependent variable and $X = [x_{ij}]_{n \times m}$ as independent variables, logistic regression is used to select and distinguish attributes. Attributes with non-significant coefficients are

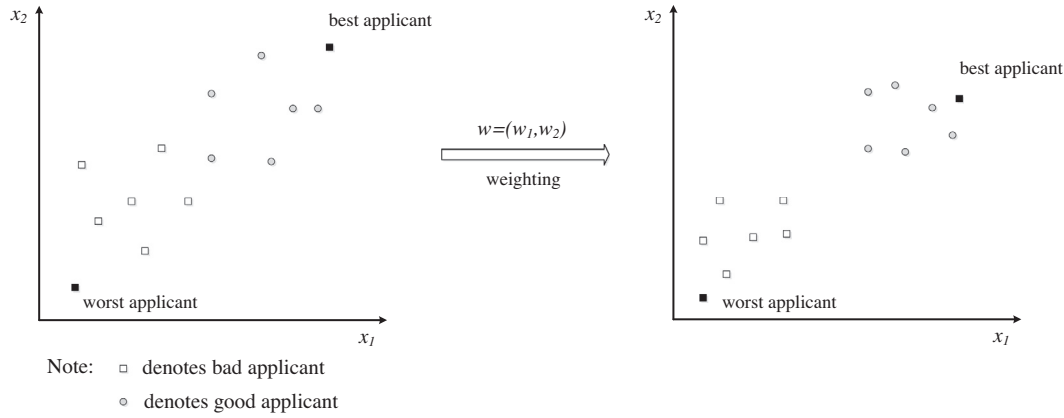


Fig. 2. The effect of the weight setting in a two dimensional space.

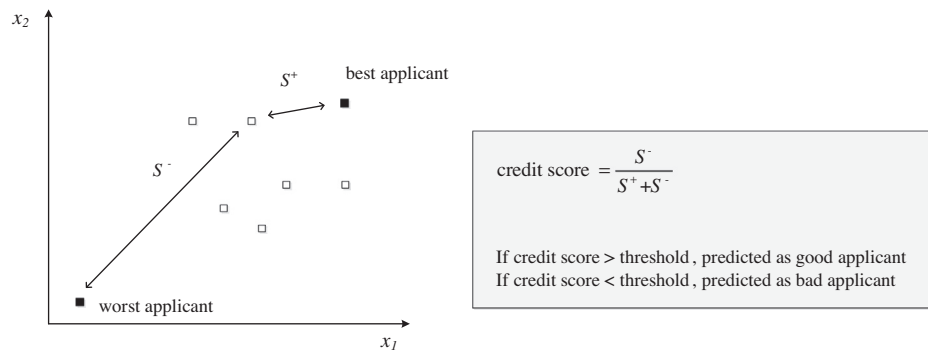


Fig. 3. The rationale of C-TOPSIS in a two dimensional space.

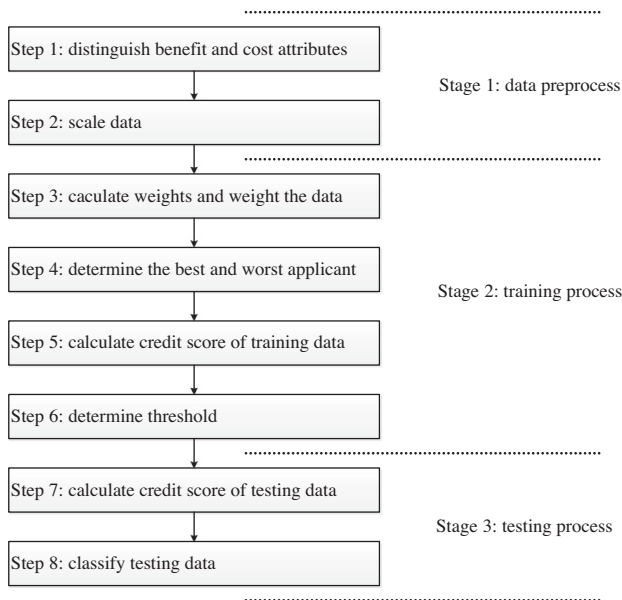


Fig. 4. The classification steps of C-TOPSIS.

omitted and attributes with significant positive or negative coefficients are regarded as benefit or cost attributes, respectively.

