Chapter 9
Evaluation Analysis

Evaluation is one of the key steps in big data analytics, which determines the merit of data analysis towards the experimental objectives. It usually relates a trade-off comparison of multiple criteria which may conflict each other or complex interpretations of the problems in nature. This chapter provides several of evaluation models of the recent studies on data science. Section 9.1 reviews three evaluation formations for the known methodologies. Section 9.1.1 describes a decision-making support for the evaluation of clustering algorithms based on multiple criteria decision making (MCDM) [1]. Section 9.1.2 is about evaluation of classification algorithms using MCDM and rank correlation [2]. Section 9.1.3 discusses the public blockchain evaluation using entropy and Technique of Order Preference Similarity to the Ideal Solution (TOPSIS) [3]. Section 9.2 outlines two evaluation methods for Software. Section 9.2.1 is about a classifier evaluation for software defect prediction [4], while Sect. 9.2.2 is about an ensemble of software defect predictors by AHP-based evaluation method [5]. Section 9.3 describes four evaluation methods for sociology and economics. Section 9.3.1 is about a delivery efficiency and supplier performance evaluation in China's E-retailing industry [6]. Section 9.3.2 is about the credit risk evaluation with Kernel-based affine subspace nearest points learning method [7]. Section 9.3.3 is a dynamic assessment method for urban eco-environmental quality evaluation [8], while Sect. 9.3.4 is an empirical study of classification algorithm evaluation for financial risk prediction [9].

9.1 Reviews of Evaluation Formations

9.1.1 Decision-Making Support for the Evaluation of Clustering Algorithms Based on MCDM

In many disciplines, the evaluation of algorithms for processing massive data is a challenging research issue. However, different algorithms can produce different or

Y. Shi, *Advances in Big Data Analytics*,
https://doi.org/10.1007/978-981-16-3607-3_9

even conflicting evaluation performance, and this phenomenon has not been fully investigated. The motivation of this section aims to propose a solution scheme for the evaluation of clustering algorithms to reconcile different or even conflicting evaluation performance. This section develops a model, called decision making support for evaluation of clustering algorithms (DMSECA), to evaluate clustering algorithms by merging expert wisdom in order to reconcile differences in their evaluation performance for information fusion during a complex decision-making process.

9.1.1.1 Clustering Algorithms

Clustering is a popular unsupervised learning technique. It aims to divide large data sets into smaller sections so that objects in the same cluster are lowly distinct, whereas objects in different clusters are lowly similar [10]. Clustering algorithms, based on similarity criteria, can group patterns, where groups are sets of similar patterns [11–13]. Clustering algorithms are widely applied in many research fields, such as genomics, image segmentation, document retrieval, sociology, bioinformatics, psychology, business intelligence, and financial analysis [14].

Clustering algorithms are usually known as the four classes of partitioning methods, hierarchical methods, density-based methods, and model-based methods [15]. Several classic clustering algorithms are proposed and reported, such as the K-means algorithm [16], k-medoid algorithm [17], expectation maximization (EM) [18], and frequent pattern-based clustering [15]. In this section, the six most influential clustering algorithms are selected for the empirical study. These are the KM algorithm, EM algorithm, filtered clustering (FC), farthest-first (FF) algorithm, make density-based clustering (MD), and hierarchical clustering (HC). These clustering algorithms can be implemented by WEKA [19].

The KM algorithm, a partitioning method, takes the input parameter k and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high, and the intercluster similarity is low. And the cluster similarity can be measured by the mean value of the objects in a cluster, which can be viewed as the centroid or center of gravity of the cluster [15].

The EM algorithm, which is considered as an extension of the KM algorithm, is an iterative method to find the maximum likelihood or maximum a posteriori estimates of parameters in statistical models, where the model depends on unobserved latent variables [20]. The KM algorithm assigns each object to a cluster.

In the EM algorithm, each object is assigned to each cluster according to a weight representing its probability of membership. In other words, there are no strict boundaries between the clusters. Thus, new means can be computed based on the weighted measures [18].

The FC applied in this work can be implemented by WEKA [19]. Like the cluster, the structure of the filter is based exclusively on the training data, and test instances will be addressed by the filter without changing their structure.

The FF algorithm is a fast, greedy, and simple approximation algorithm to the k-center problem [17], where the k points are first selected as a cluster center, and the second center is greedily selected as the point farthest from the first. Each remaining center is determined by greedily selecting the point farthest from the set of chosen centers, and the remaining points are added to the cluster whose center is the closest [16, 21].

The MD algorithm is a density-based method. The general idea is to continue growing the given cluster as long as the density (the number of objects or data points) in the neighborhood exceeds some threshold. That is, for each data point within a given cluster, the neighborhood of a given radius must contain a minimum number of points [15]. The HC algorithm is a method of cluster analysis that seeks to build a hierarchy of clusters, which can create a hierarchical decomposition of the given data sets [16, 22].

9.1.1.2 MCDM Methods

The MCDM methods, which were developed in the 1970s, are a complete set of decision analysis technologies that have evolved as an important research field of operation research [23, 24]. The International Society on MCDM defines MCDM as the research of methods and procedures concerning multiple conflicting criteria, which can be formally incorporated into the management planning process [24]. In an MCDM problem, the evaluation criteria are assumed to be independent [25, 26]. MCDM methods aim to assist decision-makers (DMs) to identify an optimal solution from a number of alternatives by synthesizing objective measurements and value judgments [27, 28]. In this section, four classic MCDM methods: the weighted sum method (WSM), grey relational analysis (GRA), TOPSIS, and PROMETHEE II are introduced as follows.

WSM

WSM [29] is a well-known MCDM method for evaluating finite alternatives in terms of finite decision criteria when all the data are expressed in the same unit [30, 31]. The benefit-to-cost-ratio and benefit-minus-cost approaches [32] can be applied to the problem of involving both benefit and cost criteria. In this section, the cost criteria are first transformed to benefit criteria. Besides, there is nominal-the-better (NB), when the value is closer to the objective value, the nominal-the-better (NB) is better. The calculation steps of WSM are as follows. First, assume n criteria, including benefit criteria and cost criteria, and m alternatives. The cost criteria are first converted to benefit criteria in the following standardization process.

1. The larger-the-better (LB): a larger objective value is better, that is, the benefit criteria, and it can be standardized as

$$x'_{ij} = \frac{x_{ij} - \min\limits_i x_{ij}}{\max\limits_i x_{ij} - \min\limits_i x_{ij}} \tag{9.1}$$

2. The smaller-the-better (SB): the smaller objective value is better, that is, the cost criteria, and it can be standardized as

$$x'_{ij} = \frac{\max\limits_i x_{ij} - x_{ij}}{\max\limits_i x_{ij} - \min\limits_i x_{ij}} \tag{9.2}$$

3. The nominal-the-better (NB): the closer to the objective value is better, and it can be standardized as

$$x'_{ij} = 1 - \frac{\left|x_{ij} - x_{ob}\right|}{\max\left\{\max\limits_i x_{ij} - x_{ob}; x_{ob} - \min\limits_i x_{ij}\right\}} \tag{9.3}$$

Finally, the total benefit of all the alternatives can be calculated as

$$A_i = \sum_{j=1}^{k} w_j x'_{ij}, \quad 1 \le i \le m, 1 \le j \le n \tag{9.4}$$

The larger WSM value indicates the better alternative.

GRA

GRA is a basic MCDM method of quantitative research and qualitative analysis for system analysis. Based on the grey space, it can address inaccurate and incomplete information. GRA has been widely applied in modeling, prediction, systems analysis, data processing, and decision-making [33]. The principle is to analyze the similarity relationship between the reference series and alternative series. The detailed steps are as follows.

Assume that the initial matrix is R:

$$\mathrm{R} = \begin{bmatrix} ccccx_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} (1 \le i \le m, 1 \le j \le n) \tag{9.5}$$

1. Standardize the initial matrix:

$$\mathrm{R}' = \begin{bmatrix} ccccx'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{bmatrix} (1 \le i \le m, 1 \le j \le n) \tag{9.6}$$

2. Generate the reference sequence x'_0:

$$x'_0 = \left(x'_0(1), x'_0(2), \ldots, x'_0(n)\right) \tag{9.7}$$

where $x'_0(j)$ is the largest and standardized value in the jth factor.

3. Calculate the differences $\Delta_{0i}(j)$ between the reference series and alternative series:

$$\Delta_{0i}(j) = |x'_0(j) - x'_{ij}|,$$
$$\Delta = \begin{bmatrix} \Delta_{01}(1) & \Delta_{01}(2) & \cdots & \Delta_{01}(n) \\ \Delta_{02}(1) & \Delta_{02}(2) & \cdots & \Delta_{02}(n) \\ \vdots & \vdots & \vdots & \vdots \\ \Delta_{0m}(1) & \Delta_{0m}(2) & \cdots & \Delta_{0m}(n) \end{bmatrix} (1 \le i \le m, 1 \le j \le n) \tag{9.8}$$

4. Calculate the grey coefficient $r_{0i}(j)$:

$$r_{0i}(j) = \frac{\min\limits_i \min\limits_j \Delta_{0i}(j) + \delta \max\limits_i \max\limits_j \Delta_{0i}(j)}{\Delta_{0i}(j) + \delta \max\limits_i \max\limits_j \Delta_{0i}(j)} \tag{9.9}$$

5. Calculate the value of grey relational degree b_i:

$$b_i = \frac{1}{n} \sum_{j=1}^{n} r_{0i}(j) \tag{9.10}$$

6. Finally, standardize the value of grey relational degree β_i:

$$\beta_i = \frac{b_i}{\sum\limits_{i=1}^{n} b_i} \tag{9.11}$$

TOPSIS

TOPSIS is one of the classic MCDM methods to rank alternatives over multicriteria. The principle is that the chosen alternative should have the shortest distance from the

positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS) [34]. TOPSIS can find the best alternative by minimizing the distance to the PIS and maximizing the distance to the NIS [35]. The alternatives can be ranked by their relative closeness to the ideal solution. The calculation steps are as follows [36]:

1. The decision matrix A is standardized:

$$a_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} \left(x_{ij}\right)^2}} \left(1 \le i \le m, 1 \le j \le n\right) \tag{9.12}$$

2. The weighted standardized decision matrix is computed:

$$\begin{aligned} &D = \left(a_{ij}{}^* w_j\right) \quad (1 \le i \le m, 1 \le j \le n) \\ &\sum_{i=1}^{m} w_j = 1 \end{aligned} \tag{9.13}$$

3. The PIS V* and the NIS V—are calculated:

$$\begin{aligned} V^* &= \left\{v_1^*, v_2^*, \ldots, v_n^*\right\} = \left\{\left(\max_i v_{ij} \,|j \in J|\right), \left(\min_i v_{ij} \,\middle|\, j \in J'\right)\right\} \\ V^- &= \left\{v_1^-, v_2^-, \ldots, v_n^-\right\} = \left\{\left(\min_i v_{ij} \,|j \in J|\right), \left(\max_i v_{ij} \,\middle|\, j \in J'\right)\right\} \end{aligned} \tag{9.14}$$

4. The distances of each alternative from PIS and NIS are determined:

$$\begin{aligned} S_i^+ &= \sqrt{\sum_{j=1}^{n} \left(V_i^j - V^*\right)^2} \quad (1 \le i \le m, 1 \le j \le n) \\ S_i^- &= \sqrt{\sum_{j=1}^{n} \left(V_i^j - V^-\right)^2} \quad (1 \le i \le m, 1 \le j \le n) \end{aligned} \tag{9.15}$$

5. The relative closeness to the ideal solution is obtained:

$$Y_i = \frac{S_i^-}{S_i^+ + S_i^-} \left(1 \le i \le m\right) \tag{9.16}$$

6. The preference order is ranked.

   The larger relative closeness indicates the better alternative.

9.1.1.3 PROMETHEE II

PROMETHEE II, proposed by Brans in 1982, uses pairwise comparisons and "values outranking relations" to select the best alternative [37]. PROMETHEE II can support DMs to reach an agreement on feasible alternatives over multiple criteria from different perspectives [38, 39]. In the PROMETHEE II method, a positive outranking flow reveals that the chosen alternative outranks all alternatives, whereas a negative outranking flow reveals that the chosen alternative is outranked by all alternatives. Based on the positive outranking flows and negative outranking flows, the final alternative can be selected and determined by the net outranking flow. The steps are as follows:

1. Normalize the decision matrix R:

$$R_{ij} = \frac{x_{ij} - minx_{ij}}{maxx_{ij} - minx_{ij}} \left(1 \le i \le n, 1 \le j \le m\right) \tag{9.17}$$

2. Define the aggregated preference indices. Let a, b ∈ A and

$$\begin{cases} \pi\left(a,b\right) = \sum_{j=1}^{k} p_j\left(a,b\right) w_j \\ \pi\left(a,b\right) = \sum_{j=1}^{k} p_j\left(b,a\right) w_j \end{cases} \tag{9.18}$$

   where A is a finite set of alternatives {a1, a2, …, an}, k is the number of criteria such that 1 ≤ k ≤ m, w_j is the weight of criterion j, and $\sum_{j=1}^{k} w_j = 1$ (1 ≤ k ≤ m). π(a, b) represents how a is preferred to b over all criteria, and $p_j(a,b)$ represents how b is preferred to a over all criteria. $p_j(a,b)$ and $p_j(b,a)$ are the preference functions of the alternatives a and b.
3. Calculate π(a, b) and π(b, a) for each pair of alternatives

   In general, there are six types of preference function. DMs must select one type of preference function and the corresponding parameter value for each criterion [40, 41].
4. Determine the positive outranking flow and negative outranking flow. The positive outranking flow is determined by

$$\phi^{+}(a) = \frac{1}{n-1} \sum_{x \in A} \pi\left(a,x\right) \tag{9.19}$$

   and the negative outranking flow is determined by

$$\phi^{-}(a) = \frac{1}{n-1} \sum_{x \in A} \pi\left(a,x\right) \tag{9.20}$$

Table 9.1 Contingency table

Partition C						Σ
		C_1	C_2	.....	C_k	
Partition P	P_1	n_{11}	n_{12}	.....	n_{1k}	N_1
	P_2	n_{21}	n_{22}	.....	n_{22}	N_2
	P_k	n_{k1}	n_{k2}	.....	n_{kk}	n_k
	Σ	n_1	n_2	.....	n_k	n

5. Calculate the net outranking flow:

$$\phi(a) = \phi^{+}(a) - \phi^{-}(a) \tag{9.21}$$

6. Determine the ranking according to the net out-ranking flow.

   Larger $\phi(a)$ is the more appropriate alternative.

9.1.1.4 Performance Measures

External measures for evaluating clustering results are more effective than internal and relative measures. Accordingly, in this study, nine clustering external measures are selected for evaluation. These are entropy, purity, micro-average precision (MAP), Rand index (RI), adjusted Rand index (ARI), F-measure (FM), Fowlkes–Mallows index (FMI), Jaccard coefficient (JC), and Mirkin metric (MM). Among them, measures of entropy and purity are widely applied as external measures in the fields of data mining and machine learning [42, 43]. The nine external measures are generated by a computer with an Intel core i5-3210M CPU @ 2.50 GHz with 8G memory. Before introducing external measures, the contingency table is described.

9.1.1.5 The Contingency Table

Given a data set D with n objects, suppose we have a partition $P = \{P_1, P_2, \ldots, P_n\}$ by some clustering method, where $\cup_{i=1}^{k} P_i = D$ and $P_i \cap P_j = \phi$, for $1 \leq \mathrm{i} \neq \mathrm{j} \leq \mathrm{k}$. According to the preassigned class labels, we can create another partition on $C = \{C_1, C_2, \ldots, C_k\}$ where $U_{i=1}^{k} C_i = D$ and $C_i \cap C_j = \phi$ for $1 \leq \mathrm{i} \neq \mathrm{j} \leq \mathrm{k}$. Let nij denote the number of objects in cluster Pi with the label of class Cj. Then, the data information between the two partitions can be displayed in the form of a contingency table, as shown in Table 9.1.

The following paragraphs define the external measures. The measures of entropy and purity are widely applied in the field of data mining and machine learning.

1. Entropy. The measure of entropy, which originated in the information-retrieval community, can measure the variance of a probability distribution. If all clusters consist of objects with only a single class label, the entropy is zero, and as the class labels of objects in a cluster become more varied, the entropy increases.

The measure of entropy is calculated as

$$E = -\sum_i \frac{n_i}{n}\left(\sum_j \frac{n_{ij}}{n_i}\log\frac{n_{ij}}{n_i}\right) \tag{9.22}$$

2. Purity. The measure of purity pays close attention to the representative class (the class with majority objects within each cluster). Purity is similar to entropy. It is calculated as

$$P = \sum_i \frac{n_{i_i}}{n}\left(\max_j \frac{n_{ij}}{n_{i_i}}\right) \tag{9.23}$$

A higher purity value usually represents more effective clustering.

3. F-Measure. The F-measure (FM) is a harmonic mean of precision and recall. It is commonly considered as clustering accuracy. The calculation of FM is inspired by the information-retrieval metric as follows:

$$\begin{aligned} F-\text{measure} &= \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \\ \text{precision} &= \frac{n_{ij}}{n_j}, \text{recall} = \frac{n_{ij}}{n_i} \end{aligned} \tag{9.24}$$

A higher value of FM generally indicates more accurate clustering.

4. Micro-average Precision. The MAP is usually applied in the information-retrieval community. It can obtain a clustering result by assigning all data objects in a given cluster to the most dominant class label and then evaluating the following quantities for each class:

(a) α(Cj): the number of objects correctly assigned to class Cj.
(b) β(Cj): the number of objects incorrectly assigned to class Cj.

The MAP measure is computed as follows:

$$\text{MAP} = \frac{\sum_j \alpha\left(C_j\right)}{\sum_j \alpha\left(C_j\right) + \beta\left(C_j\right)} \tag{9.25}$$

A higher MAP value indicates more accurate clustering.

5. Mirkin Metric. The measure of Mirkin metric (MM) assumes the null value for identical clusters and a positive value, otherwise. It corresponds to the Hamming distance between the binary vector representations of each partition [44]. The measure of MM is computed as

$$M = \sum_i n_{i.}^2 + \sum_j n_i^2 - 2\sum_i\sum_j n_{ij}^2 \tag{9.26}$$

A lower value of MM implies more accurate clustering. In addition, given a data set, assume a partition C is a clustering structure of a data set and P is a partition by some clustering method. We refer to a pair of points from the dataset as follows:

(a) SS: if both points belong to the same cluster of the clustering structure C and to the same group of the partition P
(b) SD: if the points belong to the same clusters of C and to different groups of P
(c) DS: if the points belong to different clusters of C and to the same groups of P
(d) DD: if the points belong to different clusters of C and to different groups of P

Assume that a, b, c, and d are the numbers of SS, SD, DS, and DD pairs, respectively, and that M a + b + c + d, which is the maximum number of pairs in the data set. The following indicators for measuring the degree of similarity between C and P can be defined.

6. Rand Index. The RI is a measure of the similarity between two data clusters in statistics and data clustering [45]. RI is computed as follows:

$$R = \frac{(a+d)}{M} \tag{9.27}$$

A higher value of RI indicates a more accurate result of clustering.

7. Jaccard Coefficient. The JC, also known as the Jaccard similarity coefficient (originally named the "coefficient de commutate" by Paul Jaccard), is a statistic applied to compare the similarity and diversity of sample sets [46]. JC is computed as follows:

$$J = \frac{a}{(a+b+c)} \tag{9.28}$$

A higher value of JC indicates a more accurate result of clustering.

8. Fowlkes and Mallows Index. The Fowlkes and Mallows index (FMI) was proposed by Fowlkes and Mallows [47] as an alternative for the RI. The measure of FMI is computed as follows:

$$\text{FMI} = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}} \tag{9.29}$$

A higher value of FMI indicates more accurate clustering.

9. Adjusted Rand Index. The adjusted Rand index (ARI) is the corrected-for-chance version of the measure of RI. It ranges from -1 to 1 and expresses the level of concordance between two bipartitions [48]. A value of ARI closest to 1 indicates almost perfect concordance between the two compared bipartitions, whereas a

value near −1 indicates almost complete discordance [49]. The measure of ARI is computed as:

$$\mathrm{ARI} = \frac{a - \left((a+c) + \frac{a+b}{M}\right)}{\left((a+c) + \frac{a+b}{2}\right) - \left((a+c) + \frac{a+b}{M}\right)} \tag{9.30}$$

A higher value of ARI indicates more accurate clustering.

9.1.1.6 Index Weights

In this work, the index weights of the four MCDM methods can be calculated by AHP. The AHP method, proposed by Saaty [50] is a widely used tool for modeling unstructured problems by synthesizing subjective and objective information in many disciplines, such as politics, economics, biology, sociology, management science, and life sciences [51–53]. It can elicit a corresponding priority vector according to pair-by-pair comparison values [54] obtained from the scores of experts on an appropriate scale. AHP has some problems, for example, the priority vector derived from the eigenvalue method can violate a condition of order preservation pro-posed by Costa and Vansnick [55]. However, AHP is still a classic and important approach, especially in the fields of operation research and management science [56]. AHP has the following steps:

1. Establish a hierarchical structure: a complex problem can be established in such a structure, including the goal level, criteria level, and alternative level [57].
2. Determine the pairwise comparison matrix: once the hierarchy is structured, the prioritization procedure starts for determining the relative importance of the criteria (index weights) within each level [5]. The pairwise comparison values are obtained from the scores of experts on a 1–9 scale.
3. Calculate index weights: the index weights are usually calculated by the eigen-vector method proposed by Saaty [50].
4. Test consistency: the value of 0.1 is generally considered the acceptable upper limit of the consistency ratio (CR). If the CR exceeds this value, the procedure must be repeated to improve consistency.

9.1.1.7 The Proposed Model

Clustering results can vary according to the evaluation method. Rankings can conflict even when abundant data are processed, and a large knowledge gap can exist between the evaluation results [58] due to the anticipation, experience, and expertise of all individual participants. The decision-making process is extremely complex. This makes it difficult to make accurate and effective decisions [41]. The proposed DMSECA model consists of three steps. They are as follows.

The first step usually involves modeling by clustering algorithms, which can be accomplished using one or more procedures selected from the categories of hierarchical, density-based, partitioning, and model-based methods. In this section, we apply the six most influential clustering algorithms, including EM, the FF algorithm, FC, HC, MD, and KM, for task modeling by using WEKA 3.7 on 20 UCI data sets, including a total of 18,310 instances and 313 attributes. Each of these clustering algorithms belongs to one of the four categories of clustering algorithms mentioned previously. Hence, all categories are represented.

In the second step, four commonly used MCDM methods (TOPSIS, WSM, GRA, and PROMETHEE II) are applied to rank the performance of the clustering algorithms over 20 UCI data sets based on nine external measures as the input, computed in the first step. These methods are highly suitable for the given data sets. Unsuitable methods were not selected. For example, we did not select VIKOR because its denominator would be zero for the given data sets. The index weights are determined by AHP based on the eigenvalue method. Three experts from the field of MCDM are selected and consulted as the DMs to derive the pairwise comparison values completed by the scores of experts. We randomly assign each MCDM method to five UCI data sets. We apply more than one MCDM method to analyze and evaluate the performance of clustering algorithms, which is essential.

Finally, in the third step, we propose a decision-making support model to reconcile the individual differences or even conflicts in the evaluation performance of the clustering algorithms among the 20 UCI data sets. The proposed model can generate a list of algorithm priorities to select the most appropriate clustering algorithm for secondary mining and knowledge discovery. The detailed steps of the decision-making support model, based on the 80-20 rule, are described as follows.

Step 1. Mark two sets of alternatives in a lower position and an upper position, respectively.

It is well known that the 80-20 rule reports that 80% of the results originate in 20% of the activity in most situations. The rule can be credited to Vilfredo Pareto, who observes that 80% of the wealth is usually controlled by 20% of the people in most countries. The implication is that it is better to be in the top of 20% than in the bottom of 80%. So, the 80-20 rule can be applied to focus on the analysis of the most important positions of the rankings in relation to the number of observations for predictable imbalance. The 80-20 rule indicates that the 20% of people, who are creating 80% of the results, which are highly leveraged. In this research, based on the expert wisdom originating from the 20% of people, the set of alternatives is classified into two categories, where the top of 1/5 of the alternatives is marked in an upper position, which represents more satisfactory rankings from the opinion of all individual participants involved in the algorithm evaluation process. The bottom of 1/5 is in a lower position, which represents more dissatisfactory rankings from the opinion of all individual participants. The

element marked in the upper position is calculated as follows:

$$x = \frac{n * 1}{5} \tag{9.31}$$

where n is the number of alternatives. For instance, if n 7, then $\times$ 7 $\times$ 1/5 1.4 $\approx$ 2. Hence, the second position classifies the ranking, where the first and second positions are those alternatives in the upper position, which are considered as the collective group idea of the most appropriate and satisfactory alternatives.
Similarly, the element marked in the lower position is calculated as

$$x = \frac{n * 4}{5} \tag{9.32}$$

where n is the number of alternatives. For instance, if n = 7, then 7*4/5 = 5.6 $\approx$ 6. Thus, the sixth position classifies the ranking, where the sixth and seventh positions in the lower positions are considered collectively as the worst and most dissatisfactory alternatives.

Step 2. Grade the sets of alternatives in the lower and upper positions, respectively.
A score is assigned to each position of the set of alternatives in the lower position and upper position, respectively.
The score in the lower position can be calculated by assigning a value of 1 to the first position, 2 to the second position, . . . , and x to the last position. Finally, the score of each alternative in the lower position is totaled, marked d.
Similarly, the score in the upper position can be calculated by assigning a value of 1 to the last position, 2 to the penultimate position, . . . , and x to the first position. Finally, the score of each alternative in the upper position is totaled, marked b.

Step 3. Generate the priority of each alternative.
The priority of each alternative fi, which represents the most satisfactory rankings from the opinions of all individual participants, can be determined as

$$f_i = b_i - d_i \tag{9.33}$$

where a higher value of f_i implies a higher priority.

9.1.2 Evaluation of Classification Algorithms Using MCDM And Rank Correlation

This subsection combines MCDM methods with Spearman's rank correlation coefficient to rank classification algorithms. This approach first uses several MCDM methods to rank classification algorithms and then applies Spearman's rank correlation coefficient to resolve differences among MCDM methods. Five MCDM

methods, including TOPSIS, ELECTRE III, grey relational analysis, VIKOR, and PROMETHEE II are implemented in this research.

9.1.2.1 Two MCDM Methods

In addition to GRA, TOPSIS, and PROMETHEE II methods, here two more MCDM methods are outlined as below.

ELimination and Choice Expressing REality (ELECTRE)

ELECTRE stands for ELimination Et Choix Traduisant la REalite (ELimination and Choice Expressing the REality) and was first proposed by Roy [59] to choose the best alternative from a collection of alternatives. Over the last four decades, a family of ELECTRE methods has been developed, including ELECTRE I, ELECTRE II, ELECTRE III, ELECTRE IV, ELECTRE IS, and ELECTRE TRI.

