International Journal of Information Technology & Decision Making
Vol. 11, No. 1 (2012) 197−225

DOI: 10.1142/S0219622012500095

EVALUATION OF CLASSIFICATION ALGORITHMS USING MCDM AND RANK CORRELATION*

GANG KOU†, YANQUN LU†, YI PENG†,¶ and YONG SHI‡,§

†*School of Management and Economics*
University of Electronic Science and Technology of China
Chengdu, 610054, P. R. China

‡*College of Information Science & Technology*
University of Nebraska at Omaha
Omaha, NE 68182, USA

§*CAS Research Center on Fictitious Economy*
and Data Sciences, Beijing 100080, China
¶*pengyicd@gmail.com*

Classification algorithm selection is an important issue in many disciplines. Since it normally involves more than one criterion, the task of algorithm selection can be modeled as multiple criteria decision making (MCDM) problems. Different MCDM methods evaluate classifiers from different aspects and thus they may produce divergent rankings of classifiers. The goal of this paper is to propose an approach to resolve disagreements among MCDM methods based on Spearman's rank correlation coefficient. Five MCDM methods are examined using 17 classification algorithms and 10 performance criteria over 11 public-domain binary classification datasets in the experimental study. The rankings of classifiers are quite different at first. After applying the proposed approach, the differences among MCDM rankings are largely reduced. The experimental results prove that the proposed approach can resolve conflicting MCDM rankings and reach an agreement among different MCDM methods.

Keywords: Multi-criteria decision making (MCDM); classification; Spearman's rank correlation coefficient; TOPSIS; ELECTRE; grey relational analysis; VIKOR; PROMETHEE.

1. Introduction

As one of the most important tasks in data mining and knowledge discovery (DMKD),[31] classification has been extensively studied and various algorithms have been developed over the years. Classification algorithms evaluation and selection is an active research area in the fields of machine learning, statistics, computer science, and DMKD. Rice[37] formalized algorithm selection as abstract models with the problem space, the feature space, the criteria space, the algorithm space, and the

*Authors contributed equally to this work and are alphabetically ordered by their last names.

performance measures. Nakhaeizadeh and Schnabl[26] suggested multi-criteria-based metrics to compare classification algorithms. Smith-Miles[42] treated the algorithm selection problem as a learning task and presented a unified framework that combines the cross-disciplinary developments in the algorithm selection problem. Peng *et al.*[35] applied multiple criteria decision making (MCDM) methods to rank classification algorithms and validated the approaches using software defect datasets.

The evaluation of classification algorithms normally involves two or more conflicting criteria, such as accuracy, Type-I error, Type-II error, AUC, and computation time. Thus it can be modeled as a MCDM problem.[41] Many MCDM methods have been developed and applied in a wide variety of applications.[7,8,14,19,20,23,32,48] MCDM methods rank alternatives using different approaches. Applying various MCDM methods to a sorting problem is beneficial because the ranking agreed by several MCDM methods is more trustful than one generated by single MCDM method. Though some studies shown that MCDM methods provide similar rankings of alternatives,[33,35] there are situations where different MCDM methods produce conflicting rankings. How to reconcile these differences becomes an important issue and has not been fully investigated.[34]

This paper 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. An experimental study, which chooses a wide selection of classifiers and performance measures, is conducted to validate the proposed approach over 11 public-domain binary classification datasets.

The rest of this paper is organized as follows. In Sec. 2, the proposed approach, including the selected MCDM methods and Spearman's rank correlation coefficient, is described. Section 3 presents the datasets, classification algorithms, performance measures, and the design of the experimental study. The results are discussed in Sec. 4. The final section summarizes the papers.

2. Research Methodology

Many MCDM methods have been developed to rank alternatives in different ways. While the rankings of alternatives provided by MCDM methods may in agreement sometimes, there are situations where different MCDM methods generate very different rankings. This work proposes to use Spearman's rank correlation coefficient to generate the final ranking that resolves or reduces differences among MCDM rankings. The proposed approach uses several MCDM methods to rank classifiers first; then applies Spearman's rank correlation coefficient to determine weights for MCDM methods to get secondary rankings of classifiers. This section describes the five MCDM methods used in the study and explains the details of the proposed approach.

2.1. *MCDM methods*

Ranking classification algorithms normally need to examine multiple criteria, such as accuracy, AUC, F-measure, and kappa statistic. Therefore, the task of algorithm selection can be modeled as MCDM problems. This section gives an overview of the five selected MCDM methods (i.e., TOPSIS, ELECTRE III, grey relational analysis, VIKOR, and PROMETHEE II).

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

Hwang and Yoon[15] proposed the technique for order preference by similarity to ideal solution (TOPSIS) method to rank alternatives over multiple criteria. It finds the best alternatives by minimizing the distance to the ideal solution and maximizing the distance to the nadir or negative-ideal solution.[27]

A number of extensions and variations of TOPSIS have been developed over the years. The following TOPSIS procedure adopted from Opricovic and Tzeng[29] and Olson[27] is used in the empirical study:

Step 1. Calculate the normalized decision matrix. The normalized value r_{ij} is calculated as

$$r_{ij} = x_{ij} \Big/ \sqrt{\sum_{j=1}^{J} x_{ij}^2}, \quad j = 1, \ldots, J; \;\; i = 1, \ldots, n,$$

where J and n denote the number of alternatives and the number of criteria, respectively. For alternative A_j, the performance measure of the ith criterion C_i is represented by x_{ij}.

Step 2. Develop a set of weights w_i for each criterion and calculate the weighted normalized decision matrix. The weighted normalized value v_{ij} is calculated as

$$v_{ij} = w_i x_{ij}, \quad j = 1, \ldots, J; \;\; i = 1, \ldots, n,$$

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

Step 3. Find the ideal alternative solution S^+, which is calculated as

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

where I' is associated with benefit criteria and I'' is associated with cost criteria.

Step 4. Find the negative-ideal alternative solution S^-, which is calculated as

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

Step 5. Calculate the separation measures, using the n-dimensional Euclidean distance. The separation of each alternative from the ideal solution is calculated as

$$D_j^+ = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^+)^2}, \quad j = 1, \ldots, J.$$

The separation of each alternative from the negative-ideal solution is calculated as

$$D_j^- = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^-)^2}, \quad j = 1, \ldots, J.$$

Step 6. Calculate a ratio R_j^+ that measures the relative closeness to the ideal solution and is calculated as

$$R_j^+ = D_j^- / (D_j^+ + D_j^-), \quad j = 1, \ldots, J.$$

Step 7. Rank alternatives by maximizing the ratio R_j^+.

ELimination and Choice Expressing REality (ELECTRE)

ELECTRE stands for *ELimination Et Choix Traduisant la REalité* (ELimination and Choice Expressing the REality) and was first proposed by Roy[38] 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.[11] ELECTRE III is chosen in this paper because it is appropriate for the sorting problem. The procedure can be summarized as follows[24,39,40]:

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

Grey Relational Analysis (GRA)

Grey relational analysis (GRA) is a part of grey theory, which has been proposed to handle imprecise and incomplete information in grey systems,[5] and has been proven to be suitable for selecting a best alternative. The main procedure of GRA includes generating the grey relation, defining the reference sequence, and calculating the grey relational coefficient and grey relational grade (Kuo *et al.*, 2008). The first step normalizes performance values for every alternative to get a comparability sequence. The second step defines an ideal target sequence. The third step compares the comparability sequences with the ideal sequence and the last step calculates the grey relational grade to rank alternatives. This study adopts the procedure provided by Kuo *et al.*[22]:

Step 1. Generate the grey relation:

Suppose there are m alternatives and n attributes, the ith alternative is defined as $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{in}\}$. y_{ij} is the performance value of attribute j of alternative i. Using one of the following three equations, the values of Y_i can be translated into the comparability sequence $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$:

$$x_{ij} = \frac{y_{ij} - \text{Min}\{y_{ij}, i = 1, 2, \ldots, m\}}{\text{Max}\{y_{ij}, i = 1, 2, \ldots, m\} - \text{Min}\{y_{ij}, i = 1, 2, \ldots, m\}} \quad \text{for } i = 1, 2, \ldots, m; \;\; j = 1, 2, \ldots, n, \tag{2.1}$$

$$x_{ij} = \frac{\text{Max}\{y_{ij}, i = 1, 2, \ldots, m\} - y_{ij}}{\text{Max}\{y_{ij}, i = 1, 2, \ldots, m\} - \text{Min}\{y_{ij}, i = 1, 2, \ldots, m\}} \quad \text{for } i = 1, 2, \ldots, m; \;\; j = 1, 2, \ldots, n, \tag{2.2}$$

$$x_{ij} = 1 - \frac{|y_{ij} - y_j^*| y_{ij} - \text{Min}\{y_{ij}, i = 1, 2, \ldots, m\}}{\text{Max}\{\text{Max}\{y_{ij}, i = 1, 2, \ldots, m\} - y_{ij}^*, y_{ij}^* - \text{Min}\{y_{ij}, i = 1, 2, \ldots, m\}\}} \quad \text{for } i = 1, 2, \ldots, m; \;\; j = 1, 2, \ldots, n, \tag{2.3}$$

where y_j^* represents the desired value. Equation (2.1) is used for the-larger-the-better attributes, Eq. (2.2) is used for the-smaller-the-better attributes, and Eq. (2.3) is used for the-closer-the-desired-value-y_j^*-the-better.

Step 2. Define the reference sequence:

The performance values calculated from Step 1 are ranged from 0 to 1. The ideal target sequence, also called the reference sequence, is defined as $X_0 = \{x_{01}, x_{02}, \ldots, x_{0n}\} = \{1, 1, \ldots, 1\}$. The objective of GRA is to find an alternative that has the closest comparability sequence to the reference sequence.

Step 3. Calculate the grey relational coefficient using the following equation:

$$\gamma(x_{0j}, x_{ij}) = \frac{V_{\min} + \zeta V_{\max}}{V_{ij} + \zeta V_{\max}}, \quad i = 1, 2, \ldots, m; \;\; j = 1, \ldots, n,$$

where ζ is the distinguishing coefficient, $\zeta \in [0,1]$. The distinguishing coefficient is used to expand or compress the range of the grey relational coefficient and it is defined as 0.5 in this study. x_{0j} is the reference sequence and defined as $(x_{01}, x_{01}, \ldots, x_{0J}) = (1, 1, \ldots, 1)$. $\gamma(x_{0j}, x_{ij})$ is the grey relational coefficient between x_{0j} and x_{ij}, and

$$V_{ij} = |x_{0j} - x_{ij}|,$$
$$V_{\min} = \text{Min}\{\, V_{ij}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\},$$
$$V_{\max} = \text{Max}\{\, V_{ij}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}.$$

Step 4. Calculate the grey relational grade using the following equation:

$$\Gamma(X_0, X_i) = \sum_{j=1}^{n} w_j \gamma(x_{0j}, x_{ij}), \quad \text{for } i = 1, \ldots, m,$$

where w_j is the weight of attribute j and $\sum_{j=1}^{n} w_i = 1$. The grey relational grade indicates the closeness between the ideal sequence and the comparability sequence. An alternative with the highest grey relational grade is the best choice.

VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR)

VIKOR was proposed by Opricovic[28] and Opricovic and Tzeng[30] 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.[28,49] This paper uses the following VIKOR algorithm provided by Opricovic and Tzeng[29] in the experiment:

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

$$f_i^* = \left\{ \begin{matrix} \max\limits_j f_{ij}, & \text{for benefit criteria} \\ \min\limits_j f_{ij}, & \text{for cost criteria} \end{matrix} \right\}, \quad j = 1, 2, \ldots, J,$$

$$f_i^- = \left\{ \begin{matrix} \min\limits_j f_{ij}, & \text{for benefit criteria} \\ \max\limits_j f_{ij}, & \text{for cost criteria} \end{matrix} \right\}, \quad j = 1, 2, \ldots, J,$$

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 a_j.

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

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

where w_i is the weight of ith criteria, S_j and R_j are used to formulate ranking measure.

Step 3. Compute the values $Q_j, j = 1, 2, \ldots, J$, by the relations

$$Q_j = v(S_j - S^*)/(S^- - S^*) + (1 - v)(R_j - R^*)/(R^- - R^*),$$

$$S^* = \min_j S_j, \quad S^- = \max_j S_j,$$

$$R^* = \min_j R_j, \quad R^- = \max_j R_j,$$

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' 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.

Preference Ranking Organisation METHod for Enrichment of Evaluations (PROMETHEE)

Brans[2] proposed the PROMETHEE I and PROMETHEE II in 1982. The PROMETHEE methods use pairwise comparisons and outranking relationships to choose the best alternatives. The final selection is based on the positive and negative preference flows of each alternative. The positive preference flow indicates how an alternative is outranking all the other alternatives and the negative preference flow indicates how an alternative is outranked by all the other alternatives.[3] PROMETHEE I obtains partial ranking because it does not compare conflicting actions.[4] PROMETHEE II ranks alternatives according to the net flow which equals to the balance of the positive and the negative preference flows.[43] An alternative with a higher net flow is better.[3] Since the purpose of this paper is to rank classification algorithms, PROMETHEE II is used in the experimental study.

The following PROMETHEE II procedure described by Brans and Mareschal[3] is used in the experimental study:

Step 1. Define aggregated preference indices.

Let $a, b \in A$, and let:

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

where A is a finite set of possible alternatives $\{a_1, a_2, \ldots, a_n\}$, k represents the number of evaluation criteria and w_j is the weight of each criterion. Arbitrary numbers for the weights can be assigned by the DM. The weights are then normalized to ensure that $\sum_{j=1}^{k} w_j = 1$. $\pi(a,b)$ indicates how a is preferred to b and $\pi(b,a)$ indicates how b is preferred to a. $P_j(a,b)$ and $P_j(b,a)$ are the preference functions for alternatives a and b.

Step 2. Calculate $\pi(a,b)$ and $\pi(b,a)$ for each pair of alternatives of A.

Step 3. Define the positive and the negative outranking flow as follows:

The positive outranking flow:

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

The positive outranking flow:

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

Step 4. Compute the net outranking flow for each alternative as follows:

$$\phi(a) = \phi^+(a) - \phi^-(a).$$

When $\phi(a) > 0$, a is more outranking all the alternatives on all the evaluation criteria. When $\phi(a) < 0$, a is more outranked.

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:

$$\rho_{ki} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \tag{2.4}$$

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

$$\rho_k = \frac{1}{q-1} \sum_{i=1, i \neq k}^{q} \rho_{ki}, \quad k = 1, 2, \ldots, q, \tag{2.5}$$

where q is the number of MCDM methods. The larger the ρ_k value, the more important the MCDM method is. Normalized ρ_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.

3. Experimental Study

The experiment chooses a wide selection of classification algorithms, datasets, and performance measures to test the proposed approach. The following subsections describe the data sources, classification algorithms, performance measures, and the experimental design.

3.1. *Data sources*

Eleven binary classification datasets are selected to evaluate the performance of classifiers. Table 1 displays the basic information of these datasets.

All datasets are publicly available at the UCI machine learning repository (http://archive.ics.uci.edu/ml/). These datasets are selected from a variety of disciplines, including life science, physical science, social science, and business, and represent a variation of data structures. Their sizes range from 208 to 19,020; the number of attributes varies from 4 to 73; and class distributions change from 93.6/6.4 to 52/48. By averaging the performances of classifiers across the 11 datasets, this study attempts to get a comprehensive evaluation of classification algorithms.

Table 1. Summary of UCI datasets.