Step 2: Scale data
For benefit attribute

$$r_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (8)$$

For cost attribute

$$r_{ij} = \frac{x_j^{\max} - x_{ij}}{x_j^{\max} - x_j^{\min}}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (9)$$

where $x_j^{\max} = \max\{x_{ij} | i = 1, \dots, n\}$ represents the maximum value of attribute j , $x_j^{\min} = \min\{x_{ij} | i = 1, \dots, n\}$ represents the minimum value of attribute j and r_{ij} represents the scaled data. It is noteworthy that cost attributes are transformed into benefit attributes by Eq. (9). All attributes are beneficial and scaled to $[0, 1]$ after this step. Linear normalization rather than vector normalization is employed here so that the normalized value does not depend on the evaluation unit of a criterion function [25].

3.3.2.2. Stage 2: Training process

Step 3: Calculate weights and weight the data

As the description in Section 3.2.2, the weights setting is based on the concept that good applicants should be as close to the best applicant as possible while bad applicants should be as close to the worst applicant as possible so that they can be separated as accurately as possible. This concept can be implemented by minimizing the goal function (10), which represents the sum of the average distance between good applicants and the best applicant and the average distance between bad applicants and the worst applicant. The distance here is the 2-norm of Euclidean distance for simplification.

$$\min \frac{1}{n_g} \sum_{i=1}^{n_g} \left(\sum_{j=1}^m (w_j r_{ij} - r_j^{\max})^2 \right) + \frac{1}{n_b} \sum_{i=1}^{n_b} \left(\sum_{j=1}^m (w_j r_{ij} - r_j^{\min})^2 \right) \quad (10)$$

where $r_j^{\max} = \max\{r_{ij} | i = 1, \dots, n_{\text{train}}\}$ and $r_j^{\min} = \min\{r_{ij} | i = 1, \dots, n_{\text{train}}\}$. Goal function (10) is an unconstrained convex quadratic programming problem. Take the derivative of goal function (10) with respect to w_j , and then the weights can be solved as Eq. (11).

$$w_j = \frac{\frac{1}{n_g} \sum_{i=1}^{n_g} \sum_{j=1}^m r_{ij} r_j^{\max} + \frac{1}{n_b} \sum_{i=1}^{n_b} \sum_{j=1}^m r_{ij} r_j^{\min}}{\frac{1}{n_g} \sum_{i=1}^{n_g} \sum_{j=1}^m r_{ij}^2 + \frac{1}{n_b} \sum_{i=1}^{n_b} \sum_{j=1}^m r_{ij}^2}, \quad j = 1, \dots, m \quad (11)$$

We can see that Eq. (11) is linear and has very low computation complexity. Then we can weight the whole credit data by Eq. (12).

$$v_{ij} = w_j r_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (12)$$

where v_{ij} represents the j th weighted scaled attribute value of the i th applicant.

Step 4: Determine the best applicant and the worst applicant

$$v^{\max} = \{v_j^{\max}, j = 1, \dots, m\} \quad (13)$$

$$v^{\min} = \{v_j^{\min}, j = 1, \dots, m\} \quad (14)$$

where $v_j^{\min} = \min\{v_{ij} | i = 1, \dots, n_{\text{train}}\}$ and $v_j^{\max} = \max\{v_{ij} | i = 1, \dots, n_{\text{train}}\}$. v^{\max} represents the best applicant and v^{\min} represents the worst applicant.

Step 5: Calculate credit score of training data

$$S_i^+ = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^{\max})^2}, \quad i = 1, \dots, n_{\text{train}} \quad (15)$$

$$S_i^- = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^{\min})^2}, \quad i = 1, \dots, n_{\text{train}} \quad (16)$$

where S_i^+ and S_i^- represent the distance between the i th training applicant and the best applicant and the distance between the i th training applicant and the worst applicant, respectively.

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-}, \quad i = 1, \dots, n_{\text{train}} \quad (17)$$

where C_i is credit score of the i th training applicant.

Step 6: Determine threshold

Sort the credit score from high to low, the n_g th highest score is set as the threshold score C_0 .

3.3.2.3. Stage 3: Testing process

Step 7: Calculate credit score of testing data

$$S_i^+ = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^{\max})^2}, \quad i = 1, \dots, n_{\text{test}} \quad (18)$$

$$S_i^- = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^{\min})^2}, \quad i = 1, \dots, n_{\text{test}} \quad (19)$$

where S_i^+ and S_i^- represent the distance between the i th testing applicant and the best applicant and the distance between the i th testing applicant and the worst applicant, respectively.

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-}, \quad i = 1, \dots, n_{\text{test}} \quad (20)$$

where C_i is the credit score of the i th testing applicant.

Step 8: Classify the testing data

The classification rule is:

If $C_i > C_0$, the i th testing applicant is predicted as good, otherwise if $C_i < C_0$, the i th testing applicant is predicted as bad.

