Neurocomputing 72 (2009) 3150–3157

Contents lists available at ScienceDirect

Neurocomputing

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

Bankruptcy prediction using ELECTRE-based single-layer perceptron

Yi-Chung Hu *

Department of Business Administration, Chung Yuan Christian University, Chung-Li 32023, Taiwan, ROC

ARTICLE INFO

Article history:
Received 5 November 2008
Received in revised form
6 March 2009
Accepted 11 March 2009
Communicated by G.P. Zhang
Available online 5 April 2009

Keywords:
ELECTRE
Multicriteria decision aid
Bankruptcy prediction
Genetic algorithm
Single-layer perceptron

ABSTRACT

For the outranking relation theory, the ELECTRE methods are one of the most extensively used outranking methods. To measure the degree of agreement and the degree of disagreement of the proposition "one alternative outranks another alternative", the concordance and discordance relations are usually associated with the outranking relation. Instead of the traditional single-layer perceptron (SLP) developed according to the multiple-attribute utility theory, this paper contributes to develop a novel ELECTRE-based SLP for multicriteria classification problems based on the ELECTRE methods involving pairwise comparisons among patterns. A genetic-algorithm-based method is then designed to determine connection weights. A real-world data set involving bankruptcy analysis obtained from Moody's Industrial Manuals is employed to examine the classification performance of the proposed ELECTRE-based model. The results demonstrate that the proposed model performs well compared to an arsenal of well-known classification methods involving quantitative disciplines of statistics and machine learning.

1. Introduction

The traditional single-layer perceptron (SLP) with single output neuron could be treated as a multicriteria approach to decision-aid and has been widely applied to two-class problems, such as bankruptcy analysis [14,17] which has long been an important classification problem for a business. The sigmoid function whose output ranged between 0 and 1 is usually used as the output node's transfer function. The advantage of using the sigmoid function is that errors between actual and desired outputs of individual training patterns can be easily measured [4]. Each training pattern from a finite set of alternatives, described by multiple criteria, can be categorized into one of the predefined classes, say C_1 and C_2. The desired output values of patterns in C_1 and C_2 can thus be pre-specified as, for example, 1 and 0, respectively. When the actual output value of an input pattern does not exceed a pre-specified threshold (e.g., 0.5), it can be assigned into C_2; otherwise, it can be assigned into C_1. This implies that C_1 consists of the most preferred alternatives while C_2 consists of the least preferred alternatives. Such preference information on two patterns or alternatives can be obtained through a utility function which is realized by the traditional SLP. However, for decision-aid problems, in addition to the preference relation in the utility framework, it would be interesting to obtain the preference information using the widely used outranking relation, S, which involves pairwise comparisons among patterns. This motivates us to incorporate the outranking relation into the SLP.

As for outranking relation theory (ORT) techniques, the Elimination and Choice Translating Reality (ELECTRE) methods, initially introduced by Roy [18,28], are one of the most extensively used outranking methods [20,21]. Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be two alternatives. Two binary relations associated with the outranking relation, S, are addressed in the ELECTRE methods [19]: one is the concordance relation, C_S, for which the index $C_S(a, b)$ measures the intensity of agreement with the proposition "alternative a is at least as good as alternative b" (i.e., $\mathbf{x}_a S \mathbf{x}_b$), the other is the discordance relation, D_S, for which the index $D_S(a, b)$ measures the intensity of the indications against the proposition $\mathbf{x}_a S \mathbf{x}_b$. An overall estimation of the outranking degree of $\mathbf{x}_a$ over $\mathbf{x}_b$ can be further derived by $C_S(a, b)$ and $D_S(a, b)$. Many members of the family of the ELECTRE methods have been developed for multiple-criteria decision aid [23–26]. According to C_S and D_S, Yu [22] developed the useful ELECTRE TRI method for pattern classification. Later, Lourenço and Costa [24] employed the ELECTRE TRI method to sort MOMILP nondominated solutions.

In view of the usefulness of the ELECTRE methods, the main objective of this paper is to develop a novel SLP whose transfer function is designed on the basis of the ELECTRE methods, and evaluate classification performance of the proposed SLP compared to the traditional SLP for bankruptcy prediction. The output neuron in the proposed model is represented by a set of connection weights and an ELECTRE-based transfer function. In particular, the output value is the net outranking index obtained by combining the outranking and outranked characters of an input pattern. For this, the indices of C_S and D_S on each criterion

* Tel.: +886 3 2655130.
E-mail address: ychu@cycu.edu.tw

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.03.002

can be first obtained by performing pairwise comparisons on an input pattern, say $\mathbf{x}_a$, and a training pattern, say $\mathbf{x}_b$. In order to aggregate the degree to which each criterion agrees with the proposition $\mathbf{x}_a S \mathbf{x}_b$ into an overall concordance index $C_S(a, b)$, the connection weights is interpreted as the degree of importance of the respective criteria. We employ the genetic algorithms [5–7] to develop a genetic-algorithm-based (GA-based) method to automatically determine connection weights for constructing an ELECTRE-based SLP with high classification performance.

The paper is organized as follows. Section 2 presents the concepts related to C_S and D_S involved in the ELECTRE methods, since these two relations play significant role for generating the output value of the ELECTRE-based SLP. The framework of the proposed ELECTRE-based SLP and the GA-based learning algorithm are demonstrated in Sections 3 and 4, respectively. Section 5 reports the experimental results by applying the proposed model to a real-world data set involving bankruptcy analysis, which was obtained from Moody's Industrial Manuals during the period 1975 through 1982, and comparing with other well-known classification methods involving quantitative disciplines of statistics and machine learning. It can be seen that the ELECTRE-based SLP outperforms the traditional SLP and is comparable to other classification methods. The discussion and conclusions are presented in Section 6.

