ELSEVIER

European Journal of Operational Research 136 (2002) 212–229

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Computing, Artificial Intelligence and Information Technology

Rule extraction from expert heuristics: A comparative study of rough sets with neural networks and ID3

Brenda Mak [a,*], Toshinori Munakata [b,1]

[a] *Department of Information Systems and Business Analysis, College of Business, San Francisco State University, San Francisco, CA 94132, USA*

[b] *Department of Computer and Information Science, Cleveland State University, Cleveland, OH 44115, USA*

Received 9 June 2000; accepted 4 January 2001

Abstract

The rule extraction capability of neural networks is an issue of interest to many researchers. Even though neural networks offer high accuracy in classification and prediction, there are criticisms on the complicated and non-linear transformation performed in the hidden layers. It is difficult to explain the relationships between inputs and outputs and derive simple rules governing the relationships between them. As alternatives, some researchers recommend the use of rough sets or ID3 for rule extraction. This paper reviews and compares the rule extraction capabilities of rough sets with neural networks and ID3. We apply the methods to analyze expert heuristic judgments. Strengths and weaknesses of the methods are compared, and implications for the use of the methods are suggested.

Keywords: Rough sets; Neural networks; Heuristics; Rule extraction

1. Introduction

Recent research progress in database technologies has created a significant interest in knowledge discovery in databases and data mining [1]. Knowledge discovery refers to the automation of knowledge extraction from large databases [7,8,27]. A wide variety of artificial intelligence techniques are used for rule induction from these large databases [20], and algorithms are developed to learn the regularities from the rich data [25]. These techniques include neural networks [10], ID3 [5,35], rough sets [70]. To assess the performance of the techniques, some researchers have emphasized the importance of consistency, robustness, and predictive accuracy [9,12]. Others, however, have focused on explanatory and exploratory capability, and emphasized the need to reveal data patterns that are valid, novel, useful, simple, understandable, significant [24], and interesting [3,48].

* Corresponding author. Tel.: +1-415-328-2140; fax: +1-415-405-0364.

E-mail addresses: bmak@sfsu.edu (B. Mak), munakata@cis.csuohio.edu (T. Munakata).

[1] Tel.: +216-687-3684; fax: +216-687-9354.

0377-2217/02/$ - see front matter
PII: S0377-2217(01)00062-5

Neural network is one of the most widely used artificial intelligence techniques for pattern recognition and machine learning. The neural network method is highly accurate in classification and prediction of data [4,57]. Different from the classical statistical method, it can be applied to analyze data with small sample sizes, without the need to satisfy the normal distribution assumption. Because of the non-linear transformation involved in its hidden layers, neural networks also perform well for modeling data with complicated patterns.

However, neural networks have been criticized for lack of explanatory power. In particular it is difficult to trace and explain the way the data pattern is derived, due to the complexity and non-linear nature of data transformation conducted in the multiple hidden layers. It is difficult to identify rules relating the inputs and outputs of a neural network, and assess the way each input contributes to the output of the network.

In response to these criticisms, many neural network researchers have developed ways to extract rules from neural networks. Researchers have used the weights, or connection strengths, to analyze the contribution of the inputs to the outputs and of the hidden layer elements to the outputs. For example, Setiono and Liu [46] have developed the RG algorithm to extract rules from neural networks [21]. They discretized and clustered the activation values of the hidden units and analyzed the weights connecting the inputs to hidden units and the weights connecting the hidden units to the outputs. In this way, they extracted rules relating the inputs to the hidden layer elements, and rules relating the hidden layer elements to the outputs. IF THEN ELSE rules were then extracted relating the inputs to the outputs by merging the two sets of rules. Similarly, Tan [59] also extracted rules from a neural network using the Cascade ARTMAP architecture by making inferences in multiple steps. The network was first parsed to derive the initial relationships among inputs, intermediate concepts, and outputs. These relationships were then used to set up Network A and Network B in the Cascade ARTMAP, and the relationships were further refined through mapping the two networks to one another.

Besides extracting rules to explain the relationship between inputs and outputs, researchers have also developed empirical indices to assess the contribution of inputs to outputs. Examples include the work of Garson [11], Yoon et al. [66], and Mak and Blanning [22]. These indices are measures showing the relative importance of the inputs in contributing to the outputs.

Compared to neural networks, rule extraction is relatively easy for rough sets and ID3. ID3 is a decision analysis technique based on the greedy algorithm of entropy reduction in constructing the decision tree. Attributes that would lead to substantial entropy reduction (or information gain) are included as condition attributes to partition the data and for the prediction of the decision outcome. A condition attribute that would induce the most amount of entropy reduction and information gain would be placed closer to the root and used for the next level partitioning. Sometimes filters may be set up so that only attributes with information gain greater than a certain threshold will be selected in constructing the decision tree. Variants of ID3 included Quinlan's C4.5 and C5 [44], which model both discrete and continuous variables in a decision tree. Rules extracted from the ID3, C4.5, or C5 decision trees can be used to predict new cases.