Steps 1–8 are the whole credit classification steps of C-TOPSIS.

4. Experiment

4.1. Experiment design

In this experiment, C-TOPSIS is used to classify two real world UCI data sets. In order to test the effectiveness of C-TOPSIS, results of C-TOPSIS and 7 popular approaches are compared in three aspects: accuracy, complexity and interpretability. The 7 popular approaches are linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), decision tree (DT), logistic regression (LogR), k-nearest neighbor classifier (k-NN), support vector machine (SVM) and least square support vector machine (LSSVM).

In line with previous researches [22,32,39], both data sets are split into training data (80%) and testing data (20%). SVM and LSSVM used in this experiment are non-linear SVM and LSSVM with RBF kernel. Detailed model description and procedures can be seen in [6,15,29]. Grid search and 5-fold validation [36] is used to optimize the two parameters, i.e. penalty parameter C and kernel parameter σ ($\{C, \sigma\} \in [2^{-5}, 2^{-4}, \dots, 2^{15}]$). The parameter value of k might affect the results of k-NN to some degree. A study by Enas and Choi [10] leads to the suggestion that $k \approx n^{2/8}$ or $k \approx n^{3/8}$ is reasonable, where n is the size of training data. According to the size of the data sets in the experiment, k is set as the odds from 3 to 15 ($k \in \{3, 5, \dots, 15\}$) and the k with the highest total accuracy is chosen.

All computations in this experiment are performed by MATLAB 2012a on a computer with Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz and 4 GB of main memory in Windows 7 environment.

4.2. Data description

Two widely used real world data sets, Australian Credit Approval Data Set (ACD) and German Credit Data Set (GCD), from UCI Repository of Machine Learning Databases [33] are used to check the performance of our approach. The basic information of the two data sets is shown in Table 1. The ACD consists of 307 instances of good applicants and 383 instances of bad applicants. Each applicant has 6 categorical attributes, 8 quantitative attributes and 1 class attribute (good or bad). To protect the confidentiality of the data, the attributes names and values have been changed into meaningless symbolic data. The GCD is more unbalanced and consists of 700 instances of good applicants and 300 instances of bad applicants. Total 24 attributes are used to describe the credit history, account balance, loan purpose, loan amount, employment status, personal information, age, housing, job title and class of the applicants.

Both ACD and GCD are consumer credit application data sets. A major feature of consumer credit data sets is that there exist many categorical attributes, such as present employment status, credit history and loan purpose, which cannot be dealt with by some approaches. In this paper, categorical attributes in ACD are transformed into quantitative attributes as follows. Originally, for each categorical attribute, the categories are denoted by meaningless symbols and need to be assigned proper values serving for classification. In general, the more good applicants and less bad applicants are in a category, the larger the value of the category should be. To

Table 1
Basic information of the two UCI credit data sets.

Data sets	ACD	GCD
No. of total applicants	690	1000
No. of good applicants	307	700
No. of bad applicants	383	300
No. of attributes	14	24
No. of classes	2	2

take the categorical attribute with two categories as an example, some applicants are in category 1 and the others are in category 2. Among the applicants in category 1, some of them are good and the others are bad, so the value of category 1 is defined as the ratio of number of good applicants and number of total applicants in category 1. Likewise, the value of category 2 is defined as the ratio of number of good applicants and number of total applicants in category 2. The categorical attribute with more than two categories can also be transformed like this. The above transformation method can quantify the categorical attributes easily and meaningfully. GCD does not need transformation because it has already been quantified.

4.3. Evaluation criteria

4.3.1. Accuracy evaluation

The accuracy performance is typically measured by Type 1 accuracy (T1), Type 2 accuracy (T2) and Total accuracy (T), expressed as percent of correctly classified good applicants, percent of correctly classified bad applicants and the percent of correctly classified in total, respectively [26,42]. More specifically,

$$\text{Total Accuracy (T)} = \frac{TN + TP}{TN + FP + TP + FN}$$

$$\text{Type 1 Accuracy (T1)} = \frac{TN}{TN + FP}$$

$$\text{Type 2 Accuracy (T2)} = \frac{TP}{TP + FN}$$

where TN is the number of good credit applicants correctly classified; TP is the number of bad applicants correctly classified; FP is the number of good credit applicants misclassified and FN is the number of misclassified bad applicants.

4.3.2. Complexity evaluation

Under the same computer hardware configuration, computational efficiency is mainly decided by the complexity of an approach, so in this experiment computing time is used for complexity evaluation. The more computing time an approach consumes, the more complex the approach is.