Dataset	No. of Instances	No. of Attributes	Area	Goods/Bads
Adult	48,842	14	Social	75.9/34.1
Breast-cancer (original)	699	10	Life	65.5/34.5
Breast-cancer (diagnostic)	569	32	Life	58/42
Glass	214	10	Physical	52/48
Ionosphere	351	34	Physical	64.1/35.9
Magic gamma telescope	19,020	10	Physical	64.8/35.2
Mammographic	961	5	Life	53.7/46.3
Ozone level detection	2536	73	Physical	93.6/6.4
Pima Indians diabetes	768	8	Life	65.1/34.9
Sonar	208	60	Physical	53.4/46.6
Transfusion	748	4	Business	76.2/23.8

3.2. *Classification algorithms*

Classification algorithms used in the experiment represent five categories: Trees, rules, Bayesian classifiers, lazy classifiers, and miscellaneous classifiers. All classifiers are implemented in WEKA.[45]

Trees category includes C4.5 decision tree, decision stump, random tree, and REP tree. C4.5 is a decision tree algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner[36] and is a commonly used benchmark classifier. An algorithm for generating grafted C4.5 decision trees is also included.[46] Decision stump is a simple one-level decision tree and often used with ensemble techniques.[18] Random tree builds trees by choosing a test based on number of random features at each node.[45] REP tree uses information gain to build a decision tree and reduced-error pruning (REP) to prune it.[45]

Rules category includes the zero-attribute rule, conjunctive rule, decision table, one-attribute rule, and part. The zero-attribute rule, or ZeroR, predicts the mode and is often used to measure the performance of other algorithms. The conjunctive rule learns a single best rule to predict a class value and assign uncovered test instances to the default class value of the uncovered training instances.[45] Decision table builds a decision table majority classifier by selecting the right feature subsets. Instances not covered by a decision table can be determined by the nearest-neighbor method. The one-attribute rule, or OneR, finds association rules using just one attribute.[17] Part generates partial C4.5 decision trees and obtains rules from the partial trees.[10]

Bayesian classifiers category includes Bayesian network[47] and Naïve Bayes.[6] Both classifiers 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. Two forms of Naïve Bayes classifier are included in this study: the standard probabilistic Naïve Bayes classifier and an incremental Naïve Bayes classifier.[45]

IB1, a basic instance-based learner provided by WEKA, is chosen to represent lazy classifiers. An unknown instance is assigned to the same class as the training instance that is the closest to it measured by Euclidean distance.

In addition to the above mentioned classifiers, Adaboost M1, hyperpipe, and voting feature interval (VFI) are included. Adaboost M1 algorithm changes weights of training instances in each iteration to force learning algorithms to put more emphasis on previously misclassified instances.[12] Hyperpipe classifier constructs a hyperpipe for each category and assigns a new instance to the category that most contains the instance.[45] VFI classifier creates a set of feature intervals and records class counts for each interval. The class having the highest vote is the predicted class.[13]

3.3. *Performance measures*

Widely used performance measures in classification are accuracy, true positive rate, true negative rate, mean absolute error (MAE), precision, F-measure, the area under receiver operating characteristic (AUC), kappa statistic, and computation time. Definitions of these measures are provided as follows.

- Overall accuracy: Accuracy is the percentage of correctly classified instances.

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

  where TN, TP, FN, and FP stand for true negative, true positive, false negative, and false positive, respectively.
- True Positive (TP): TP is the number of correctly classified positive instances. TP rate is also called sensitivity measure.

$$\text{True Positive rate/Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

- False Positive (FP): FP is the number of positive instances that is misclassified as negative class. FP rate is called Type-I error.

$$\text{False Positive rate/Type-I error} = \frac{\text{FP}}{\text{FP} + \text{TN}}.$$

- True Negative (TN): TN is the number of correctly classified negative instances. TN rate is also called specificity measure.

$$\text{True Negative rate/Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}.$$

- False Negative (FN): FN is the number of negative instances that is misclassified as positive class. FN rate is also called Type-II error.

$$\text{False Negative rate/Type-II error} = \frac{\text{FN}}{\text{FN} + \text{TP}}.$$

- Mean absolute error (MAE): This measures how much the predictions deviate from the true probability. $P(i, j)$ is the estimated probability of i module to be of class j taking values in [0, 1].[9]

$$\text{MAE} = \frac{\sum_{j=1}^{c} \sum_{i=1}^{m} |f(i,j) - P(i,j)|}{m \cdot c}.$$

- Precision: This is the number of classified fault-prone modules that actually are fault-prone modules.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

- F-measure: It is the harmonic mean of precision and recall. F-measure has been widely used in information retrieval.[1]

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

- AUC: ROC stands for receiver operating characteristic, which shows the tradeoff between TP rate and FP rate. AUC represents the accuracy of a classifier. The larger the area, the better the classifier.
- Kappa statistic (KapS): This is a classifier performance measure that estimates the similarity between the members of an ensemble in multi-classifiers systems.[21]

$$\text{KapS} = \frac{P(A) - P(E)}{1 - P(E)}.$$

P(A) is the accuracy of the classifier and P(E) is the probability that agreement among classifiers is due to chance.

$$\text{P(E)} = \frac{\sum_{k=1}^{c} \left(\left[\sum_{j=1}^{c} \sum_{i=1}^{m} f(i,k) C(i,j) \right] \cdot \left[\sum_{j=1}^{c} \sum_{i=1}^{m} f(i,j) C(i,k) \right] \right)}{m^2},$$

where m is the number of modules and c is the number of classes. $f(i, j)$ is the actual probability of i module to be of class j. $\sum_{i=1}^{m} f(i, j)$ is the number of modules of class j. Given threshold θ, $C_\theta(i, j)$ is 1 if j is the predicted class for i obtained from $P(i, j)$; otherwise it is 0.[9]

- Training time: The time needed to train a classification algorithm or ensemble method.
- Test time: The time needed to test a classification algorithm or ensemble method.

3.4. *Experimental design*

The experiment was carried out according to the following process:

Input: 11 binary classification datasets

Output: Rankings of classifiers

Step 1. Prepare target datasets: Select and transform relevant features; data cleaning; data integration.

Step 2. Train and test the selected classification algorithms on a randomly sampled partitions (i.e., 10-fold cross-validation) using WEKA 3.7.[16]

Step 3. Evaluate classification algorithms using TOPSIS, GRA, VIKOR, PROMETHEE II, and ELECTRE III. MCDM methods are implemented using MATLAB 7.0.[25]

Step 4. Generate the first ranking of classification algorithms provided by each MCDM method. If there are disagreements among MCDM methods, go to Step 5; otherwise, end the process.

Step 5. Solve the formulas (2.4) and (2.5) to get the weights of MCDM methods.

Step 6. Recalculate the rankings of classifiers using the MCDM methods with the ranking produced in Step 4 and the normalized weights obtained in Step 5 as inputs.

END

Classification algorithm selection problem involves both benefit and cost criteria. A criterion is called the *benefit* because the higher a classification algorithm scores in terms of the corresponding criterion, the better the algorithm is. Seven of the 10 performance metrics used in this study are benefit criteria. They are accuracy, kappa statistic, TP rate, TN rate, precision, F-measure, and AUC. On the other hand, a criterion is called the *cost* because the higher a classification algorithm scores in terms of the corresponding criterion, the worse the algorithm is.[44] Three performance metrics used in this study are cost criteria: MAE, training, and testing time. For TOPISIS, ELECTRE III, and PROMETHEE II, criteria or performance measures are assigned equal weights.

The proposed approach is tested on two individual UCI datasets and all 11 datasets. Classification algorithms are first applied to two UCI datasets (i.e., adult and magic gamma telescope) and evaluated using the proposed ranking approach. Then the performances of these classifiers on the 11 UCI datasets are averaged and ranked using the proposed ranking approach. The classification results and the first and secondary rankings of classifiers are reported in the following section.