2. Outranking relations

Let n be the number of criteria. Each pattern is a vector evaluated by n criteria such that $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$. From the viewpoint of the multiple-attribute utility theory (MAUT), $U(\mathbf{x}_i) > U(\mathbf{x}_j)$ holds if and only if $\mathbf{x}_i$ is preferred to $\mathbf{x}_j$ (i.e., $\mathbf{x}_i \succ \mathbf{x}_i$), where $U(\mathbf{x}_i)$ and $U(\mathbf{x}_j)$ are the utilities of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Thus, the preference relation among patterns is described by the utility function. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the additive utility function is the most commonly used form:

$$U(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i \tag{1}$$

where $0 \leqslant w_1, w_2, \ldots, w_n \leqslant 1$. These constants represent the weights of the attributes and sum up to one:

$$\sum_{i=1}^{n} w_i = 1 \tag{2}$$

For the traditional SLP, although the sum of connection weights may not be 1, its output value can also be interpreted as the synthetic evaluation or utility of the corresponding input pattern. As mentioned above, the larger the output value, the more possible an input pattern can be assigned into C_1. This means that the most preferred alternatives constitute C_1 and the least preferred alternatives constitute C_2 such that C_1 and C_2 are defined in an ordinal way (i.e., $C_1 \succ C_2$).

The foundations of the ORT have been set by Bernard Roy through the development of the ELECTRE methods [20]. In contrast to the MAUT, the outranking relations provide preference information by performing pairwise comparisons among alternatives. Given $\mathbf{x}_i \in C_1$ and $\mathbf{x}_j \in C_2$, the ordering of the classes (i.e., $C_1 \succ C_2$) for the outranking relations implies that $\mathbf{x}_i$ is at least as good as $\mathbf{x}_j$ [29]. The development of the outranking relation S involves the establishment of concordance and discordance relations. In ELECTRE I and ELECTRE II, the concordant set C_S^{ij} and the discordant set D_S^{ij} are defined as follows:

$$C_S^{ij} = \{k | x_{ik} \geqslant x_{jk}\} \tag{3}$$

$$D_S^{ij} = \{k | x_{ik} < x_{jk}\} \tag{4}$$

It is clear that C_S^{ij} is composed of all attributes for which $\mathbf{x}_i$ is better than or equal to $\mathbf{x}_j$, and D_S^{ij} is composed of all attributes for which $\mathbf{x}_i$ is worse than $\mathbf{x}_j$. The partial concordance index $C_{S,r}(i,j)$ which measures the intensity of agreement with the proposition $\mathbf{x}_i S \mathbf{x}_j$ on the basis of criterion r is defined as follows:

$$c_{S,r}(i,j) = \begin{cases} w_r, & \text{if } r \in C_S^{i,j} \\ 0, & \text{if } r \notin C_S^{i,j} \end{cases} \tag{5}$$

$C_{S,1}(i, j)$, $C_{S,2}(i, j), \ldots, C_{S,n}(i, j)$ are aggregated to obtain an overall concordance index $C_S(i, j)$ measuring the overall agreement or the degree of agreement with the proposition $\mathbf{x}_i S \mathbf{x}_j$ is defined as

$$D_{S,r}(i,j) = \sum_{r=1}^{n} C_{S,r}(i,j) \tag{6}$$

Among those criteria in D_S^{ij}, the partial discordance index $D_{S,r}(i, j)$ which measures the intensity of the indications against the proposition $\mathbf{x}_i S \mathbf{x}_j$ on the basis of criterion r is defined as follows:

$$D_{S,r}(i,j) = \frac{|x_{ir} - x_{jr}|}{\sum_{k \in D_S^{ij}} |x_{ik} - x_{jk}|}, \quad \text{if } r \in D_s^{ij} \tag{7}$$

If $r \notin D_S^{ij}$, then $D_{S,r}(i, j)$ is specified as 0.

Once the concordance and discordance indices are estimated, $S(i, j)$, which represents the overall estimation of the outranking degree of $\mathbf{x}_i$ over $\mathbf{x}_j$, is defined as follows [20]:

$$S(i,j) = C_S(i,j) \prod_{l \in F} \frac{1 - D_{S,l}(i,j)}{1 - C_S(i,j)} \tag{8}$$

where F denotes the set of criteria for which the discordance index is larger than the concordance index:

$$F = \{l | D_{S,l}(i,j) > C_S(i,j)\} \tag{9}$$

$S(i, j)$ ranges in the interval $[0, 1]$. When no criteria are discordant, $S(i, j)$ is equal to $C_S(i, j)$. Let $\neg$ denote the logical negation operation. $S(i, j) = 0$ indicates that $\mathbf{x}_i S \mathbf{x}_j$ does not hold (i.e., $\neg(\mathbf{x}_i S \mathbf{x}_j)$), whereas $S(i, j) = 1$ indicates that $\mathbf{x}_i S \mathbf{x}_j$ holds necessarily.

3. ELECTRE-based single-layer perceptron

The outranking indices $S(i, j)$ for the pairwise comparison ($\mathbf{x}_i$, $\mathbf{x}_j$) constitute the basis of determining the classification of $\mathbf{x}_i$. Let T denote the set of training patterns. As shown in Fig. 1, the connection weights of the proposed SLP is interpreted as the degree of importance of the respective criteria. The sum of connection weights is thus equal to 1. When an input pattern $\mathbf{x}_i$ is presented to the proposed SLP, the corresponding actual value (i.e., S_i) generated by the output node of the SLP is the net outranking index, whereas S_i is defined as follows:

$$s_i = \frac{1}{m}(S_i^+ - S_i^-) = \frac{1}{m} \sum_{\mathbf{x}_j \in T} (S(i,j) - S(j,i)) \tag{10}$$

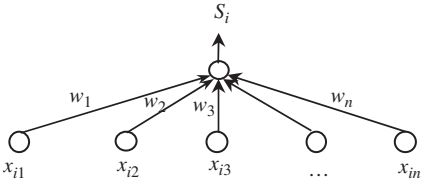

Fig. 1. Architecture of the ELECTRE-based SLP.