4.3.3. Interpretability evaluation

Interpretability decides whether the classification results of an approach can be explained clearly to the customers. Therefore, in this experiment, when an applicant is classified as bad by an approach, if we can explain which attributes cause the failure of the application, then the approach is considered as an interpretable approach. On the contrary, if we cannot explain, then the approach is a black-box and believed to be an approach with bad interpretability.

4.4. Accuracy comparison

Classification accuracy is the basic and decisive aspect in choosing the credit classification approach because a classification approach with low accuracy is useless. In order to check the accuracy performance of C-TOPSIS, we compare C-TOPSIS with 7 other major popular credit classification models on the two UCI data sets. Accuracy is the average of 100 runs. Table 2 shows the accuracy and rank of C-TOPSIS and 7 popular approaches. Approaches are ranked by total accuracy.

Table 2 shows that the ranks of the some approaches are not consistent on the two data sets. Some approaches rank high on one data set while low on the other data set. Furthermore, none of the approaches performs best on both data sets. Specifically,

Table 2

Accuracy results of C-TOPSIS and some popular approaches.

Approaches	ACD				GCD			
	T1 (%)	T2 (%)	T (%)	Rank	T1 (%)	T2 (%)	T (%)	Rank
LDA	92.28	80.64	85.79	5	72.37	71.38	72.08	3
QDA	65.70	91.37	80.02	7	66.82	69.57	67.64	8
DT	80.39	85.39	83.18	6	78.54	49.56	69.85	7
LogR	86.91	85.73	86.25	3	88.31	49.34	76.62	1
k-NN	56.01	79.84	69.31	8	92.78	19.17	70.70	6
SVM	88.54	83.99	86.00	4	75.00	64.26	71.78	5
LSSVM	89.03	85.01	86.78	1	76.36	61.78	71.99	4
C-TOPSIS	84.68	88.00	86.52	2	82.55	58.94	75.47	2

Note: (1) Accuracy is the average of 100 runs. (2) Approaches are ranked by total accuracy. (3) Parameter k of k-NN approach is set as 5 for ACD and 13 for GCD.

total accuracy of LSSVM is 86.78% on ACD and it ranks 1st and yet its accuracy on GCD is 71.99% and it ranks 4th. On the contrary, total accuracy of LogR on GCD is 76.62% and it ranks 1st, however, its accuracy on ACD is 86.25% and it ranks 3rd.

Besides, k-NN QDA and DT have relatively poor performance on both data sets, ranking the last, second last and third last, respectively. Specifically, total accuracy of k-NN is 69.31% on ACD and 70.70% on GCD. Total accuracy of QDA is 80.02% on ACD and 67.64% on GCD. Total accuracy of DT is 83.18% on ACD and 69.85% on GCD.

Total accuracy of C-TOPSIS is 86.52% on ACD and 75.47% on GCD, which ranks 2nd on both data sets. The accuracy performance of C-TOPSIS is comparatively outstanding for it ranks the top on both data sets simultaneously.

4.5. Complexity comparison

Computational efficiency is very important in application because a time-consuming approach entails higher hardware as well as time cost. In order to check the complexity performance of C-TOPSIS, the computing time of 100 runs for C-TOPSIS and the 7 popular approaches on ACD and GCD are recorded respectively. The average, standard deviation, maximum and minimum of computing time in seconds are shown in Tables 3 and 4. Approaches are ranked by average computing time.

Tables 3 and 4 show that only the ranks of LDA and QDA are inconsistent on the two data sets. LDA takes 9.80×10^{-4} s on ACD, 2.62×10^{-3} s on GCD and ranks 2nd and 3rd. QDA ranks 3rd and 2nd. SVM and LSSVM need the longest computing time and rank last and second last, respectively. Specifically, SVM consumes 1.84×10^3 s on ACD and 3.69×10^3 s on GCD while LSSVM consumes 5.23×10^1 s on ACD and 9.32×10^1 s on GCD.

Table 3

Computing time results of C-TOPSIS and some popular approaches on ACD.