4. Results

4.1. *Classification results*

4.1.1. *Magic gamma telescope dataset*

Table 2 shows the testing results of the magic gamma telescope dataset using the selected classifiers. Each classifier is measured by the ten performance metrics defined in Sec. 3.3. The best result of each performance measure is highlighted in boldface. Adaboost M1 algorithm performances well on this dataset and there is no classification algorithm achieves the best results across all measures.

Table 2. Classification results of magic gamma telescope dataset.

Classifiers	Accuracy	Kappa	MAE	TP	TN	Precision	F-measure	AUC	Train	Testing
Bayes net	0.7892	0.4510	0.2380	0.9483	0.4501	0.7862	0.8597	0.8550	0.5365	0.0139
Naïve Bayes	0.7444	0.3303	0.2625	0.9197	0.3704	0.7570	0.8304	0.7568	0.1563	0.0203
Incremental Naïve Bayes	0.7444	0.3303	0.2625	0.9197	0.3704	0.7570	0.8304	0.7568	0.1688	0.0234
IB1	0.8178	0.5692	0.1822	0.8897	0.6644	0.8498	0.8693	0.7770	0.0109	23.0875
AdaBoost M1	**0.8694**	**0.6915**	**0.1289**	0.9273	0.7460	0.8862	**0.9063**	**0.9133**	57.1016	0.0500
HyperPipes	0.6953	0.0624	0.4995	0.9989	0.0477	0.6911	0.8170	0.5238	0.0125	0.0047
VFI	0.6956	0.0640	0.4960	0.9988	0.0491	0.6913	0.8171	0.5259	0.0313	0.0156
Conjunctive rule	0.7269	0.4440	0.3370	0.6848	**0.8167**	**0.8888**	0.7733	0.7508	23.2016	0.0031
Decision table	0.8337	0.5917	0.2441	0.9334	0.6211	0.8402	0.8843	0.8782	4.7797	0.0109
OneR	0.7181	0.3248	0.2819	0.8279	0.4840	0.7739	0.7999	0.6560	0.1547	0.0047
PART	0.8545	0.6519	0.1952	0.9269	0.7001	0.8689	0.8965	0.8995	7.0563	0.0094
ZeroR	0.6808	0.0000	0.4347	**1.0000**	0.0000	0.6808	0.8101	0.5000	**0.0047**	**0.0016**
Decision stump	0.7170	0.4112	0.3556	0.6989	0.7557	0.8592	0.7708	0.7273	0.2531	0.0016
C4.5	0.8566	0.6579	0.1894	0.9261	0.7083	0.8714	0.8978	0.8667	2.7203	0.0063
Grafted C4.5	0.8591	0.6631	0.1874	0.9298	0.7083	0.8718	0.8998	0.8679	3.8156	0.0031
Random tree	0.8259	0.6003	0.1741	0.8703	0.7313	0.8736	0.8719	0.8008	0.9109	0.0094
REP tree	0.8563	0.6558	0.1948	0.9292	0.7008	0.8689	0.8980	0.8762	0.7813	0.0031

4.1.2. *Adult dataset*

Table 3 summarizes the testing results of the adult dataset using the classifiers. The best result of each performance measure is highlighted in boldface. Similar to the magic gamma telescope data, there is no classification algorithm that has the best results on all measures, and the performances of classifiers on this dataset are more divergent than on the magic gamma telescope data. Adaboost M1 algorithm, which performances significantly better than other classifiers on the magic gamma telescope data, only achieves the best MAE on the adult dataset.

4.1.3. *Average of 11 datasets*

Table 4 summarizes the testing results of the adult dataset using the selected 17 classifiers. The best result of each performance measure is highlighted in boldface. Similar to the previous two datasets, best performance measures are achieved by difference classification algorithms.

4.2. *MCDM rankings*

Following the proposed approach and the experimental design, the five MCDM methods are applied to rank 17 classification algorithms using the classification results of the magic gamma telescope, adult, and the average of all 11 datasets.

4.2.1. *Magic gamma telescope dataset*

The ranking scores of classifiers generated by TOPSIS, GRA, VIKOR, PROMETHEE II, and ELECTRE III for the magic gamma telescope dataset are summarized in Table 5. Each MCDM method provides a value and ranking for each classifier based on their performances on the testing data.

For the magic gamma telescope data, five MCDM methods all rank Adaboost M1 algorithm as the best classifier. However, their rankings for IB1 are quite different. While TOPSIS ranks IB1 as the second best classifier for this datasets, VIKOR, PROMETHEE, and ELECTRE rank it as the 8th classifier. In fact, the five MCDM methods agree on only three classifiers: Adaboost M1, HyperPipes, and ZeroR.

4.2.2. *Adult data*

The ranking scores of classifiers generated by TOPSIS, GRA, VIKOR, PROMETHEE II, and ELECTRE III for the adult dataset are summarized in Table 6. The five MCDM methods agree on no classifier over the adult dataset. The ranking results can be expected since the performances of the classifiers on the adult data are divergent (see Table 3).

4.2.3. *Average of 11 datasets*

Table 7 reports the rankings of classifiers generated by the five MCDM methods based on their average classification results on 11 datasets. Similar to the adult dataset, the five MCDM methods cannot reach an agreement on any classifier.

Table 3. Classification results of adult dataset.

Classifiers	Accuracy	Kappa	MAE	TP	TN	Precision	F-measure	AUC	Train	Testing
Bayes net	0.8380	0.5944	0.1766	0.8504	0.7986	0.9301	0.8885	**0.9163**	0.6078	0.0203
Naïve Bayes	0.8348	0.5019	0.1730	0.9343	0.5211	0.8602	0.8957	0.8923	0.2422	0.0313
Incremental Naïve Bayes	0.8348	0.5019	0.1730	0.9343	0.5211	0.8602	0.8957	0.8923	0.2547	0.0344
IB1	0.7936	0.4299	0.2064	0.8686	0.5571	0.8608	0.8647	0.7129	0.0328	98.4328
AdaBoost M1	0.8356	0.5426	**0.1617**	0.8996	0.6340	0.8857	0.8926	0.8788	117.2766	0.0703
HyperPipes	0.7641	0.0304	0.4993	**1.0000**	0.0203	0.7629	0.8655	0.5305	0.0313	0.0078
VFI	0.7561	0.4598	0.4457	0.7302	**0.8375**	**0.9341**	0.8196	0.8715	0.0672	0.0344
Conjunctive rule	0.7592	0.0000	0.3656	**1.0000**	0.0000	0.7592	0.8631	0.5000	3.0234	0.0078
Decision table	0.8572	0.5671	0.2070	0.9514	0.5603	0.8722	0.9100	0.8995	9.9828	0.0156
OneR	0.8084	0.2839	0.1916	0.9985	0.2093	0.7992	0.8878	0.6039	0.2766	0.0078
PART	0.8503	0.5752	0.1828	0.9183	0.6360	0.8884	0.9031	0.8656	105.7125	0.0984
ZeroR	0.7592	0.0000	0.3656	**1.0000**	0.0000	0.7592	0.8631	0.5000	**0.0109**	0.0094
Decision stump	0.7592	0.0000	0.2933	**1.0000**	0.0000	0.7592	0.8631	0.7592	0.3094	**0.0047**
C4.5	0.8622	0.6001	0.1926	0.9358	0.6299	0.8886	0.9116	0.8891	7.2891	0.0109
Grafted C4.5	**0.8625**	**0.6011**	0.1922	0.9362	0.6303	0.8887	**0.9118**	0.8890	11.4063	0.0109
Random tree	0.8105	0.4804	0.1922	0.8765	0.6027	0.8743	0.8754	0.7431	0.5375	0.0063
REP tree	0.8481	0.5582	0.2057	0.9278	0.5971	0.8790	0.9027	0.8709	0.8375	0.0094

Table 4. Average classification results of all datasets.