There are two main steps of ELECTRE methods: the first step is the construction of one or several outranking relations; the second step is an exploitation procedure that identifies the best compromise alternative based on the outranking relation obtained in the first step.[60] ELECTRE III is chosen in this section because it is appropriate for the sorting problem. The procedure can be summarized as follows [59, 61, 62]:

Step 1. Define a concordance and discordance index set for each pair of alternatives

$$A_j \text{ and } A_k,\ j,k = 1,\ldots,m;\ i \neq k$$

Step 2. Add all the indices of an alternative to get its global concordance index C_{ki}.
Step 3. Define an outranking credibility degree $\sigma_s(A_i, A_k)$; by combining the discordance indices and the global concordance index.
Step 4. Define two outranking relations using descending and ascending distillation. Descending distillation selects the best alternative first and the worst alternative last. Ascending distillation selects the worst alternative first and the best alternative last.
Step 5. Alternatives are ranked based on ascending and descending distillation processes.

VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR)

VIKOR was proposed by Opricovic [63] and Opricovic and Tzeng [64] for multicriteria optimization of complex systems. The multicriteria ranking index, which is based on the particular measure of closeness to the ideal alternative, is introduced to rank alternatives in the presence of conflicting criteria. This section

uses the following VIKOR algorithm provided by Opricovic and Tzeng in the experiment:

Step 1. Determine the best f_i^* and the worst f_i^- values of all criterion functions, $i = 1, 2, \cdots, n$.

$$f_i^* = \left\{ \begin{matrix} \max_j f_{ij}, \text{for benefit criteria} \\ \min_j f_{ij}, \text{for cost criteria} \end{matrix} \right\}, j = 1, 2, \ldots, J,$$
$$f_i^- = \left\{ \begin{matrix} \min_j f_{ij}, \text{for benefit criteria} \\ \max_j f_{ij}, \text{for cost criteria} \end{matrix} \right\}, j = 1, 2, \ldots, J,$$

where J is the number of alternatives, n is the number of criteria, and f_{ij} is the rating of ith criterion function for alternative aj.

Step 2. Compute the values S_j and R_j; $j = 1, 2, \cdots, J$, by the relations

$$S_j = \sum_{i=1}^{n} w_i \left(f_i^* - f_{ij}\right)\left(f_i^* - f_i^-\right)$$
$$R_j = \max_i \left[w_i \left(f_i^* - f_{ij}\right)\left(f_i^* - f_i^-\right)\right]$$

where wi is the weight of ith criteria, Sj and Rj are used to formulate ranking measure.

Step 3. Compute the values Qj; $j = 1, 2, \cdots, J$, by the relations

$$Q_j = v\left(S_j - S^*\right)\left(S^- - S^*\right) + (1 - v)\left(R_j - R^*\right)\left(R^- - R^*\right)$$
$$S^* = \min_j S_j, S^- = \max_j S_j$$
$$R^* = \min_i R_j, R^- = \max_j R_j$$

where the solution obtained by S is with a maximum group utility, the solution obtained by R is with a minimum individual regret of the opponent, and v is the weight of the strategy of the majority of criteria. The value of v is set to 0.5 in the experiment.

Step 4. Rank the alternatives in decreasing order. There are three ranking lists: S; R, and Q.

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

(a) $Q(a'') - Q(a') \geq 1(J - 1)$; (b) Alternative a 0 is ranked the best by S or/and R.

If only the condition (b) is not satisfied, alternatives a' and a'' are proposed as compromise solutions, where a′′ is ranked the second by Q. If the condition (a) is not satisfied, alternatives a'; $a'' \ldots$; a^M are proposed as compromise solutions, where a^M is ranked the Mth by Q and is determined by the relation $Q(a^M) - Q(a') < 1(J - 1)$ for maximum M.

9.1.2.2 Spearman's Rank Correlation Coefficient

Spearman's rank correlation coefficient measures the similarity between two sets of rankings. The basic idea of the proposed approach is to assign a weight to each MCDM method according to the similarities between the ranking it generated and the rankings produced by other MCDM methods. A large value of Spearman's rank correlation coefficient indicates a good agreement between a MCDM method and other MCDM methods.

The proposed approach is designed to handle conflicting MCDM rankings through three steps. In the first step, a selection of MCDM methods is applied to rank classification algorithms. If there are strong disagreements among MCDM methods, the different ranking scores generated by MCDM methods are used as inputs for the second step.

The second step utilizes Spearman's rank correlation coefficient to find the weights for each MCDM method. Spearman's rank correlation coefficient between the kth and ith MCDM methods is calculated by the following equation:

$$\rho_{ki} = 1 - \frac{6\sum d_i^2}{n\left(n^2 - 1\right)} \tag{9.34}$$

where n is the number of alternatives and di is the difference between the ranks of two MCDM methods. Based on the value of k_i, the average similarities between the kth MCDM method and other MCDM methods can be calculated as

$$\rho_k = \frac{1}{q-1}\sum\nolimits_{i=1, i\neq k}^{q} \rho_{ki}, k = 1, 2, \ldots, q, \tag{9.35}$$

where q is the number of MCDM methods. The larger the k value, the more important the MCDM method is. Normalized k values can then be used as weights for MCDM methods in the secondary ranking.

The third step uses the weights obtained from the second step to get secondary rankings of classifiers. Each MCDM method is applied to re-rank classification algorithms using ranking scores produced by MCDM methods in the first step and the weights obtained in the second step.

- The detailed experimental study of this method can be found in [2]

9.1.3 Public Blockchain Evaluation Using Entropy and TOPSIS

This subsection aims to make a comprehensive evaluation of public blockchains from multiple dimensions. Three first-level indicators and eleven second-level indicators are designed to evaluate public blockchains. The technique for order

preferences by similarity to an ideal solution (TOPSIS) method is used to rank public blockchains, and the entropy method is used to determine the weights of each dimension. Since Bitcoin has an absolute advantage, a let-the-first-out (LFO) strategy is proposed to reduce the criteria of the positive ideal solution and make a more reasonable evaluation.

9.1.3.1 Proposed Evaluation Model

Evaluation Indicator

With the increasing requirement of performance, more and more blockchains are designed by new technology. Technology is an important indicator to evaluate public blockchains, but technology is not everything. The popularity is a key factor to measure a platform or system, and the blockchain is the same. For example, the second global public blockchain technology assessment index shows that Bitcoin ranks 17th, but Bitcoin is still one of the most popular blockchains.

Therefore, two indicators are designed to measure the popularity of public blockchains. One is recognition, which is the degree of acceptance of public blockchains by developers and others. The greater the acceptance, the better the blockchain. The other is activity, which measures the activity of developers and others. When developers stop maintaining and improving a blockchain, or people stop talking about it, the blockchain is no longer popular. Developers and other people can be considered separately, but they are under the same indicator in this section because of the same topic. Figure 9.1 shows the first-level indicators and their second-level indicators.

Technology

The basic technology (I_{11}) and the applicability (I_{12}) are the first and the second second-level indicators of technology respectively. These two indicators are quantified by the expert scoring method. Since CCID has established a technology assessment index for public blockchains, this section will reference its scoring results for the two indicators. The basic technology mainly examines the realization function, basic performance, safety and degree of centralization of public blockchains. The applicability focuses on the application scenarios, the number of wallets, the ease of use, and the development support on the chain.

The TPS (I_{13}) is the most important indicator of public blockchain networks. The TPS of Bitcoin and Ethereum are 7 and 20 respectively, while the TPS of VISA is 2000. A blockchain's TPS depends on its consensus algorithm, and the POW consensus algorithm makes the TPS of Bitcoin and Ethereum small.

In November 2017, Ethereum launched a pet cat game called CryptoKitties. Since December 3, 2017, pending transactions at Ethereum have skyrocketed. CryptoKitties accounted for more than 10% of the activity in Ethereum, resulting in

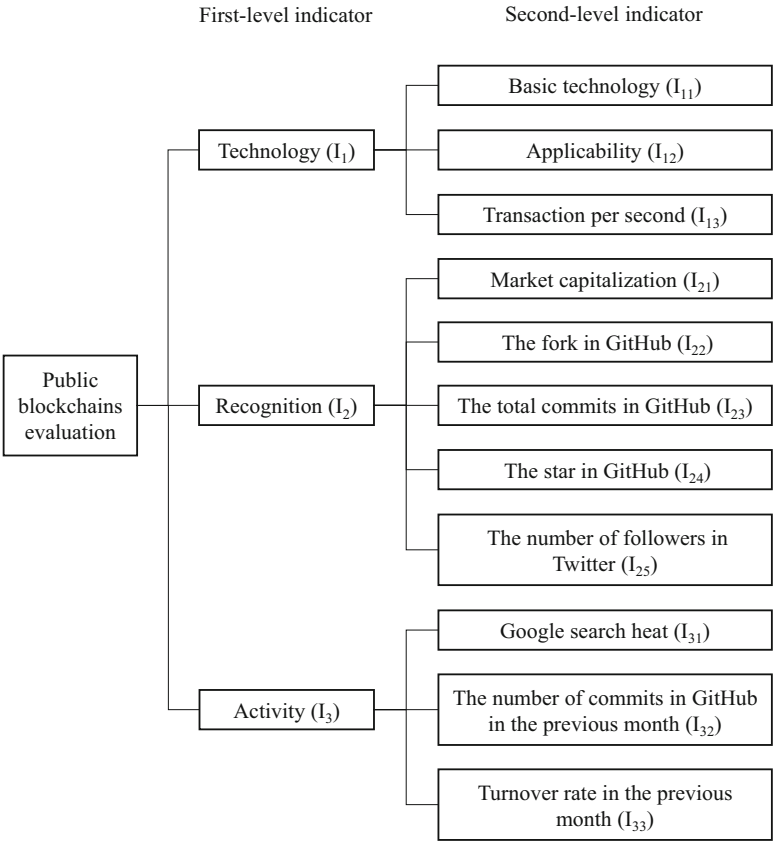

Fig. 9.1 The evaluation indicators for public blockchains evaluation

serious congestion in the Ethereum network. The gas fee, also called transaction fee, is required to be paid to the miners to run a particular transaction or contract. With the congestion of the Ethereum network, the gas fee will increase. As can be seen in Fig. 9.2, the gas fee increases rapidly since December 3, 2017. Additionally, the congestion appears again in the Ethereum network since June 30, 2018, because of the principles of FCoin GPM listing. These high transaction costs show the congestion in the Ethereum network. Since people pay most attention to the TPS nowadays, the TPS is independent of the I_{11} as the third second-level indicator of technology.

However, even if the TPS needs to be upgraded to solve the congestion problem, too large TPS is meaningless. For example, if 2000 TPS is enough to handle the daily transactions, there is no difference between 5000 TPS and million TPS. In this case, the hyperbolic tangent function is introduced to reduce the benefits of the

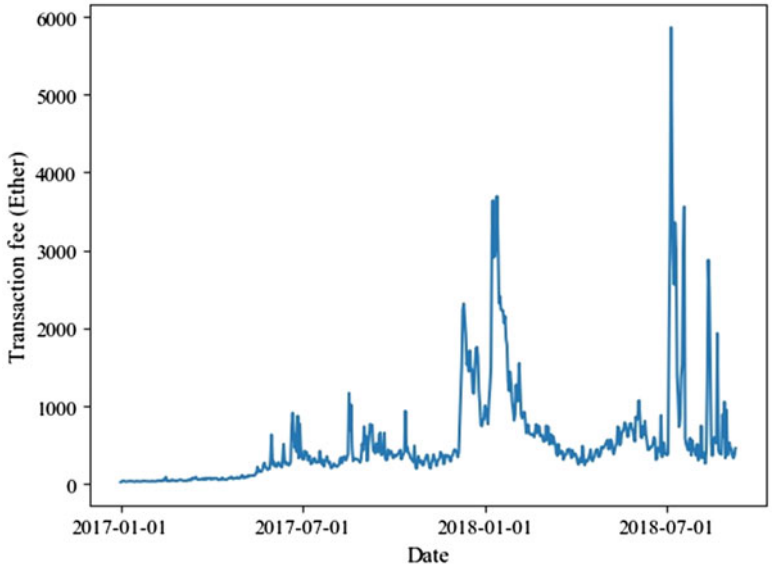

Fig. 9.2 The transaction fee of Ethereum network

increased TPS:

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}, x = \frac{\text{TPS}}{\alpha} \tag{9.36}$$

where α is a scale factor and set to 2000 in this section.

Recognition

The market capitalization (I_{21}) is the first second-level indicator of recognition. The market capitalization of a company is the result of the transaction price of the company's stock in the securities market multiplied by the total share capital, reflecting the company's asset value, profitability value, and growth value. Similarly, the market capitalization of a public blockchain is the result of the transaction price of the public blockchain's coin in the cryptocurrency market multiplied by the total number of coins. It reflects the blockchain's use value and growth value. Once a blockchain is not recognized and no longer used, its value will be zero.

The fork (I_{22}), the total commits (I_{23}), and the star (I_{24}) in GitHub are the second, third, and fourth second-level indicator of recognition respectively. A basic technical feature of the blockchain is the shared ledger, which requires multiple participation and cooperation. Due to the openness and transparency of the open source, the open source of blockchain not only quickly obtain the recognition and trust of partners, but also quickly gather a number of outstanding talents for continuous

developments. The fork in GitHub represents the number of people who recognize or want to contribute to the blockchain; the total commits in GitHub represent the improvements of the blockchain; the star in GitHub represents the number of developers who like the blockchain.

The number of followers in Twitter (I_{25}) is the fifth second-level indicator of recognition. Twitter is one of the most famous online news and social networking service. The blockchains always have Twitter accounts to post news to the public, and the followers of a public blockchain's Twitter account represent the people who care and recognize the public blockchain.

Activity

The Google search heat in the previous month (I_{31}) is the first second-level indicator of activity. In the search market, Google handles around 90% of searches worldwide. The popularity of search terms over time and across various regions of the world can be compared in Google Trends. The Google search heat of a public blockchain is the sum of its name's search heat and its short name's search heat.

The number of commits in GitHub in the previous month (I_{32}) is the second second-level indicator of activity. It reflects the improvements of blockchains in the previous month.

The turnover rate in the previous month (I_{33}) is the third second-level indicator of activity. The turnover rate is the frequency of coins traded in the market in a certain period of time. The higher the turnover rate, the more active the transactions of cryptocurrency and the more popular the public blockchain. Generally, a high turnover rate means good liquidity of the cryptocurrency.

Evaluation Process

The choice of indicators weights is an important step in the TOPSIS. The entropy method is an objective method to calculate weights based on the objective information of indicators [65]. An indicator with small entropy value means the indicator is important and has a large weight [66]. The entropy is calculated as follows:

$$e_j = -\frac{1}{\ln n}\sum_{i=1}^{n} p_{ij}\ln p_{ij}, \quad p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \tag{9.37}$$

where x_{ij} is the jth normalized indicator value of the ith public blockchain. Then the degree of divergence (d_j) and the weight (w_j) can be calculated as follows:

$$d_j = 1 - e_j \tag{9.38}$$

$$w_j = \frac{d_j}{\sum_{j=1}^{m} d_j} \tag{9.39}$$

The TOPSIS ranks public blockchains according to their relative proximities calculated by the distance from the positive ideal solution and the distance from the negative ideal solution [67]. The steps for the TOPSIS are described below. The first step is to normalize the indicator matrix:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \tag{9.40}$$

With the weights obtained by the entropy method, the weighted normalization matrix is calculated as follows:

$$v = r \cdot \mathrm{diag}(w) \tag{9.41}$$

where $\mathrm{diag}(w)$ is a diagonal matrix where the diagonal elements are the weights w. Then the positive ideal solution (A^+) and the negative ideal solution (A^-) can be obtained:

$$A^+ = \left\{ \left(\max_i v_{ij} | j \in J_1 \right), \left(\min_i v_{ij} | j \in J_2 \right) | i = 1, 2, \ldots, n \right\} = \left\{ v_1^+, v_2^+, \ldots, v_j^+, \ldots, v_m^+ \right\} \tag{9.42}$$

$$A^- = \left\{ \left(\min_i v_{ij} | j \in J_1 \right), \left(\max_i v_{ij} | j \in J_2 \right) | i = 1, 2, \ldots, n \right\} = \left\{ v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_m^- \right\} \tag{9.43}$$

where J_1 and J_2 are the benefit and the cost indicators respectively. The distance of each indicator from A^+ and A^- can be calculated as follows:

$$S_i^+ = \sqrt{\sum_{j=1}^{m} \left(v_{ij} - v_j^+ \right)^2}, i = 1, 2, \ldots, n \tag{9.44}$$

$$S_i^- = \sqrt{\sum_{j=1}^{m} \left(v_{ij} - v_j^- \right)^2}, i = 1, 2, \ldots, n \tag{9.45}$$

The relative proximity of each public blockchain to the ideal solution can be calculated as follows:

$$C_i^* = \frac{S_i^-}{S_i^+ + S_i^-}, i = 1, 2, \ldots, n \tag{9.46}$$

Lastly, the public blockchains can be ranked by their relative proximities.

The relative proximities are based on the positive ideal solution and the negative ideal solution. If the relative proximity of the first place is much larger than that of the second place, then some indicator values of the first place are much larger than those of the second place. In this case, even if the second place is much better than the third place, the advantage will become very small under the absolute advantage

of the first place. Since the positive ideal solution cannot be achieved by other items, it is better to reduce the criteria of the positive ideal solution. Therefore, a let-the-first-out (LFO) strategy is proposed to make a more reasonable evaluation. In the LFO, if the relative proximity of the first place is much larger than that of the second place, the position of the first place is retained and the other items are re-evaluated.

- The data analysis can be found in [3].

9.2 Evaluation Methods for Software

9.2.1 Classifier Evaluation for Software Defect Prediction

This subsection integrates traditional feature selection methods and multi-criteria decision making (MCDM) methods to improve the accuracy and reliability of defect prediction models and evaluate the performances of software defect detection models.

9.2.1.1 Research Methodology

Results of empirical studies on software defect prediction models do not always converge. Myrtveit et al. [68] analyzed some empirical software engineering studies and identified three factors that may contribute to the divergence: a single sample dataset, choice of accuracy indicators, and cross validation. They concluded that a crucial step in software defect prediction is the design of research procedures.

The inputs are four public-domain software defect datasets provided by the NASA IV&V Facility Metrics Data Program (MDP) repository. Feature selection and classification are conducted in four steps. First, feature selection is conducted using traditional techniques. Features are then ranked using the proposed feature selection method. The third step employs MCDM methods to evaluate feature selection techniques and choose the better performed techniques. In the last step, the selected features are used in the classification to predict software defects. The performances of classifiers are also evaluated using MCDM methods and a recommendation of classifiers for software defect prediction is made based on their accuracy and reliability.

Multiple criteria decision making (MCDM) aims at solving decision problems with multiple objectives and often conflictive constraints [40, 68, 69]. Five MCDM methods, i.e., DEA (BCC model), ELECTRE, PROMETHEE, TOPSIS, and VIKOR, are used in the experimental study to evaluate algorithms.

For feature selection algorithms, output components include seven attributes:

LOC_COMMENTS (The number of lines of comments in a module),
HALSTEAD_PROG_TIME (The halstead programming time metric of a module),
MAINTENANCE_SEVERITY (Maintenance Severity),
NODE_COUNT (Number of nodes found in a given module),
NUM_OPERATORS (The number of operators contained in a module),
NUM_UNIQUE_OPERATORS (The number of unique operators contained in a module),
PERCENT_COMMENTS (Percentage of the code that is comments).

All other attributes are input components. For classification algorithms, input component is false positive rate and output components include the area under receiver operating characteristic (AUC), precision, F-measure, and true positive rate.

9.2.1.2 Experimental Study

Data Sources

The data used in this study are modified public-domain software defect datasets provided by the NASA IV&V Facility Metrics Data Program (MDP) repository [70]. The structures of the datasets are summarized in Table 9.2.

CM is from a science instrument written in a C code with approximately 20 kilo-source lines of code (KLOC). KC is about the collection, processing and delivery of satellite metadata and is written in Java with 18 KLOC. PC is flight software from an earth orbiting satellite written in a C code with 26 KLOC. UC is dynamic simulator for attitude control systems. Forty common attributes are selected for each dataset.

Discussion of Results

Table 9.3 summarizes the feature weights for each dataset. Features that are highly ranked in one or two dataset may have low rankings in other datasets, such as attribute 4, 9, and 27. This indicates that performances of feature selection techniques vary at different datasets. It also shows a need for evaluation of feature selection techniques.

Table 9.2 Dataset structures

Dataset	Number of instances	Normal instances	Bug instances
CM	568	425	143
KC	804	495	309
PC	4472	3718	754
UC	10, 064	9285	779

Table 9.3 Feature weights for the four datasets

Attributes	CM Data		KC Data		PC Data		UC Data	
	W	R	W	R	W	R	W	R
att1	0.57	7	0.44	24	0.60	12	0.68	3
att2	0.26	37	0.40	30	0.35	39	0.22	39
att3	0.64	5	0.59	8	0.63	8	0.47	28
att4	0.27	36	0.57	9	0.95	1	0.74	2
att5	0.57	8	0.51	15	0.55	17	0.64	9
att6	0.48	21	0.30	39	0.43	31	0.48	26
att7	0.41	26	0.55	12	0.51	21	0.40	35
att8	0.44	23	0.33	37	0.65	7	0.67	4
att9	0.68	3	0.35	33	0.47	27	0.31	38
att10	0.33	33	0.64	2	0.69	4	0.62	13
att11	0.50	19	0.48	19	0.52	20	0.47	29
att12	0.33	32	0.57	10	0.55	18	0.62	12
att13	0.56	10	0.51	16	0.45	29	0.60	14
att14	0.51	18	0.44	23	0.63	9	0.60	15
att15	0.52	17	0.34	36	0.40	35	0.42	34
att16	0.24	39	0.51	14	0.49	24	0.45	30
att17	0.49	20	0.49	18	0.56	16	0.66	6
att18	0.56	9	0.41	29	0.29	40	0.18	40
att19	0.29	35	0.61	4	0.43	34	0.43	33
att20	0.43	25	0.60	5	0.77	2	0.79	1
att21	0.47	22	0.47	21	0.43	32	0.53	19
att22	0.52	14	0.44	25	0.49	25	0.47	27
att23	0.43	24	0.43	26	0.43	33	0.59	17
att24	0.52	16	0.39	31	0.49	23	0.53	21
att25	0.54	12	0.49	17	0.59	13	0.52	22
att26	0.55	11	0.42	28	0.58	14	0.55	18
att27	0.72	1	0.43	27	0.39	37	0.40	36
att28	0.62	6	0.54	13	0.46	28	0.33	37
att29	0.38	30	0.34	35	0.40	36	0.49	23
att30	0.65	4	0.38	32	0.65	6	0.65	8
att31	0.24	38	0.32	38	0.45	30	0.48	24
att32	0.37	31	0.45	22	0.62	10	0.60	16
att33	0.40	28	0.25	40	0.47	26	0.43	32
att34	0.32	34	0.35	34	0.50	22	0.53	20
att35	0.70	2	0.64	3	0.68	5	0.62	11
att36	0.52	13	0.59	7	0.57	15	0.65	7
att37	0.41	27	0.56	11	0.61	11	0.64	10
att38	0.52	15	0.47	20	0.36	38	0.43	31
att39	0.40	29	0.65	1	0.73	3	0.66	5
att40	0.21	40	0.59	6	0.53	19	0.48	25

W for Weight, R for Rank

Table 9.4 MCDM evaluation of classifiers for CM dataset

	DEA	ELECTRE	PROMETHEE	TOPSIS	VIKOR
Naïve Bayes	2	3	2	2	2
Logistic	8	7	6	6	1
RBFNetwork	7	5	7	5	6
SMO	6	9	4	8	5
IB1	5	8	8	9	9
FLR	1	1	1	1	3
DecisionTable	3	6	9	3	4
RIPPER	9	2	3	7	7
C4.5	4	4	5	4	8

Table 9.5 MCDM evaluation of classifiers for KC dataset

	DEA	ELECTRE	PROMETHEE	TOPSIS	VIKOR
Naïve Bayes	5	5	3	2	7
Logistic	1	2	2	1	1
RBFNetwork	7	4	4	4	9
SMO	6	6	6	5	8
IB1	9	9	5	7	6
FLR	4	1	1	3	3
DecisionTable	3	3	7	6	2
RIPPER	2	8	9	9	5
C4.5	8	7	8	8	4

The five MCDM methods are applied to evaluate the 11 feature selection techniques.

Tables 9.4, 9.5, 9.6, and 9.7 summarize the evaluation results of the nine classifiers on the four datasets. The rankings of classifiers vary with different datasets. Even within a dataset, different MCDM methods may produce divergent rankings for the same classifier. For example, RIPPER was ranked the second best classifier by ELECTRE and the worst classifier by DEA for CM dataset. In general, FLR outperforms other classifiers. It was ranked the best classifier by at least two MCDM methods for every dataset. SMO achieves good performances on PC and UC, which are larger than CM and KC. The performances of other classifiers on the four software defect datasets are rather mixed.

9.2.2 Ensemble of Software Defect Predictors: An AHP-Based Evaluation Method

This subsection evaluates the quality of ensemble methods for software defect prediction with the analytic hierarchy process (AHP) method. The AHP is a

Table 9.6 MCDM evaluation of classifiers for PC dataset

	DEA	ELECTRE	PROMETHEE	TOPSIS	VIKOR
Naïve Bayes	9	9	3	4	7
Logistic	8	6	7	7	5
RBFNetwork	2	3	4	3	3
SMO	1	1	2	2	1
IB1	5	8	9	9	9
FLR	4	2	1	1	2
DecisionTable	3	4	6	6	6
RIPPER	7	5	5	5	8
C4.5	6	7	8	8	4

Table 9.7 MCDM evaluation of classifiers for UC dataset

	DEA	ELECTRE	PROMETHEE	TOPSIS	VIKOR
Naïve Bayes	5	8	3	4	6
Logistic	3	4	5	5	3
RBFNetwork	2	5	4	3	2
SMO	1	2	2	2	1
IB1	8	7	8	8	7
FLR	4	1	1	1	5
DecisionTable	7	3	7	6	4
RIPPER	6	9	6	7	8
C4.5	9	6	9	9	9

multicriteria decision-making approach that helps decision makers structure a decision problem based on pairwise comparisons and experts' judgments. Three popular ensemble methods (bagging, boosting, and stacking) are compared with 12 well-known classification methods using 13 performance measures over 10 public-domain datasets from the NASA Metrics Data Program (MDP) repository.[70] The classification results are then analyzed using the AHP to determine the best classifier for software defect prediction task.