Approaches	Time (s)				Rank
	Average	SD	Max	Min	
LDA	9.80×10^{-4}	3.99×10^{-5}	1.32×10^{-3}	9.51×10^{-4}	2
QDA	1.21×10^{-3}	3.94×10^{-5}	1.74×10^{-3}	1.17×10^{-3}	3
DT	2.80×10^{-2}	1.53×10^{-3}	3.25×10^{-2}	2.35×10^{-2}	6
LogR	6.51×10^{-3}	1.55×10^{-4}	7.94×10^{-3}	6.01×10^{-3}	5
k-NN	2.97×10^{-3}	7.17×10^{-5}	3.64×10^{-3}	2.88×10^{-3}	4
SVM	1.84×10^3	7.05×10^1	2.03×10^3	1.71×10^3	8
LSSVM	5.23×10^1	4.41×10^{-1}	5.44×10^1	5.19×10^1	7
C-TOPSIS	5.98×10^{-4}	4.27×10^{-5}	1.11×10^{-3}	5.64×10^{-4}	1

Note: (1) Average is the average computing time of 100 runs in seconds. (2) Approaches are ranked by average computing time. (3) Parameter k of k-NN approach is set as 5 for ACD. (4) SD is the abbreviation of standard deviation.

Table 4

Computing time results of C-TOPSIS and some popular approaches on GCD.

Approaches	Time (s)				Rank
	Average	SD	Max	Min	
LDA	2.62×10^{-3}	1.91×10^{-3}	1.48×10^{-2}	1.53×10^{-3}	3
QDA	1.79×10^{-3}	2.40×10^{-4}	6.00×10^{-3}	1.73×10^{-3}	2
DT	8.30×10^{-2}	3.06×10^{-3}	9.29×10^{-2}	7.52×10^{-2}	6
LogR	9.02×10^{-3}	5.66×10^{-4}	1.87×10^{-2}	8.31×10^{-3}	5
k-NN	6.40×10^{-3}	1.18×10^{-4}	7.29×10^{-3}	6.19×10^{-3}	4
SVM	3.69×10^3	1.75×10^2	4.05×10^3	3.26×10^3	8
LSSVM	9.32×10^1	7.96×10^{-1}	9.63×10^1	9.16×10^1	7
C-TOPSIS	6.70×10^{-4}	3.97×10^{-5}	1.06×10^{-3}	6.30×10^{-4}	1

Note: (1) Average is the average computing time of 100 runs in seconds. (2) Approaches are ranked by average computing time. (3) Parameter k of k-NN approach is set as 13 for GCD. (4) SD is the abbreviation of standard deviation.

Besides, LDA, QDA, DT, LogR, k-NN and C-TOPSIS consume observably less time than SVM and LSSVM. The computing time of SVM is almost 1×10^6 times of LDA, QDA, DT, LogR, k-NN and C-TOPSIS. Furthermore, compared with SVM, LSSVM is much time-saving.

C-TOPSIS ranks 1st on both data sets, specifically, consuming 5.98×10^{-4} s on ACD and 6.70×10^{-4} s on GCD. As expected, the results show that C-TOPSIS has high computational efficiency because the whole process of C-TOPSIS just involves linear computation.

4.6. Interpretability comparison

Besides accuracy and complexity, interpretability is always taken into consideration in classification of loan applicants since they have the rights to know why they are rejected. According to the interpretability evaluation criterion (Section 4.3), we grade C-TOPSIS and the other 7 approaches with 1 and 0. Approaches that meet the evaluation criterion are graded 1 while approaches that do not meet the criterion are graded 0. On the basis of the score, approaches are categorized into 2 ranks; rank 1 contains the interpretable approaches with score 1 while rank 2 contains the approaches with score 0. The interpretability score and rank of every approach are shown in Table 5.

From Table 5 we can see that DT and C-TOPSIS meet the interpretability evaluation criterion and are graded 1, LDA, QDA, LogR, k-NN, SVM and LSSVM do not meet it and are graded 0. DT can clearly show a lucid and visualized classification tree. According to the classification tree, applicants can easily get to know which attributes affect the classification results, which attributes make little contribution to the results and which attributes fail to meet the requirement and cause rejection. So DT meets the interpretability evaluation criterion and is graded 1. C-TOPSIS is easy to understand, explain and use. If an applicant is classified as bad,

Table 5

Interpretability score of C-TOPSIS and some popular approaches.

Approaches	Interpretability score	Rank
LDA	0	2
QDA	0	2
DT	1	1
LogR	0	2
k-NN	0	2
SVM	0	2
LSSVM	0	2
C-TOPSIS	1	1

Note: (1) Score 1 is for approaches which meet the interpretability evaluation criterion while score 0 is for approaches which do not meet the criterion. (2) Rank 1 contains approaches with score 1 while rank 2 contains approaches with score 0.

we can compare attributes to the best applicant, figure out which attributes are too far from the best and need improvement. So C-TOPSIS is also graded 1.