where m denotes the number of training patterns, S_i^+ represents the outranking character of $\mathbf{x}_i$ over all training patterns, and S_i^- represents the outranked character of $\mathbf{x}_i$ by all training patterns. S_i ranges in the interval $[-1, 1]$. Actually, S_i^+ consists of the outranking character of $\mathbf{x}_i$ over the training patterns of C_1 and the outranking character of $\mathbf{x}_i$ over the training patterns of C_2, whereas S_i^- consists of the outranked character of $\mathbf{x}_i$ by the training patterns of C_1 and the outranked character of $\mathbf{x}_i$ by the training patterns of C_2. The related properties of S_i are described as follows:

(1) $S_i = -1$ (i.e., $S_i^+ = 0$, $S_i^- = 1$) indicates that $\mathbf{x}_i$ does not outrank any training pattern, whereas $S_i = 1$ (i.e., $S_i^+ = 1$, $S_i^- = 0$) indicates that $\mathbf{x}_i$ outranks all training patterns.
(2) An input pattern from C_1 is expected to strongly outrank the training patterns of C_1 and those of C_2, and is expected to be weakly outranked by training patterns of C_1 and those of C_2. That is, an input pattern that belongs to C_1 is expected to have a high S_i^+ and a low S_i^-.
(3) An input pattern from C_2 is expected to be strongly outranked by training patterns of C_1 and those of C_2, and is expected to weakly outrank training patterns of C_1 and those of C_2. That is, an input pattern that belongs to C_2 is expected to have a low S_i^+ and a high S_i^-.
(4) Given any two patterns $\mathbf{x}_a$ and $\mathbf{x}_b$ such that $\mathbf{x}_{ak} \geqslant \mathbf{x}_{bk}$ $(1 \leqslant k \leqslant n)$, since $C_S(a, j) \geqslant C_S(b, j)$ and $D_{S,r}(a, j) \leqslant D_{S,r}(b, j)$ for all $\mathbf{x}_j \in T$, where $r \in D_S^{ij}$, $S_a^+ \geqslant S_b^+$ and $S_a^- \leqslant S_b^-$ holds. This leads to $S_a \geqslant S_b$. The categorization of $\mathbf{x}_a$ into C_2 and $\mathbf{x}_b$ into C_1 can be avoided by a cut-off point.

Patterns with the net outranking index higher than the cut-off point θ are assigned into C_1, whereas patterns with the net outranking index lower than the cut-off point are assigned into C_2. In other words, the higher the S_i generated by the ELECTRE-based transfer function, the more possible that $\mathbf{x}_i$ is assigned into C_1, otherwise, $\mathbf{x}_i$ is assigned into C_2. Thus, it is reasonable for us to assign the expected output value of 1 to a pattern from C_1, and that of -1 to a pattern from C_2. The training error of the proposed model can be computed by the actual and desired outputs of respective training patterns:

$$E = \sum_{i=1}^{m} (d_i - S_i)^2 \tag{11}$$

where d_i is the desired output of $\mathbf{x}_i$. To effectively reduce E, connection weights are required to be determined approximately.

4. Genetic-algorithm-based learning algorithm

4.1. Outline of ELECTRE-based SLP

The proposed ELECTRE-based SLP does not involve any complicated mechanism for tuning parameter specifications. Its construction involves basic genetic operations such as selection, crossover and mutation. Let n_{size} and n_{max} denote the population size and the total number of generations, respectively. By using the GA, the outline of the proposed SLP is as follows.

Algorithm. ELECTRE-based SLP learning algorithm

Step 1: Normalization of attribute value

The concordance and discordance indices can be meaningless if the measurement scales of respective attributes are different. To eliminate the effect resulting from the measurement scales, normalization is required to be performed for each performance value of an input pattern $\mathbf{x}$:

$$x_j' = \frac{x_j - mi_j}{ma_j - mi_j}, \quad j = 1, 2, \ldots, n \tag{12}$$

where ma_j and mi_j are the maximum and the minimum values of the domain interval of x_j, respectively. That is, our classification problem is defined in the n-dimensional pattern space $[0, 1]^n$ with continuous attributes.

Step 2: Initialization

Generate an initial population of n_{size} chromosomes.

Step 3: Evaluate each chromosome

Decode each chromosome in the current population and compute the corresponding fitness value.

Step 4: Generate new strings by genetic operations

(1) *Selection*: $\frac{1}{2}n_{size}$ pairs of chromosomes can be selected from the current population for mating.
(2) *Crossover*: It is determined whether crossover is to be performed on two selected substrings with a crossover probability. If so, the substring are swapped at the crossover point chosen randomly.
(3) *Mutation*: The mutation operator is performed for each bit of the newly generated binary chromosomes in the next population with a mutation probability.

Step 5: Elitist strategy

Randomly remove n_{del} $(0 \leqslant n_{del} \leqslant n_{size})$ newly generated chromosomes from the n_{size} strings generated by the above operations. Insert n_{del} chromosomes with the maximum fitness in the previous population into the current one.

Step 6: Termination test

Terminate the algorithm if n_{max} generations have been generated; otherwise, return to *Step* 3.

Evaluation of each chromosome, coding method and genetic operations related to the above algorithm are explained in the following sections in detail.

4.2. Evaluation of each chromosome

The net outranking indices of respective training patterns and the training error of the proposed SLP can be obtained by a set of connection weights. Each substring of the connection weights can be directly decoded as a real value ranging from zero to one, but the decoded value of the substring of the cut-off point ranges in the interval $[-1, 1]$. Our problem is to construct a SLP, denoted by *ES*, with high classification power and low training error by the ELECTRE methods. By introducing positive weights w_{CA} and w_V, the fitness function $f(ES)$ is defined as follows:

$$f(ES) = w_{NC} \cdot NC(ES) + w_E \cdot \frac{1}{1 + E(ES)} \tag{13}$$

where $NC(ES)$ and $E(ES)$ denote the number of correctly classified training patterns by *ES* and the training error of *ES*, respectively.