9.2.2.1 Ensemble Methods

Ensemble learning algorithms construct a set of classifiers and then combine the results of these classifiers using some mechanisms to classify new data records [71]. Experimental results have shown that ensembles are often more accurate and robust to the effects of noisy data, and achieve lower average error rate than any of the constituent classifiers [15, 72–75].

How to construct good ensembles of classifiers is one of the most active research areas in machine learning, and many methods for constructing ensembles have been proposed in the past two decades [76]. Dietterich [71] divides these methods into five groups: Bayesian voting, manipulating the training examples,

manipulating the input features, manipulating the output targets, and injecting randomness. Several comparative studies have been conducted to examine the effectiveness and performance of ensemble methods. Results of these studies indicate that bagging and boosting are very useful in improving the accuracy of certain classifiers [77], and their performances vary with added classification noise. To investigate the capabilities of ensemble methods in software defect prediction, this study concentrates on three popular ensemble methods (i.e. bagging, boosting, and stacking) and compares their performances on public-domain software defect datasets.

Bagging

Bagging combines multiple outputs of a learning algorithm by taking a plurality vote to get an aggregated single prediction [78]. The multiple outputs of a learning algorithm are generated by randomly sampling with replacement of the original training dataset and applying the predictor to the sample. Many experimental results show that bagging can improve accuracy substantially. The vital element in whether bagging will improve accuracy is the instability of the predictor [78]. For an unstable predictor, a small change in the training dataset may cause large changes in predictions [79]. For a stable predictor, however, bagging may slightly degrade the performance [78].

Researchers have performed large empirical studies to investigate the capabilities of ensemble methods. For instance, Bauer and Kohavi [77] compared bagging and boosting algorithms with a decision tree inducer and a NaÏve Bayes inducer. They concluded that bagging reduces variance of unstable methods and leads to significant reductions in mean-squared errors. Dietterich [72] studied three ensemble methods (bagging, boosting, and randomization) using decision tree algorithm C4.5 and pointed out that bagging is much better than boosting when there is substantial classification noise.

In this subsection, bagging is generated by averaging probability estimates [16].

Boosting

Similar to bagging, boosting method also combines the different decisions of a learning algorithm to produce an aggregated prediction [80]. In boosting, however, weights of training instances change in each iteration to force learning algorithms to put more emphasis on instances that were predicted incorrectly previously and less emphasis on instances that were predicted correctly previously. Boosting often achieves more accurate results than bagging and other ensemble methods. However, boosting may overfit the data and its performance deteriorates with classification noise.

This study evaluates a widely used boosting method, AdaBoost algorithm, in the experiment. AdaBoost is the abbreviation for adaptive boosting algorithm because

it adjusts adaptively to the errors returned by classifiers from previous iterations [73, 81]. The algorithm assigns equal weight to each training instance at the beginning. It then builds a classifier by applying the learning algorithm to the training data. Weights of misclassified instances are increased, while weights of correctly classified instances are decreased. Thus, the new classifier concentrates more on incorrectly classified instances in each iteration.

Stacking

Stacking generalization, often abbreviated as stacking, is a scheme for minimizing the generalization error rate of one or more learning algorithms [82]. Unlike bagging and boosting, stacking can be applied to combine different types of learning algorithms. Each base learner, also called "level 0" model, generates a class value for each instance. The predictions of level-0 models are then fed into the level-1 model, which combines them to form a final prediction [16].

Another ensemble method used in the experiment is voting, which is a simple average of multiple classifiers probability estimates provided by WEKA [16].

9.2.2.2 Selected Classification Models

As a powerful tool that has numerous applications, classification methods have been studied extensively by several fields, such as machine learning, statistics, and data mining [83]. Previous studies have shown that an ideal ensemble should consist of accurate and diverse classifiers. [84] Therefore, this study selects 12 classifiers to build ensembles. They represent five categories of classifiers (i.e., trees, functions, Bayesian classifiers, lazy classifiers, and rules) and were implemented in WEKA.

For trees category, we chose classification and regression tree (CART), NaÏve Bayes tree, and C4.5. Functions category includes linear logistic regression, radial basis function (RBF) network, sequential minimal optimization (SMO), and Neural Networks. Bayesian classifiers include Bayesian network and NaÏve Bayes. K-nearest-neighbor was chosen to represent lazy classifiers. For rules category, decision table and Repeated Incremental Pruning to Produce Error Reduction (RIPPER) rule induction were selected.

Classification and regression tree (CART) can predict both continuous and categorical dependent attributes by building regression trees and discrete classes, respectively [85]. NaÏve Bayes tree is an algorithm that combines NaÏve Bayes induction algorithm and decision trees to increase the scalability and interpretability of NaÏve Bayes classifiers [86]. C4.5 is a decision tree algorithm that constructs decision trees in a top–down recursive divide-and-conquer manner [87].

Linear logistic regression models the probability of occurrence of an event as a linear function of a set of predictor variables [88]. Neural network is a collection of artificial neurons that learns relationships between inputs and outputs by adjusting the weights. RBF network [89] is an artificial neural network that uses radial basis

functions as activation functions. The centers and widths of hidden units are derived using k-means, and the outputs obtained from the hidden layer are combined using logistic regression [16]. SMO is a sequential minimal optimization algorithm for training support vector machines (SVM) [90, 91].

Bayesian network and NaÏve Bayes both model probabilistic relationships between the predictor variables and the class variable. While NaÏve Bayes classifier [92] estimates the class-conditional probability based on Bayes theorem and can only represent simple distributions, Bayesian network is a probabilistic graphic model and can represent conditional independencies between variables [93].

K-nearest-neighbor [94] classifies a given data instance based on learning by analogy. That is, it assigns an instance to the closest training examples in the feature space.

Decision table selects the best-performing attribute subsets using best-first search and uses cross-validation for evaluation [95]. RIPPER [96] is a sequential covering algorithm that extracts classification rules directly from the training data without generating a decision tree first.

Each of stacking and voting combines all classifiers to generate one prediction. Since bagging and boosting are designed to combine multiple outputs of a single learning algorithm, they are applied to each of the 12 classifiers and produced a total of 26 aggregated outputs.

9.2.2.3 The Analytic Hierarchy Process (AHP)

The analytic hierarchy process is a multicriteria decision-making approach that helps decision makers structure a decision problem based on pairwise comparisons and experts' judgments [97, 98]. Saaty [99] summarizes four major steps for the AHP. In the first step decision makers define the problem and decompose the problem into a three-level hierarchy (the goal of the decision, the criteria or factors that contribute to the solution, and the alternatives associated with the problem through the criteria) of interrelated decision elements [100]. The middle level of criteria might be expanded to include subcriteria levels. After the hierarchy is established, the decision makers compare the criteria two by using a fundamental scale in the second step. In the third step, these human judgments are converted to a matrix of relative priorities of decision elements at each level using the eigenvalue method. The fourth step calculates the composite or global priorities for each decision alternatives to determine their ratings.

The AHP has been applied in diverse decision problems, such as economics and planning, policies and allocations of resources, conflict resolution, arms control, material handling and purchasing, manpower selection and performance measurement, project selection, marketing, portfolio selection, model selection, politics, and environment [101]. Over the last 20 years, the AHP has been studied extensively and various variants of the AHP have been proposed. [102–105].

In this study, the decision problem is to select the best ensemble method for the task of software defect prediction. The first step of the AHP is to decompose

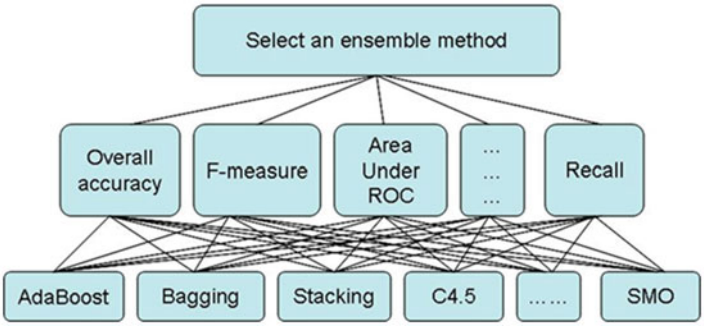

Fig. 9.3 An AHP hierarchy for the ensemble selection problem

the problem into a decision hierarchy. As shown in Fig. 9.3, the goal is to select an ensemble method that is superior to other ensemble methods over public-domain software defect datasets through the comparison of a set of performance measurements. The criteria are performance measures for classifiers, such as overall accuracy, F-measure, area under ROC (AUC), precision, recall, and Kappa statistic. The decision alternatives are ensembles and individual classification methods, such as AdaBoost, bagging, stacking, C4.5, SMO, and NaÏve Bayes. Individual classifiers are included as the decision alternatives for the purpose of comparisons.

In step 2, the input data for the hierarchy, which is a scale of numbers that indicates the preference of decision makers about the relative importance of the criteria, are collected. Saaty [97] provides a fundamental scale for this purpose, which has been validated theoretically and practically. The scale ranges from 1 to 9 with increasing importance. Numbers 1, 3, 5, 7, and 9 represent equal, moderate, strong, very strong, and extreme importance, respectively, while 2, 4, 6, and 8 indicate inter-mediate values. This study uses 13 measures to assess the capability of ensembles and individual classifiers. Previous works have proved that the AUC is the most informative and objective measurement of predictive accuracy [106] and is an extremely important measure in software defect prediction. Therefore, it is assigned a number of 9. The F-measure, mean absolute error, and overall accuracy are very important measures, but less important than the AUC. The true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), precision, recall, and Kappa statistic are strongly important classification measures that are less important than the F-measure, mean absolute error, and overall accuracy. Training and test time refer to the time needed to train and test a classification algorithm or ensemble method, respectively. They are useful measures in real-time software defect identification. Since this study is not aimed at real-time software defect identification problem, they are included to measure the efficiency of ensemble methods and are given the lowest importance.

The third step of the AHP computes the principal eigenvector of the matrix to estimate the relative weights (or priorities) of the criteria. The estimated priorities are obtained through a two-step process: (1) raise the matrix to large powers

(square); (2) sum and normalize each row. This process is repeated until the difference between the sums of each row in two consecutive rounds is smaller than a prescribed value. After obtaining the priority vector of the criteria level, the AHP method moves to the lowest level in the hierarchy, which consists of ensemble methods and classification algorithms in this experiment. The pairwise comparisons at this level compare learning algorithms with respect to each performance measure in the level immediately above. The matrices of comparisons of the learning algorithms with respect to the criteria and their priorities are analyzed and summarized in Sect. 9.2.1.2. The ratings for the learning algorithms are produced by aggregating the relative priorities of decision elements [107].

- The data analysis can be found in [5]

9.3 Evaluation Methods for Sociology and Economics

9.3.1 Delivery Efficiency and Supplier Performance Evaluation in China's E-Retailing Industry

This subsection focuses on overall and sub-process supply chain efficiency evaluation using a network slacks-based measure model and an undesirable directional distance model. Based on a case analysis of a leading Chinese B2C firm W, a two-stage supply chain structure covering procurement-stock and inventory-sale management is constructed.

In Chinese B2C e-commerce websites, two typical operation models are widely taken based on different strategic positioning. One is the third-party platform model which provides an e-commerce platform, technical support, advertising and marketing services for franchises. The leading B2C e-commerce platform in China is Taobao.com and Tmall.com. Their business revenue stems mainly from commissions and service. Another model is called the self-operated model, which has a logistics system for transferring and distributing goods. Examples include companies such as Jingdong, Dangdang, Amazon, Yihaodian and Suning. The source of their profits is that sales revenues decrease purchasing costs. According to a research report from IResearch, a leading internet consultant company and online media in China, platform model companies like Tmall accounts for most of B2C e-commerce market share, as shown in Fig. 9.4.

However, with the ongoing rapid growth of e-commerce in virtual markets, logistics has become the largest bottleneck of e-commerce's constant development. Most e-commerce players take the third party logistics (3PL) model in the initial development because of its advantage in reducing operations costs and capital investment. Because 3PL is either contractual or out-sourced logistics concentrating on regional operations, with business expansion, the drawbacks of 3PL are gradually arising. For example, lost packages and theft are common when using 3PL. Frequent overstocking during holidays and promotion days are also often disclosed due to the

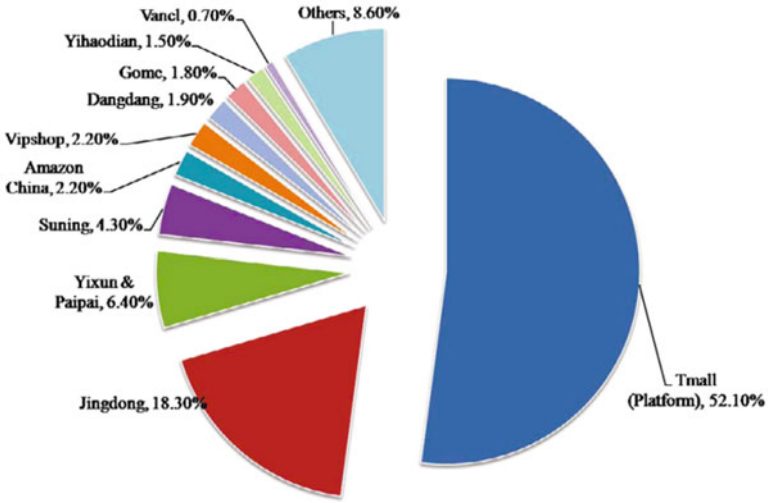

Fig. 9.4 Market share of major Chinese B2C e-commerce players in 2013

insufficient shipping capacity of 3PL. 3PL services are offered to both suppliers and customers while self-operated logistics are often built by B2C websites to improve service quality and "last mile delivery" efficiency through control of every section of the supply chain, from warehouse to consumer. As a result, a hybrid form of logistics combining 3PL and self-managed logistics is currently a popular topic of study.

From an e-retail supply chain perspective, whatever business you are in, suppliers and vendors play a crucial role in your company's success. The merchandise quality and richness provided by suppliers determine the popularity of goods, which in turn affect inventory turnover and sales. Based on that, e-retail supply chain process can be generally divided into two stages—procurement-stock management and inventory-sale control. The first sub-stage, procurement-stock management, represents "the first mile delivery" efficiency of e-retail. The second sub-process, stock-sale control describes supplier performance due to the conversion of inventory into sales revenue, as shown in Fig. 9.5. It should be noted that the overall supply chain efficiency is measured without considering internal link activities or intermediate variables.

9.3.1.1 Case, Research Problem and Data

W firm, one of China's leading B2C e-commerce firms, is chosen as our research case. The reasons are given as follows:

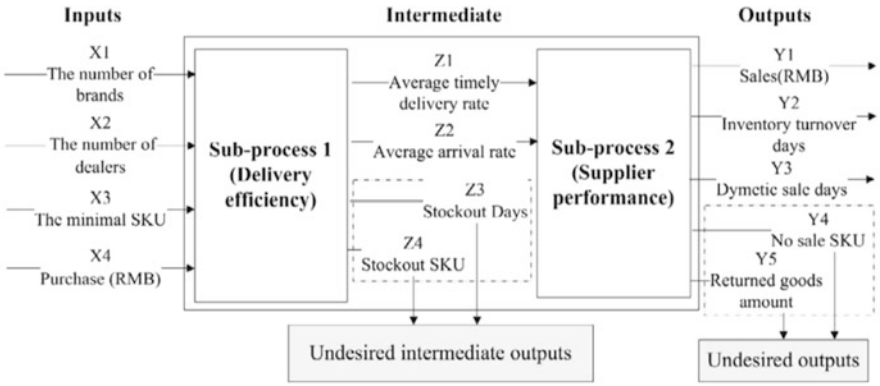

Fig. 9.5 E-commerce procurement-inventory-sale supply chain structure

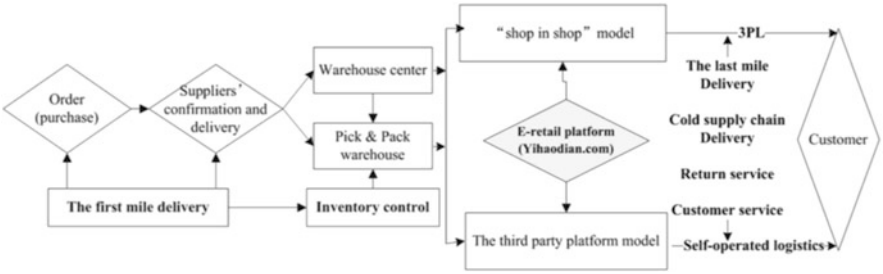

Fig. 9.6 E-retail supply chain for W firm

Firstly, W firm has established a nationwide supply chain network and has an industry-leading supply chain management system in the Chinese B2C e-commerce sector.

Secondly, W firm has the ability to realize a full online operation based on its open supply chain platform which aims to serve traditional enterprises who would like to tap into the e-commerce sector but lack online operating ability. It is similar to the third party platform model in regards to covering an integrated online operations service, improving suppliers' supply chain efficiency and reducing operations costs by system integration, cloud-based marketing, promotion tools, logistics, warehousing and information services.

Thirdly, from "the last mile delivery", those suppliers who choose the "shop in shop" model sell their merchandise by third party logistics (3PL), while running business operations on independently. For contrast, those suppliers choosing the third party platform model only need to provide their merchandise to the platform of W firm, while online operations-related activities are executed by W firm. E-retail supply chain for W firm is described in Fig. 9.6.

In conclusion, the operations model of the suppliers in W firm can be clearly divided into the third party platform model and self-operated model, which are two predominant e-business models in china. The third party platform model and self-operated model offer different "last mile delivery" choices for e-commerce players. Thus, this case can be used to analyze the following questions:

1. What causes overall e-retail supply chain inefficiency? "The first mile delivery" or "the last mile delivery"?
2. How do self-operated mode and the third party platform mode affect supply chain efficiency respectively?
3. What is the way forward for product category and Geographic expansion for major Chinese B2C e-commerce players?
4. Which is better for e-retail supply chains: Self-logistics, 3PL or the hybrid model?

Accordingly, the data of more than 2400 suppliers covering purchasing cost, the lead time, inventory, sale, delivery and returned goods were collected from W firm. Excluding incomplete data, 1229 suppliers of the "shop in shop" model and 899 suppliers of the third party platform model were obtained. Nine major product categories are included in this data set, and the research methods are described in detail.

9.3.1.2 Research Methodology

Network Slacks-Based Measure of Efficiency (NSBM)

Suppose there are n DMUs (j = 1, 2, ..., n) consisting of k divisions (k = 1, 2, ..., k) in a supply chain. mk and rk represent the number of inputs and outputs of Division k, respectively. The set of links leading from Division h to division k is defined as $L(k.\ h)$. Accordingly, the production possibility set (x^k, y^k, z^k, h) under the assumption of variable returns-to-scale (VRS) production is defined by

$$
\begin{aligned}
&x^k \geq \sum_{j=1}^{n} x_j^k \lambda_j^k, k = 1, 2, \cdots, k \\
&y^k \leq \sum_{j=1}^{n} x_j^k \lambda_j^k, k = 1, 2, \cdots, k \\
&z^{k,h} = \sum_{j=1}^{n} z_j^{k,h} \lambda_j^k, \forall k, h \text{ (as outputs from } k \text{ and inputs to } h), \\
&\sum_{j=1}^{n} \lambda_j^k = 1, \forall k, \lambda_j^k \geq 0, \forall j, k
\end{aligned}
$$

where, $\lambda^k \in R_+^n$ is the intensity vector corresponding to Division k (k = 1, 2, ..., n).

For the evaluated DMU0 (0 = 1, 2, ..., n), in the case of linking activities determined freely while keeping continuity between input and output, non-oriented

overall efficiency can be represented as:

$$\rho^* = \min_{\lambda^k, s^{k-}, s^{k+}} \frac{\sum_{k=1}^{k} w^k \left[1 - \frac{1}{m^k}\left(\sum_{i=1}^{m_k} \frac{s_i^{k-}}{x_{is}^k}\right)\right]}{\sum_{k=1}^{k} w^k \left[1 - \frac{1}{r^k}\left(\sum_{r=1}^{r_k} \frac{s_r^{k+}}{y_{ro}^k}\right)\right]} \tag{9.47}$$

$$\text{s.t.} \begin{cases} x_o^k = X^k \lambda^k + s^{k-} \\ y_o^k = Y^k \lambda^k - s^{k+} \\ \lambda^k = 1 \\ X^k = \left(x_1^k, x_2^k, \cdots, x\right) \in R^{m_k} \times n \\ Y^k = \left(y_1^k, y_2^k, \cdots, y_n^k\right) \in R^{r_k} \times n \\ z^{k,h} \lambda^h = z^{k,h} \lambda^k, (\forall k, h) \\ z^{k,h} = \left(z_1^{k,h}, z_2^{k,h}, \cdots, z_n^{k,h}\right) \in R^{t_{k,h}} \times n \\ \lambda^k \geq 0, s^{k-} \geq 0, s^{k+} \geq 0, \forall k \end{cases} \tag{9.48}$$

where $\sum_{k=1}^{k} w^k$, $w^k \geq 0$ $(\forall k)$, and w^k is the relative weight of division k defined by the decision makers. Non-oriented division efficiency score can be calculated by the below:

$$\rho_k = \frac{1 - \frac{1}{m_k}\left(\sum_{i=1}^{m_k} \frac{s_i^{k-*}}{x_{io}^k}\right)}{1 - \frac{1}{r_k}\left(\sum_{r=1}^{r_k} \frac{s_r^{k+*}}{y_{ro}^k}\right)}, k = 1, 2, \cdots, k \tag{9.49}$$

s^{k-*} and s^{k+*} are the excessive inputs and short outputs for the above Eq. (9.47).

Undesirable Output Directional Distance Function Model

It is important for a retail supply chain to effectively manage inventory and avoid returned purchases. It is therefore reasonable to extend the network slack-based measure (NSBM) to incorporate undesirable outputs so that it can give a comprehensive and accurate evaluation on delivery efficiency and supplier performance in a given e-retail supply chain.

The usual technical efficiency measurement is based on input and output distance functions, which cannot simultaneously contract undesirable/bad outputs and inputs and expand good/desirable outputs. Directional distance function is a generalized form of the radial model, and it allows us to explicitly increase the desirable outputs and simultaneously decrease undesirable outputs and inputs. To see this let good outputs be denoted by $y \in R_+^M$, bad or undesirable outputs by $b \in R_+^J$, and inputs by $x \in R_+^N$,. Suppose there are k (k = 1, 2, …, K) DMUs in an e-retail supply chain. Each DMU uses input $x^k = \left(x_1^k, x_2^k, \cdots, x_N^k\right) \in R_+^N$ to jointly produce

desirable/good outputs $y\hat{}k = \left(y_1^k, y_2^k, \cdots, y_M^k\right) \in R_+^M$ and undesirable/bad outputs $b\hat{}k = \left(b_1^k, b_2^k, \cdots, b_J^k\right) \in R_J^+$. For a specific DMU0, a more generalized form of directional distance function is denoted by Chambers et al. [85] as follows:

$$\theta = \min \frac{1-\frac{1}{m}\sum_{i=1}^{m} w_i \alpha g_{xi0} x_{i0}}{1+o_d \frac{1}{s_d}\sum_{d=1}^{s_d} w_d \beta g_{yd0} y_{d0} - o_u \frac{1}{s_u}\sum_{u=1}^{s_u} w_u \gamma g_{yu0} y_{u0}} \tag{9.50}$$

$$\text{s.t.} \begin{cases} X\lambda + \alpha g^x \leq x_0 \\ Y^d\lambda - \beta g_y^d \geq y_0^d \\ Y^u\lambda + \gamma g_y^u \leq y_0^u \end{cases} \tag{9.51}$$

with $\sum_{i=1}^{m} w_i = m$, $\sum_{d=1}^{s_d} w_d = s_d$, $\sum_{u=1}^{s_u} w_u = s_u,$, $o_u + o_d = 1$, where m, sd, and su denote the number of inputs, desirable (good) outputs and undesirable (bad) outputs respectively. x0 and y0 are the inputs and outputs of the evaluated DMU0. w_i, w_d, and w_u separately express the weights of inputs, desirable (good) outputs and undesirable (bad) outputs defined by decision makers. g_x and g_y represent the direction vector of inputs and outputs defined by decision makers. ou and od refer to the overall weight of undesirable (bad) and desirable (good) outputs defined by decision makers.

Noted that α, β, γ represent the expansion rate for desirable output items, contraction rate for undesirable output items and input items respectively, and α, β, γ are not necessarily the same value. Namely, it allows for different proportional contraction and expansion rate for inputs, undesirable outputs and desirable outputs.

Performance assessed by directional distance model can be flexibly applied to different analysis purposes. For example, if the direction is chosen by setting$g = (-gx, gy, -gb) = (-xk, yk, -bk)$, the efficiency score represents how much the percentage needed to be improved in good outputs, bad outputs and inputs [78]. If instead the direction is set by $g = (-gx, gy, -gb) = (-1, 1, -1)$, the solution value can be interpreted as the net improvement in performance in the case of feasible expansion in good outputs and feasible contraction in bad outputs and inputs [107].

Here we choose the measurement based on the observed data, namely$g = (-gx, gy, -gb) = (-xk, yk, -bk)$, because we would like to observe the potential proportionate change in good outputs, bad outputs and inputs.

9.3.1.3 Variables Description

Input-Output Variables Description in the First Sub-process

As a non-parametric method for converting multi-inputs into multi-outputs, how to choose suitable input-output variable combination is crucial for DEA efficiency evaluation. Thus, in order to give an accurate efficiency measurement, it is necessary to give a reasonable input-output variable description based on e-retail supply chain network structure. Unlike in traditional retail, data mining techniques make demand

forecasts possible. An e-commerce supply chain therefore starts with procurement management based on demand forecast. Purchasing plays an important role in cost saving and making profit. The way of orders is scheduled and the resultant lead time directly determines the performance of downstream activities and inventory levels. As a result, order-related input and output variables such as the selection of the right supplier, product variety, purchasing cost, average arrival rate, on time delivery rate are considered in the first sub-process of e-retail supply chain.

The number of brands and stock keeping unit (SKU) describe a variety and richness of the products in e-retail [108, 109]. Higher variety will lead to an increase in consumer's utility, which in turn affects inventory turnover and finally results in an increase in gross margin [110]. Additionally, the number of dealers determines the size of the suppliers and purchasing cost denotes the total financial inputs. Therefore, the number of brands, the number of dealers, the minimal stock keeping unit (SKU) and purchasing cost can be considered as the initial inputs of procurement-delivery management.

Furthermore, gross margin is associated with stockout costs. In practice, stock-outs will lead consumer to switch retailers on subsequent shopping trips due to poor shopping experience [111]. As a result, higher stockouts mean higher lost profits. Hence, an important task of procurement managers is to reduce stockout SKUs and shorten stockout days. Accordingly, the variables of stockout SKUs and stockout days are considered as undesirable outputs in the first sub-process of e-retail supply chain performance measurement.