In LogR, the logarithm of odds (probability of being good/probability of being bad) is the dependent variable of the regression while the independent variables are the attributes. If the regression coefficient is positive, then the higher the attribute is, the more are the chances that the applicant would be classified as good, and vice versa. Nevertheless, if an applicant is rejected, we cannot clearly point out the exact causative attributes. The rationale of k-NN is easy to explain and understand. The applicant is rejected or approved according to the class of the k most similar applicant. If more than half of the k applicants are bad applicants, the applicant is classified as bad applicant, and vice versa. However, if an applicant is classified as bad, it is not easy to figure out which attributes are responsible for it. LDA classifies the applicants by projecting the data set into a lower dimension space. The rationale of LDA is relatively easy to understand and yet we cannot distinguish which attributes cause rejection of the application. QDA is more complicated than LDA. Therefore, LDA and QDA are graded 0. SVM and LSSVM are well known as the black-box model [1,2,13]. We cannot distinguish which attributes lead to the applicant's failure, so SVM and LSSVM are graded 0.

Therefore, only DT and C-TOPSIS are in the 1st rank and the remaining 6 approaches are in the 2nd rank. C-TOPSIS is very competitive here. This result is reasonable and not surprising for C-TOPSIS is easy to understand and explain based on the rationale of TOPSIS.

4.7. Comprehensive comparison

In order to check whether C-TOPSIS can balance accuracy, complexity and interpretability, we discuss the comprehensive performance of C-TOPSIS in this section. Table 6 shows the accuracy, complexity and interpretability ranks of C-TOPSIS and the 7 other approaches.

From Table 6 we can see that some approaches perform well in accuracy, such as LogR and LSSVM. Some approaches have advantages in complexity, such as LDA and QDA. Some approaches excel in interpretability, such as DT. Only C-TOPSIS ranks among the top 3 in all the three aspects.

LDA ranks 2nd and 3rd in complexity, however, it ranks 5th and 3rd in accuracy and is in 2nd rank interpretability. QDA is just like LDA, which has advantages of low complexity while accuracy and interpretability are weak. DT is in 1st rank in interpretability, however, it ranks 6th and 7th in accuracy and 6th in complexity. LogR ranks 1st and 3rd in accuracy, but 5th in complexity and is in 2nd rank in interpretability. k-NN ranks 8th and 6th in accuracy, 4th in complexity and in 2nd rank in interpretability which means its performance in all three aspects is poor. SVM ranks 4th and 5th in accuracy, 8th in complexity and is in 2nd rank in interpretability.

Table 6

Accuracy, complexity and interpretability rank of C-TOPSIS and some popular approaches.

Approaches	Rank				
	Accuracy		Complexity		Interpretability
	ACD	GCD	ACD	GCD	
LDA	5	3	2	3	2
QDA	7	8	3	2	2
DT	6	7	6	6	1
LogR	3	1	5	5	2
k-NN	8	6	4	4	2
SVM	4	5	8	8	2
LSSVM	1	4	7	7	2
C-TOPSIS	2	2	1	1	1

As expected, SVM performs relatively well in accuracy, but performs poor in complexity and interpretability because of its time-consuming parameter optimization and opaque classification process. LSSVM ranks 1st and 4rd in accuracy, 7th in complexity and is in 2nd rank in interpretability. As a modified SVM, LSSVM performs better in accuracy and complexity, however, the interpretability has not been improved.

Finally, C-TOPSIS ranks 2nd in accuracy, 1rd in complexity and is in 1st rank in interpretability. Only C-TOPSIS ranks among the top 3 in all three aspects. Therefore, from the above analysis, it can be concluded that compared with the 7 popular approaches, C-TOPSIS is the most balanced approach and performs very competitively in all the three aspects. This paper may offer a novel competitive alternative approach for credit classification.

5. Conclusion and future research

This study is, to the best of our knowledge, the first paper to propose a systematic classification approach based on the rationale of TOPSIS. The proposed approach named C-TOPSIS can balance accuracy, complexity and interpretability well and is found to be a competitive approach for credit classification. Our findings in the experiment provide compelling evidence of the advantages of C-TOPSIS.

In the experiment, by using two widely-used UCI credit data sets, C-TOPSIS and 7 popular approaches are compared in terms of accuracy, complexity and interpretability. LDA and QDA perform well in complexity but accuracy and interpretability are their weak points. DT wins out in interpretability but its accuracy and complexity are unacceptable. LogR performs well in accuracy but its complexity and interpretability are not competitive. k-NN performs poor in all the three aspects. SVM performs the worst in complexity while its accuracy and interpretability are also not so satisfactory. LSSVM is competitive in accuracy, yet its complexity and interpretability are also very poor. Only C-TOPSIS ranks among the top 3 in all the three aspects, which verifies that C-TOPSIS can balance accuracy, complexity and interpretability well.