4.3. Coding

Let P_i denote the population generated in the ith generation. Since the above-mentioned parameter specifications (i.e., w_1, $w_2,\ldots,w_n$, θ) cannot be easily pre-specified by decision-makers, they are automatically determined by the GA. Each chromosome corresponds to an ELECTRE-based SLP. The binary chromosome $k(1\leqslant k\leqslant n_{size})$ in $P_i(1\leqslant i\leqslant n_{max})$ is represented by $w^i_{k1}w^i_{k2}\ldots w^i_{k,n}\theta^i_k$. $(n+1)$ binary substrings constitute a string, and n_{size} chromosomes make up a population.

The length of a substring is dependent on the domain interval of the corresponding variable and the required precision [9]. That is, if the domain interval of a variable has the width of τ_1, and the corresponding required precision is τ_2 decimal places, then τ_3 bits are required to code such a variable if $2^{\tau_3-1}<\tau_1 10^{\tau_2}<2^{\tau_3}$ holds. For instance, if the width of the domain interval of a variable is 1 (i.e., $\tau_1=1.0$), and the required precision is three decimal places (i.e., $\tau_2=3$) for each variable, then 10 bits are required to code this variable (i.e., $\tau_3=10$).

4.4. Genetic operations

An initial population containing n_{size} chromosomes is generated and inserted into P_1. Each gene in the binary chromosome is randomly assigned as either one or zero, with a probability of 0.5. By evaluating the fitness value of each chromosome in P_i, the genetic operators including reproduction, crossover, and mutation [5,6,35,36] are iterated until n_{size} new chromosomes are generated in P_{i+1}. When a stopping condition is satisfied, the proposed classifier is terminated. In addition, the best chromosome with maximum fitness value among the successive generations is taken as the desired solution to examine the generalization ability of the proposed model, since such a chromosome may not appear in the final generation. In this paper, n_{max} is used as the stopping condition. It should be noted that the specification of the stopping condition can consider the available computation time [9,10].

When the fitness values of individual chromosomes are obtained, in order to generate new strings in the next population, the binary tournament selection is taken into account. Two strings are randomly selected from the current population, and the one with the maximum fitness can be placed in the mating pool. This process can be repeated n_{size} times until there are n_{size} strings in the mating pool. Thus, n_{size} pairs of chromosomes can be selected for mating. The crossover and mutation are further applied to a selected parent to reproduce children by altering the chromosomal makeup of two parents.

The crossover operator with a pre-specified probability Pr_c is used for exchanging partial information between two substrings in the selected parent. A larger crossover probability allows more solution space to be explored. Each crossover point in a substring is chosen randomly. Thus, there are n+1 crossover points for a selected pair. Two newly generated chromosomes are generated to replace their parent strings by inserting these two new chromosomes into P_{i+1}. Then, the mutation operator is performed on each bit of strings newly generated by the crossover operation. Each gene in a string could be thus changed either from zero to one or from one to zero with the mutation probability Pr_m. In order to avoid generating much random perturbation, a low mutation rate is often considered. Additionally, if there are two or more different chromosomes with maximum fitness value in P_i, then one chromosome can be randomly selected from these best chromosomes as an elite individual.

5. Application to bankruptcy prediction

5.1. Data description

It is known that bankruptcy prediction has long been an important classification problem for a business. To examine the classification performance and the potential applications of the proposed ELECTRE-based SLP, we taken into account the sample bankruptcy prediction data obtained from Moody's Industrial Manuals during the period 1975 through 1982. The data set comprises 129 firms where 65 went bankrupt. Each pattern is described by the five explanatory variables (i.e., $n=5$) provided in [11]: Working capital/total assets, retained earnings/total assets, EBIT/total assets, MVE/total debt, and Sales/total assets, where EBIT is earnings before interest and tax, and MVE is the market value of equity.

The hold-out method is commonly employed to examine the testing performance of each classification method [30]. In the hold-out method, the given data set are randomly partitioned into training and test sets. Four splits of the collected samples are considered as follows:

(1) 80% training patterns and 20% test patterns (80/20 partition)
(2) 70% training patterns and 30% test patterns (70/30 partition)
(3) 60% training patterns and 40% test patterns (60/40 partition)
(4) 50% training patterns and 50% test patterns (50/50 partition)

The training set is employed to derive the classifier, whose classification accuracy rate is estimated with the test set. Since the result may be dependent on a partition set, by random sampling, we perform 10 random splits of the given data set, partitioning the data into training and test sets for each proportion of the partition. That is, the hold-out method is repeated 10 times for each proportion. The overall accuracy is taken as the average of the accuracies obtained from 10 estimates [8,30].

5.2. Parameters for genetic operations

The computer programs are coded by Delphi 7.0 on a Pentium 4 personal computer with Microsoft Windows XP, 512 MB RAM and a clock rate of 3.20 GHz. Since there is no best set of values of parameters for GA [9], the pre-specified values of parameter specifications used in the learning algorithm are determined by a general suggestion introduced in [9]. That is, customized parameter tuning is not considered for training the FIFLN. Parameter specifications for the ELECTRE-based SLP are pre-specified as follows:

(1) $n_{size}=50$: Since the most common sizes of population vary from 50 individuals to 500 individuals, $n_{size}=50$ is acceptable.
(2) $n_{max}=500$: It is considered that a sufficient evolution of the GA is required.
(3) $n_{del}=2$: From the experimental results reported in [16], it is seen that only two or three elite chromosomes are sufficient to generate better results. Thus, two elite chromosomes are taken into account.
(4) The length of each substring is specified to be 10: The required precision of three decimal places for each parameter is taken into account.
(5) $\mathrm{Pr}_c=1.0$ and $\mathrm{Pr}_m=0.05$.
(6) $w_{NC}=1.0$, $w_E=0.1$: The reason is that the classification power of a classifier is more important than any other factors [10].