Classifiers	Accuracy	Kappa	MAE	TP	TN	Precision	F-measure	AUC	Train	Testing
Bayes net	0.8470	0.6274	0.1723	0.8289	**0.8308**	0.8471	0.8333	**0.8967**	0.1251	0.0041
Naïve Bayes	0.8093	0.5430	0.1973	0.7951	0.7799	0.8085	0.7868	0.8732	0.0449	0.0072
Incremental Naïve Bayes	0.8093	0.5430	0.1973	0.7951	0.7799	0.8085	0.7868	0.8732	0.0479	0.0085
IB1	0.8416	0.5874	0.1530	0.8239	0.7664	0.8202	0.8193	0.7951	0.0054	11.3416
AdaBoost M1	0.8641	0.6382	**0.1439**	0.8307	0.8120	0.8480	0.8344	0.8857	16.1057	0.0115
HyperPipes	0.6909	0.3141	0.4683	**0.9717**	0.3411	0.6792	0.7729	0.7019	0.0048	0.0016
VFI	0.7417	0.4762	0.3860	0.7972	0.7190	0.7683	0.7456	0.7907	0.0115	0.0061
Conjunctive rule	0.7810	0.3887	0.2787	0.7282	0.7084	0.7628	0.7322	0.7198	2.4281	0.0013
Decision table	0.8576	0.5651	0.2086	0.7986	0.7543	0.7828	0.7875	0.8517	1.9513	0.0028
OneR	0.8068	0.4333	0.1921	0.7617	0.6570	0.7764	0.7519	0.7094	0.0469	0.0014
PART	**0.8746**	**0.6573**	0.1543	0.8573	0.7948	**0.8566**	**0.8518**	0.8719	10.4043	0.0099
ZeroR	0.6450	0	0.3979	0.5455	0.4545	0.3869	0.4500	0.5000	**0.0014**	0.0010
Decision stump	0.7913	0.4066	0.2664	0.7481	0.7050	0.7799	0.7532	0.7889	0.0614	**0.0007**
C4.5	0.8654	0.6345	0.1646	0.8436	0.7825	0.8430	0.8404	0.8516	0.9801	0.0016
Grafted C4.5	0.8666	0.6354	0.1636	0.8427	0.7826	0.8441	0.8404	0.8509	1.4746	0.0013
Random tree	0.8255	0.5388	0.1734	0.8134	0.7272	0.8021	0.8045	0.8009	0.1466	0.0017
REP tree	0.8453	0.5548	0.2007	0.8184	0.7264	0.8215	0.8128	0.8197	0.1560	0.0011

Table 5. Rankings of MCDM methods for magic gamma telescope data.

Classifiers	TOPSIS		GRA		VIKOR		PROMETHEE II		ELECTRE III	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Bayes net	0.5265	9	5.5843	9	0.6113	10	0.75	9	11.1942	9
Naïve Bayes	0.4256	12	4.7899	13	0.6278	11	−1.875	11	−27.5446	12
Incremental Naïve Bayes	0.4256	11	4.7899	12	0.6040	9	−2.625	12	−27.1662	10
IB1	0.6600	2	6.7878	6	0.5868	8	0.875	8	35.9843	8
AdaBoost M1	0.7382	1	8.8461	1	0.0000	1	8.625	1	101.1193	1
HyperPipes	0.2761	16	4.1691	16	0.9857	16	−6.125	16	−98.4829	16
VFI	0.2767	15	4.1735	15	0.8651	15	−4.375	14	−77.8855	15
Conjunctive rule	0.4578	10	5.4732	10	0.7994	12	−0.75	10	−27.2394	11
Decision table	0.6068	7	6.4750	8	0.5549	3	2.875	6	60.1665	5
OneR	0.3610	14	4.4232	14	0.8458	13	−5.25	15	−77.642	14
PART	0.6467	3	7.3049	3	0.5285	2	4	4	75.6119	2
ZeroR	0.2696	17	4.1239	17	1.0000	17	−7.25	17	−121.073	17
Decision stump	0.4036	13	4.9035	11	0.8549	14	−4	13	−58.1242	13
C4.5	0.6379	5	7.2553	4	0.5586	4	4.125	3	66.5695	3
Grafted C4.5	0.6425	4	7.3411	2	0.5771	5	5	2	63.8372	4
Random tree	0.5968	8	6.5079	7	0.5851	6	2.875	7	43.5262	7
REP tree	0.6339	6	7.2392	5	0.5851	7	3.125	5	57.1483	6

Table 6. Rankings of MCDM methods for adult data.

Classifiers	TOPSIS		GRA		VIKOR		PROMETHEE II		ELECTRE III	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Bayes net	0.6137	6	7.2591	5	0.4691	5	4	5	48.6014	1
Naïve Bayes	0.5864	9	6.4345	9	0.4678	4	1.875	7	31.6551	9
Incremental Naïve Bayes	0.5864	8	6.4345	8	0.4678	3	0.875	8	32.0072	8
IB1	0.5543	10	5.8467	10	0.5661	9	−2.75	13	−22.94	11
AdaBoost M1	0.6801	2	7.2773	4	0.1670	2	4.125	4	60.4784	3
HyperPipes	0.2872	17	4.2222	17	1.0000	17	−4.625	15	−88.3356	16
VFI	0.4344	13	5.5427	12	0.8248	13	−2.25	11	−27.5881	12
Conjunctive rule	0.2997	15	4.2848	15	0.9848	15	−4.125	14	−83.5514	15
Decision table	0.6260	5	7.0922	6	0.5127	8	3.625	6	57.0849	5
OneR	0.4547	12	5.3386	13	0.7806	12	−2.375	12	−33.7779	13
PART	0.6951	1	7.3911	3	0.0000	1	4.875	2	72.9711	1
ZeroR	0.2990	16	4.2790	16	0.9870	16	−6.75	17	−105.551	17
Decision stump	0.3556	14	4.6251	14	0.9161	14	−5.75	16	−81.2669	14
C4.5	0.6367	4	7.3992	2	0.4930	7	4.75	3	59.7748	4
Grafted C4.5	0.6409	3	7.4269	1	0.4891	6	5.625	1	63.0099	2
Random tree	0.5250	11	5.5949	11	0.6966	11	−1.75	10	−17.8663	10
REP tree	0.6069	7	6.6787	7	0.5903	10	0.625	9	35.2947	7

Table 7. Rankings of MCDM methods for all datasets.

Classifiers	TOPSIS		GRA		VIKOR		PROMETHEE II		ELECTRE III	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Bayes net	0.8884	3	9.0336	1	0.0247	3	4.625	2	46.0865	1
Naïve Bayes	0.8194	6	7.9796	7	0.1398	7	0.625	8	21.8364	8
Incremental Naïve Bayes	0.8193	7	7.9790	8	0.1398	8	−0.75	11	21.8346	9
IB1	0.7017	13	7.6511	11	0.6095	14	2	6	13.9034	11
AdaBoost M1	0.7286	10	8.5554	5	0.5633	13	4.375	5	29.8199	5
HyperPipes	0.5390	16	6.3320	16	0.7704	16	−3.5	14	−65.2184	16
VFI	0.6665	14	6.7763	14	0.4853	12	−4	15	−28.7941	12
Conjunctive rule	0.6661	15	6.4191	15	0.3738	12	−5.875	17	−56.0399	15
Decision table	0.8240	5	7.8190	10	0.1382	6	−0.375	10	25.909	7
OneR	0.7175	11	7.0047	12	0.2726	9	−3.25	12	−29.6082	13
PART	0.8033	9	8.7858	3	0.2775	10	5.25	1	35.8156	4
ZeroR	0.3507	17	4.7839	17	1.0000	17	−5.75	16	−119.8643	17
Decision stump	0.7141	12	6.9303	13	0.2998	11	−3.375	13	−32.141	14
C4.5	0.8922	1	8.8054	2	0.0078	1	4.625	2	44.2543	3
Grafted C4.5	0.8900	2	8.7759	4	0.0109	2	4.625	2	44.4001	2
Random tree	0.8188	8	7.8354	9	0.1152	5	−0.375	9	20.6171	10
REP tree	0.8334	4	7.9911	6	0.0963	4	1.125	7	27.189	6

Results of the three tests indicate that different MCDM methods may have strong disagreements. Therefore, the next step applies Spearman's rank correlation coefficient to generate secondary rankings in an attempt to resolve the inconsistency.