Rule distillation is also relatively efficient with rough sets compared to neural networks. The rough set method was introduced by Zdzislaw Pawlak for the analysis and pattern discovery in databases, in particular for data that are ambiguous or incomplete [37,39,40]. In the rough set methodology, a database is regarded as a decision table, which is made up of the universe of discourse, a family of equivalence relations over the universe, condition attributes and decision attributes. The rule discovery process in rough set analysis involves simplifying the decision tables with elimination of superfluous attributes and values of attributes, and finding out simple rules relating the condition and decision attributes. The measures assessing the contribution of condition attributes in affecting the decision attributes include dependency [36] and significance [15].

Since its introduction, the rough set method has increasingly been applied to derive rules, and to

provide reasoning and discover relationships in qualitative, incomplete, or imprecise data. This capability is especially important for business analysis, as a lot of case data in business is incomplete and imprecise. How do rough sets compare to neural networks in extracting rules from business heuristic data? Can rough sets model data as accurately as neural networks? Can rough sets discover rules much more efficiently than neural networks from expert heuristic judgments?

The objective of this paper is to compare the rule extraction capability of neural networks to rough sets and ID3. In particular we are interested in comparing the three methods in their classification accuracy, ease of rule extraction methods, and the way they analyze the contribution of inputs (condition attributes) to outputs (decision attributes). We apply these methods to analyze expert strategic judgments on new product entry. We begin in the following section with a brief explanation of the rough set concepts. In Section 3 we compare the rule extraction mechanisms involved in the three methods and derive an index to measure the contribution of condition attributes for rough sets. Section 4 explains the results of an empirical study to compare the three methods. Sections 5 and 6 discuss the results and implications for future research.

2. The rough set method

Introduced by Pawlak in the early 1980s [38], the rough set theory is a relatively new mathematical and artificial intelligence technique dealing with ordinary sets and relations. Its target objective can be somewhat similar to that of fuzzy theory [68], which deals with uncertain or approximate reasoning. But the two techniques differ concerning their approaches and objectives to target problems. The major objective of fuzzy set theory is to deal with complex problems by allowing gradual changes and descriptive expressions. The generic fuzzy set theory does not have learning capability and the forms of fuzzy membership functions are assumed in the analysis [36]. On the other hand, the major objective of rough set theory is to distill rules from data and make sense out of complex data. The membership functions are computed empirically from data.

According to Pawlak, the rough set theory is built on the assumption that information can be associated with every object in the universe. Objects characterized by the same amount of information are similar, or indiscernible, to one another. A set of indiscernible objects is called the elementary set and a crisp set refers to a union of some elementary sets. Objects that are not made up of elementary sets belong to a rough, or imprecise, set. They can be characterized by the lower and upper approximations of rough set. The lower approximation consists of objects that belong to the set with certainty while the upper approximation contains all objects that may possibly belong to the set.

The concepts of indiscernible relations and approximations can best be illustrated as follows. Consider an information system that has a rule base made up of expert judgments. Experts are given the values of condition attributes for market scenarios, and are asked to specify the new product entry strategies they would use for each of these scenarios. Based on the judgments of the experts, an information table could be made up relating the new product entry decisions to the market scenarios. In Table 1, seven cases are presented showing three condition attributes: G, F, L, and one decision attribute K. The associated values and meaning of the attributes are: G, demand growth rate (high or low), F, Financial strength (strong or weak), L, Cost of development (high or low), and K, entry strategy (GO or NOGO).

Each new product entry case can be characterized in terms of the three condition attributes G, F, and L. Cases 4 and 5 are indiscernible in terms of attributes G, F, and L, since the values for these attributes are the same. Similarly, cases 1 and 6 are indiscernible in terms of F and L, and so are cases 4 and 7.

The entry strategies for the first four cases are GO, that is, launch the new product, while the strategies for the rest are NOGO, that is, do not launch the product. Let us try to describe set $\{1,2,3,4\}$ and $\{5,6,7\}$ in terms of attributes G, F, and L. As observed from cases 4 and 5, which have the same attribute values but differ in their entry

Table 1
The new product entry decision

Case	Demand growth rate (G)	Financial strength (F)	Cost of development (L)	Entry strategy (K)
1	High	Strong	Low	GO
2	High	Strong	High	GO
3	High	Weak	Low	GO
4	High	Weak	High	GO
5	High	Weak	High	NOGO
6	Low	Strong	Low	NOGO
7	Low	Weak	High	NOGO

strategies, we understand no exact answer can be derived based on the attribute relationships shown in Table 1. However, approximate answers may be obtained. We may conclude that cases 1, 2, and 3 surely use a GO strategy and belong to the set $\{1,2,3,4\}$, while cases 6 and 7 surely use an NOGO strategy and belong to the set $\{5,6,7\}$. Thus the set $\{1,2,3\}$ is the lower approximation of the set $\{1,2,3,4\}$. On the other hand, cases $1,2,3,4,5$ may possibly employ a GO strategy. Thus the set $\{1,2,3,4,5\}$ is the upper approximation of the set $\{1,2,3,4\}$. The set $\{4,5\}$, the boundary region of the set $\{1,2,3,4\}$, is the difference between the upper and lower approximations.