It is crucial that purchasing management is not something stand-alone, but has close links with the measurement of overall supply chain performance. Thus, average arrival rate and on-time delivery rate are used to measure procurement-delivery efficiency. They are the outputs in the first sub-process and the inputs in the second sub-process of e-retail supply chain. The detail input-output variables are described in Table 9.8.

Input-Output Variables Description in the Second Sub-process

Efficient procurement-stock performance can accelerate inventory turnover and promote sales. It is easier for e-commerce players to turn their capital into inventory, but it is difficult for them to turn their inventory into money. According to a statistics of Slywotzky [112], there are 95% of the time used for storage, loading and transportation in a commodity production and sales process. Hence, inventory turnover plays a crucial role in supply chain efficiency measurement. Generally speaking, shorter turnover times mean greater capacity to turn stock into revenue. Accelerating inventory turnover means an increase in the liquidity of capital. Based on that, average days to turnover inventory is considered as one of outputs in the second sub-process of e-retail supply chain. It should be noted that average days to turn over inventory refers to the number of days it takes to sell all on-hand inventory,

Table 9.8 Input and output variables description in the procurement sub-process

Procurement		Variables	Description	Metrics
Inputs	Supplier selection	The number of brands (X1)	The suppliers offer the variety of brand	Material quality
		the number of dealers (X2)	The same brand owns the different dealers	Supplier scale
		The minimal SKU (X3)	The minimal stock keeping units offered by suppliers	Merchandise richness
	Purchasing volume	Purchasing cost (X4)	The total order volume	Total purchasing cost
Desirable outputs		Average on-time delivery rate (Z1)	The ratio of on-time delivery to delayed delivery	Just-in-time delivery performance
Undesirable outputs		Average arrival rate (Z2)	The ratio of the actual delivered products by supplier to those ordered by purchasing managers	Just-in-time delivery performance
		Stockout days (Z3)	The short days of the SKUs when SKUs is below the safety stock	Inventory level
		Stockout SKU (Z4)	SKUs without being offered or restocked by suppliers	Inventory level

and can be calculated by the following formula:

$$\text{days to turnover inventory} = 365/\text{inventory turnover}$$

A change in inventory is a response to the change in sales, while dynamic sale is a key for inventory turnover. In practice, dynamic sale days is often used to illustrate inventory change and judge whether the merchandise is popular or not. In general, shorter dynamic sale days mean faster inventory turnover and less unmarketable goods. The unmarketable goods will lead to the loss of sales revenue due to an increase in stock costs. In e-retail, another loss of sales revenue can be attributed to consumer returned goods. Therefore, when associated with average days to turn inventory and sales revenue, dynamic sale days are considered as the output variables, while no-sale SKU and users' returned goods amount are chosen as undesirable output variables of supplier performance measurement in the second sub-process of e-retail supply chain. The detail input and output variables' illustration is shown in Table 9.9.

9.3.1.4 Empirical Results

E-Retail Efficiency of "the First Mile Delivery" and "the Last Mile Delivery"

Procurement-stock sub-process of e-retail supply chain is called as "the first mile delivery" due to its nature of affecting inventory management. It is the first section of e-retail supply chain, and its performance directly affects subsequent inventory and sales. Therefore, we give more weight to the first stage of e-retail supply chain than to the second stage. According to network slacks-based measure (NSBM) model, for a specific division k, the weight w1k of procurement-stock sub-process is given 0.6 and w2k of inventory-sale sub-process is given 0.4. Associated with the directional distance model with undesirable output, the weights wd of desirable (good) outputs is denoted as 0.6 and the weights wd of undesirable (bad) outputs is denoted as 0.4. We simultaneously run the above two models using the software of MaxDEA 6.2, and the results are given in Fig. 9.7.

As shown in Fig. 9.7, efficiency scores of the procurement-stock stage (Node 1) are lower than those of inventory-sale stage (Node 2). We can hence conclude that it is procurement-stock conversion inefficiency that results in W firm's overall supply chain inefficiency. The process from purchasing to putting in stock is named "first mile delivery", which is essential to developing a healthy buyer-supplier relationship and improving inventory control level.

Specifically, the suppliers of the "shop in shop" model have higher overall supply chain efficiency in kitchen and cleaning products than others due to higher purchasing-stock efficiency in the first sub-process of supply chain. In contrast, the suppliers of the third party platform model achieve better stock-sale performance in kitchen and cleaning products than others but it has low overall supply chain efficiency due to the poor performance in purchasing-stock efficiency, referring to

Table 9.9 Input and output variables description in the inventory-sale sub-process

Inventory-sale process	Variables	Description	Metrics
Inputs	Average on-time delivery rate (Z1)	The ratio of on-time delivery to delayed delivery	Just-in-time delivery performance
	Average arrival rate (Z2)	The ratio of the actual delivered products by supplier to those ordered by purchasing managers	
Desirable Outputs	Average days to turn inventory (Y1)	Inventory turnover in days equals 365 days divided by inventory turnover	Measure the value of capital movement
	Dynamic sale days (Y2)	The number of days it takes for an SKU sold	Evaluate the change of inventory and the popularity of merchandise
	Sales revenue (Y3)	Revenue from goods sold	Financial performance
Undesirable outputs	Unsaleable SKUs (Y4)	The merchandise without sold	Measure the losses of unmarketable goods
	Users' returned goods amount (Y5)	The amount of returned goods by consumers	Measure the losses of poor users' satisfaction

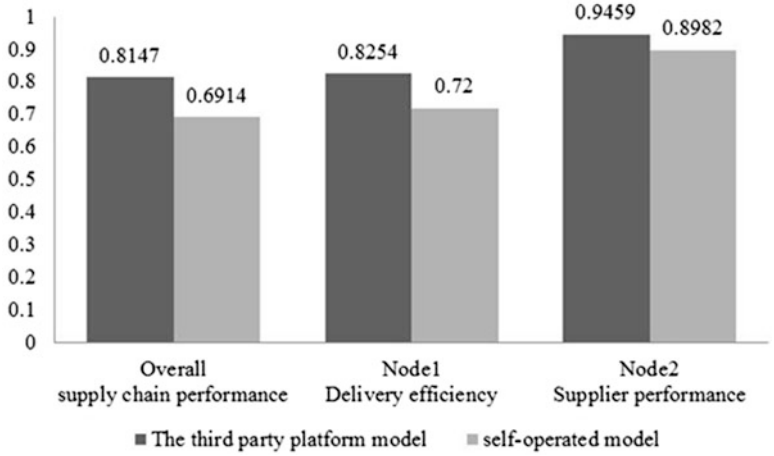

Fig. 9.7 E-retail procurement efficiency and supplier performance

Table 9.10. For this discussion, we can conclude that purchasing-stock efficiency plays a more key role in affecting overall supply chain efficiency. This conclusion further verifies the finding in Fig. 9.7.

Product Categories Expansion and Efficiency Analysis

As China's leading B2C e-commerce online supermarket, W firm has more advantages in fast moving consumer goods (FMGG) like food and drink, as shown in Fig. 9.8. In line with strategic positioning of W firm, this finding displays its core business focus on online supermarket and the concept of "the home". It is this strategic positioning that creates a barrier to potential competitors entering, thus affording a competitive advantage compared with other B2C websites such as dangdang, Suning and Redbaby. As a result, this unique positioning has allowed W firm to quickly build a loyal customer base and win a first-mover advantage.

However, with growing orders, one-stop shopping of "the home" becomes more and more important for attracting customers. Thus, W firm gradually expands its product categories from FMCG products to electronics, apparel, auto parts, maternity, and household products. In general, all major Chinese B2C e-commerce websites experience similar product categories expansion, namely starting with a narrow, vertical product line then expanding to a broad range of categories. For example, Dangdang started with books and Jingdong with digital products. Then, with growing user and market demands, all of them are in pursuit of all-categories expansion. In other words, Chinese B2C e-commerce websites experience a development of transferring from a vertical model to an integrated model.

Table 9.10 Overall and sub-process efficiency comparison for two different supply chain model

Categories	Overall supply chain efficiency	Stage 1 Purchasing-Stock efficiency	Stage 2 Stock-Sale performance
Self-operated model with a third party logistics (3PL) (shop in shop)			
Auto parts	0.4779	0.5392	0.8299
Beauty and personal care	0.6273	0.6576	0.8741
Computer and digital	0.7061	0.7263	0.9207
Food and drink	0.7190	0.7437	0.9097
Health products	0.5701	0.5840	0.9068
Household	0.6281	0.6656	0.8737
Home appliances	0.7015	0.7537	0.8763
Kitchen and cleaning	0.7548	0.8045	0.8697
Toys, mom and baby	0.6241	0.6408	0.8917
All	0.6914	0.7200	0.8982
Third-party platform model with a self-logistics			
Auto parts	0.8021	0.8080	0.9520
Beauty and personal care	0.7770	0.7893	0.9240
Computer and digital	0.8145	0.8374	0.9249
Food and drink	0.8395	0.8496	0.9531
Health products	0.7807	0.7857	0.9427
Household	0.7854	0.7987	0.9396
Home appliances	0.8284	0.8329	0.9556
Kitchen and cleaning	0.7973	0.8009	0.9573
Toys, mom and baby	0.8329	0.8427	0.9520
All	0.8147	0.8254	0.9459

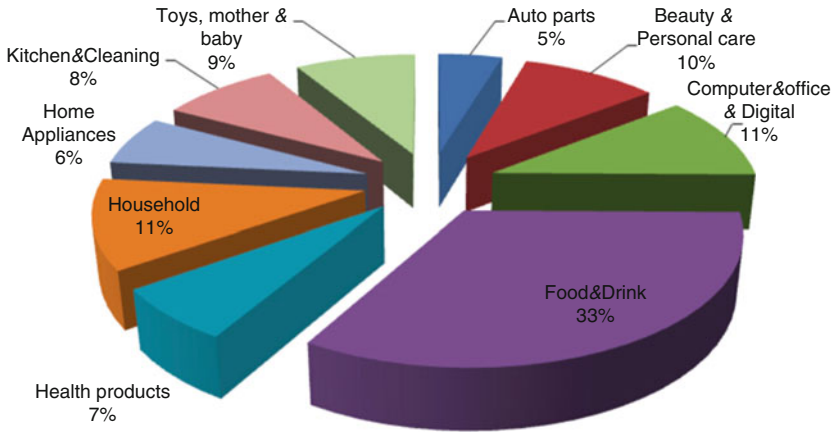

Fig. 9.8 The distribution of overall efficient supplier in different product categories

9.3.1.5 Operations Model Comparison

By the way of third party platform model, the "last mile delivery" fleet serves shops settled on the W platform while simultaneously serving merchants who sell their products on their own web page or other market platforms. The full operations service effectively reduces "the last mile delivery" cost and has allowed W firm to create higher supplier performance in the second sub-process of supply chain, referring to Fig. 9.9. However, which model is more efficient in the first stage known as "first mile delivery", self-operated model or platform model?

From inventory management, too much stock will increase inventory cost while too little stock will affect stockout rate. Thus, it is necessary for an integrated platform to make automated procurement decisions. Figure 9.9 describes inventory management for W firm. It can be seen that a purchase order would be automatically issued and sent to the suppliers when inventory dropped below a defined safety stock, and then the order will be filled by the suppliers [113]. In this way, W firm can record the delivery time, receiving and shelving information and process payment. Therefore, it can be seen in Fig. 9.9 that platform model presents higher procurement-stock efficiency scores than the self-operated (shop in shop) model.

Is the platform model efficient for all product categories?

In response to this question, we compare the "last mile delivery" efficiency of different product categories for the platform model and self-operated model, referring to Fig. 9.10. The results show self-operated (shop in shop) model performs better in computer and Office and digital, food and drink and healthy products. This is because of the high values of computer and Office and digital, and the shorter shelf life of food and drink and healthy products, which determine their priority in order of handling, picking, stockout-compensation and delivery. Furthermore, from the consumer's demand, products such as food and drink and healthy products are often bought based on the temporary needs of customers. Thus it is more suitable for these products to be delivered from regional distribution centers, while self-operated model is more helpful to reduce these product's delivery cost. This is also the reason

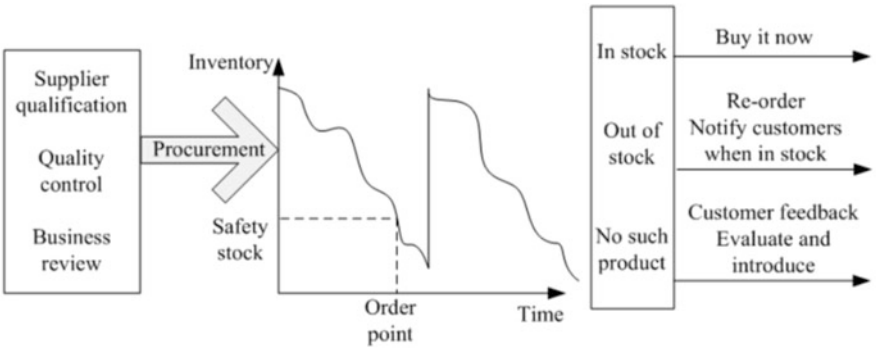

Fig. 9.9 Inventory management for W firm

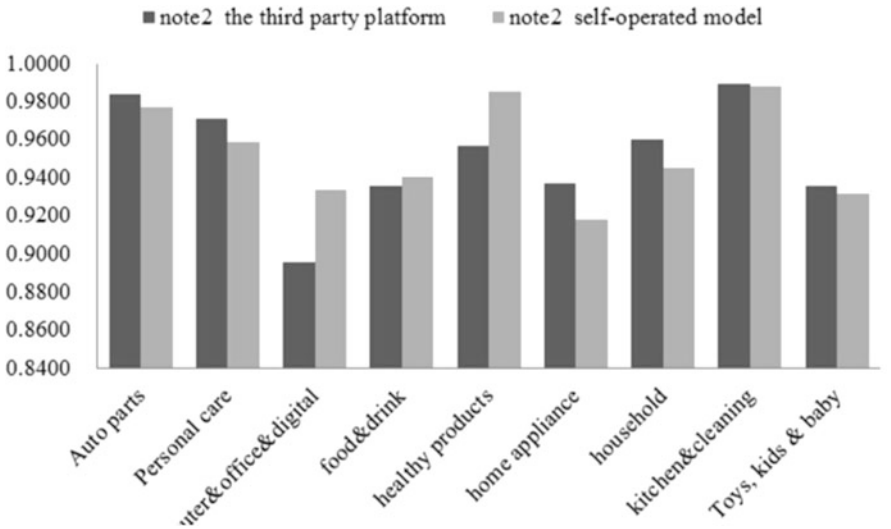

Fig. 9.10 Supplier performance comparison in different product categories and operations model

why Jingdong, the top Chinese self-operated B2C e-commerce website, starts with 3C (Computer, Communication, and Consumer electronic) products.

For the above discussion, we can conclude that the third party platform model generally performs better than self-operated model, due to its higher efficiency in "first mile delivery" and "last mile delivery". However, from a product categories perspective, self-operated model has greater efficiency in computer and Office and digital, food and drink and healthy products than the third party platform model due to these products' characteristics of regional demand and delivery.

Geographic Expansion and Efficiency Evaluation on 3PL and Self-Operated Model

As e-commerce continues its rapid growth into virtually every market sector, retailers are eager to expand their presence online to capture this market share. According to a research report of i-Research, a leading organization focusing on in-depth research in China's internet industry, China's business-to-consumer (B2C) market is to CNY 666.1 billion in 2013, accounting for 36.2% of online shopping market, and has become a formidable force. However, because B2C is an e-commerce model directly facing the customers, the "last mile delivery" is a crucial challenge for improving users' online shopping experience. Therefore, it is very important for e-commerce players to improve the "first mile delivery" (from order to warehouse) and the "last mile delivery" (from warehouse to consumer).

Starting with a large selection spanning many different product categories is a great challenge for the supply chain capacity of W firm. Although the FMCG category contributes to increasing traffic and consumer stickiness due to its nature of meeting daily needs, how to pick, pack and delivery these small items is a constant struggle. For example, by 2013, W firm had about 2,000,000 SKUs, which is 100 times that of a traditional supermarket, and each order of W firm has an average of 10 merchandises while each order of Jingdong has less than 2 merchandises. So it is stringent on warehouse design and the method of choosing food and drink supply chain. Most importantly, food and drink require faster inventory turnover due to their shorter shelf life. As a result, procurement-inventory-sale-delivery decisions needs to be automated as much as possible.

Like most B2C e-commerce players, W firm initially took 3PL delivery service model for the purpose of saving cost. But initial on-time delivery was only 90% and customer returns reached over 3% [113]. Coupled with growing orders, 3PL struggles to keep up with this growth. Therefore, the self-built logistics system becomes essential. In light of Amazon China's centralized distribution model, W firm controls all decisions from its headquarters and builds multiple distribution centers. A new "line-haul + regional distribution center + last mile delivery" model is taken. It is noted that the centralized distribution model serves nationwide consumers with the same selection on one website utilizing transshipment between warehouses to ensure the availability of products from all warehouses. In contrast, the decentralized distribution model offers different selections from local branch websites and delivers products from local distribution centers to consumers.

In the term of warehousing expansion, W firm has built five large warehousing centers covering Beijing, Shanghai, Guangzhou, Wuhan and Chengdu. By the way self-established logistics system and the third party platform operations model, W firm has borne fruit with a drastically enhanced customer experience and a 10% improvement in consumer satisfaction. The results in Table 9.10 verify that the third party platform model with self-operated logistics has better delivery efficiency, supplier performance and supply chain efficiency than self-operated (shop in shop) model.

In summary, both the self-operated model and the third platform model are more efficient in supplier performance than that in purchasing-stock efficiency, as shown in Fig. 9.10 and Table 9.10. Thus, it is urgent for W firm to strengthen their "first mile delivery" efficiency because the "first mile delivery" plays a more crucial role in supplier selection and inventory control. From an e-commerce logistics view, self-operated logistics can improve service quality and efficiency through controlling each section from warehouse to consumers, including "the last mile delivery" and is hence more efficient in the coordination of supply chain. But the complicated supply chain network and growing product categories make most e-retail players tend towards a hybrid form of 3PL and self-logistics.

9.3.2 *Credit Risk Evaluation with Kernel-Based Affine Subspace Nearest Points Learning Method*

This subsection presents a novel kernel-based method named kernel affine subspace nearest point (KASNP) method for credit evaluation. KASNP method is an extension of a new method named affine subspace nearest point method (ASNP) [114, 115] by kernel trick. Compared with SVM, KASNP is an unconstrained optimal problem, which avoids the convex quadratic programming process and directly computes the optimum solution by training set. On three credit datasets, our experimental results show that KASNP is more effective and competitive.

9.3.2.1 Affine Subspace Nearest Point Algorithm

The idea of affine subspace nearest point algorithm is derived from the geometric SVM and its nearest-points problem. Here we first give a brief overview of the geometric interpretation and the nearest point problem of SVM in original space.

Nearest Point Problem of SVM

Given a set S, co(S) denotes the convex hull of S, and is the set of convex combinations of all elements of S:

$$\mathrm{co}\,(\boldsymbol{S}) = \left\{ \sum_k \alpha_k \boldsymbol{x}_k | \boldsymbol{x}_k \in \boldsymbol{s}, \alpha_k \geq 0, \sum_k \alpha_k = 1 \right\} \tag{9.52}$$

For the linearly separable binary case, given training data, $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, $\ldots$, $(\boldsymbol{x}_l, y_l)$, $\boldsymbol{x}_i \in \boldsymbol{R}^d$, $y_i \in \{+1, -1\}$, $i = 1, \ldots, l$, yi is the class label, i.e. $\boldsymbol{S}_1 = \{(\boldsymbol{x}_i, y_i) | \, y_i = +1\}$ and $\boldsymbol{S}_2 = \{(\boldsymbol{x}_i, y_i) | \, y_i = -1\}$, then the convex hulls of the two sets are

$$\mathrm{co}\,(\boldsymbol{S}_1) = \left\{ \sum_{i:y_i=+1} \alpha_i \boldsymbol{x}_i | \sum_{i:y_i=+1} \alpha_i = 1, \alpha_i \geq 0 \right\} \tag{9.53}$$

$$\mathrm{co}\,(\boldsymbol{S}_2) = \left\{ \sum_{i:y_i=-1} \alpha_i \boldsymbol{x}_i | \sum_{i:y_i=-1} \alpha_i = 1, \alpha_i \geq 0 \right\} \tag{9.54}$$

As we know, the aim of normal SVM is to find the hyperplane, which separates training data without errors and maximizes the distance (called margin) from the closest vectors to it. In fact, from geometric view, the optimal separating hyperplane is just the one that is orthogonal to and bisects the shortest line segment joining the convex hulls of two sets, and the optimal problem of SVM is equivalent to finding the nearest point problem in the convex hulls [116]. The geometric interpretation

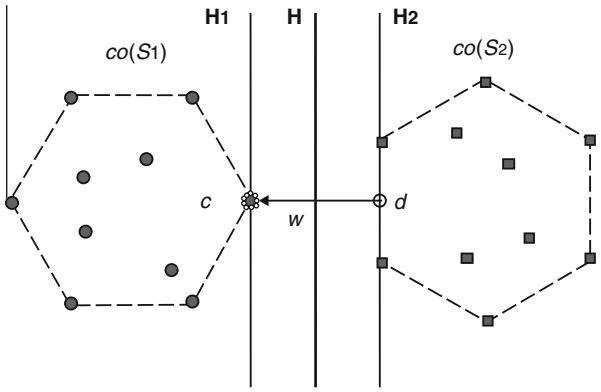

Fig. 9.11 The geometric interpretation and nearest point problem of SVM. co(S1) and co(S2) are two smallest convex sets (convex hulls) shown with dashed lines which contain each class. c and d are the nearest points on them

and nearest point problem (NNP) of SVM can be easily understood by Fig. 9.11.

$$\begin{aligned} &\min_{\alpha} \left\| \sum_{i:y_i=+1} \alpha_i \boldsymbol{x}_i - \sum_{i:y_i=-1} \alpha_i \boldsymbol{x}_i \right\|^2 \\ &\text{s.t.} \sum_{i:y_i=+1} \alpha_i = 1, \sum_{i:y_i=-1} \alpha_i = 1 \\ &\alpha_i \geq 0, i = 1, \ldots, l \end{aligned} \tag{9.55}$$

If $\boldsymbol{\alpha}^* = \left(\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*\right)$ is the solution to the convex quadratic optimization Eq. (9.55), then the nearest points in two convex hulls are $\boldsymbol{c} = \sum_{i:y_i=+1} \alpha_i^* \boldsymbol{x}_i$ and $\boldsymbol{d} = \sum_{i:y_i=-1} \alpha_i^* \boldsymbol{x}_i$. Constructing the decision boundary $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$ to be the perpendicular bisector of the line segment joining the two nearest points means that w lies along the line segment and the midpoint p of the line segment satisfies the function $f(x) = 0$. w and p can be computed by c and d: $w = c\ d$, $p = (1/2)(c + d)$, then b = w p. In the end, the classification discriminant function can be written as: f(x) = sgn(w x + b), where sgn( ) is the sign function.

Similar to the above process of the geometric method of SVM, ASNP method [114] extends the areas searched for the nearest points from the convex hulls in SVM to affine subspaces, and constructs the decision hyperplane separating the affine subspaces with equivalent margin.

9.3.2.2 Affine Subspace Nearest Points (ASNP) Algorithm

Definition 9.1 (Affine subspace). Lee and Seung [117] Given a sample set $S = \{x_1, \ldots, x_m\}$, $x_i \in R^d$, the affine subspace spanned by S can be written as

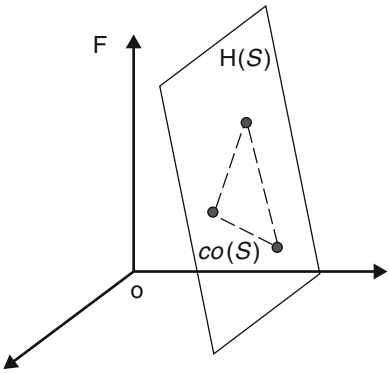

Fig. 9.12 The affine subspace H(S) created by the three samples set S. F is the space three samples lie in. The inner area of the triangle shown with dashed lines is the convex hull co(S), whereas the minimum hyperplane that contains the triangle is the affine subspace H(S)

Eq. (9.56) or Eq. (9.57):

$$H(\boldsymbol{S}) = \left\{ \sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i | \sum_{i=1}^{m} \alpha_i = 1 \right\} \tag{9.56}$$

$$H(\boldsymbol{S}) = \left\{ \boldsymbol{x}_0 + \sum_{i=1}^{m} \alpha_i (\boldsymbol{x}_i - \boldsymbol{x}_0) \right\}, \boldsymbol{x}_0 \in H(\boldsymbol{S}) \tag{9.57}$$

For Eq. (9.56), we can get rid of the constraint $\sum_{i=1}^{m} \alpha_i = 1$ by taking a point in H(S) as a new origin x_0. Therefore the equivalent of Eq. (9.56) can be written as Eq. (9.57). We can let x_0 be the average of all samples, $x_0 = \frac{1}{m} \sum_{i=1}^{m} x_i$.

In order to interpret the affine subspace, we simply depict the affine subspace in geometry, see, for example in Fig. 9.12.

Compared with the convex hull co(S), the affine subspace contains the convex hull, but is not constrained by $\alpha_i \geq 0$ (see Eq. 9.56). The convex hull only contains the interpolations of the basis vectors, whereas the affine subspace contains not only the convex hull but also the linear extrapolations.

For a binary-class problem with training sets $S_1 = \{x_1, x_2, \ldots, x_m\}$ and $S_2 = \{x_{m+1}, x_{m+2}, \ldots, x_n\}$. Two affine subspaces respectively spanned by them are

$$H(\boldsymbol{S}_1) = \left\{ \sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i | \sum_{i=1}^{m} \alpha_i = 1 \right\} \tag{9.58}$$

$$H(\boldsymbol{S}_2) = \left\{ \sum_{i=m+1}^{n} \alpha_i \boldsymbol{x}_i | \sum_{i=m+1}^{n} \alpha_i = 1 \right\} \tag{9.59}$$

Then the problem of finding the closest points in affine subspaces can be written as the following optimization problem:

$$\begin{array}{l} \min_{\boldsymbol{\alpha}} \left\| \sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i - \sum_{i=m+1}^{n} \alpha_i \boldsymbol{x}_i \right\|^2 \\ \text{s.t.} \sum_{i=1}^{m} \alpha_i = 1, \sum_{i=m+1}^{n} \alpha_i = 1, i = 1, \ldots, l \end{array} \tag{9.60}$$

Compared with Eq. (9.56), Eq. (9.60) is not under constraint $\alpha_i \geq 0$ which can be also converted into an unconstrained optimal problem as follows:

As Eq. (9.57) is represented, Eqs. (9.58) and (9.59) can be written in unconstrained Eqs. (9.61) and (9.62).

$$H(\boldsymbol{S}_1) = \left\{\overline{\boldsymbol{u}}_1 + \sum_{i=1}^{m} \alpha_i (\boldsymbol{x}_i - \overline{\boldsymbol{u}}_1)\right\} \tag{9.61}$$

$$H(\boldsymbol{S}_2) = \left\{\overline{\boldsymbol{u}}_2 + \sum_{i=m+1}^{n} \alpha_i (\boldsymbol{x}_i - \overline{\boldsymbol{u}}_2)\right\} \tag{9.62}$$

where $\overline{\boldsymbol{u}}_1 = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$ and $\overline{\boldsymbol{u}}_2 = \left(\frac{1}{n-m}\right)\sum_{i=m+1}^{n} \boldsymbol{x}_i$.