4.3. *Secondary MCDM rankings*

The goal of the secondary ranking is to determine a set of weights for MCDM methods by solving the formulas (2.4) and (2.5). Before the computation, the ranking scores of the MCDM methods need to be standardized. The ranking scores of TOPSIS, GRA, PROMETHEE II, and ELECTRE III are standardized using (x-min)/(max-min); and the Q ranking scores of VIKOR are standardized using (max-x)/(max-min). The standardized MCDM ranking scores for the magic gamma telescope, adult, and the 11 datasets are presented in Tables 8–10, respectively.

4.3.1. *Magic gamma telescope dataset*

The standardized ranking scores are used to calculate the weights and normalized weights of the MCDM methods. The results are presented in Table 11.

Based on the standardized ranking scores and the normalized weights of the MCDM methods, the secondary rankings of classifiers on the magic gamma telescope dataset are computed and summarized in Table 12. The five MCDM methods provide the same rankings for 11 classifiers. Rankings of TOPSIS and GRA agree on all the classifiers, and differ from VIKOR and PROMETHEE II on only two classifiers. The degrees of disagreements on the rankings of classifiers among MCDM methods are also reduced.

Table 8. Standardized MCDM ranking scores for magic gamma telescope dataset.

Classifiers	TOPSIS	GRA	VIKOR	PROMETHEE II	ELECTRE III
Bayes net	0.5482	0.3093	0.3887	0.5039	0.5953
Naïve Bayes	0.3328	0.141	0.3722	0.3386	0.4209
Incremental Naïve Bayes	0.3329	0.141	0.396	0.2913	0.4226
IB1	0.833	0.5641	0.4132	0.5118	0.7069
AdaBoost M1	1	1	1	1	1
HyperPipes	0.0138	0.0096	0.0143	0.0709	0.1017
VFI	0.0153	0.0105	0.1349	0.1811	0.1944
Conjunctive rule	0.4018	0.2857	0.2006	0.4094	0.4223
Decision table	0.7197	0.4979	0.4451	0.6378	0.8157
OneR	0.195	0.0634	0.1542	0.126	0.1955
PART	0.8048	0.6736	0.4715	0.7087	0.8852
ZeroR	0	0	0	0	0
Decision stump	0.2859	0.1651	0.1451	0.2047	0.2833
C4.5	0.7859	0.6631	0.4414	0.7165	0.8445
Grafted C4.5	0.7958	0.6813	0.4229	0.7717	0.8322
Random tree	0.6983	0.5049	0.4149	0.6378	0.7408
REP tree	0.7775	0.6597	0.4149	0.6535	0.8021

Table 9. Standardized MCDM ranking scores for adult dataset.

Classifiers	TOPSIS	GRA	VIKOR	PROMETHEE II	ELECTRE III
Bayes net	0.8005	0.9476	0.5309	0.8687	0.8635
Naïve Bayes	0.7336	0.6903	0.5322	0.697	0.7686
Incremental Naïve Bayes	0.7336	0.6903	0.5322	0.6162	0.7705
IB1	0.6549	0.5069	0.4339	0.3232	0.4628
AdaBoost M1	0.9633	0.9533	0.833	0.8788	0.93
HyperPipes	0	0	0	0.1717	0.0964
VFI	0.361	0.412	0.1752	0.3636	0.4367
Conjunctive rule	0.0308	0.0195	0.0152	0.2121	0.1232
Decision table	0.8307	0.8956	0.4873	0.8384	0.911
OneR	0.4106	0.3484	0.2194	0.3535	0.402
PART	1	0.9888	1	0.9394	1
ZeroR	0.0289	0.0177	0.013	0	0
Decision stump	0.1678	0.1257	0.0839	0.0808	0.136
C4.5	0.8569	0.9914	0.507	0.9293	0.9261
Grafted C4.5	0.8672	1	0.5109	1	0.9442
Random tree	0.583	0.4283	0.3034	0.404	0.4912
REP tree	0.7838	0.7665	0.4097	0.596	0.789

Table 10. Standardized MCDM ranking scores for all datasets.

Classifiers	TOPSIS	GRA	VIKOR	PROMETHEE II	ELECTRE III
Bayes net	0.9929	1	0.983	0.9438	1
Naïve Bayes	0.8654	0.752	0.8669	0.5843	0.8539
Incremental Naïve Bayes	0.8654	0.7518	0.8669	0.4607	0.8539
IB1	0.6481	0.6747	0.3936	0.7079	0.8061
AdaBoost M1	0.6978	0.8875	0.4401	0.9213	0.902
HyperPipes	0.3478	0.3643	0.2314	0.2135	0.3293
VFI	0.5831	0.4688	0.5187	0.1685	0.5488
Conjunctive rule	0.5824	0.3848	0.6311	0	0.3846
Decision table	0.874	0.7142	0.8685	0.4944	0.8784
OneR	0.6773	0.5226	0.7331	0.236	0.5439
PART	0.8358	0.9417	0.7282	1	0.9381
ZeroR	0	0	0	0.0112	0
Decision stump	0.671	0.5051	0.7056	0.2247	0.5286
C4.5	1	0.9463	1	0.9438	0.989
Grafted C4.5	0.9959	0.9394	0.9968	0.9438	0.9898
Random tree	0.8644	0.7181	0.8917	0.4944	0.8465
REP tree	0.8914	0.7547	0.9108	0.6292	0.8861

Table 11. Weights of MCDM methods for magic gamma telescope dataset.

	TOPSIS	GRA	VIKOR	PROMETHEE II	ELECTRE III
Weights	0.93811	0.95343	0.93689	0.95772	0.9663
Normalized weights	0.1974	0.20062	0.19714	0.20152	0.20333

Table 12. Secondary MCDM rankings for magic gamma telescope dataset.

Classifiers	TOPSIS		GRA		VIKOR		PROMETHEE II		ELECTRE III	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Bayes net	0.4712	9	0.4902	9	0.606	9	−0.0246	9	1.4191	9
Naïve Bayes	0.3284	11	0.4267	11	0.7632	11	−0.3252	12	−4.0101	11
Incremental Naïve Bayes	0.3246	12	0.4254	12	0.7654	12	−0.2506	11	−4.1612	12
IB1	0.5973	7	0.5760	7	0.4815	7	0.3232	7	6.0744	8
AdaBoost M1	1	1	1.0000	1	0	1	1	1	14.9350	1
HyperPipes	0.0562	16	0.3433	16	0.9674	16	−0.875	16	−13.9071	16
VFI	0.1306	15	0.3603	15	0.9343	15	−0.7248	15	−11.7990	15
Conjunctive rule	0.3501	10	0.4351	10	0.7153	10	−0.1997	10	−3.2142	10
Decision table	0.6157	6	0.5855	6	0.457	6	0.3995	6	7.0028	6
OneR	0.1533	14	0.3700	14	0.8887	14	−0.6256	14	−10.1468	14
PART	0.6929	2	0.6519	2	0.4013	2	0.7749	2	8.9851	2
ZeroR	0	17	0.3333	17	1	17	−1	17	−16.0002	17
Decision stump	0.2227	13	0.3905	13	0.8059	13	−0.4745	13	−7.4233	13
C4.5	0.6767	4	0.6349	4	0.4252	3	0.6509	4	8.6411	4
Grafted C4.5	0.6844	3	0.6448	3	0.4288	4	0.7009	3	8.9066	3
Random tree	0.595	8	0.5655	8	0.4836	8	0.25	8	6.5769	7
REP tree	0.6509	5	0.6115	5	0.4524	5	0.4006	5	8.1209	5

Table 13. Weights of MCDM methods for adult dataset.