Table 1
Classification accuracy rates (%) of different classification methods for test patterns.

Classification method	Partition			
	80/20	70/30	60/40	50/50
LDA	86.92	85.05	84.24	85.47
Probit method	85.38	89.95	89.49	89.03
LOGIT	86.92	90.46	89.11	88.41
SLP	77.69	81.19	78.40	82.23
MLP	81.15	80.93	80.35	84.70
FIFLN	91.15	89.18	90.47	87.48
SVM	90.39	84.53	90.51	88.39
ELECTRE TRI (pessimistic)	85.38	83.76	82.36	84.08
ELECTRE TRI (optimistic)	81.92	86.08	83.14	82.84
ELECTRE-based SLP	91.54	91.24	92.02	89.95

5.3. Classification methods

The experimental results are reported by an extensive comparison of the proposed model on the above-mentioned data set with other well-known classification methods. The following methods are considered: the linear discriminant analysis (LDA), the logistic regression model (LOGIT), the probit method [12], the traditional SLP, the multi-layer perceptron (MLP), the support vector machine (SVM), the ELECTRE TRI method [22,24,27], and the fuzzy integral-based functional-link net (FIFLN) [17], which is a non-additive version of the functional-link net [1–3]. All these methods cover quantitative disciplines of statistics and machine learning. Hu and Tseng [17] reported the results obtained by the LDA, the LOGIT, the probit method, the SLP, the MLP and FIFLN. The results of SVM are obtained by performing the popular machine learning software WEKA 3.5.8. We summarize the average accuracy rates obtained by the above classification methods in Table 1. In particular, since many parameters are not easily specified by decision-makers, we employ the GA to implement the ELECTRI TRI method.

5.3.1. GA-based ELECTRE TRI method

The ELECTRE TRI method also employed the credibility index introduced in Eq. (8) (i.e., $S(i, j)$) to indicate the degree of credibility in the proposition $\mathbf{x}_i S \mathbf{x}_j$. A cut-off point $\lambda (0 \leqslant \lambda \leqslant 1)$ is given to decide whether $\mathbf{x}_i S \mathbf{x}_j$ holds or not. That is, $\mathbf{x}_i S \mathbf{x}_j$ holds when $S(i, j) > \lambda$ In other words, $\neg(\mathbf{x}_i S \mathbf{x}_j)$ holds when $S(i, j) \leqslant \lambda$. Although development of the proposed method and the ELECTRE TRI method are based on the ELECTRE methods, there are significant differences between the ELECTRE TRI method and the proposed model:

(1) In the proposed method, pairwise comparisons are performed among an input pattern and a set of training data, whereas in the ELECTRE TRI method, the same are performed among an input pattern and a set of reference profiles. The reference profiles are fictitious and delimit different classes.
(2) As depicted in Fig. 2, in the ELECTRE TRI method, the concordance and discordance indices are defined by three parameters of criterion $r (1 \leqslant r \leqslant n)$: preference (i.e., p_r), indifference (i.e., q_r) and veto (i.e., v_r) thresholds. The characteristic of such a discordance index is that, given $\mathbf{x}_a$ and $\mathbf{x}_b$, $\neg(\mathbf{x}_a S \mathbf{x}_b)$ holds when $x_{br} - x_{ar}$ is higher than the veto of criterion r (i.e., v_r). It is seen that, if $C_{S,r}(a, b) \neq 0$, then $D_{S,r}(a, b) = 0$, and vice versa.
(3) In the ELECTRE TRI method, two procedures are employed to assign the class label to a pattern: one is the pessimistic procedure, and the other is the optimistic procedure. For a two-class problem, there is a reference profile, say $\mathbf{f} = (f_1, f_2, \ldots, f_n)$. In the pessimistic procedure, the rule is that if $\mathbf{x}_i S \mathbf{f}$ holds then categorize the pattern into C_1, otherwise (i.e., $\neg(\mathbf{x}_i S \mathbf{f})$) categorize the pattern into C_2. In the optimistic procedure, the rule is that if $\mathbf{f} S \mathbf{x}_i$ and $\neg(\mathbf{x}_i S \mathbf{f})$ hold then categorize the pattern into C_2; otherwise categorize the pattern into C_1.

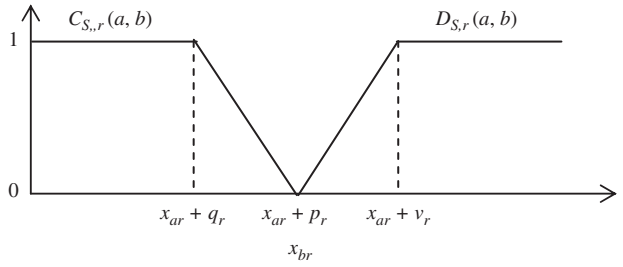

Fig. 2. The concordance and discordance indices in the ELECTRE TRI method.

The key issue of performing the ELECTRE TRI method is to determine the above-mentioned preferential parameters (i.e., criteria weights, preference, indifference, veto, profile) [20]. Since it is not possible to elicit optimal parameter specifications from the decision-makers, to construct a classifier denoted by *EST* using the ELECTRE TRI method, a GA-based learning algorithm with the objective of maximizing the number of correctly classified training patterns is implemented for computer simulation.

$$f(EST) = w_{NC} \cdot NC(EST) \tag{14}$$

where $NC(EST)$ denotes the number of correctly classified training patterns by *EST*. In the coding scheme, the binary chromosome k $(1 \leqslant k \leqslant n_{size})$ in P_i $(1 \leqslant i \leqslant n_{max})$ is represented by $w^i_{k1} w^i_{k2} \ldots w^i_{k,n}$ $p^i_{k1} p^i_{k2} \ldots p^i_{k,n}$ $q^i_{k1} q^i_{k2} \ldots q^i_{k,n}$ $v^i_{k1} v^i_{k2} \ldots v^i_{k,n}$ $f^i_{k1} f^i_{k2} \ldots f^i_{kn} \lambda^i_k$, where p^i_{kr}, q^i_{kr}, v^i_{kr}, and f^i_{kr} denote the preference, indifference, veto, and attribute value of the profile in the criterion r, respectively. That is, $5n+1$ binary substrings constitute a string. In practice, the outline of the learning algorithm for the ELECTRE TRI method is the same as those introduced in the previous section. For the ELECTRE TRI method, w_E is not required and the other parameter setting is the same as those for the ELECTRE-based SLP.