So Eq. (9.60) can be rewritten as

$$\min_{\alpha} \left\| \left(\boldsymbol{u}_1 + \sum_{i=1}^{m} \alpha_i (\boldsymbol{x}_i - \overline{\boldsymbol{u}}_1)\right) - \left(\boldsymbol{u}_2 + \sum_{i=m+1}^{n} \alpha_i (\boldsymbol{x}_i - \overline{\boldsymbol{u}}_2)\right) \right\|^2 \tag{9.63}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}^T$.

Equation (9.63) is an unconstrained optimal problem, which can be computed directly, and $\boldsymbol{\alpha}$ is

$$\alpha = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{+} \boldsymbol{A}^{\mathrm{T}} (\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2) \tag{9.64}$$

Or

$$\alpha = \left(\boldsymbol{A}^{T}\boldsymbol{A} + \sigma \boldsymbol{I}\right)^{-1} \boldsymbol{A}^{T} (\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2) \tag{9.65}$$

where $\boldsymbol{A} = ((\overline{\boldsymbol{u}}_1 - \boldsymbol{x}_1), \ldots, (\overline{\boldsymbol{u}}_1 - \boldsymbol{x}_m), (\boldsymbol{x}_{m+1} - \overline{\boldsymbol{u}}_2), \ldots, (\boldsymbol{x}_n - \overline{\boldsymbol{u}}_2))$, and $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{+}$ is the pseudo-inverse of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$; $\sigma \geq 0$, and I is n*n identity Matrix.

Then the two nearest points in affine subspaces are

$$c = \overline{\boldsymbol{u}}_1 + \sum_{i=1}^{m} \alpha_i (\boldsymbol{x}_i - \overline{\boldsymbol{u}}_1) \tag{9.66}$$

$$\boldsymbol{d} = \overline{\boldsymbol{u}}_2 + \sum_{i=m+1}^{n} \alpha_i (\boldsymbol{x}_i - \overline{\boldsymbol{u}}_2) \tag{9.67}$$

The midpoint of the line segment joining c and d is p = (1/2) (c + d). Similar to the nearest point problem of SVM, the decision boundary w x + b = 0 is the perpendicular bisector of the line segment. Thus, w = c − d and b = −w*p. Correspondingly, the decision function is.

$$\begin{aligned} f(\boldsymbol{x}) &= \operatorname{sgn}(\boldsymbol{w} \cdot \boldsymbol{x} + b) \\ &= \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i (\boldsymbol{x}_i \cdot \boldsymbol{x}) - (12)\sum_{i=1}^{n}\sum_{j=1}^{n} y_i \alpha_i \alpha_j \left(\boldsymbol{x}_i \cdot \boldsymbol{x}_j\right)\right) \end{aligned} \tag{9.68}$$

From the above process, we can see that ASNP computing the nearest points in the affine subspaces avoids convex quadratic programming routine and can directly obtain the optimum solution as Eq. (9.67) or Eq. (9.68).

We have introduced the linear ASNP above. But in real world, some data distribution is more complex and nonlinear. When convex hulls intersect (i.e. nonlinearly separating), the distance of nearest points from convex hulls will be zero. Similar with that, when the affine subspaces intersect, the distance in ASNP will also be zero. For the nonlinear distribution data, SVM introduces kernel trick to transform the nonlinear problem to a linear problem (i.e. convex hulls are non-intersection) theoretically. Now kernel method has been widely applied in classification problem, and it has been an effective method for nonlinear or complex data problems. In order to deal with nonlinear problems, we extend the ASNP algorithm to a nonlinear KASNP algorithm by the kernel trick in this section.

9.3.2.3 Kernel Affine Subspace Nearest Points (KASNP) Algorithm

Kernel Method and Kernel Trick

Kernel method [91, 118] is an algorithm that, by replacing the inner product with an appropriate positive definite function, implicitly performs a nonlinear mapping U of the input data from Rd into a high-dimensional feature space H. To compute dot products of (U(x) U(x0)), we employ kernel representation of the form k(x, x0) = (U(x) U(x0)), which allows us to compute the value of the dot products in H without having to actually carry out the map U.

Cover's theorem states that if the transformation is nonlinear and the dimensionality of the feature space is high enough, then the input space may be transformed into a new feature space where the patterns are linearly separable with high probability [119]. That is, when the decision function is not a linear function of the data, the data can be mapped from the input space into a high dimensional feature space by a nonlinear transformation. In this high dimensional feature space, a generalized optimal separating hyperplane is constructed. This nonlinear transformation just can be performed in an implicit way through the kernel methods. Thus the basic principle behind kernel-based algorithms is that a nonlinear mapping is used to extend the input space into a higher-dimensional feature space. Implementing a linear algorithm in the feature space then corresponds to a nonlinear version of the algorithm in the original input space. Kernel-based classification algorithms, primarily in Support Vector Machines (SVM), have gained a great deal of popularity in machine learning fields [91, 118, 120, 121].

Common choices of kernel function are the linear kernel k(x, y) = (x y), the polynomial kernel k(x, y) = (1 + (x y))d, and the radial basis function (RBF) kernel k(x, y) = exp (1/2)(kx yk/r)2 and the sigmoid kernel k(x, y) = tanh(b(x y) c). In this section, we adopt linear kernel and RBF kernel for experiments.

Kernel Affine Subspace Nearest Points (KASNP) Algorithm

Suppose a nonlinear mapping U of the input data in Rd into a high-dimensional feature space H. In space H, we construct the ASNP classifier. Similar to the linear case (see Eq. 9.63), the optimal problem of the closest points in H can be written as the following optimization problem:

$$\min_{\alpha}\left\|\left(\boldsymbol{u}_1+\sum_{i=1}^{m}\alpha_i\left(\Phi\left(\boldsymbol{x}_i\right)-\overline{\boldsymbol{u}}_1\right)\right)-\left(\boldsymbol{u}_2+\sum_{i=m+1}^{n}\alpha_i\left(\Phi\left(\boldsymbol{x}_i\right)-\overline{\boldsymbol{u}}_2\right)\right)\right\|^2 \tag{9.69}$$

Where $\overline{\boldsymbol{u}}_1=\frac{1}{m}\sum_{i=1}^{m}\Phi\left(\boldsymbol{x}_i\right),\overline{\boldsymbol{u}}_2=\frac{1}{n-m}\sum_{i=m+1}^{n}\Phi\left(\boldsymbol{x}_i\right)$.

Let $\boldsymbol{A}=\left(\overline{\boldsymbol{u}}_1-\Phi\left(\boldsymbol{x}_1\right),\ldots,\overline{\boldsymbol{u}}_1-\Phi\left(\boldsymbol{x}_m\right),\Phi\left(\boldsymbol{x}_{m+1}\right)-\overline{\boldsymbol{u}}_2,\ldots,\Phi\left(\boldsymbol{x}_n\right)-\overline{\boldsymbol{u}}_2,\right.$ Formula (9.69) can written as

$$\min_{\alpha} f\left(\boldsymbol{\alpha}\right)=\min_{\alpha}\left\|\left(\overline{\boldsymbol{u}}_1-\overline{\boldsymbol{u}}_2\right)-\boldsymbol{A}\alpha\right\|^2 \tag{9.70}$$

By solving $\frac{\partial f}{\partial\alpha}=0$, we have

$$\boldsymbol{A}^T\boldsymbol{A}\alpha=\boldsymbol{A}^T\left(\overline{\boldsymbol{u}}_1-\overline{\boldsymbol{u}}_2\right) \tag{9.71}$$

In Eq. (9.71) $\boldsymbol{A}^T\boldsymbol{A}$ *and* $\boldsymbol{A}^T\left(\overline{\boldsymbol{u}}_1-\overline{\boldsymbol{u}}_2\right)$ can be cast in terms of dot products $(\boldsymbol{\Phi}(\boldsymbol{x}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{x}_j))$ as follows:

$$\boldsymbol{A}^T\boldsymbol{A}=\left(\boldsymbol{M}^T\boldsymbol{F}+\boldsymbol{E}\right)^T\left(\Phi^T\Phi\right)\left(\boldsymbol{M}^T\boldsymbol{F}+\boldsymbol{E}\right) \tag{9.72}$$

$$\boldsymbol{A}^T\left(\overline{\boldsymbol{u}}_1-\overline{\boldsymbol{u}}_2\right)=\left(\boldsymbol{M}^T\boldsymbol{F}+\boldsymbol{E}\right)^T\left(\Phi^T\Phi\right)\boldsymbol{F}^T\boldsymbol{m}^T \tag{9.73}$$

Where $\boldsymbol{\Phi}=(\boldsymbol{\Phi}(\boldsymbol{x}_1),\ldots,\boldsymbol{\Phi}(\boldsymbol{x}_m),\boldsymbol{\Phi}(\boldsymbol{x}_{m+1}),\ldots,\boldsymbol{\Phi}(\boldsymbol{x}_n))$,

$$\boldsymbol{M}=\begin{pmatrix}\frac{1}{m} & 0\\ 0 & \frac{1}{n-m}\end{pmatrix}\begin{pmatrix}1\cdots1\ 0\cdots0\\ 0\cdots0\ 1\cdots1\end{pmatrix}_{2\times0},$$

$$F=\begin{pmatrix}1\cdots1\ 0\cdots0\\ 0\cdots0\ 1\cdots1\end{pmatrix}_{2\times n},m=\left(\frac{1}{m},\frac{1}{n-m}\right),$$

$$E = \begin{bmatrix} -1 & & & & & \\ & \ddots & & & & \\ & & -1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix},,$$

$$\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \begin{pmatrix} (\Phi(\boldsymbol{x}_1) \cdot \Phi(\boldsymbol{x}_1)) & \cdots & (\Phi(\boldsymbol{x}_1) \cdot \Phi(\boldsymbol{x}_n)) \\ \vdots & \ddots & \vdots \\ (\Phi(\boldsymbol{x}_n) \cdot \Phi(\boldsymbol{x}_1)) & \cdots & (\Phi(\boldsymbol{x}_n) \cdot \Phi(\boldsymbol{x}_n)) \end{pmatrix}.$$

Employing kernel representations of the form $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j))$, $\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi}$ is

$$\boldsymbol{K} = \Phi^T \Phi = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & k(\boldsymbol{x}_n, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{pmatrix}$$

Equations (9.72) and (9.73) can be kernelized:

$$\boldsymbol{A}^T \boldsymbol{A} = \left(\boldsymbol{M}^T \boldsymbol{F} + \boldsymbol{E}\right)^T \boldsymbol{K} \left(\boldsymbol{M}^T \boldsymbol{F} + \boldsymbol{E}\right) \tag{9.74}$$

$$\boldsymbol{A}^T (\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2) = \left(\boldsymbol{M}^T \boldsymbol{F} + \boldsymbol{E}\right)^T \boldsymbol{K} \boldsymbol{F}^T \boldsymbol{m}^T \tag{9.75}$$

So we can directly obtain the solution α of Eq. (9.69):

$$\boldsymbol{\alpha} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^+ \left(\boldsymbol{A}^T (\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2)\right) \tag{9.76}$$

or

$$\boldsymbol{\alpha} = \left(\boldsymbol{A}^T \boldsymbol{A} + \sigma \boldsymbol{I}\right)^{-1} \left(\boldsymbol{A}^T (\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2)\right) \tag{9.77}$$

where $\boldsymbol{A}^T\boldsymbol{A}^+$ is pseudo-inverse of $\boldsymbol{A}^T\boldsymbol{A}$; $\sigma \geq 0$, and I is n*n identity Matrix.

After getting the optimal solution α, two nearest point c and d can be represented by α:

$$\begin{aligned} \boldsymbol{c} &= \overline{\boldsymbol{u}}_1 + \textstyle\sum_{i=1}^m \alpha_i (\Phi(\boldsymbol{x}_i) - \overline{\boldsymbol{u}}_1) \\ &= \textstyle\sum_{i=1}^m \sum \left(\frac{1}{m}\left(1 - \sum_{i=1}^m \alpha_i\right) + \alpha_i\right) \Phi(\boldsymbol{x}_i) \end{aligned} \tag{9.78}$$

$$\begin{aligned} \boldsymbol{d} &= \overline{\boldsymbol{u}}_2 + \sum_{i=m+1}^{n} \alpha_i \left(\Phi\left(\boldsymbol{x}_i\right) - \overline{\boldsymbol{u}}_2\right) \\ &= \sum_{i=m+1}^{n} \left(\frac{1}{n-m}\left(1 - \sum_{i=m+1}^{n} \alpha_i\right) + \alpha_i\right) \Phi\left(\boldsymbol{x}_i\right) \end{aligned} \tag{9.79}$$

then, w, p and b can be written as:

$$\boldsymbol{w} = \boldsymbol{c} - \boldsymbol{d} = \Phi \boldsymbol{v}_1 \tag{9.80}$$

$$\boldsymbol{p} = (12)(\boldsymbol{c} + \boldsymbol{d}) = \frac{1}{2} \Phi \boldsymbol{v}_2 \tag{9.81}$$

$$b = -\boldsymbol{w} \cdot \boldsymbol{p} = -\frac{1}{2} v_1^T \Phi^T \Phi v_2 = -\frac{1}{2} v_1^T \boldsymbol{K} v_2 \tag{9.82}$$

where

$$\boldsymbol{v}_1 = \begin{pmatrix} \frac{1}{m}\left(1 - \sum_{i=1}^{m} \alpha_i\right) + \alpha_1 \\ \frac{1}{m}\left(1 - \sum_{i=1}^{m} \alpha_i\right) + \alpha_m \\ \frac{-1}{n-m}\left(1 - \sum_{i=m+1}^{n} \alpha_i\right) - \alpha_{m+1} \\ \frac{-1}{n-m}\left(1 - \sum_{i=m+1}^{n} \alpha_i\right) - \alpha_n \end{pmatrix} \tag{9.83}$$

$$\boldsymbol{v}_1 = \begin{pmatrix} \frac{1}{m}\left(1 - \sum_{i=1}^{m} \alpha_i\right) + \alpha_1 \\ \frac{1}{m}\left(1 - \sum_{i=1}^{m} \alpha_i\right) + \alpha_m \\ \frac{-1}{n-m}\left(1 - \sum_{i=m+1}^{n} \alpha_i\right) - \alpha_{m+1} \\ \frac{-1}{n-m}\left(1 - \sum_{i=m+1}^{n} \alpha_i\right) - \alpha_n \end{pmatrix} \cdot Z \tag{9.84}$$

So the decision boundary $(w \cdot \Phi(x)) + b = 0$ is

$$\boldsymbol{v}_2^T \boldsymbol{k}_x - \frac{1}{2} \boldsymbol{v}_2^T K \boldsymbol{v}_1 = 0 \tag{9.85}$$

Where $\boldsymbol{k}_x = \boldsymbol{\Phi}^T \boldsymbol{\Phi}(\boldsymbol{x}) = (k(\boldsymbol{x}_1, \boldsymbol{x}), k(\boldsymbol{x}_2, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_n, \boldsymbol{x}))^T$.
The decision function $f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w} \cdot \boldsymbol{\Phi}(\boldsymbol{x}) + b)$ is

$$f\left(\boldsymbol{x}\right) = \operatorname{sgn}\left(\boldsymbol{w} \cdot \Phi\left(\boldsymbol{x}\right) + b\right) = \operatorname{sgn}\left(\boldsymbol{v}_2^T \boldsymbol{k}_x - \frac{1}{2} \boldsymbol{v}_2^T \boldsymbol{K} \boldsymbol{v}_1\right) \tag{9.86}$$

According to the previous descriptions, the overall process of KASNP learning algorithm can be summarized into the following three steps:

Step 1: Computing the optimal solution $\boldsymbol{\alpha}$ of the nearest points problem of KASNP by training set:

$$\boldsymbol{\alpha} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{+} \left(\boldsymbol{A}^T \left(\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2\right)\right) \text{ or } \boldsymbol{\alpha} = \left(\boldsymbol{A}^T \boldsymbol{A} + \sigma \boldsymbol{I}\right)^{-1} \left(\boldsymbol{A}^T \left(\overline{\boldsymbol{u}}_1 - \overline{\boldsymbol{u}}_2\right)\right)$$

Step 2: Constructing decision boundary by $\boldsymbol{\alpha}$:

$$\boldsymbol{v}_2^T \boldsymbol{k}_x - \frac{1}{2}\boldsymbol{v}_2^T \boldsymbol{K}\boldsymbol{v}_1 = 0$$

Correspondingly, the decision function is

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\boldsymbol{v}_2^T \boldsymbol{k}_x - \frac{1}{2}\boldsymbol{v}_2^T \boldsymbol{K}\boldsymbol{v}_1\right)$$

Step 3: Testing a sample $\boldsymbol{y}$,

$$\text{If } f(\boldsymbol{y}) \geq 0, \boldsymbol{y} \in \text{the class of } \boldsymbol{S}_1; \text{ otherwise, } \boldsymbol{y} \in \text{the class of } \boldsymbol{S}_2$$

9.3.2.4 Two-Spiral Problem Test

2D two-spiral classification is a classical nonlinear problem and has been particularly popular for testing novel statistical pattern recognition classifiers. The problem is a difficult test case for learning algorithms [122, 123] and is known to give neural networks severe problems, but it can be successfully solved by nonlinear kernel SVMs [124, 125]. In this section, we also tested our KASNP with RBF kernel $k(x, y) = \exp\left(\frac{1}{2}\right)(x - y/\sigma)^2$ on a 2D two-spiral dataset accessible from the Carnegie Mellon repository [126]. The benchmark dataset, download from http://www.cgi.cs.cmu.edu/afs/cs.cmu.edu/project/vairepository/ai/areas/ai/areas/neural/bench/cmu/0.html, has two classes of spiral-shaped training data points, with 97 points for each, and is illustrated in Fig. 9.13. In order to visualize the separating surface by KASNP, the nodes of a 2D grid (0.05 space per grid) are tested and marked with different color (gray and white) to show their class. Figure 9.14 shows the decision region by KASNP. The parameter r of RBF kernel for KASNP is 0.8.

In Fig. 9.14, our KASNP constructs a smooth nonlinear spiral-shaped separating surface for the 2D two-spiral dataset, which implies that the KASNP classification method can achieve an excellent generalization for nonlinear data.

9.3.2.5 Credit Evaluation Applications and Experiments

Credit risk evaluation is a very typical classification problem to identify "good" and "bad" creditors. In this section, we apply KASNP for credit risk evaluation. To test the efficacy of our proposal KASNP for creditor evaluation, we compare it with SVM by linear kernel and RBF kernel on three real world credit datasets: Australian credit dataset, German credit dataset and a major US credit dataset.

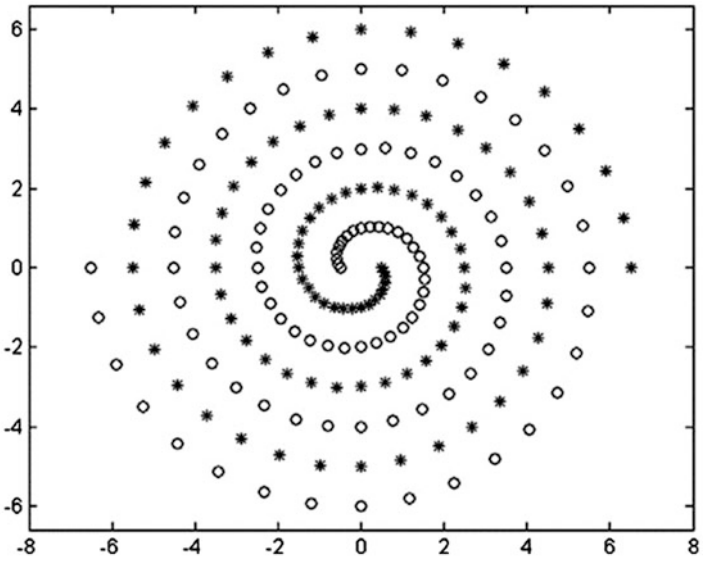

Fig. 9.13 2D two-spiral dataset: "o" spiral 1, "*" spiral 2

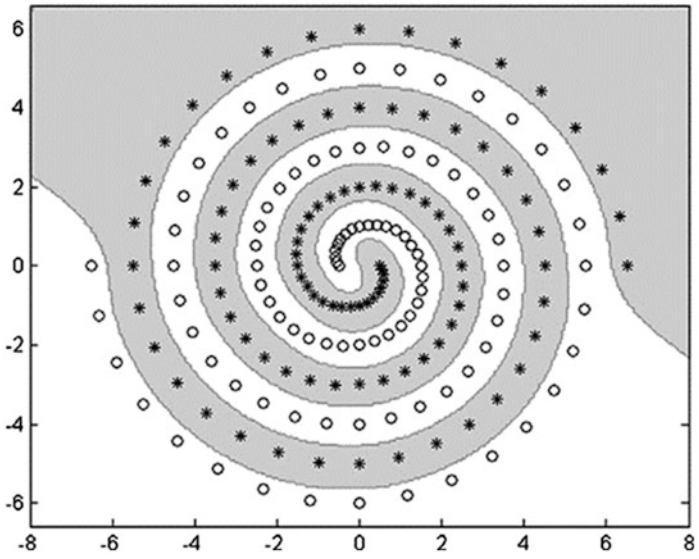

Fig. 9.14 The separation generated by RBF kernel KASNP

The compared linear kernel KASNP is equivalent to original ASNP method [114], that is, ASNP method is a special case of KASNP when kernel function is linear kernel.

Experiment Design

In our experiments, three accuracies will be tested to evaluate the classifiers, "Good" accuracy, "Bad" accuracy and Total accuracy:

$$"Good" Accuracy = \frac{number\ of\ correctly\ classified\ "Good"\ samples\ \ in\ test\ set}{number\ of\ "Good"\ samples\ in\ test\ set}$$

$$"Bad" Accuracy = \frac{number\ of\ correctly\ classified\ "Bad"\ samples\ \ in\ test\ set}{number\ of\ "Bad"\ samples\ in\ test\ set}$$

$$Total\ Accuracy = \frac{number\ of\ correct\ classification\ in\ test\ set}{number\ of\ samples\ in\ test\ set}$$

where "Good" accuracy and "Bad" accuracy respectively measure the capacity of the classifiers to identify "Good" or "Bad" clients. In the real world, for the special purposes to prevent the credit fraud, the accuracy of classification for the risky class must be improved to reach an acceptable standard but not excessively affecting the accuracy of classification for other classes. Thus, improving "Bad" accuracy is one of the most important tasks in credit scoring [127].

In our experiments of each dataset, we randomly select p (p = 40, 60, 80, ..., 180) samples from each class to train the compared classifiers and the remaining for the test. We repeat the test 20 times and report the mean of "Bad", "Good" and Total accuracies for each compared classifiers. All of our experiments are carried out on Matlab 7.0 platform. The convex quadratic programming problem of SVM is solved utilizing Matlab optimal tools. The experimental results on three credit datasets are separately given in the following subsections.

Results on Australian Credit Dataset

The Australian credit dataset from the UCI Repository of Machine Learning Databases (http://archive.ics.uci.edu/ml/) contains 690 instances of MasterCard applicants, 307 of which are classified as positive and 383 as negative. Each instance has 14 attributes, and all attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. With the number variety (40, 60, ..., 180) of randomly selected training samples per class, the "Bad" accuracy, "Good" accuracy and total accuracy comparisons of different methods on Australian credit dataset, are shown in Tables 9.11, 9.12, and 9.13 respectively. Parameter r of RBF kernel is set to 50,000 for both RBF SVM and RBF KASNP, and the penalty constant C of SVM is ∞.

In above experimental results, for "Bad" accuracy, nonlinear classifiers RBF SVM and RBF KASNP outperform other two linear classifiers, and RBF KASNP is better than RBF SVM. For "Good" accuracy, linear kernel KASNP is the best

Table 9.11 "Bad" accuracy (%) comparisons of different methods on Australian dataset

Number of training data per class	"Bad" accuracy (%) comparisons on Australian dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	79.65	84.50	81.97	86.90
60	83.08	85.20	82.06	88.05
80	81.01	86.07	81.34	88.18
100	84.12	87.37	81.27	87.60
120	83.71	86.71	81.48	87.51
140	82.12	87.14	81.40	87.43
160	82.38	87.00	80.25	87.02
180	79.48	86.77	80.07	86.26

Table 9.12 "Good" accuracy (%) comparisons of different methods on Australian dataset

Number of training data per class	"Good" accuracy (%) comparisons on Australian dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	81.85	73.95	89.76	72.73
60	87.00	74.98	89.74	76.84
80	85.15	78.28	91.43	79.52
100	83.31	79.44	91.69	81.06
120	84.87	81.36	91.90	82.86
140	83.68	82.49	91.32	84.07
160	84.25	84.05	92.24	85.14
180	84.76	83.86	91.85	86.22

Table 9.13 Total accuracy (%) comparisons of different methods on Australian dataset

Number of training data per class	Total accuracy (%) comparisons on Australian dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	80.61	79.89	85.38	80.70
60	84.78	80.77	85.39	83.19
80	82.78	82.74	85.66	84.47
100	83.78	84.02	85.67	84.84
120	84.19	84.49	85.81	85.58
140	82.76	85.24	85.44	86.06
160	83.12	85.82	85.01	86.27
180	81.52	85.65	84.61	86.24

of all classifiers, and its "Good" accuracy can get 89.74–92.24% (see Table 9.12). From the total accuracy comparisons, KASNP dominates SVMs. Linear KASNP can reach the highest total accuracy when the number of training samples $p = 40, \ldots, 120$, and RBF KASNP is the best one when $p = 140, 160, 180$ (see Table 9.13).