However, since the data sets used in this study are limited and this study is the first paper that attempts to propose a systematic classification approach based on the rationale of TOPSIS, some questions still remain unresolved. For example, more data sets need to be tested in the future to further verify the effectiveness of C-TOPSIS. Since weights and threshold can significantly affect the classification result, the weight setting and threshold determination in this paper may be reconsidered. The results of these ongoing research works will be reported in the near future.

Acknowledgments

This research has been supported by grants from the National Natural Science Foundation of China (71071148), Key Research Program of Institute of Policy and Management, Chinese Academy of Sciences (Y201171Z05) and Youth Innovation Promotion Association of the Chinese Academy of Sciences. The authors also would like to thank the anonymous reviewers for their valuable suggestions, especially the suggestions concerning the rationale and intrinsic problem of TOPSIS.

References

- [1] N. Barakat, A.P. Bradley, Rule extraction from support vector machines: a review, *Neurocomputing* 74 (1–3) (2010) 178–190.
- [2] N. Barakat, A.P. Bradley, M.N.H. Barakat, Intelligent support vector machines for diagnosis of diabetes mellitus, *Information Technology in Biomedicine* 14 (4) (2010) 1114–1120.
- [3] M. Behzadian, S.K. Otaghsara, M. Yazdani, J. Ignatius, A state-of-the-art survey of TOPSIS applications, *Expert Systems with Applications* 39 (2012) 13051–13069.
- [4] S. Chatterjee, S. Barcun, A nonparametric approach to credit scoring, *Journal of the American Statistical Association* 65 (1970) 150–154.
- [5] Y.S. Chen, C.H. Cheng, Hybrid models based on rough set classifiers for setting credit rating decision rules in the global banking industry, *Knowledge-Based Systems* 30 (2013) 224–239.
- [6] C. Cortes, V. Vapnik, Support vector networks, *Machine Learning* 20 (1995) 273–287.
- [7] J.N. Crook, D.B. Edelman, L.C. Thomas, Recent developments in consumer credit risk assessment, *European Journal of Operational Research* 183 (2007) 1447–1465.
- [8] M. Doumpos, C. Zopounidis, V. Gofinopoulou, Additive support vector machines for pattern classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37 (3) (2007) 540–550.
- [9] D. Durand, Risk Elements in Consumer Installment Financing, National Bureau of Economic Research, New York, 1941.
- [10] G.G. Enas, S.C. Choi, Choice of the smoothing parameter and efficiency of k-nearest neighbor classification, *Computers & Mathematics with Applications* 12 (1986) 235–244.
- [11] L. Han, L.Y. Han, H.W. Zhao, Orthogonal support vector machine for credit scoring, *Engineering Applications of Artificial Intelligence* 26 (2013) 848–862.
- [12] D.J. Hand, W.E. Henley, Statistical classification methods in consumer credit scoring: a review, *Journal of the Royal Statistical Society: Series A* 160 (3) (1997) 523–541.
- [13] A.B. Hens, M.K. Tiwari, Computational time reduction for credit scoring: an integrated approach based on support vector machine and stratified sampling method, *Expert Systems with Applications* 39 (2012) 6774–6781.
- [14] F. Hoffmann, B. Baesens, C. Mues, et al., Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms, *European Journal of Operational Research* 177 (2007) 540–555.
- [15] C.W. Hsu, C.C. Chang, C.J. Lin, A Practical Guide to Support Vector Classification, 2010. <<http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>>.
- [16] C.L. Hwang, K. Yoon, Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey, Springer-Verlag, Berlin and New York, 1981.
- [17] Y.T. iC, M. Yurdakul, Development of a quick credibility scoring decision support system using fuzzy TOPSIS, *Expert Systems with Applications* 37 (2010) 567–574.
- [18] Y.B. Ju, A.H. Wang, Emergency alternative evaluation under group decision makes: a method of incorporating DS/AHP with extended TOPSIS, *Expert Systems with Applications* 39 (2012) 1315–1323.
- [19] L.J. Kao, C.C. Chiu, F.Y. Chiu, A Bayesian latent variable model with classification and regression tree approach for behavior and credit scoring, *Knowledge-Based Systems* 36 (2012) 245–252.
- [20] H. Li, H. Adeli, J. Sun, J.G. Han, Hybridizing principles of TOPSIS with case-based reasoning for business failure prediction, *Computers and Operations Research* 38 (2011) 409–419.
- [21] J.P. Li, G. Li, D.X. Sun, C.F. Lee, Evolution strategy based adaptive Lq penalty support vector machines with Gauss kernel for credit risk analysis, *Applied Soft Computing* 12 (8) (2012) 2675–2682.
- [22] J.P. Li, L.W. Wei, G. Li, W.X. Xu, An evolution strategy-based multiple kernels multi-criteria programming approach: the case of credit decision making, *Decision Support System* 51 (2011) 292–298.
- [23] D. Martens, B. Baesens, T.V. Gestel, J. Vanthienen, Comprehensive credit scoring models using rule extraction from support vector machines, *European Journal of Operational Research* 183 (2007) 1466–1476.
- [24] D. Mehta, The formulation of credit policy models, *Management Science* 15 (1968) 30–50.
- [25] S. Opricovic, G.H. Tzeng, Compromise solution by MCDM methods: a comparative analysis of VIKOR and TOPSIS, *European Journal of Operational Research* 156 (2004) 445–455.
- [26] Y. Peng, G.X. Wang, G. Kou, Y. Shi, An empirical study of classification algorithm evaluation for financial risk prediction, *Applied Soft Computing* 11 (2) (2011) 2906–2915.
- [27] Y. Peng, G. Kou, Y. Shi, Z.X. Chen, A multi-criteria convex quadratic programming model for credit data analysis, *Decision Support System* 44 (2008) 1016–1030.
- [28] N.Y. Secme, A. Bayrakdaroglu, C. Kahraman, Fuzzy performance evaluation in Turkish banking Sector using Analytic Hierarchy Process and TOPSIS, *Expert Systems with Applications* 36 (2009) 11699–11709.
- [29] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, *Neural Processing Letter* 9 (1999) 293–300.
- [30] L.C. Thomas, D.B. Edelman, J.N. Crook, Credit Scoring and its Applications, Society for Industrial and Applied Mathematics Press, Philadelphia, 2002.
- [31] L.C. Thomas, R.W. Oliver, D.J. Hand, A survey of the issues in consumer credit modeling research, *Journal of Operational Research Society* 56 (2005) 1006–1015.
- [32] C.F. Tsai, Feature selection in bankruptcy prediction, *Knowledge-Based Systems* 22 (2009) 120–127.
- [33] UCI Machine Learning Repository, Irvine, School of Information and Computer Science, University of California, CA. <<http://archive.ics.uci.edu/ml/datasets.html>>.