5.4. Comparative study

The generalization ability of each classification method is evaluated. The average testing performances of different classification methods are summarized in Table 1. As can be seen, the ELECTRE-based SLP gives satisfactory performance compared with all the other methods in all four splits of the analysis. In 80/20, 60/40 and 50/50 partitions, the testing performances obtained by the proposed model are higher than those obtained by the other classification methods. In the 70/30 partition, the proposed model has an average accuracy of 91.24% which is slightly inferior to that of the quadratic interval logit model. The proposed model provides higher accuracy, with the differences ranging between 0.39% (compared to FIFLN) and 13.85% (compared to SLP) in the 80/20 partition, ranging between 0.78% (compared to the LOGIT [13]) and 10.05% (compared to SLP) in the 70/30 partition, ranging between 1.51% (compared to SVM) and 13.62% (compared to SLP) in the 60/40 partition, and ranging between 0.92% (compared to the probit method) and 7.72% (compared to the functional-link net) in the 50/50 partition.

We employ the Friedman test [32] with the post-hoc tests, which was recommended by Demšar [31], to perform statistical

Table 2
Classification accuracy rates (%) of different classification methods for training patterns.

Classification method	Partition			
	80/20	70/30	60/40	50/50
LDA	89.13	87.69	88.14	89.11
Probit method	92.82	92.57	92.65	90.98
LOGIT	92.43	91.46	91.88	90.51
SLP	78.45	79.93	80.93	81.96
MLP	98.35	97.78	97.94	97.82
FIFLN	92.14	92.24	92.91	93.93
SVM	93.69	94.23	93.42	94.55
ELECTRE TRI (pessimistic)	88.93	88.80	88.64	89.11
ELECTRE TRI (optimistic)	88.64	88.24	88.39	89.25
ELECTRE-based SLP	94.71	94.40	93.75	95.24

comparisons of different classification methods over multiple partitions. The non-parametric Friedman test ranks the classification methods for each partition separately, the best performing method getting the rank of 1, the second best ranks 2 and so on. In case of ties, average ranks can be assigned. For instance, for 80/20 partition, the ELECTRE-based SLP and the FIFLN are ranked first and second, respectively. As a result, the ELECTRE-based SLP gets the ranks of 1 for all partitions. Thus, the average rank of the ELECTRE-based SLP is 1.0. The average rank of other classification methods can be obtained in a similar manner. Let r_j, k_1 and k_2 denote the average rank of the jth classification method, the number of classification methods and the number of data sets used, respectively. Under the null hypothesis, which is that the average ranks of classification methods are equal to each other, a statistic F_F distributed according to the F distribution with k_1-1 and $(k_1-1)(k_2-1)$ degrees of freedom is formulated by the Friedman statistic χ_F^2 [33].

$$F_F = \frac{(k_2-1)\chi_F^2}{k_2(k_1-1)-\chi_F^2} \tag{15}$$

where χ_F^2 is defined as

$$\chi_f^2 = \frac{12k_2}{k_1(k_1+1)}\left[\sum_{j=1}^{k_1} r_j^2 - \frac{k_1(k_1+1)^2}{4}\right] \tag{16}$$

$\chi_F^2 = 44.97$ can be thus derived by the data from Table 1 with $k_1 = 10$ and $k_2 = 4$. Then, $F_F = 16.04$ is easily obtained by χ_F^2. Since F_F is greater than the critical value $F(9, 39)$ at the 5% level, the null hypothesis is rejected. We thus proceed with a post-hoc test, the Nemenyi test [34].

The classification performance of two classification methods is significantly different if the difference of their average ranks is not below the critical difference at the α level

$$CD = q_\alpha\sqrt{\frac{k_1(k_1+1)}{6k_2}} \tag{17}$$

CD is equal to 6.25 when $q_{0.10} = 2.920$ (i.e., $\alpha = 0.10$) and $k_1 = 10$. We can identify that classification performance of the ELECTRE-based SLP is significantly better than that of the traditional SLP ($CD = 9.00$), the MLP ($CD = 7.50$), and the ELECTRE TRI method with the pessimistic ($CD = 6.63$) and the optimistic procedures ($CD = 6.25$). The fitting ability is further shown in Table 2. As can be seen, although the MLP outperforms the other classification methods, it suffers from over-fitting. Over-fitting for the ELECTRE-based SLP should not be a serious problem.

5.5. Simulations on other data sets

By the 10-fold cross-validation (10-CV), the generalization abilities of the ELECTRE-based SLP, the ELECTRE TRI method and the traditional SLP are further examined through computational experiments on two data sets, including the Wisconsin breast-cancer data consisting of 699 patterns with nine criteria and the Pima Indian diabetes data consisting of 768 patterns with eight criteria. For 10-CV, nine subsets are used as the training patterns and are tested on the single remaining subset. This procedure repeats until each of the ten subsets is tested. The 10-CV independently performs ten times using different partitions of a data set. The above data sets are available from the UCI machine learning repository on http://www.ics.uci.edu/~mlearn/MLRepository.html.