Table 9.14 "Bad" accuracy (%) comparisons of different methods on German dataset

Number of training data per class	"Bad" accuracy (%) comparisons on German dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	65.87	67.08	67.12	67.15
60	67.90	68.77	67.08	67.60
80	69.64	70.20	69.73	70.66
100	71.47	69.92	71.35	71.53
120	70.92	71.81	72.28	72.36
140	71.06	72.47	73.59	73.16
160	71.29	72.75	71.46	73.75
180	73.13	72.13	72.42	72.83

Table 9.15 "Good" accuracy (%) comparisons of different methods on German dataset

Number of training data per class	"Good" accuracy (%) comparisons on German dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	64.91	68.83	66.56	68.89
60	67.73	69.16	66.95	71.09
80	68.75	69.75	69.56	69.60
100	68.23	69.83	69.38	69.89
120	69.89	69.59	68.88	69.58
140	69.63	69.96	69.22	69.83
160	70.94	70.56	70.85	71.31
180	70.33	70.57	70.40	70.66

Results on German Credit Dataset

The German credit dataset from the UCI Repository of Machine Learning Databases (http://archive.ics.uci.edu/ml/) concludes 1000 instances, 700 instances of credit-worthy applicants and 300 instances whose credit should not be extended. For each instance, 24 numerical attributes describe the credit history, account balances, loan purpose, loan amount, employment status, and personal information. The different accuracy comparisons of the classifiers on German dataset are given in Tables 9.11, 9.12, and 9.13 respectively. The parameter r of RBF kernel for SVM and KASNP is set to r = 20,000, and the penalty constant C of SVM is set to 1.

From the experimental results in Tables 9.14, 9.15, and 9.16, we can see that our proposed RBF KASNP is slightly better than others. RBF KASNP has five highest accuracies (when p = 40, 80, 100, 120, 160) in "Bad" accuracy comparison, and six best results (when p = 40, 60, 80, 100, 160, 180) for "Good" clients identification. For total accuracy, RBF KASNP continuously achieves the highest accuracy in eight comparison results.

Table 9.16 Total accuracy (%) comparisons of different methods on German dataset

Number of training data per class	Total accuracy (%) comparisons on German dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	65.18	68.34	66.72	68.40
60	67.77	69.05	66.99	70.14
80	68.98	69.87	69.60	69.88
100	69.04	69.86	69.87	70.30
120	70.13	70.12	69.68	70.24
140	69.95	70.51	70.19	70.57
160	71.01	71.01	70.98	71.82
180	70.85	70.86	70.78	71.07

Table 9.17 "Bad" accuracy (%) comparisons of different methods on USA dataset

Number of training data per class	"Bad" accuracy (%) comparisons on USA dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	63.97	65.34	61.83	81.32
60	65.82	66.01	68.35	82.44
80	67.37	69.99	71.89	83.33
100	66.32	70.69	74.41	83.29
120	68.07	69.43	77.40	82.82
140	67.58	71.64	78.14	84.59
160	69.27	73.21	78.39	84.14
180	73.13	74.44	79.57	84.37

Results on USA Credit Dataset

The last credit card dataset used in our experiments is provided by a major U.S. bank. It contains 6000 records and 66 derived attributes. Among these 6000 records, 960 are bankruptcy accounts and 5040 are "good" status accounts [128]. The "Bad", "Good" and total accuracy comparisons of the classifiers are shown in Tables 9.17, 9.18, and 9.19 respectively. Parameter r of RBF kernel of SVM and KASNP is r = 10,000, and the penalty constant C of SVM is C = 1.

Comparing the results reported in Tables 9.17, 9.18, and 9.19, we find the following results: (1) RBF KASNP is superior to other classifiers in finding "Bad" clients. As we can see from Table 9.17, only using 80 training samples (40 per class), RBF KASNP can achieve best "Bad" classification results 81.32% which is at least higher 15% than the accuracies of other approaches. (2) For identifying "Good" clients, four approaches have not clear difference, and RBF SVM and linear KASNP respectively have four best results in Table 9.18. (3) From the general view (see Table 9.19), the two KASNP approaches dominate SVMs. RBF KASNP performs the best when p = 40, ..., 120, and linear KASNP outperforms the others when p = 140, 160, 180.

Table 9.18 "Good" accuracy (%) comparisons of different methods on USA dataset

Number of training data per class	"Good" accuracy (%) comparisons on USA dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	67.12	67.62	59.13	66.11
60	66.46	67.84	65.73	67.15
80	66.65	66.35	68.33	67.15
100	67.02	67.97	67.40	67.45
120	69.34	69.72	68.36	68.00
140	68.04	68.79	69.44	67.13
160	66.59	68.66	70.52	67.73
180	61.38	68.93	70.18	67.69

Table 9.19 Total accuracy (%) comparisons of different methods on USA dataset

Number of training data per class	Total accuracy (%) comparisons on USA dataset			
	Linear SVM	RBF SVM	Linear KASNP	RBF KASNP
40	67.81	67.27	59.55	68.48
60	66.44	67.56	66.13	69.49
80	67.39	66.90	68.86	69.59
100	66.92	68.37	68.44	69.80
120	69.15	69.68	69.68	70.16
140	67.98	69.20	70.69	69.63
160	66.97	69.30	71.63	70.04
180	63.01	69.69	71.48	69.99

9.3.2.6 Discussion

From above experimental results of three credit datasets, we can conclude that as a whole the proposed KASNP is comparable with SVM for creditor classification. As we know, the capacity of finding "Bad" clients is an important measure for credit risk evaluation approaches. From "Bad" accuracy comparison experimental results in Tables 9.11, 9.12, and 9.13, we note that our proposed KASNP with RBF kernel can achieve the best performance for identifying "Bad" creditors. Especially for US dataset, KASNP obviously outperformed other approaches. In total performance, RBF KASNP also performed better than SVMs. Thus, RBF KASNP classifier made a better risky classification performance. Moreover, we also note that, for "Good" clients identification, linear KASNP is a good classifier. Especially on Australian dataset, linear KASNP obtained wonderful "Good" accuracies, while its "Bad" accuracies also kept acceptable standard.

9.3.3 A Dynamic Assessment Method for Urban Eco-Environmental Quality Evaluation

This subsection provides an urban eco-environmental quality assessment system with a dynamic assessment of the Yangtze River Delta and the Pearl River Delta economic zones are proposed and analyzed.

9.3.3.1 Related Works

Assessment of Urban Eco-Environmental Quality

With the rapid surge in urbanization around the world, there are a series of urban eco-environmental problems. In 1962, Carson described the destruction of urban eco-environment in Silent Spring for the first time, which led to the wide-range attention. In 1971, the United Nations Educational, Scientific and Cultural Organization developed the 'Man and the Biosphere' research project, which focused on the eco-environment of human settlements and carried out the urban research subject in human ecology theories and views [129]. Schneider pointed out: 'in contrast with common sense of many urban sociologist and environmentalists, that the urban basic issues are not clean air and water, not endangered species or environment, not energy, nor the urban housing construction and renovation investment, but the association structure of the human environment—the city, it is necessary to build up a harmonious developing city to solve the problem' [29]. In 1984, Yanitsky established a human residence where economy, society and nature are coordinated in development. In 1998, Bohm studied the special urban development process of Vienna in Australia. Although the number of population has not changed significantly, the residential area, road area, and energy consumption have increased significantly, and urban green space reduced significantly. The United Nations human environment and development conference held in Rio de Janeiro, Brazil, pointed out that environmental issue will be the largest challenge in the twenty-first century. The urban eco-environmental quality problem has been an active research fields for years [115, 130–132].

Sensitivity Analysis

Multi-attribute evaluation (MAE) is used in assessment when the known options available are fixed, and the number of the evaluation alternatives are limited [133]. The reliability of the evaluation results is tested in the sensitivity analysis. For a limited alternative set, there are two parameters to determine their ranking of the alternatives: one is the relative importance among attributes, that is, attribute weights; and the other is attribute value correspondent to each alternative.

The early studies of the sensitivity analysis focused on the key attribute weights [134, 135]. Starr [136], Isaacs [137], Fishbum [138] and Evans [139], studied the maximum regional-changed issues when the alternative order remained constant. French and Insua [140] determined the potential competitors in the current optimal solution with the minimum distance method. Masuda [141] and Armacost and Hosseini [142] studied the sensitivity analysis on the analytic hierarchy process (AHP). Ringuest [143] studied the distance sensitivity analysis between the set closest to the original weight and original weight when the optimal solution remained unchanged.

Urban Eco-Environmental Quality Index System

Here, an Urban Eco-Environmental Quality Index System is proposed to assess urban eco-environmental development and quality level.

To build an Urban Eco-Environmental Quality Index System, the following principles should be followed.

People-oriented principle. The core of urban eco-environment is 'human', who is both the creator and the bearer of urban eco-environment. Therefore, the assessment index system should not only reflect on what are closely related with people's living, but also reflect the objective and subjective experience on the environment.

Comprehensiveness principle. The construction of the assessment index system must reflect all aspects of urban eco-environment, including the living conditions, natural environment, social environment, and infrastructure indicators, as well as all aspects of urban environment.

Representative principle. The assessment index system should reflect the main features of urban eco-environment. Both qualitative indicators and quantitative indicators should be included.

9.3.3.2 Selecting Indicators

According to the previous studies [144–146], we selected 25 comprehensive evaluation index, from four perspectives—population ecological indicators, nature ecological indicators, economy ecological indicators, and society ecological indicators to establish the index system, which includes both the cost-based indicators and efficiency-based indicators [147]. The details of all indicators are shown in Table 9.20.

These indicators are collected from the 'China City Statistical Yearbook' and the 'China Statistical Yearbook for Regional Economy', in order to increase the comparability of the index, we unify the indicators to the relative ratio, such as

Table 9.20 Urban eco-environmental quality index system

Factors	Subfactors
Population ecological indicators	Natural population growth rate (%) population density (person/km)
Nature ecological indicators	Percentage of hospital doctors in urban population (%)
	Percentage of college students in urban population (%)
	Percentage of industrial waste water up to the discharge standards (%)
Economy ecological indicators	Industrial waste gas treatment rate (%)
	Industrial solid waste comprehensive utilization rate (%)
	Urban sewage treatment rate (%)
	Domestic garbage treatment rate (%)
	Percentage of comprehensive utilization value of waste products in gross regional product (%)
	Green area per person (square meter/person) green coverage rate in completed area (%)
	Unemployment rate (%)
Society ecological indicators	Public library collection per 100 people (book, part/100 people) percentage of the internet users in urban population (%)
	Household water consumption per person
	(ton/person)
	Household electricity consumption per person (kilowatt hour/person)
	Bus per 10,000 people (bus/10,000 people)
	Urban road area per person (square meter/person)
	Percentage of urban construction land in urban area (%)
	Percentage of tertiary industries in gross regional product (%)
	Gross regional product per person (RMB/person)
	Gross regional product growth rate (%)
	Percentage of investment in science and education in fiscal expenditure (%)
	Average wage of staff and workers (RMB/person)

$$\text{percentage of hospital doctors in urban population} = \frac{\text{hosptial doctors}}{\text{urban population}} \times 100\%$$

$$\text{percentage of investment in science and education in fiscal expenditure} = \frac{\text{investment in science and education}}{\text{fiscal expenditure}} \times 100\%$$

Evaluation Method

The proposed evaluation method includes three steps: The first step is the data preprocessing, the second step is the Dynamic Assessment, and the third step is the sensitivity analysis.

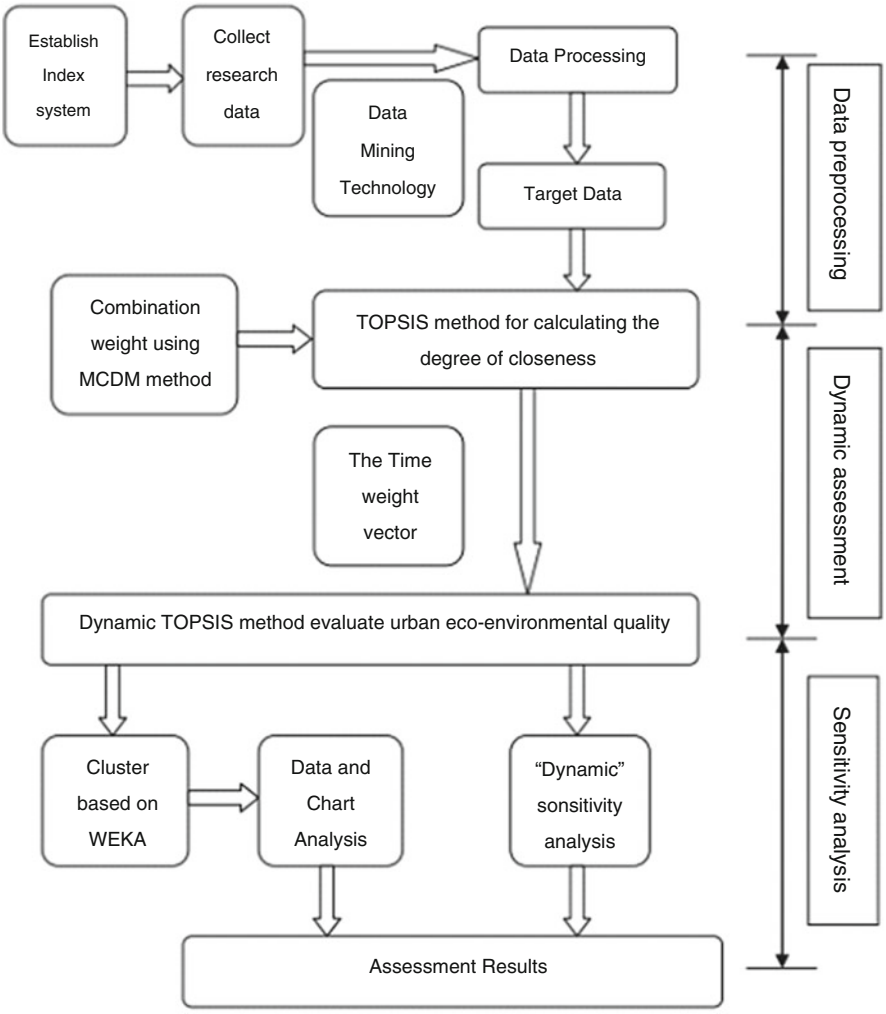

Fig. 9.15 The evaluation framework flow chart

In data preprocessing, evaluation index system is setup and data is processed. The evaluation index system is based on ecological theory, and advices of experts. In data processing, data is cleaned and transformed. A Dynamic Assessment model to evaluate the urban eco-environmental quality is proposed. The sensitivity of attributes weights and values are analyzed.

Figure 9.15 shows the structure of the proposed evaluation model. In the following subsections, we will present the details of the models and methods in proposed framework.

Multi-criteria Decision Making Method

Multi-criteria decision making method (MCDM) is a decision making analysis method, which has been developed since 1970s. MCDM is the study of methods and procedures by which concerns about multiple conflicting criteria can be formally incorporated into the management planning process and the optimum one can be identified from a set of alternatives. In the following subsections, MCDM related methods, Entropy Method, Grey Relation Analysis (GRA) and Technique for order preference by similarity to ideal solution (TOPSIS), which are integrated in this research, are discussed.

Entropy Method

In this research, we introduced the concept of entropy to measure the information, which is a term in information theory, also known as the average amount of information. The index weight is calculated by the Entropy Method. According to the degree of index dispersion, the weight of all indicators is calculated by information entropy. Entropy method is highly reliable and can be easily adopted in information measurement. The calculation steps are as follows:

Suppose we have a decision matrix B with m alternatives and n indicators:

1. In matrix B, feature weight p_{ij} is of the ith alternative to the jth factor:

$$\mathrm{p}_{ij} = y_{ij} \sum_{i=1}^{m} y_{ij} \ (1 \le i \le m, 1 \le j \le n) \tag{9.87}$$

2. The output entropy e_j of the jth factor becomes

$$e_j = -k \sum_{i=1}^{m} p_{ij} \ \ln p_{ij} \ (k = 1 \ \ln m; 1 \le j \le n) \tag{9.88}$$

3. Variation coefficient of the jth factor: g_j can be defined by following equation:

$$g_j = 1 - e_j, \ \ (1 \le j \le \mathrm{n}) \tag{9.89}$$

Note that the larger g_j is, the higher the weight should be.

4. Calculate the weight of entropy α_j:

$$\alpha_j = g_j \sum_{j=1}^{m} g_j, \ \ (1 \le j \le n) \tag{9.90}$$

Grey Relational Analysis Method

Grey relational analysis is a part of grey theory, which can handle imprecise and incomplete information in grey systems. GRA only requires small sample

data, simple calculation and the precision is quite high. Specifically, weights are calculated as [148].

Suppose we have the initial matrix R

$$\mathrm{R} = \begin{bmatrix} x_1 & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

1. Standardize the raw matrix R

$$\mathrm{R} = \begin{bmatrix} x'_1 & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{bmatrix} \tag{9.91}$$

2. Generate the reference sequence x'_0

$$x'_0 = \left(x'_0(1), x'_0(2), \cdots, x'_0(n)\right) \tag{9.92}$$

$x'_0(j)$ is the largest and normalized value in the jth factor.

3. Calculate the difference $\Delta_{0i}(j)$ between the normalize sequences and the reference sequence x'_0

$$\Delta_{0i}(j) = |x'_0(j) - x'_{ij}|$$

$$\Delta = \begin{bmatrix} \Delta_{01}(1) & \Delta_{01}(2) & \cdots & \Delta_{01}(n) \\ \Delta_{02}(1) & \Delta_{02}(2) & \cdots & \Delta_{02}(n) \\ \vdots & \vdots & \vdots & \vdots \\ \Delta_{0m}(1) & \Delta_{0m}(2) & \cdots & \Delta_{0m}(n) \end{bmatrix} \tag{9.93}$$

4. Compute the grey coefficient: $r_{0i}(j)$

$$r_{0i}(j) = \frac{\min_{i=1}^{n} \min_{j=1}^{m} \Delta_{0i}(j) + \delta \max_{i=1}^{n} \max_{j=1}^{m} \Delta_{0i}(j)}{\Delta_{0i}(j) + \delta \max_{i=1}^{n} \max_{j=1}^{m} \Delta_{0i}(j)} \tag{9.94}$$

where δ is a distinguished coefficient. Usually, the value of d often is set to 0.5, to offer moderate distinguishing effects and good stability.

5. Obtain the grey relational degree value: b_i

$$b_i = \frac{1}{n} \sum\nolimits_{j=1}^{n} r_{0i}(j) \tag{9.95}$$

6. Finally, calculate the weight of GRA: β_i

$$\beta_i = \frac{b_i}{\sum_{i=1}^{n} b_i} \tag{9.96}$$

In this research, we use Entropy and the GRA method to calculate the normalized weight of the indicators.

Technique for Order Preference by Similarity to Ideal Solution Method

Technique for order preference by similarity to ideal solution TOPSIS was initially developed to rank alternatives over multiple criteria. TOPSIS finds the best alternatives by minimizing the distance to the ideal solution and maximizing the distance to the nadir or negative-ideal solution [34]. All alternative solutions can be ranked according to their closeness to the ideal solution. Because its first introduction, a number of extensions and variations of TOPSIS have been developed over the years. The calculation steps are as follows:

1. Calculate the normalized decision matrix A. The normalized value a_{ij} is calculated as

$$a_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} \left(x_{ij}\right)^2}} \left(1 \le i \le m, 1 \le j \le n\right) \tag{9.97}$$

2. Calculate the weighted normalized decision matrix

$$D = \left(a_{ij} * w_j\right) \left(1 \le i \le \mathrm{m}, 1 \le \mathrm{j} \le \mathrm{n}\right) \tag{9.98}$$

where w_j is the weight of the ith criterion, and $\sum_{j=1}^{n} w_j = 1$.

3. Calculate the ideal solution V^* and the negative ideal solution V^-

$$\begin{aligned} V^* &= \left\{v_1^*, v_2^*, \cdots, v_n^*\right\} = \left\{\left(\max_i v_{ij} | j \in J\right), \left(\min_i v_{ij} | j \in J'\right)\right\} \\ V^- &= \left\{v_1^-, v_2^-, \cdots, v_n^-\right\} = \left\{\left(\min_i v_{ij} | j \in J\right), \left(\max_i v_{ij} | j \in J'\right)\right\} \end{aligned} \tag{9.99}$$

4. Calculate the separation measures, using the m-dimensional Euclidean distance

$$\begin{aligned} S_i^+ &= \sqrt{\sum_{j=1}^{n} \left(V_i^j - V^*\right)^2} \left(1 \le i \le m, 1 \le j \le n\right) \\ S_i^- &= \sqrt{\sum_{j=1}^{n} \left(V_i^j - V^-\right)^2} \left(1 \le i \le m, 1 \le j \le n\right) \end{aligned} \tag{9.100}$$

5. Calculate the relative closeness to the ideal solution

$$Y_i = \frac{S_i^-}{S_i^+ + S_i^-} \ (1 \le i \le m) \tag{9.101}$$

where $Y_i \in (0, 1)$. The larger Y_i is, the closer the alternative is to the ideal solution.

6. Rank the preference order

The larger TOPSIS value, the better the alternative.

Dynamic Assessment Method

Dynamic assessment has been introduced by Feuesrtein in the 'theory, tools, techniques of learning potential assessment—the dynamic assessment on hysteresis operators' in 1979. The root of its theory can be traced back to 'the zone of proximal development' by Vygotsky [149]. Over time and the accumulation of the data, people have many chronological sequence data of the plane data table series, called 'time series data sheet.' Comprehensive evaluation with time series data, its parameter values are dynamic, which is defined as 'dynamic comprehensive evaluation' problem [150].

Dimension Reduction for Time Series Data

With the proposed dynamic TOPSIS model, the three-dimensional time series data is reduced to two-dimensional data using the time–weight vector described in the following subsection. The time-weighted vector $\mathrm{w} = (\mathrm{w}_1, \mathrm{w}_2, \mathrm{w}_\mathrm{n})$ T represents the degree of emphasis on different time, according to different criteria. The 'time–weight vector entropy' I is given as $I = -\sum_{k=1}^{p} w_k \ \ln \ w_k$, and the 'time degree' T is $T = \sum_{k=1}^{p} w_k \frac{p-k}{p-1}$, where p is the number of years.

The 'time degree' T indicates the degree to which the aggregation operator values a time interval. It can take a value between 0 and 1 to reflect the attitude of a decision maker as shown in Table 9.21. $\mathrm{T} = 0$ implies that time weighted vector w becomes $(0, 0, \ldots, 1)$ and the element with the latest time value obtains the largest weight. $\mathrm{T} = 1$ implies that time weighted vector w becomes $(1, 0, \ldots, 0)$ and the element with the earliest time value obtains the largest weight. $\mathrm{T} = 0.5$ implies that data elements of different years have the same importance.

The criterion to determine the time–weight vector is that in the condition of a given 'time degree' T, to mine sample information as much as possible and consider different information of evaluated samples in the timing. The time weighted vector

Table 9.21 The Mean of the time degree T

T value	Illustration
0.1	The recent data is most important
0.3	The recent data is more important
0.5	The data is of equal importance
0.7	The earlier data is more important
0.9	The earlier data is most important
0.2, 0.4, 0.6, 0.8	Intermediate values between adjacent scale values

can be calculated:

$$\begin{cases} \text{MAX}\left(-\sum_{k=1}^{p} w_k \ln w_k\right) \\ s.t. T = \sum_{k=1}^{p} w_k \frac{p-k}{p-1} \\ \sum_{k=1}^{p} w_k = 1, w_k \in [0, 1], k = 1, 2, \cdots, p \end{cases} \tag{9.102}$$

9.3.3.3 Dynamic Technique for Order Preference by Similarity to Ideal Solution Evaluation Method

The dynamic TOPSIS evaluation method based on a dynamic assessment model is used to assess eco-environmental quality, and the proposed method considers the time weight vector to construct three-dimensional time series data [151]. In this model, through the MCDM (TOPSIS), the two-dimensional data is reduced to one-dimensional data to dynamically assess the quality of the urban eco-environment. The steps of proposed dynamic assessment method are as follows:

1. Determine the evaluation index system, according to the ecological theory.
2. Data preprocessing and standardization.
3. Use multi-attribute evaluation method to determine the combination weight.
4. Use MCDM: TOPSIS method to assess the level of urban eco-environmental quality from 2005 to 2009.
5. Create a dynamic assessment model as

$$Z = \alpha_1 Y_1 + \alpha_2 Y_2 + \cdots \alpha_i Y_i + \cdots + \alpha_n Y_n \ (i = 1, 2, \cdots n) \tag{9.103}$$

Where Y_i is defined in Eq. (9.101) used by TOPSIS method to determine relative closeness degree of the urban eco-environmental quality each year. ai is defined in Eq. (9.102) and is the time–weight vector w_i.

Calculate the utility value of urban eco-environmental quality.

Dynamic Sensitivity Analysis

There are two aspects of sensitivity analysis—one is the sensitivity analysis of attribute weight, and the other is the sensitivity analysis of attribute value. However, previous studies on sensitivity analysis are static assessment, which does not show the influence of time [152].

The Dynamic sensitivity analysis is to consider the influence of the Dynamic time weight vector for decision-maker to make the final decisions. Because of the uncertainty of the time–weight vector, the assessment results are uncertain. It is necessary and critical to do sensitivity analysis of dynamic assessment method.

Assume that the weight w_k of index T_k has small fluctuations w_k, the changes in weight value are defined as $w_k^* = w_k + \Delta w_k$, whereas the other weights remain unchanged. After the normalization, we obtain

$$\begin{aligned} w_k' &= \frac{w_k}{w_1+w_2+\cdots w_k+\Delta w_k+\cdots w_n} \\ &= \frac{w_i}{(w_1+w_2+\cdots w_k^*+\cdots w_n)(k=1,2,\cdots n)} \end{aligned} \tag{9.104}$$

The stable range of the index T_k is

$$\begin{cases} \Delta w_k > -w_k, y_{ik} = y_{tk} \\ -w_k < \Delta w_k < \sum_{j=1}^{n} \frac{(y_{ij}-y_{tj})w_k}{y_{tj}-y_{ij}}, y_{ik} < y_{tk} \\ \Delta w_k > \max\left[\sum_{j=1}^{n} \frac{(y_{ij}-y_{tj})w_k}{y_{tj}-y_{ij}}, -w_k\right], y_{ik} > y_{tk} \end{cases} \tag{9.105}$$

K-Means Clustering Algorithm

Clustering analysis divides data set into several different classes, making the data in the same class as similar as possible, but in the different class, as dissimilar as possible [10]. The higher the degree of similarities among similar objects and the more differences among the dissimilar objects, the better the cluster quality.

Cluster is 'the process of dividing physical or abstract objects into similar object classes' [15]. The steps of the K-means cluster algorithm are as follows:

1. Put n objects into k non-empty set.
2. Select random seed value as the current center of clusters.
3. Assign each object with the nearest seed value.
4. Repeat the second step, until there are no new assignments.

In this study, we complete the K-means clustering method by using the WEKA software [16], the specific processes are showed in Fig. 9.16.

The data of empirical study is collected from the 'China City Statistical Yearbook' and 'China Statistical Year-book for Regional Economy' between 2005 and 2009 in [8].