- [34] S. Vukovic, B. Delibasic, A. Uzelac, M. Suknovic, A case-based reasoning model that uses preferences theory functions for credit scoring, *Expert System with Applications* 39 (9) (2012) 8389–8395.
- [35] G. Wang, J. Ma, L.H. Huang, K.Q. Xu, Two credit scoring models based on dual strategy ensemble trees, *Knowledge-Based Systems* 26 (2012) 61–68.
- [36] L.W. Wei, Z.Y. Chen, J.P. Li, Evolution strategies based adaptive Lp LS-SVM, *Information Sciences* 181 (2011) 3000–3016.
- [37] J.C. Wiginton, A note on the comparison of logit and discriminant models of consumer credit behavior, *Journal of Quantitative Analysis* 15 (1980) 757–770.
- [38] D.S. Wu, D.L. Olson, A TOPSIS data mining demonstration and application to credit scoring, *International Journal of Data Warehousing and Mining* 2 (3) (2006) 1–10.
- [39] T.C. Wu, M.F. Hsu, Credit risk assessment and decision making by a fusion approach, *Knowledge-Based Systems* 35 (2012) 102–110.
- [40] P. Yao, Y.H. Lu, Neighborhood rough set and SVM based hybrid credit scoring classifier, *Expert System with Applications* 38 (2011) 11300–11304.
- [41] C.C. Yeh, F.Y. Lin, C.Y. Hsu, A hybrid KMV model, random forests and rough set theory approach for credit rating, *Knowledge-Based Systems* 33 (2012) 166–172.
- [42] L.A. Yu, S.Y. Wang, K.K. Lai, Credit risk assessment with a multistage neural network ensemble learning approach, *Expert Systems with Applications* 34 (2008) 1434–1444.
- [43] L.A. Yu, X. Yao, S.Y. Wang, K.K. Lai, Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection, *Expert Systems with Application* 38 (2011) 15392–15399.
- [44] L.G. Zhou, K.K. Lai, Credit scoring models with AUC maximization based on weighted SVM, *International Journal of Information Technology & Decision Making* 8 (4) (2009) 677–696.