With the same parameter values as in the computer simulation for the thyroid data, the average classification rate of the ELECTRE-based SLP is 96.31%, and that of the traditional SLP, the ELECTRE TRI method with the pessimistic and the optimistic procedures are 96.09%, 95.12% and 95.07%, respectively, for the Wisconsin breast-cancer data. For the diabetes data, the average result of the ELECTRE-based SLP on is 74.86%, and that of the traditional SLP, the ELECTRE TRI method with the pessimistic and the optimistic procedures are 73.63%, 73.37% and 73.09%, respectively. The experimental results show that the ELECTRE-based SLP outperforms the traditional SLP and the ELECTRE TRI method.

5.6. Parameter specifications examination

First, we examine the significance of the crossover and mutation operations in the ELECTRI-based SLP by specifying either the crossover probability or the mutation probability as zero. The average classification rates are obtained by 10 independent trials with all 129 firms used as training data. The average classification rate of the ELECTRI-based SLP with both genetic operations (i.e., crossover and the mutation) over 10 trials is 94.38% after 500 generations, but with no crossover, it is 94.23%. Thus, it seems that the crossover operation does not have an effect on the training performance. Additionally, the average classification rate over 10 trials with no mutation is 78.29% after 500 generations. Since the result is much inferior to that obtained by the ELECTRI-based SLP with both genetic operations, it is

Table 3
Classification accuracy rates with various crossover probabilities.

Crossover probability	Classification rate (%)
0.4	94.69
0.5	94.30
0.6	94.30
0.7	94.77
0.8	94.53

Table 4
Classification accuracy rates with various mutation probabilities.

Mutation probability	Classification rate (%)
0.05	94.38
0.10	94.31
0.20	93.61
0.30	93.46
0.50	93.07
0.80	93.23

Table 5
Classification accuracy rates with various population sizes.

Population size	Classification rate (%)
20	94.30
50	94.38
80	94.46
100	94.84
200	94.99

reasonable to infer that the construction of the ELECTRI-based SLP is mainly driven by the mutation operation.

For the collected data of 129 patterns, computer simulations are further performed by changing one of the three parameter values (i.e., crossover probability, mutation probability, and the population size). The other parameters are specified in the same manner as in Section 5.2. The average classification rates obtained by 10 trials for each parameter specification are summarized in Tables 3–5. It can be seen that the ELECTRI-based SLP works well in a wide range of parameter values. This shows the robustness of the ELECTRI-based SLP with respect to the crossover probability, the mutation probability, and the population size.

This study examines the CPU time of the proposed SLP, using all the given samples in a data set as training data, obtained by a single trial. The ELECTRE-based SLP requires about 2, 28 and 42 min for the bankruptcy prediction data, the Wisconsin breast-cancer data and the diabetes data, respectively. The ELECTRE-based SLP seems to require long computing time for a pattern classification problem with larger size.

6. Discussion and conclusions

This paper presents a novel neural network named as the ELECTRE-based SLP with GA-based learning for addressing multi-criteria classification problems. The basis of the traditional SLP is the MAUT, whereas the ELECTRE-based SLP is developed using the outranking relations that involve pairwise comparison among patterns. Furthermore, each connection weight of the ELECTRE-based SLP can be interpreted as the degree of importance of the corresponding criterion, whereas the interpretation of each connection weight for the traditional SLP may be meaningless. The experimental results show that the ELECTRE-based SLP outperforms the traditional SLP in bankruptcy prediction.

The known ELECTRE TRI method is particularly addressed in this paper. As mentioned above, in the proposed model, the pairwise comparisons are performed between an input pattern that need to be classified and a set of training data, whereas the ELECTRE TRI method perform the pairwise comparisons between an input pattern and a set of reference profiles that distinguish the classes. We also develop a learning algorithm for the ELECTRE TRI method to automatically determine the values of thresholds. The experimental results also show that the ELECTRE-based SLP outperforms the ELECTRE TRI method.

In comparison with an arsenal of well-known classification methods involving multivariate techniques, machine learning tools and neural networks, the experimental results from an application to a real-world data set involving bankruptcy prediction are quite encouraging in terms of the classification performance obtained by the ELECTRE-based SLP. It seems that the ELECTRE-based SLP is an effective tool for financial distress analysis. Even so, it is not necessary to conclude which method is best, since there is no such thing as the "best" classifier [8]. The evaluation of the classification performance of the ELECTRE-based SLP is necessary considering additional real-world data sets.

Although the focus of this paper is the application of the proposed ELECTRI-based SLP to bankruptcy prediction, which is a dichotomous classification problem, it is necessary to extend the framework to other multi-class problems. A possible method is to consider the so-called One-Class-in-One-Network structure [15], where an ELECTRI-based SLP is devoted to one class only. The other method is to use multiple cut points. For two-class cases, as mentioned above, patterns with the net outranking index higher than the cut-off point θ are assigned into C_1, whereas patterns with the net outranking index lower than the cut-off point are assigned into C_2. The larger the net outranking index, the more preferred it is for a dichotomous classification problem. In general, for α-class cases (i.e., C_1, C_2,…,C_α) ($\alpha \geqslant 3$), $\alpha-1$ cut-off points, t_1, t_2,…,$t_{\alpha-1}$, are required to perform the classification of patterns. Each cut-off point is specified between -1 and 1. The classification of an input pattern $\mathbf{x}$ is performed according to the following rules [20]:

(1) $\mathbf{x}$ is assigned into C_1 when $S(\mathbf{x}) \geqslant t_1$
(2) $\mathbf{x}$ is assigned into C_r when $t_r \leqslant S(\mathbf{x}) < t_{r-1}$, $2 \leqslant r \leqslant \alpha-1$
(3) $\mathbf{x}$ is assigned into C_α when $S(\mathbf{x}) < t_{\alpha-1}$

where $S(\mathbf{x})$ denotes the net outranking index with respect to $\mathbf{x}$. To estimate cut-off points, t_1, t_2,…,$t_{\alpha-1}$ can be coded as individual substrings in the learning algorithm. Applications of the proposed SLP to multi-class problems will be explored in our future study.