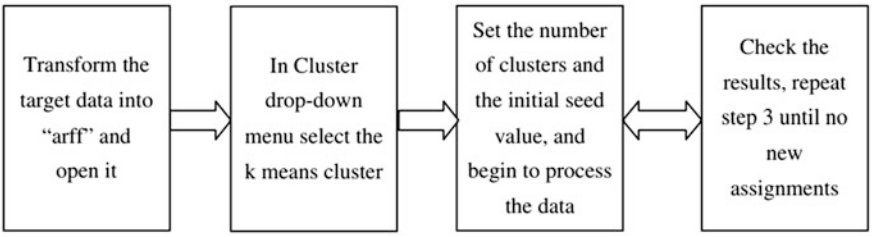

Fig. 9.16 K-means clustering algorithm based on WEKA flow chart

9.3.4 An Empirical Study of Classification Algorithm Evaluation for Financial Risk Prediction

This subsection is to develop an approach to evaluate classification algorithms for financial risk prediction. It constructs a performance score to measure the performance of classification algorithms and introduces MCDM methods to rank the classifiers. An empirical study is designed to assess nine classification algorithms using five performance measures over seven real-life credit risk and fraud risk datasets from six countries. For each performance measure, a performance score is calculated for each selected classification algorithm. The classification algorithms are then ranked using three MCDM methods (i.e., TOPSIS, PROMETHEE, and VIKOR) based on the performance scores.

Another problem in financial risk detection is that the knowledge gap [58] between the results classification methods can provide and taking actions based on them remains large. The lack of interaction between industry practitioners and academic researchers makes it hard to discover financial risks or opportunities and hence weakens the value that classification methods may bring to financial risk detection. To deal with the knowledge gap problem, this section combines the classification results, the knowledge discovery in database (KDD) process, and the concept of chance discovery to build a knowledge-rich financial risk management process in an attempt to increase the usefulness of classification results in financial risk prediction.

9.3.4.1 Evaluation Approach for Classification Algorithms

This section develops a two-step process to evaluate classification algorithms for financial risk prediction. In the first step, a performance score is created for each selected classification algorithm. The second step applies three MCDM methods (i.e., TOP-SIS, PROMETHEE, and VIKOR) to rank the selected classification algorithms using the performance scores as inputs. This section describes how the performance scores are calculated and gives an overview the three MCDM methods used in the study.

Performance Score

There are a variety of measures for classification algorithms and these measures have been developed to evaluate very different things. Some studies have shown that the classification algorithm achieves the best performance according to a given measure on a dataset, may not be the best method using a different measure [106, 153]. In addition, characteristics of datasets, such as size, class distribution, or noise, can affect the performance of classifiers. Hence, evaluating the performance of classification algorithms using one or two measures on one or two datasets often proves to be inadequate.

Based on these two considerations, this study constructs a performance metric that assesses the quality of classifiers using a set of measures on a collection of financial risk datasets in an attempt to give a comprehensive evaluation of classification algorithms. The basic idea of this performance metric is similar to ranking methods, which use experimental results generated by a set of classification algorithms on a set of datasets to rank those algorithms [154]. Specifically, it resembles the significant wins (SW) ranking method by conducting pairwise comparisons of classifiers using tests of statistical significance.

Selection of Performance Measures

Accuracy and error rates are important measures of classification algorithms in financial risk prediction. This work utilizes overall accuracy, precision, true positive rate, true negative rate, and the area under the receiver operating characteristic curve (AUC) to build the performance score. The following paragraphs define and describe these measures.

- Accuracy is the percentage of correctly classified instances [15]. It is one the most widely used classification performance metrics.

$$\text{overall accuracy} = \frac{\text{TN} + \text{TP}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

  where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. TP and TN are defined below. FP is the number of non-fault-prone instances that is misclassified as fault-prone class. FN is the number of fault-prone instances that is misclassified as non-fault-prone class.
- Precision is the number of classified positive or abnormal instances that actually are positive instances.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- TP (true positive) is the number of correctly classified positive or abnormal instances. TP rate measures how well a classifier can recognize abnormal records. It is also called sensitivity measure. In the case of financial risk detection,

abnormal instances are bankrupt, fraudulent or erroneous accounts. A classifier with a higher TP rate can help financial institutions reduce their potential credit losses than a classifier with a lower TP rate.

$$\text{true positive rate/sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- TN (true negative) is the number of correctly classified negative or normal instances. TN rate measures how well a classifier can recognize normal records. It is also called specificity measure.

$$\text{true negative rate/specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- ROC stands for receiver operating characteristic, which shows the tradeoff between TP rate and FP rate [15]. The area under the ROC (AUC) represents the accuracy of a classifier. The larger the area, the better the classifier.

Calculation of the Performance Score

The performance score is generated by conducting paired t tests with a significance level of 5% for each classifier. The goal of a paired statistical significance test is to evaluate whether the superior or inferior performance of one classifier over another is statistically significant. The performance score for each classifier is calculated as follows:

Step 1: for each dataset, compare the tenfold cross-validation results of individual performance measure for two classifiers. The null hypothesis is that the two classifiers are the same. If the paired statistical significance (0.05) test indicates that one classifier is better than the other classifier, the performance scores of the superior classifier and the inferior classifier equals to 1 and -1, respectively. If the paired statistical significance (0.05) test indicates that the null hypothesis cannot be rejected, then the performance scores for both classifiers equal to 0 in this case.

Step 2: repeat Step 1 for all classifier pairs for the dataset tested in Step 1. Then we get performance scores of all classifiers for the specific dataset and specific performance measure.

Step 3: repeat Steps 1 and 2 for other datasets included in the experiment. The sum of performance scores from all datasets is the performance score of this classifier for the current performance measure. The larger the score is, the better the classifier performs in this measure.

Step 4: repeat Steps 1, 2 and 3 for other four performance measures to get the performance scores of all classifiers for all performance measures.

MCDM Methods

To evaluate classification algorithms, normally more than one criterion needs to be examined, such as accuracy, AUC, and misclassification rate. Thus algorithm selection can be modeled as multiple criteria decision making (MCDM) problems [155]. This subsection uses three MCDM methods, i.e., TOPSIS, PROMETHEE, and VIKOR, and explains how they can be used to rank classification algorithms.

Experiment

The experiment is designed to validate the proposed two-step evaluation approach using nine classification methods over seven real-life credit risk and fraud risk datasets from six countries. The first and second parts of this section give an overview of classification algorithms and financial risk datasets used in the empirical study. The third and fourth parts describe the experimental design and the evaluation results.

9.3.4.2 Classification Algorithms

The classification algorithms used in the experiment include eight well-known classification techniques and ensemble method. The eight classification methods are Bayesian Network [93], Naïve Bayes [92], support vector machine (SVM) [90], linear logistic regression [156], k-nearest neighbor [94], C4.5 [87], Repeated Incremental Pruning to Produce Error Reduction (RIPPER) rule induction [96], and radial basis function (RBF) network [89]. All algorithms were implemented using Weka 3.6, a free data mining software package [16].

Bayesian Network and Naïve Bayes both model probabilistic relationships between predictor variables and the class variable. While Naïve Bayes classifier estimates the class-conditional probability based on Bayes theorem and can only represent simple distributions, Bayesian Network is a probabilistic graphic model and can represent conditional independencies between variables. SVM classifier uses a nonlinear mapping to transform the training data into a higher dimension and search for the linear optimal separating hyperplane, which is then used to separate data from different classes [15]. Linear logistic regression models the probability of occurrence of an event as a linear function of a set of predictor variables. k-nearest neighbor classifies a given data instance based on learning by analogy, that is, assigns it to the closest training examples in the feature space. C4.5 is a decision tree algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner. RIPPER is a sequential covering algorithm that extracts classification rules directly from the training data without generating a decision tree first [15]. RBF network is an artificial neural network that uses radial basis functions as activation functions.

In addition to the eight classification techniques, ensemble method was included in the experiment. An ensemble consists of a set of individually trained classifiers

whose predictions are combined when classifying novel instances. There are two fundamental elements of ensembles: a set of properly trained classifiers and an aggregation mechanism that organizes these classifiers into the output ensemble. This study uses the vote algorithm in Weka to perform the ensemble method. Vote combines classifiers by averaging their probability estimates [16].

9.3.4.3 Financial Risk Datasets

The datasets used in this study come from six countries and represent four aspects of financial risk: credit approval (credit card application), credit behavior, bankruptcy risk, and fraud risk.

German Credit Card Application Dataset (UCI MLR)

The German credit card application dataset comes from UCI machine learning databases. It contains 1000 instances with 24 predictor variables and 1 class variable (UCI). The 24 variables describe the status of existing checking account, credit history, education level, employment status, personal status, age, and so on. The class variable indicates whether an application is accepted or declined. Seventy percent of the instances are accepted applications and 30% are declined instances.

Australian Credit Card Application Dataset [87]

This dataset was provided by a large bank and concerns consumer credit card applications. It has 690 instances with 15 predicator variables plus 1 class variable. The class variable indicates whether an application is accepted or declined. 55.5% of the instances are accepted applications and 44.5% are declined instances.

USA Credit Cardholders' Behavior Dataset [157]

The dataset was from a major US bank and contains 6000 credit card data with 64 predictor variables plus 1 class variable. Each instance has a class label indicating its credit status: either good or bad. Eighty-four percent of the data are good accounts and 16% are bad accounts. Good indicates good status accounts and bad indicates accounts with late payments, delinquency, or bankruptcy. The predictor variables describe account balance, purchase, payment, cash advance, interest charges, date of last payment, times of cash advance, and account open date.

China Credit Cardholders' Behavior Dataset

This dataset was collected by a commercial bank in China and contains 5456 credit card data with 13 attributes. These attributes describe credit cardholders' daily balance, abnormal usage, limit usage rate, first time used, revoking pay, suspend pay, transactions detail, and personal information. Each record in the dataset has a class label denotes the status of a credit card account: either good or bad. There are 91.9% good accounts and 8.1% bad accounts.

Japanese Bankruptcy Dataset [158]

This set collects 37 bankrupt Japanese firms and 111 non-bankrupt Japanese firms from various sources during the post-deregulation period of 1989–1999. Final sample firms are ones traded in the First Section of Tokyo Stock Exchange, and their financial data are available from 2000 PACAP database for Japan compiled by the Pacific-Basin Capital Market (PACAP) Research Center at the University of Rhode Island. Each case has 13 predictor variables and 1 class variable (bankrupt or non-bankrupt). The predictor variables describe financial state and performance of firms.

Korean Bankruptcy Dataset [159]

This dataset collects bankrupt firms in Korea from 1997 to 2003 from public sources. It consists of 65 bankrupt and 130 non-bankrupt firms whose data are available and publicly trading firms in the Korean Stock Exchange. Each case has 13 predictor variables with one class variable (bankrupt or non-bankrupt).

Insurance Dataset [160]

The data was provided by an anonymous US corporation. Each record concerns about an insurance claim. The set has 18,875 instances with 103 variables. A binary class attribute indicates whether an instance is a normal claim or abnormal claim. There are 353 abnormal claims and 18,522 normal claims. The abnormal instances represent fraudulent or erroneous claims and were manually collected and verified.

9.3.4.4 Experimental Design

The calculation process of the performance score and the three MCDM methods were applied to the nine classifiers over the seven financial risk datasets. The experiment was carried out according to the following process:

Input: a financial risk related dataset.
Output: ranking of classification algorithms.
Step 1: understand business requirements, dataset structure and data mining task.
Step 2: prepare target datasets: select and transform relevant features; data cleaning; data integration. Communicate any findings during data preparation with domain experts.
Step 3: train and test multiple classification models in randomly sampled partitions (i.e., tenfold cross-validation) using Weka 3.6 [19].
Step 4: calculate the performance scores following the process discussed in section "Performance Score".
Step 5: evaluate classification algorithms using TOPSIS, PROMETHEE II, and VIKOR. The performance scores for each classifier obtained from Step 4 are used as inputs to the MCDM methods. All the MCDM methods are implemented using MATLAB.
Step 6: generate three separate tables of the final ranking of classification algorithms provided by each MCDM method.
Step 7: discuss the results with domain experts. Explore potential chance(s) from data mining results. Go back to Step 1 if new business questions are raised during the process.
END

Measures have different importance in financial risk prediction. For example, false negative (FN) is the number of positive or abnormal instances that is misclassified as normal class. Since positive instances are bankrupt, fraudulent or erroneous accounts in financial risk detection, a classifier with a high FN rate can cause huge lost to creditors. Thus FN measure should have higher importance in financial risk prediction than other measures, such as false positive measure [161]. Another important measure in financial risk prediction is AUC because it selects optimal models independently from the class distribution and the cost associated with each class.

Weights of each performance measure used in TOPISIS, PROMETHEE, and VIKOR are defined according to these findings from previous research. In this study, FN rate is not included because it equals to one minus TP rate. The importance of FN rate in financial risk prediction is then reflected in the weight of TP rate. The weights of the five performance measures are defined as: TP rate and AUC are set to 10 and other three measures (i.e., over-all accuracy, precision, and TN rate) are set to 1. The weights are normalized and the sum of all weights equal to 1.

9.3.4.5 Results and Discussion

The results of test set overall accuracy, precision, AUC, TP rate, and TN rate of all classifiers on the seven datasets are reported in Table 9.22. In the dataset column of Table 9.22, Australian indicates the Australian credit card application data; USA indicates the credit cardholders' behavior data from the United States; China refers

to the credit cardholders' behavior data collected from a Chinese bank; IN indicates the insurance data; German indicates the German credit card application data; and Japan and Korea indicate the Japanese and the Korean bankruptcy data, respectively. The nine classification methods were applied to each dataset using tenfold cross-validation. For each dataset, the best result of a specific performance measure is highlighted in boldface.

When the distribution of classes is highly skewed, as in the IN dataset (1.87% abnormal instances versus 98.13% normal cases), Naïve Bayes and Bayesian Network outperform other classifiers. Naïve Bayes has the highest TP rate (0.9065), which indicates that it captured 90.65% of the abnormal records, while Bayesian Network achieves a good TN rate (0.8291). Although SVM and RBF network got perfect overall accuracy (100%), they failed to identify any abnormal behavior (TP = 0 and FN = 1). For evenly distributed dataset, such as the Australian data, all classifiers have good over-all accuracy and AUC. For small datasets, such as the Japanese bankruptcy data, no classifier produces satisfactory results on AUC and TP rate. However, SVM and ensemble obtained good AUC and TP rate for the small size Korea bankruptcy dataset. For medium sized datasets, such as the credit cardholders' behavior datasets, linear logistic generates the best AUC, while Naïve Bayes and SVM produce the best TP rates. There is no classification algorithm which achieves the best results across all measures for a single dataset or has the best outcomes for a single performance measure across all datasets.

Based on the classification results presented in Table 9.22, the performance scores of all classifiers are calculated following the process discussed in Sect. 9.3.4.6 and the results are summarized in Table 9.23. For each performance measure, the best result generated by a classification algorithm is highlighted in boldface and italic. Since the performance scores are generated by conducting paired t tests with a significance level of 5% for all classifier pairs across all datasets, a classification algorithm with the highest performance score indicates that it performs significantly better than other classifiers for that specific performance measure over the seven datasets. Similar to the classification results reported in Table 9.22, no classifier has the highest performance scores for all five measures and classifiers with the best scores on some measures may perform poorly on other measures. For example, SVM achieves the best performance scores on overall accuracy and TN rate, but its scores on precision and AUC are quite low. Therefore the MCDM methods are introduced to provide a final ranking of classification algorithms.

The ranking of classifiers generated by TOPSIS, PROMETHEE II, and VIKOR is summarized in Tables 9.23, 9.24, 9.25, and 9.26, respectively. The results of TOPSIS and PROMETHEE are straightforward: the higher the ranking, the better the classifier. Linear logistic, Bayesian Network, and ensemble methods are the top-three ranked classifiers using the TOPSIS approach. The same set of classifiers is ranked as the top-three classifiers by the PROMETHEE II, however, the order of Bayesian Network and ensemble is reversed.

Since VIKOR provides compromised solutions, the ranking of classifiers needs to be determined by the Step 5 of the VIKOR algorithm.

Table 9.22 Classification results

Dataset	Algorithm	Overall accuracy	Precision	Area under ROC	True positive rate	True negative rate
Australian	Bayesian Network	0.8522	**0.8596**	0.9143	0.7980	0.8956
Australian	Naïve Bayes	0.7725	0.8571	0.8978	0.5863	**0.9217**
Australian	SVM	0.8551	0.7867	0.8622	**0.9251**	0.7990
Australian	Linear logistic	**0.8623**	0.8313	**0.9312**	0.8664	0.8590
Australian	K nearest neighbor	0.7942	0.7653	0.7922	0.7752	0.8094
Australian	C4.5	0.8348	0.8271	0.8346	0.7948	0.8668
Australian	RBF network	0.8304	0.8493	0.8995	0.7524	0.8930
Australian	RIPPER rule induction	0.8522	0.8213	0.8714	0.8534	0.8512
Australian	Ensemble	0.8551	0.8439	0.99	0.8274	0.8773
USA	Bayesian Network	0.7055	0.3366	0.8424	0.8656	0.6750
USA	Naïve Bayes	0.6933	0.3280	0.8395	**0.8740**	0.6589
USA	SVM	0.8372	0.4738	0.5632	0.1604	**0.9661**
USA	Linear logistic	**0.8532**	**0.5785**	**0.8539**	0.3031	0.9579
USA	K nearest neighbor	0.8028	0.3830	0.6327	0.3802	0.8833
USA	C4.5	0.8170	0.4156	0.6245	0.3542	0.9052
USA	RBF network	0.8400	0.0000	0.8256	0.0000	**1.0000**
USA	RIPPER rule induction	0.8443	0.5212	0.6380	0.3333	0.9417
USA	Ensemble	0.8382	0.4929	0.8432	0.3990	0.9218
China	Bayesian Network	0.9111	0.9805	0.9388	0.9216	0.7909
China	Naïve Bayes	0.8645	**0.9822**	0.9102	0.8684	**0.8205**
China	SVM	0.9417	0.9507	0.9359	**0.9878**	0.4159
China	Linear logistic	0.9426	0.9555	**0.9453**	0.9835	0.4773
China	K nearest neighbor	0.9263	0.9598	0.7505	0.9601	0.5409
China	C4.5	0.9443	0.9622	0.8593	0.9779	0.5614
China	RBF network	0.9247	0.9374	0.9113	0.9840	0.2477
China	RIPPER rule induction	0.9351	0.9576	0.7419	0.9727	0.5068
China	Ensemble	**0.9472**	0.9661	0.9229	0.9769	0.6091
IN	Bayesian Network	0.8261	0.0694	0.8361	0.6686	0.8291
IN	Naïve Bayes	0.3368	0.0250	0.7307	**0.9065**	0.3260
IN	SVM	0.9813	0.0000	0.5000	0.0000	**1.0000**
IN	Linear logistic	0.9809	0.0000	0.7546	0.0000	0.9996
IN	K nearest neighbor	0.9723	0.2300	0.5961	0.2040	0.9870
IN	C4.5	0.9786	0.3641	0.6656	0.1898	0.9937
IN	RBF network	0.9813	0.0000	0.7097	0.0000	**1.0000**
IN	RIPPER rule induction	0.9806	0.4444	0.5774	0.1586	0.9962
IN	Ensemble	**0.9817**	**0.5745**	**0.8443**	0.0765	0.9989
German	Bayesian Network	0.7250	0.5654	0.7410	0.3600	0.8814
German	Naïve Bayes	0.7550	0.6104	0.7888	**0.5067**	0.8614
German	SVM	**0.7740**	**0.6667**	0.6938	0.4933	**0.8943**
German	Linear logistic	0.7710	0.6578	0.7919	0.4933	0.8900

(continued)

Table 9.22 (continued)

Dataset	Algorithm	Overall accuracy	Precision	Area under ROC	True positive rate	True negative rate
German	K nearest neighbor	0.6690	0.4485	0.6064	0.4500	0.7629
German	C4.5	0.7190	0.5388	0.6607	0.4400	0.8386
German	RBF network	0.7400	0.5840	0.7520	0.4633	0.8586
German	RIPPER rule induction	0.7340	0.5720	0.6557	0.4500	0.8557
German	Ensemble	0.7620	0.6476	**0.7980**	0.4533	**0.8943**
Japan	Bayesian Network	0.7568	0.5135	0.7292	**0.5135**	0.8378
Japan	Naïve Bayes	0.7432	0.4857	0.7197	0.4595	0.8378
Japan	SVM	0.7500	0.0000	0.5000	0.0000	**1.0000**
Japan	Linear logistic	0.7770	0.5667	0.7290	0.4595	0.8829
Japan	K nearest neighbor	0.7770	0.5714	0.6595	0.4324	0.8919
Japan	C4.5	0.7162	0.4242	0.5270	0.3784	0.8288
Japan	RBF network	0.7162	0.3810	0.6533	0.2162	0.8829
Japan	RIPPER rule induction	0.7365	0.4706	0.6193	0.4324	0.8378
Japan	Ensemble	**0.7905**	**0.6667**	**0.7424**	0.3243	0.9459
Korea	Bayesian Network	0.8667	**0.8095**	0.8773	0.7846	**0.9077**
Korea	Naïve Bayes	0.7744	0.7059	0.8168	0.5538	0.8846
Korea	SVM	**0.8718**	0.7778	0.8682	**0.8615**	0.8769
Korea	Linear logistic	0.8462	0.7692	0.8749	0.7692	0.8846
Korea	K nearest neighbor	0.8154	0.7101	0.7993	0.7538	0.8462
Korea	C4.5	0.8359	0.7797	0.7948	0.7077	0.9000
Korea	RBF network	0.8256	0.7460	0.8033	0.7231	0.8769
Korea	RIPPER rule induction	0.8667	0.7826	0.8577	0.8308	0.8846
Korea	Ensemble	0.8564	0.7681	**0.9026**	0.8154	0.8769

Table 9.23 Performance scores of classifiers

Classifier/measure	Overall accuracy	Precision	AUC	TP rate	TN rate
Bayesian Network	−19	8	23	5	−4
Naïve Bayes	−28	8	24	2	3
SVM	22	−20	−27	1	13
Linear logistic	22	6	32	4	6
K nearest neighbor	−26	−13	−36	−2	−23
C4.5	−4	5	−26	1	−7
RBF network	4	−22	3	−22	5
RIPPER rule induction	10	9	−23	8	−4
Ensemble	19	19	30	3	11

The classifier with the first position in the ranking list by Q cannot be proposed as the compromise solution because the condition (a) $Q(a'') - Q(a') \geq 1(J-1)$ is not satisfied. Therefore, alternatives a', a'', *and* a''' are proposed as compromise solutions, since a is the maximum number of alternative determined by the relation

Table 9.24 Results of the TOPSIS approach

Classifier	TOPSIS
Linear logistic	0.891293
Bayesian Network	0.874166
Ensemble	0.866155
Naïve Bayes	0.815243
RIPPER rule induction	0.638725
C4.5	0.544801
SVM	0.542099
K nearest neighbor	0.457113
RBF network	0.283217

Table 9.25 Results of the PROMETHEE II approach

Classifier	PROMETHEE II
Linear logistic	0.711957
Ensemble	0.532609
Bayesian Network	0.413043
RIPPER rule induction	0.353261
Naïve Bayes	0.190217
C4.5	−0.43478
SVM	−0.44022
RBF network	−0.46739
K nearest neighbor	−0.8587

Table 9.26 Results of the VIKOR approach

Classifier	VIKOR Q	VIKOR S	VIKOR R
Linear logistic	0.00055	0.080211	0.057971
Ensemble	0.027268	0.090276	0.072464
Bayesian Network	0.070517	0.168871	0.057545
Naïve Bayes	0.137489	0.205328	0.086957
RIPPER rule induction	0.628727	0.393233	0.351662
SVM	0.76261	0.520044	0.377238
C4.5	0.765376	0.533903	0.370844
RBF network	0.971288	0.688997	0.434783
K nearest neighbor	0.979134	0.698862	0.434783

$Q(a^M) - Q(a') < 1(J-1)$. That is, the rankings of linear logistic, Bayesian Network, and ensemble methods are in closeness according to VIKOR.

The results of Tables 9.23, 9.24, 9.25, and 9.26 indicate that TOPSIS, PROMETHEE II, and VIKOR provide similar top-ranked classification algorithms for financial risk prediction.

9.3.4.6 Knowledge-Rich Financial Risk Management Process

Even though classification has become a crucial tool in financial risk prediction, most studies focus on developing algorithms or improving existing algorithms that can identify suspicious patterns and have not paid enough attention to the involvement of end users and the actionability of the classification results [83]. This is mainly due to two reasons: (1) the difficulty in accessing real-life financial risk data and (2) limited access to domain experts and background information. The lack of interaction between industry practitioners and academic researchers makes it hard to discover financial risks or opportunities and hence weaken the value that classification methods may bring to financial risk detection.

In an attempt to improve the usefulness of classification results and increase the probability of identifying unusual chances in financial risk analysis, this section proposes a knowledge-rich financial risk management process (Fig. 9.17). Chance discovery (CD) is defined as "the awareness of a chance and the explanation of its significance" [162]. Ohsawa and Fukuda [162] suggested three keys to chance discovery: communicating the significance of an event; enhancing user's awareness of an event's utility using mental imagery; and revealing the causalities of rare events using data mining methods. Figure 9.17 combines the knowledge discovery in database (KDD) process model [113], the chance discovery process [162], and the CRISP-DM process model [163]. It emphasizes three keys to chance discovery and knowledge-rich data mining: users, communication and data mining techniques. Users refer to domain experts and decision makers. Domain experts are knowledgeable of the field information, data collection procedures and meaning of variables. With the assistance of data miners, domain experts can gain insights of financial risk data from different aspects and potentially observe new chances. To turn the identified knowledge into financial or strategic advantages, decision makers, who understand the operational and strategic goals of a company, are required to provide feedbacks on the importance of the potential new chances and determine what actions should be taken. Moving back and forth between steps is always required. The cyclical nature is illustrated by the outer circle of the chance discovery process in Fig. 9.17.