	TOPSIS	GRA	VIKOR	PROMETHEE II	ELECTRE III
Weights	0.94332	0.93658	0.86305	0.92862	0.93137
Normalized weights	0.20494	0.20347	0.1875	0.20174	0.20234

Table 14. Secondary MCDM rankings for adult dataset.

Classifiers	TOPSIS		GRA		VIKOR		PROMETHEE II		ELECTRE III	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Bayes net	0.7782	5	0.7461	5	0.2885	5	0.4491	5	8.2889	5
Naïve Bayes	0.6838	7	0.6202	8	0.3493	7	0.2421	7	4.6275	7
Incremental Naïve Bayes	0.6677	8	0.6090	9	0.3575	8	0.1677	8	4.1789	9
IB1	0.4797	10	0.4949	10	0.5834	10	−0.2025	11	−2.1822	10
AdaBoost M1	0.9032	2	0.8574	2	0.0867	2	0.7484	3	10.8850	2
HyperPipes	0.0857	16	0.3467	16	0.9783	16	−0.9243	16	−14.7981	16
VFI	0.3597	12	0.4381	12	0.6947	13	−0.3988	12	−5.9346	12
Conjunctive rule	0.1084	15	0.3534	15	0.9505	15	−0.7248	15	−11.7694	15
Decision table	0.767	6	0.7382	6	0.3143	6	0.379	6	8.2099	6
OneR	0.3536	13	0.4357	13	0.675	12	−0.4257	13	−6.1287	13
PART	0.9726	1	0.9737	1	0	1	0.9239	1	11.6903	1
ZeroR	0.0163	17	0.3360	17	0.9933	17	−0.9255	17	−14.9999	17
Decision stump	0.124	14	0.3624	14	0.8942	14	−0.6754	14	−10.4321	14
C4.5	0.7966	4	0.8068	4	0.2791	4	0.6308	4	8.9130	4
Grafted C4.5	0.8056	3	0.8440	3	0.2656	3	0.8063	2	9.2100	3
Random tree	0.4485	11	0.4773	11	0.5851	11	−0.1995	10	−3.9423	11
REP tree	0.6624	9	0.6216	7	0.4158	9	0.1293	9	4.1838	8

Table 15. Weights of MCDM methods for all datasets.

	TOPSIS	GRA	VIKOR	PROMETHEE II	ELECTRE III
Weights	0.86213	0.88358	0.77604	0.80178	0.88603
Normalized weights	0.2048	0.2099	0.18435	0.19047	0.21048

Table 16. Secondary MCDM rankings for all datasets.

Classifiers	TOPSIS		GRA		VIKOR		PROMETHEE II		ELECTRE III	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Bayes net	0.9752	1	0.9718	1	0.0000	1	0.8789	1	9.6360	1
Naïve Bayes	0.7700	6	0.7141	7	0.2725	6	0.2255	6	4.7491	6
Incremental Naïve Bayes	0.7367	10	0.7017	10	0.3435	10	0.0529	10	4.0773	9
IB1	0.6455	11	0.6026	11	0.4226	11	−0.2743	11	−0.4077	11
AdaBoost M1	0.7411	8	0.7265	6	0.3379	9	0.1646	8	3.5535	10
HyperPipes	0.3067	16	0.4179	16	0.6964	15	−0.8274	16	−11.8771	15
VFI	0.4673	14	0.4896	14	0.6361	14	−0.5731	14	−8.6187	14
Conjunctive rule	0.4142	15	0.4696	15	0.7488	16	−0.7285	15	−14.1241	16
Decision table	0.7427	7	0.7071	8	0.3245	7	0.1979	7	4.0955	7
OneR	0.5392	12	0.5375	12	0.5622	12	−0.3066	12	−5.6241	12
PART	0.8647	4	0.8393	4	0.1463	4	0.4564	4	7.2822	4
ZeroR	0.0048	17	0.3338	17	1.0000	17	−0.9762	17	−15.0000	17
Decision stump	0.5255	13	0.5281	13	0.5754	13	−0.4546	13	−6.0896	13
C4.5	0.9653	2	0.9559	2	0.0060	2	0.8735	2	9.4418	2
Grafted C4.5	0.9629	3	0.9510	3	0.0110	3	0.7749	3	9.3748	3
Random tree	0.7402	9	0.7026	9	0.3263	8	0.0677	9	4.0790	8
REP tree	0.7954	5	0.7463	5	0.2362	5	0.4485	5	5.4522	5

4.3.2. *Adult dataset*

Table 13 summarizes the weights and normalized weights of the five MCDM methods for the adult data.

The secondary rankings of classifiers on adult dataset are reported in Table 14. Ten classifiers got the same rankings on this dataset. The degrees of disagreements on the rankings of classifiers are greatly reduced.

4.3.3. *Average of 11 datasets*

The weights and normalized weights of the five MCDM methods for all 11 datasets are shown in Table 15.

The secondary rankings of classifiers on all the dataset are reported in Table 16. The five MCDM methods generate the same rankings on 10 classifiers. Rankings of classifiers produced by TOPSIS and PROMETHEE II are the same. The degrees of disagreements on the rankings of classifiers are greatly reduced compared with the original MCDM rankings.

The results of magic gamma telescope, adult, and all 11 datasets indicate that the secondary rankings of the MCDM methods are now in strong agreement. Specifically, classifiers that got the same rankings have increased from three to 11 for the magic gamma telescope dataset. For adult and the average of 11 datasets, the agreements of the five MCDM methods on classifiers have changed from zero to 10. Though the rankings of classifiers are not identical, the differences among the five MCDM methods have been largely reduced.

5. Conclusion

MCDM methods are feasible tools for selecting classification algorithms because the task of algorithms selection normally involves more than one criterion and can be modeled as MCDM problems. Since different MCDM methods evaluate classifiers from different aspects, they may produce divergent rankings.

This paper proposed an approach that uses Spearman's rank correlation coefficient to reconcile conflicting rankings generated by different MCDM methods for classification algorithms. The key of the proposed approach is to determine a weight for each MCDM method according to the similarities between the ranking it generated and the rankings produced by other MCDM methods. An MCDM method that has a larger Spearman's rank correlation coefficient is considered more important than one with a smaller Spearman's rank correlation coefficient because it has better agreements with other MCDM methods.

An experimental study was designed to validate the proposed approach. First, 17 classifiers were applied to 11 binary UCI classification datasets and the classification results were measured using 10 performance measures. Second, five MCDM methods (i.e., TOPSIS, GRA, VIKOR, PROMETHEE II, and ELECTRE III) were used to evaluate the classification algorithms based on their performances on the

measures over magic gamma telescope data, adult data, and the collection of all 11 datasets. The rankings produced by the five MCDM methods for the three data are quite different. Third, the weights of MCDM methods were calculated using Spearman's rank correlation coefficient equations. The normalized weights and the rankings generated at the second stage were used as inputs to produce the secondary rankings of classifiers. The results indicate that the approach proposed can provide a compatible ranking when different MCDM techniques disagree.

Acknowledgments

This research has been partially supported by grants from the National Natural Science Foundation of China (#70901015 for Gang Kou, #70901011 and #71173028 for Yi Peng, and #70921061 for Yong Shi).