Acknowledgement

The author would like to thank the anonymous referees for their valuable comments. This research is partially supported by the National Science Council of Taiwan under grant NSC 97-2410-H-033-015-MY2.

References

[1] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, Reading, 1989.
[2] Y.H. Pao, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (5) (1992) 76–79.
[3] G.H. Park, Y.H. Pao, Unconstrained word-based approach for off-line script recognition using density-based random-vector functional-link net, Neurocomputing 31 (1-4) (2000) 45–65.
[4] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, NJ, 1997.
[5] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[6] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms: Concepts and Designs, Springer, London, 1999.
[7] A.J.F. Rooij, L.C. Jain, R.P. Johnson, Neural Network Training Using Genetic Algorithms, World Scientific, Singapore, 1996.
[8] L.I. Kuncheva, Fuzzy Classifier Design, Physica-Verlag, Heidelberg, 2000.
[9] A. Osyczka, Evolutionary Algorithms for Single and Multicriteria Design Optimization, Physica-Verlag, NY, 2002.
[10] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Transactions on Systems, Man, and Cybernetics 35 (2) (2005) 359–365.
[11] E.L. Altman, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, Journal of Finance 23 (1968) 596.
[12] M. Zmijewski, Methodological issues related to the estimation of financial distress prediction models, Journal of Accounting Research 22 (1) (1984) 59–82.
[13] F.M. Tseng, L. Lin, A quadratic interval logit model for forecasting bankruptcy, Omega 33 (1) (2005) 85–91.
[14] B. Andrea, M. Sergio, T. Pietro, Neural Networks for Economic and Financial Modelling, International Thomson Computer Press, London, 1996.
[15] S.Y. Kung, Digital Neural Networks, Prentice-Hall, Englewood Cliffs, 1993.
[16] T. Murata, H. Ishibuchi, H. Tanaka, Multi-objective genetic algorithm and its applications to flowshop scheduling, Computers and Industrial Engineering 30 (4) (1996) 957–968.
[17] Y.C. Hu, F.M. Tseng, Functional-link net with fuzzy integral for bankruptcy prediction, Neurocomputing 70 (16-18) (2007) 2959–2968.

[18] B. Roy, The outranking approach and the foundations of ELECTRE methods, Theory and Decision 31 (1991) 49–73.
[19] P. Perny, Multicriteria filtering methods based on concordance and non-discordance principles, Annals of Operations Research 80 (1998) 137–165.
[20] M. Doumpos, C. Zopounidis, Multicriteria Decision Aid Classification Methods, Kluwer, Dordrecht, 2002.
[21] K.P. Yoon, C.L. Hwang, Multiple Attribute Decision Making: An Introduction, Sage Publications, London, 1995.
[22] W. Yu, ELECTRE TRI: Aspects methodologiques et manuel d'utilisation, Document du Lamsade No. 74, Universite de Paris-Dauphine, 1992.
[23] A. Teixeira de Almeida, Multicriteria decision model for outsourcing contracts selection based on utility function and ELECTRE method, Computers and Operations Research 34 (12) (2007) 3569–3574.
[24] R.P. Lourenço, J.P. Costa, Using ELECTRE TRI outranking method to sort MOMILP nondominated solutions, European Journal of Operational Research 153 (2) (2004) 271–289.
[25] G. Aiello, M. Enea, G. Galante, A multi-objective approach to facility layout problem by genetic search algorithm and ELECTRE method, Robotics and Computer Integrated Manufacturing 22 (5-6) (2006) 447–455.
[26] C. Zopounidis, M. Doumpos, Multicriteria classification and sorting methods: a literature review, European Journal of Operational Research 138 (2) (2002) 229–246.
[27] L. Dias, V. Mousseau, J. Figueira, J. Clímaco, An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI, European Journal of Operational Research 138 (2) (2002) 332–348.
[28] B. Roy, Classement et choix en présence de points de vue multiple (la méthode Electre), Revue d'Informatique et le Recherche Opérationelle 8 (1968) 57–75.
[29] M. Doumpos, C. Zopounidis, A multicriteria classification approach based on pairwise comparisons, European Journal of Operational Research 158 (2004) 378–389.
[30] J.W. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2006.
[31] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[32] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Annals of Mathematical Statistics 11 (1940) 86–92.
[33] R.L. Iman, J.M. Davenport, Approximations of the critical region of the Friedman statistic, Communications in Statistics (1980) 571–595.
[34] P.B. Nemenyi, Distribution-free multiple comparisons, Ph.D. Thesis, Princeton University, 1963.
[35] P.C. Pendharkar, J.A. Rodger, An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification, Computers and Operations Research 31 (2004) 481–498.
[36] P.C. Pendharkar, J.A. Rodger, Technical efficiency-based selection of learning cases to improve forecasting accuracy of neural networks under monotonicity assumption, Decision Support Systems (2003) 117–136.

Yi-Chung Hu received the B.S. degree in information and computer engineering from the Chung Yuan Christian University, Chung-Li, Taiwan, the M.S. degree in computer and information science from the National Chiao Tung University, Hsinchu, Taiwan, and the Ph.D. degree in information management from the National Chiao Tung University, Hsinchu, Taiwan, in 1991, 1993, and 2003, respectively. From October 1993 to May 1998, he worked as an engineer in the Aerospace Industry Development Corporation, Taichung, Taiwan. From June 1998 to January 2000, he worked as an assistant researcher in the Chung Shan Institute of Science and Technology, Taoyan, Taiwan. He was an assistant professor in the Department of Business Administration at the Chung Yuan Christian University, Chung-Li, Taiwan, from August 2003 to July 2006. He is currently an associate professor in the Department of Business Administration at the Chung Yuan Christian University, Chung-Li, Taiwan. His research interests include soft computing, multiple-criteria decision making, data mining and electronic commerce.