This study chose the insurance data as an example to examine the proposed process. The business objective(s) of this project was to develop classification model(s) to assist human inspection of suspicious claims. After the business objective has been deter-mined, the dataset was preprocessed for classification task. During the preparation stage, two issues were brought up by the data miners: first, there are several attributes with missing values for all the instances in the dataset; second, the definitions of four attributes are conflicting. From the data miner's point of view, an attribute with completely missing values is useless in data mining tasks and should be simply removed. But from the domain expert's perspective, this is an unusual situation and represents a potential chance for operational improvement. Any attribute stored in the database was designed to capture relevant information and an attribute with complete missing value may indicate errors in the data

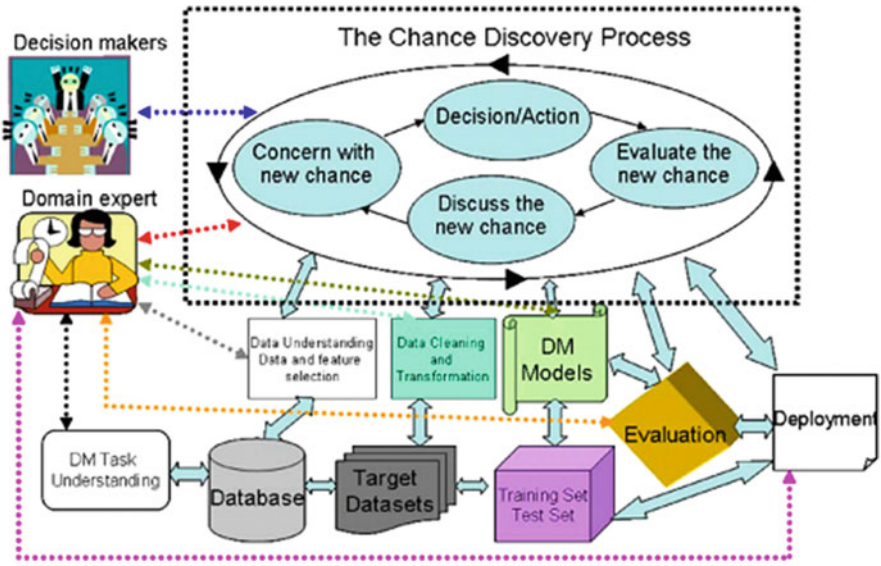

Fig. 9.17 Knowledge-rich financial risk management process

collecting process. After careful examination, domain experts found out the reasons for missing values and took corrective actions.

Then nine classifiers were applied to the insurance data using tenfold cross-validation. A classifier with low false negative (FN) rate can minimize insurance fraud risk because FN rate denotes the percentage of high-risk claims that were misclassified as normal claims. For this dataset, Naïve Bayes has the lowest FN rate ($1 - 0.9065 = 0.00935$). Because it achieves the lowest FN rate and provides classification results that can be easily understood and used by domain experts, Naïve Bayes was chosen as the decision classifier. This model can be used to predict high-risk claim; narrow down the size suspicious records; and accelerate the claim-handling process. The classification results obtained from data mining step can further be analyzed to provide additional insights about the data. For instance, if some general features of high- or low-risk claims can be identified from the classification results, it may help the insurance company to establish profiles for each type of claims, which potentially may bring profits to the company.

To summarize, the empirical study demonstrates that introducing the concept of chance discovery into the KDD process can help users choose the most appropriate classifier, promote the awareness of previously unnoticed chances, and increase the usefulness of data mining results.

References

1. Wu, W., Xu, Z., Kou, G., Shi, Y.: Decision-making support for the evaluation of clustering algorithms based on MCDM. Complexity. **2020**, 9602526 (2020)
2. Kou, G., Lu, Y., Peng, Y., Shi, Y.: Evaluation of classification algorithms using MCDM and rank correlation. Int. J. Inf. Technol. Decis. Making. **11**(01), 197–225 (2012)
3. Tang, H., Shi, Y., Dong, P.: Public blockchain evaluation using entropy and topsis. Expert Syst. Appl. **117**, 204–210 (2019)
4. Kou, G., Peng, Y., Shi, Y., Wu, W.: Classifier evaluation for software defect prediction. Stud. Informatics Contr. **21**(2), 118 (2012)
5. Peng, Y., Kou, G., Wang, G., Wu, W., Shi, Y.: Ensemble of software defect predictors: an ahp-based evaluation method. Int. J. Inf. Technol. Decis. Making. **10**(01), 187–206 (2011)
6. Shi, Y., Yang, Z., Yan, H., Tian, X.: Delivery efficiency and supplier performance evaluation in China's e-retailing industry. J. Syst. Sci. Complex. **30**(2), 392–410 (2017)
7. Zhou, X., Jiang, W., Shi, Y., Tian, Y.: Credit risk evaluation with kernel-based affine subspace nearest points learning method. Expert Syst. Appl. **38**(4), 4272–4279 (2011)
8. Kou, G., Wu, W., Zhao, Y., Peng, Y., Yaw, N.E., Shi, Y.: A dynamic assessment method for urban eco-environmental quality evaluation. J. Multi-Criteria Decis. Anal. **18**(1–2), 23–38 (2011)
9. Peng, Y., Wang, G., Kou, G., Shi, Y.: An empirical study of classification algorithm evaluation for financial risk prediction. Appl. Soft Comput. **11**(2), 2906–2915 (2011)
10. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
11. Chen, L., Xu, Z., Wang, H., Liu, S.: An ordered clustering algorithm based on k-means and the promethee method. Int. J. Mach. Learn. Cybern. **9**(6), 917–926 (2018)
12. Wu, J., Chen, J., Xiong, H., Xie, M.: External validation measures for k-means clustering: a data distribution perspective. Expert Syst. Appl. **36**(3), 6050–6061 (2009)
13. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
14. Paul, A.K., Shill, P.C.: New automatic fuzzy relational clustering algorithms using multi-objective nsga-II. Inf. Sci. **448**, 112–133 (2018)
15. Han, J., Kamber, M., Pei, J.: Data Mining Concepts and Techniques, 3rd edn. Morgan Kaufmann, Burlington, MA (2011)
16. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Practical Machine Learning Tools and Techniques, p. 578. Morgan Kaufmann, Burlington, MA (2005)
17. Hochbaum, D.S., Shmoys, D.B.: A best possible heuristic for the kcenter problem. Math. Oper. Res. **10**(2), 180–184 (1985)
18. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The kdd process for extracting useful knowledge from volumes of data. Commun. ACM. **39**(11), 27–34 (1996)
19. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
20. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. Ser. B Methodological. **39**(1), 1–22 (1977)
21. Kumar, M., et al.: An optimized farthest first clustering algorithm. In: 2013 Nirma University International Conference on Engineering (NUiCONE), pp. 1–5. IEEE, New York (2013)
22. Dasgupta, S., Long, P.M.: Performance guarantees for hierarchical clustering. J. Comput. Syst. Sci. **70**(4), 555–569 (2005)
23. Hamdan, S., Cheaitou, A.: Supplier selection and order allocation with green criteria: an MCDM and multi-objective optimization approach. Comput. Oper. Res. **81**, 282–304 (2017)
24. Peng, Y., Shi, Y.: Multiple criteria decision making and operations research. Ann. Oper. Res. **197**(1), 1–4 (2012)
25. Wang, Z., Xu, Z., Liu, S., Tang, J.: A netting clustering analysis method under intuitionistic fuzzy environment. Appl. Soft Comput. **11**(8), 5558–5564 (2011)

26. Yang, J.L., Chiu, H.N., Tzeng, G.H., Yeh, R.H.: Vendor selection by integrated fuzzy MCDM techniques with independent and interdependent relationships. Inf. Sci. **178**(21), 4166–4183 (2008)
27. He, J., Zhang, Y., Shi, Y., Huang, G.: Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring. IEEE Trans. Knowl. Data Eng. **22**(6), 826–838 (2010)
28. Shi, Y., Zhang, L., Tian, Y., Li, X.: Intelligent Knowledge: A Study Beyond Data Mining. Springer, New York (2015)
29. Schneider, K.R.: On the Nature of Cities. Josey-Bass Publishers, San Francisco (1979)
30. Fishburn, P.: Additive Utilities with Incomplete Product Set: Applications to Priorities and Assignments. ORSA Publication, Baltimore, MD (1967)
31. Triantaphyllou, E.: Multi-criteria decision making methods. In: Multicriteria decision making methods: a comparative study, pp. 5–21. Springer, New York (2000)
32. Triantaphyllou, E., Baig, K.: The impact of aggregating benefit and cost criteria in four mcda methods. IEEE Trans. Eng. Manag. **52**(2), 213–226 (2005)
33. Wu, W., Kou, G., Peng, Y.: Group decision-making using improved multi-criteria decision making methods for credit risk analysis. Filomat. **30**(15), 4135–4150 (2016)
34. Jahanshahloo, G.R., Lotfi, F.H., Izadikhah, M.: Extension of the topsis method for decision-making problems with fuzzy data. Appl. Math. Comput. **181**(2), 1544–1551 (2006)
35. Cheng, S., Hwang, C.: Fuzzy Multiple Attribute Decision Making: Methods and Applications Lecture Notes in Economics and Mathematical Systems. Springer, New York (1992)
36. Opricovic, S., Tzeng, G.H.: Compromise solution by MCDM methods: a comparative analysis of vikor and topsis. Eur. J. Oper. Res. **156**(2), 445–455 (2004)
37. Brans, J.P., Mareschal, B.: Promethee methods. In: Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 163–186. Springer, New York (2005)
38. Hermans, C.M., Erickson, J.D.: Multicriteria decision analysis: overview and implications for environmental decision making. Ecol. Econ. Sustain. Watershed Manage. **7**, 213–228 (2007)
39. Kuang, H., Kilgour, D.M., Hipel, K.W.: Grey-based promethee II with application to evaluation of source water protection strategies. Inf. Sci. **294**, 376–389 (2015)
40. Ergu, D., Kou, G., Peng, Y., Shi, Y.: A simple method to improve the consistency ratio of the pair-wise comparison matrix in anp. Eur. J. Oper. Res. **213**(1), 246–259 (2011)
41. Kou, G., Wu, W.: Multi-criteria decision analysis for emergency medical service assessment. Ann. Oper. Res. **223**(1), 239–254 (2014)
42. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques (2000)
43. Zhao, Y., Karypis, G., Fayyad, U.: Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Disc. **10**(2), 141–168 (2005)
44. Mirkin, B.: Mathematical Classification and Clustering, vol. 11. Springer Science & Business Media, Berlin (1996)
45. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
46. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. Sci. Nat. **44**, 223–270 (1908)
47. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. J. Am. Stat. Assoc. **78**(383), 553–569 (1983)
48. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
49. Badescu, D., Boc, A., Diallo, A.B., Makarenkov, V.: Detecting genomic regions associated with a disease using variability functions and adjusted rand index. BMC Bioinformatics. **12**, 1–10 (2011)
50. Saaty, R.W.: The analytic hierarchy process—what it is and how it is used. Math. Modell. **9**(3–5), 161–176 (1987)
51. Takahashi, I.: Ahp applied to binary and ternary comparisons. J. Oper. Res. Soc. Jpn. **33**(3), 199–206 (1990)

52. Tyagi, S., Agrawal, S., Yang, K., Ying, H.: An extended fuzzyahp approach to rank the influences of socialization-externalization-combination-internalization modes on the development phase. Appl. Soft Comput. **52**, 505–518 (2017)
53. Wu, W., Kou, G., Peng, Y., Ergu, D.: Improved ahp group decision making for investment strategy selection. Technol. Econ. Dev. Econ. **18**(2), 299–316 (2012)
54. Yu, C.S.: A gp-ahp method for solving group decision-making fuzzy ahp problems. Comput. Oper. Res. **29**(14), 1969–2001 (2002)
55. eCosta, C.A.B., Vansnick, J.C.: A critical analysis of the eigenvalue method used to derive priorities in ahp. Eur. J. Oper. Res. **187**(3), 1422–1428 (2008)
56. Ertay, T., Ruan, D., Tuzkaya, U.R.: Integrating data envelopment analysis and analytic hierarchy for the facility layout design in manufacturing systems. Inf. Sci. **176**(3), 237–262 (2006)
57. Amiri, M.P.: Project selection for oil-fields development by using the ahp and fuzzy topsis methods. Expert Syst. Appl. **37**(9), 6218–6224 (2010)
58. Domingos, P.: Toward knowledge-rich data mining. Data Min. Knowl. Disc. **15**(1), 21–28 (2007)
59. Roy, B., et al.: ELECTRE III: un algorithme de classement basé sur une représentation floue des préférences en présence de critères multiples. Cahires du CERO. **20**(1), 3–24 (1978)
60. Figueira, J.R., Mousseau, V., Roy, B.: Electre methods. In: Multiple Criteria Decision Analysis, pp. 155–185. Springer, New York (2016)
61. Milani, A.S., Shanian, A., El-Lahham, C.: Using different electre methods in strategic planning in the presence of human behavioral resistance. J. Appl. Math. Decis. Sci. **2006** (2006)
62. Roy, B., Bouyssou, D.: Aide multicritère à la décision: méthodes et cas. Economica, Paris (1993)
63. Opricovic, S.: Multicriteria optimization of civil engineering systems. Faculty of Civil Engineering, Belgrade 2(1), 5–21 (1998)
64. Opricovic, S., Tzeng, G.H.: Multicriteria planning of post-earthquake sustainable reconstruction. Comput. Aided Civ. Inf. Eng. **17**(3), 211–220 (2002)
65. Mao, N., Song, M., Deng, S.: Application of topsis method in evaluating the effects of supply vane angle of a task/ambient air conditioning system on energy utilization and thermal comfort. Appl. Energy. **180**, 536–545 (2016)
66. Ding, X., Chong, X., Bao, Z., Xue, Y., Zhang, S.: Fuzzy comprehensive assessment method based on the entropy weight method and its application in the water environmental safety evaluation of the Heshangshan drinking water source area, three gorges reservoir area, China. Water. **9**(5), 329 (2017)
67. Behzadian, M., Otaghsara, S.K., Yazdani, M., Ignatius, J.: A state-of the-art survey of topsis applications. Expert Syst. Appl. **39**(17), 13051–13069 (2012)
68. Myrtveit, I., Stensrud, E., Shepperd, M.: Reliability and validity in comparative studies of software prediction models. IEEE Trans. Softw. Eng. **31**(5), 380–391 (2005)
69. Kou, G., Shi, Y., Wang, S.: Multiple criteria decision making and decision support systems-guest editor's introduction. Decis. Support. Syst. **51**(2), 247–249 (2011)
70. Chapman, M., Callis, P., Jackson, W.: Metrics data program, NASA IV and V facility (2004). http://mdp.ivv.nasa.gov
71. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach. Learn. **40**(2), 139–157 (2000)
72. Dietterich, T.G.: Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems, pp. 1–15. Springer, New York (2000)
73. Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. In: ICML, vol. 96, pp. 148–156. Citeseer (1996)
74. Ting, K.M., Zheng, Z.: A study of adaboost with naive Bayesian classifiers: weakness and improvement. Comput. Intell. **19**(2), 186–200 (2003)
75. Wilson, T., Wiebe, J., Hwa, R.: Recognizing strong and weak opinion clauses. Comput. Intell. **22**(2), 73–99 (2006)

76. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. J. Artif. Intell. Res. **11**, 169–198 (1999)
77. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach. Learn. **36**(1), 105–139 (1999)
78. Breiman, L., et al.: Heuristics of instability and stabilization in model selection. Ann. Stat. **24**(6), 2350–2383 (1996)
79. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
80. Schapire, R.E.: The strength of weak learnability. Mach. Learn. **5**(2), 197–227 (1990)
81. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of online learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
82. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
83. Peng, Y., Kou, G., Shi, Y., Chen, Z.: A descriptive framework for the field of data mining and knowledge discovery. Int. J. Inf. Technol. Decis. Making. **7**(04), 639–682 (2008)
84. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10), 993–1001 (1990)
85. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth International Group, Belmont, CA (1984) Google Scholar
86. Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Kdd, vol. 96, pp. 202–207 (1996)
87. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Elsevier, Amsterdam (2014)
88. Dong, M.: The improvement of the method of topsis in synthetic queme & sensitivity analysis. Syst. Eng. Theory Pract. **5** (1993)
89. Bishop, C.M., et al.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
90. Platt, J.: Sequential minimal optimization: a fast algorithm for training support vector machines (1998)
91. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Science & Business Media, Berlin (2013)
92. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Mach. Learn. **29**(2), 103–130 (1997)
93. Weiss, S.M., Kulikowski, C.A.: Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Morgan Kaufmann Publishers Inc, Burlington, MA (1991)
94. Dasarathy, B.V.: Nearest Neighbor (nn) Norms: Nn Pattern Classification Techniques. IEEE Computer Society Tutorial, Los Alamitos (1991)
95. Kohavi, R.: The power of decision tables. In: European Conference on Machine Learning, pp. 174–189. Springer, New York (1995)
96. Cohen, W.W.: Fast effective rule induction. In: Machine Learning Proceedings 1995, pp. 115–123. Elsevier, Amsterdam (1995)
97. Saaty, T.L.: How to make a decision: the analytic hierarchy process. Eur. J. Oper. Res. **48**(1), 9–26 (1990)
98. Saaty, T.L., Sagir, M.: Extending the measurement of tangibles to intangibles. Int. J. Inf. Technol. Decis. Making. **8**(01), 7–27 (2009)
99. Saaty, T.L.: Decision making with the analytic hierarchy process. Int. J. Serv. Sci. **1**(1), 83–98 (2008)
100. Saaty, T.L.: A scaling method for priorities in hierarchical structures. J. Math. Psychol. **15**(3), 234–281 (1977)
101. Zahedi, F.: The analytic hierarchy process—a survey of the method and its applications. Interfaces. **16**(4), 96–108 (1986)
102. Despotis, D.K., Derpanis, D.: A min–max goal programming approach to priority derivation in ahp with interval judgements. Int. J. Inf. Technol. Decis. Making. **7**(01), 175–182 (2008)
103. Ho, W.: Integrated analytic hierarchy process and its applications—a literature review. Eur. J. Oper. Res. **186**(1), 211–228 (2008)

104. Li, H.L., Ma, L.C.: Ranking decision alternatives by integrated dea, ahp and gower plot techniques. Int. J. Inf. Technol. Decis. Making. **7**(02), 241–258 (2008)
105. Sugihara, K., Tanaka, H.: Interval evaluations in the analytic hierarchy process by possibility analysis. Comput. Intell. **17**(3), 567–579 (2001)
106. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recogn. Lett. **30**(1), 27–38 (2009)
107. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.R.: Model-based evaluation of clustering validation measures. Pattern Recogn. **40**(3), 807–824 (2007)
108. Chamberlin, E.H.: Product heterogeneity and public policy. Am. Econ. Rev. **40**(2), 85–92 (1950)
109. Lancaster, K.: The economics of product variety: a survey. Mark. Sci. **9**(3), 189–206 (1990)
110. Gaur, V., Kesavan, S.: The effects of firm size and sales growth rate on inventory turnover performance in the US retail sector. In: Retail Supply Chain Management, pp. 25–52. Springer, New York (2015)
111. Fitzsimons, G.J.: Consumer response to stockouts. J. Consum. Res. **27**(2), 249–266 (2000)
112. Slywotzky, A.J.: The age of the choiceboard. Harv. Bus. Rev. **78**(1), 40–40 (2000)
113. Fisher, M.: Yihaodian: The no. 1 store. The Wharton School Case Study (2012)
114. Zhou, X., Jiang, W., Tian, Y., Zhang, P., Nie, G., Shi, Y.: A new kernel-based classification algorithm. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 1094–1099. IEEE, New York (2009)
115. Zhu, X., Yang, X., Liu, T.: On mechanism for eco-environmental quality of Jiangsu province. Econ. Geogr. **24**(4), 473–476 (2004)
116. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.: A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Trans. Neural Netw. **11**(1), 124–136 (2000)
117. Lee, D.D., Seung, H.S.: Unsupervised learning by convex and conic coding. In: Advances in Neural Information Processing Systems, pp. 515–521. Princeton University, Princeton, NJ (1997)
118. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152 (1992)
119. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. Electron. Comput. **3**, 326–334 (1965)
120. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468), pp. 41–48. IEEE, New York (1999)
121. Schölkopf, B., Smola, A., Müller, K.R.: Kernel principal component analysis. In: International Conference on Artificial Neural Networks, pp. 583–588. Springer, New York (1997)
122. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. Tech. Rep. (1990)
123. Osowski, S., Brudzewski, K.: Fuzzy self-organizing hybrid neural network for gas analysis system. IEEE Trans. Instrum. Meas. **49**(2), 424–428 (2000)
124. Osowski, S., Siwek, K., Markiewicz, T.: MLP and SVM networks-a comparative study. In: Proceedings of the 6th Nordic Signal Processing Symposium, 2004. NORSIG 2004, pp. 37–40. IEEE, New York (2004)
125. Xu, Q., Pei, W., Yang, L., He, Z.: Support vector machine tree based on feature selection. In: International Conference on Neural Information Processing, pp. 856–863. Springer, New York (2006)
126. Fahlman, S.: Cmu benchmark collection for neural net learning algorithms. In: Machine-Readable Data Repository. School of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA (1993)
127. Li, A., Shi, Y., He, J.: Mclp-based methods for improving "bad" catching rate in credit cardholder behavior analysis. Appl. Soft Comput. **8**(3), 1259–1265 (2008)
128. He, J., Liu, X., Shi, Y., Xu, W., Yan, N.: Classifications of credit cardholder behavior by using fuzzy linear programming. Int. J. Inf. Technol. Decis. Making. **3**(04), 633–650 (2004)

129. Rester F, V.H.A.: Ecology and iaanniug in melnopolitan area a sensitivity model (1980)
130. Alberti, M., Waddell, P.: An integrated urban development and ecological simulation model. Integr. Assess. **1**(3), 215–227 (2000)
131. Van Kamp, I., Leidelmeijer, K., Marsman, G., De Hollander, A.: Urban environmental quality and human well-being: towards a conceptual framework and demarcation of concepts; a literature study. Landsc. Urban Plan. **65**(1-2), 5–18 (2003)
132. Yu, L., Yin, W.: Application research of analytic hierarchy process (arp) in urban ecotope quality evaluation. Sichuan Environ. **21**(4), 38–40 (2002)
133. Zeleny, M.: MCDM: Past Decade and Future Trends: A Source Book of Multiple Criteria Decision Making, vol. 1. Jai Press, London (1984)
134. Anderson Jr., W.T., Cox III, E.P., Fulcher, D.G.: Bank selection decisions and market segmentation: determinant attribute analysis reveals convenience- and service-oriented bank customers. J. Mark. **40**(1), 40–45 (1976)
135. Myers, J.H., Alpert, M.I.: Determinant buying attitudes: meaning and measurement. J. Mark. **32**(4 part 1), 13–20 (1968)
136. Starr, M.K.: A discussion of some normative criteria for decisionmaking under uncertainty. Ind. Manage. Rev. **8**(1), 71 (1966)
137. Isaacs, H.H.: Sensitivity of decisions to probability estimation errors. Oper. Res. **11**(4), 536–552 (1963)
138. Fishburn, P.C.: Analysis of decisions with incomplete knowledge of probabilities. Oper. Res. **13**(2), 217–237 (1965)
139. Evans, J.R.: Sensitivity analysis in decision theory. Decis. Sci. **15**(2), 239–247 (1984)
140. Simon French, D.R.I.: Partial information and sensitivity analysis in multi-objective decision making. In: Lockett, A.G., Islei, G. (eds.) Improving Decision Making in Organisations Lecture Notes in Economics and Mathematical Systems. Springer, Berlin (1989)
141. Masuda, T.: Hierarchical sensitivity analysis of priority used in analytic hierarchy process. Int. J. Syst. Sci. **21**(2), 415–427 (1990)
142. Armacost, R.L., Hosseini, J.C.: Identification of determinant attributes using the analytic hierarchy process. J. Acad. Mark. Sci. **22**(4), 383–392 (1994)
143. Ringuest, J.L.: Lp-metric sensitivity analysis for single and multiattribute decision analysis. Eur. J. Oper. Res. **98**(3), 563–570 (1997)
144. Dale, V.H., Beyeler, S.C.: Challenges in the development and use of ecological indicators. Ecol. Indic. **1**(1), 3–10 (2001)
145. Gu, C.C.G.: Study on index system of eco-city assessment. Natural Ecol. Conserv. **8**, 24–38 (2001)
146. Whitford, V., Ennos, A.R., Handley, J.F.: "City form and natural process"—indicators for the ecological performance of urban areas and their application to Merseyside, UK. Landsc. Urban Plan. **57**(2), 91–103 (2001)
147. Nakhaeizadeh, G., Schnabl, A.: Development of multi-criteria metrics for evaluation of data mining algorithms. In: KDD, pp. 37–42 (1997)
148. Kao, P., Hocheng, H.: Optimization of electrochemical polishing of stainless steel by grey relational analysis. J. Mater. Process. Technol. **140**(1–3), 255–259 (2003)
149. Tan, O.-S., Seng, A.S.-H.: Cognitive Modifiability in Learning and Assessment. Cengage Learning, Singapore (2008)
150. Ren, R.W.H.: Multi-Variable Data Analysis. National Defense Industry Press, Beijing (1997)
151. Guo, Y., Yao, Y., Yi, P.: Method and application of dynamic comprehensive evaluation. Syst. Eng. Theory Pract. **27**(10), 154–158 (2007)
152. Zuo, J.: Discussion multi-criteria decision making method of sensitivity analysis. Syst. Eng. Theory Pract. **7**(3), 1–11 (1987)
153. Ali, S., Smith, K.A.: On learning algorithm selection for classification. Appl. Soft Comput. **6**(2), 119–138 (2006)
154. Brazdil, P.B., Soares, C.: A comparison of ranking methods for classification algorithm selection. In: European Conference on Machine Learning, pp. 63–75. Springer, New York (2000)

155. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**(1–2), 1–39
156. Le Cessie, S., Van Houwelingen, J.C.: Ridge estimators in logistic regression. J. R. Stat. Soc. Ser. C Appl. Stat. **41**(1), 191–201 (1992)
157. Kou, G., Peng, Y., Shi, Y., Wise, M., Xu, W.: Discovering credit cardholders' behavior by multiple criteria linear programming. Ann. Oper. Res. **135**(1), 261–274 (2005)
158. Kwak, W., Shi, Y., Eldridge, S.W., Kou, G.: Bankruptcy prediction for Japanese firms: using multiple criteria linear programming data mining approach. Int. J. Bus. Intell. Data Mining. **1**(4), 401–416 (2006)
159. Kwak, W., Shi, Y., Kou, G.: Bankruptcy prediction for Korean firms after the 1997 financial crisis: using a multiple criteria linear programming data mining approach. Rev. Quant. Finan. Acc. **38**(4), 441–453 (2012)
160. Peng, Y., Kou, G., Sabatka, A., Matza, J., Chen, Z., Khazanchi, D., Shi, Y.: Application of classification methods to individual disability income insurance fraud detection. In: International Conference on Computational Science, pp. 852–858. Springer, New York (2007)
161. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. J. Oper. Res. Soc. **54**(6), 627–635 (2003)
162. Ohsawa, Y., Fukuda, H.: Chance discovery by stimulated groups of people. Application to understanding consumption of rare food. J. Contingencies Crisis Manage. **10**(3), 129–138 (2002)
163. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R., et al.: Crisp-dm 1.0: Step-by-Step Data Mining Guide, vol. 9, p. 13. SPSS Inc, Chicago, IL (2000)