References

1. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval* (Addison Wesley, 1999).
2. J. P. Brans, L'ingénièrie de la décision, Elaboration d'instruments d'aide à la décision. La méthode PROMETHEE, in *L'aide à la décision: Nature, Instruments et Perspectives d'Avenir*, eds. R. Nadeau and M. Landry (Presses de l'Université Laval, Québec, Canada, 1982), pp. 183−213.
3. J. P. Brans and B. Mareschal, PROMETHEE methods, in *Multiple Criteria Decision Analysis: State of the Art Surveys*, eds. J. Figueira, V. Mousseau and B. Roy (Springer, New York, 2005), pp. 163−195.
4. J. P. Brans and B. Mareschal, How to decide with PROMETHEE (1994), http://www.visualdecision.com/Pdf/How%20to%20use%20PROMETHEE.pdf.
5. J. Deng, Control problems of grey systems, *Systems and Control Letters* **1** (1982) 288−294.
6. P. Domingos and M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning* **29**(203) (1997) 103−130.
7. D. Ergu, G. Kou, Y. Peng and Y. Shi, A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP, *European Journal of Operational Research* **213**(1) (2011) 246−259, doi:10.1016/j.ejor.2011.03.014.
8. D. Ergu, G. Kou, Y. Shi and Y. Shi, Analytic network process in risk assessment and decision analysis, *Computers & Operations Research* (2011), doi:10.1016/j.cor.2011.03.005.
9. C. Ferri, J. Hernandezorallo and R. Modroiu, An experimental comparison of performance measures for classification, *Pattern Recognition Letters* **30**(1) (2009) 27−38.
10. E. Frank and I. H. Witten, Generating accurate rule sets without global optimization, *Fifteenth Int. Conf. Machine Learning* (1998), pp. 144−151.
11. J. Figueira, V. Mousseau and B. Roy, ELECTRE Methods, in *Multiple Criteria Decision Analysis: State of the Art Surveys*, eds. J. Figueira, V. Mousseau and B. Roy (Springer, New York, 2005), pp. 131−153.
12. Y. Freund and R. E. Schapire, Experiments with a new boosting algorithm, in *Proc. Int. Conf. Machine Learning* (Morgan Kaufmann, San Francisco, 1996), pp. 148−156.
13. G. Demiroz and A. Guvenir, Classification by voting feature intervals, *Machine Learning: ECML-97*, Lecture Notes in Computer Science, Vol. 1224 (1997), pp. 85−92.

14. P. Gupta, M. K. Mehlawat and A. Saxena, Asset portfolio optimization using fuzzy mathematical programming, *Information Sciences* **178**(6) (2008) 1734−1755.
15. C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making Methods and Applications* (Springer, Berlin Heidelberg, 1981).
16. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, The WEKA data mining software: An update, *SIGKDD Explorations* **11**(1) (2009) 10−18.
17. R. C. Holte, Very simple classification rules perform well on most commonly used datasets, *Machine Learning* **11** (1993) 63−91.
18. W. Iba and P. Langley, Induction of one-level decision trees, in *Proc. Ninth Int. Conf. on Machine Learning* (ML, 1992), Aberdeen, Scotland, UK, July 1−3, 1992 (Morgan Kaufmann 1992, ISBN 1-55860-247-X), pp. 233−240.
19. G. Kou, Y. Peng, Y. Shi, M. Wise and W. Xu, Discovering credit cardholders' behavior by multiple criteria linear programming, *Annals of Operations Research* **135**(1) (2005) 261−274.
20. G. Kou and C. Lou, Multiple factor hierarchical clustering algorithm for large scale web page and search engine clickstream data, *Annals of Operations Research* (2010), doi: 10.1007/s10479-010-0704-3.
21. L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms* (Wiley, 2004).
22. Y. Kuo, T. Yang and G. W. Huang, The use of grey relational analysis in solving multiple attribute decision-making problems, *Computers & Industrial Engineering* **55** (2008) 80−93.
23. S. K. Lee, Y. H. Cho and S. H. Kim, Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations, *Information Sciences* **180**(11) (2010) 2142−2155.
24. A. S. Milani, A. Shanian and C. El-Lahham, Using different ELECTRE methods in strategic planning in the presence of human behavioral resistance, *Journal of Applied Mathematics and Decision Sciences* **2006** (2006) 19, doi:10.1155/JAMDS/2006/10936.
25. MATLAB, The MathWorks, Inc., Natick, MA 01760 (2005), http://www.mathworks.com/products/matlab/.
26. G. Nakhaeizadeh and A. Schnabl, Development of multi-criteria metrics for evaluation of data mining algorithms, in *Proc. Third Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)* (Newport Beach, California, 1997), pp. 37−42.
27. D. L. Olson, Comparison of weights in TOPSIS models, *Mathematical and Computer Modelling* **40**(7−8) (2004) 721−727.
28. S. Opricovic, Multicriteria optimization of civil engineering systems, Faculty of Civil Engineering, Belgrade (1998).
29. S. Opricovic and G. H. Tzeng, Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS, *European Journal of Operational Research* **156**(2) (2004) 445−455.
30. S. Opricovic and G. H. Tzeng, Multicriteria planning of post-earthquake sustainable reconstruction, *Computer-Aided Civil and Infrastructure Engineering* **17**(3) (2002) 211−220.
31. Y. Peng, G. Kou, Y. Shi and Z. Chen, A descriptive framework for the field of data mining and knowledge discovery, *International Journal of Information Technology & Decision Making* **7**(4) (2008) 639−682.
32. Y. Peng, G. Kou, Y. Shi and Z. Chen, A multi-criteria convex quadratic programming model for credit data analysis, *Decision Support Systems* **44**(4) (2008) 1016−1030.

33. Y. Peng, G. Kou, G. Wang, W. Wu and Y. Shi, Ensemble of software defect predictors: An AHP-based evaluation method, *International Journal of Information Technology & Decision Making* **10**(1) (2011) 187−206, doi:10.1142/S0219622011004282.
34. Y. Peng, G. Kou, G. Wang and Y. Shi, FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms, *Omega* **39**(6) (2011) 677−689, doi: 10.1016/j.omega.2011.01.009.
35. Y. Peng, G. Wang and H. Wang, User preferences based software defect detection algorithms selection using MCDM, *Information Sciences* (2010), doi:10.1016/j. ins.2010.04.019.
36. J. R. Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kaufmann, 1993).
37. J. Rice, The algorithm selection problem, *Advances in Computers* **15** (1976) 65−118.
38. B. Roy, Classement et choix en presence de points de vue multiples (la methode ELECTRE) *R.I.R.O* **8** (1968) 57−75.
39. B. Roy and D. Bouyssou, *Aide Multicritere a la Decision: Methodes et cas* (Economica, Paris, 1993).
40. B. Roy, ELECTRE III: Un algorithme de classements fonde sur une representation flue des preferences en presence de ctriteres multiples, *Cahiers du CERO* **20**(1) (1978) 3−24.
41. L. Rokach, Ensemble-based classifiers, *Artificial Intelligence Review* **33**(1−2) (2009) 1−39.
42. K. A. Smith-Miles, Cross-disciplinary perspectives on meta-learning for algorithm selection, *ACM Computing Surveys* **41**(1) (2008), Article No.: 6, doi:10.1145/ 1456650.1456656.
43. B. Soylu, Integrating PROMETHEE II with the Tchebycheff function for multi criteria decision making, *International Journal of Information Technology & Decision Making* **9**(4) (2010) 525−545.
44. E. Triantaphyllou and K. Baig, The impact of aggregating benefit and cost criteria in four MCDA methods, *IEEE Transactions on Engineering Management* **52**(2) (2005) 213−226.
45. I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. (Morgan Kaufmann, San Francisco, 2005).
46. G. I. Webb, Decision tree grafting from the all-tests-but-one partition, in *Proc. Sixteenth Int. Joint Conf. Artificial Intelligence* (*IJCAI'99*), Stockholm, Sweden, July 31−August 6 (1999), pp. 702−707.
47. S. M. Weiss and C. A. Kulikowski, *Computer Systems that Learn: Classification and Predication Methods from Statistics, Neural Nets. Machine Learning and Expert Systems* (Morgan Kaufmann, 1991).
48. J. L. Yang, H. N. Chiu and G. Tzeng, Vendor selection by integrated fuzzy MCDM techniques with independent and interdependent relationships, *Information Sciences* **178**(21) (2008) 4166−4183.
49. Y. Ou Yang, H. Shieh, J. Leu and G. Tzeng, A Vikor-based multiple criteria decision method for improving information security risk, *International Journal of Information Technology & Decision Making* **8**(2) (2009) 267−287.