Let U be a finite set of objects called the universe. If $R \subseteq U \times U$ is an equivalence relation on U, then $S = (U,R)$ is called an *approximation space*. If $u, v \in U$ and $(u,v) \in R$, we say that u and v are *indistinguishable* in S. R is called an *indiscernibility relation*. Let $R^* = \{X_1, X_2, \ldots, X_n\}$ denote the partition induced by R, where X_i is an equivalence class of R. X_i is also called an *elementary set* of S. Any finite union of elementary sets is called a *definable set*. Let X be any subset of U. Then we define the following:

$$\underline{S}(X) = \cup_{Xi \subseteq X} X_i$$
the *lower approximation* of X in S,
$$\bar{S}(X) = \cup_{Xi \cap X \neq \emptyset} X_i$$
the *upper approximation* of X in S.

$\underline{S}(X)$ is the union of all the elementary sets of S, where each elementary set is totally included (i.e., a subset) in X. $\bar{S}(X)$ is the union of all the elementary sets of S, where each elementary set contains at least one element in X. Using the lower and upper approximations discussed above, we can characterize the approximation space $S = (U,R)$ in terms of the concept X with three distinct regions defined as follows:

1. the *positive region*: $\mathrm{POS}_S(X) = \underline{S}(X)$,
2. the *boundary region*: $\mathrm{BND}_S(X) = \bar{S}(X) - \underline{S}(X)$,
3. the *negative region*: $\mathrm{NEG}_S(X) = U - \bar{S}(X)$.

Let C be the set of all condition attributes and let $A \subseteq C$, i.e., A is a set of condition attributes. For example, $A =$ {Demand growth rate, Financial strength, Cost of market development}. Let B be the set of all decision attributes and let $B \subseteq D$, i.e., B is a set of decision attributes. For example, $B =$ {Entry Strategy}. Let $\tilde{A}$ be an equivalence relation on U such that $\tilde{A} = \{(u,v) | u$ and v have the same value for every attribute in $A\}$. Similarly, we define an equivalence relation $\tilde{B}$ for set B. Let $A^* = \{X_1, \ldots, X_n\}$ and $B^* = \{Y_1, \ldots, Y_m\}$ denote the partitions on U induced by the equivalence relations $\tilde{A}$ and $\tilde{B}$, respectively. To determine the extent that partition B^* can be approximated by partition A^*, we define the *positive*, *boundary*, and *negative* regions of the partition B^* as follows:

$$\mathrm{POS}_S(B^*) = \cup_{Y_j \in B^*} \underline{S}(Y_j) = \cup_{Y_j \in B^*} [\cup_{X_i \subseteq Y_j} X_i],$$

$$\mathrm{BND}_S(B^*) = \cup_{Y_j \in B^*} (\bar{S}(Y_j) - \underline{S}(Y_j))$$
$$= \cup_{Y_j \in B^*} [\cup_{X_i \cap Y_j \neq \phi} X_i - \cup_{X_i \subseteq Y_j} X_i],$$

$$\mathrm{NEG}_S(B^*) = U - \cup_{Yj \in B^*} (\bar{S}(Y_j))$$
$$= U - \cup_{Y_j \in B^*} [\cup_{X_i \cap Y_j \neq \phi} X_i].$$

If the boundary region is the empty set, that is, BNDs $(B^*) = \phi$, the set B^* is denoted as the crisp set with respect to S. Alternatively, if BNDs $(B^*) \neq \phi$, then the set B^* is called the rough set with respect to S. The coefficient α_s (B^*) is known as the

accuracy of approximation and is given by the ratio of the cardinality of the lower approximation, $|\underline{S}(X)|$, to the cardinality of the upper approximation, $|\overline{S}(X)|$, and $0 \leqslant \alpha_s(B^*) \leqslant 1$. If $\alpha_s(B^*) = 1$, B^* is crisp with respect to S. If $\alpha_s(B^*) \leqslant 1$, B^* is rough with respect to S. The accuracy of the set $\{1, 2, 3, 4\}$ shown in Table 1 is 3/5 or 0.6.

Besides identifying indiscernibility relationships and equivalent classes to approximate data, data reduction can also be achieved through keeping the attributes that are required to preserve the indiscernibility relation. Let B be a non-empty subset of C, the set of condition attributes. B is said to be a *reduct* of C if B is a maximal independent set of condition attributes [29]. There is no superfluous attribute in B and all attributes are indispensable. Features found in the intersection of all reducts of the information system are known as the core of the information system.

One critical challenge in the development of decision tables is the choice of condition attributes to be included. Reduct approximation is one approach employed to solve the problem. The method involves computing reducts for some random subsets of the universe of a given information system and finding the most stable reducts that occur most frequently. These conditional attribute sets that occur "sufficiently often" as reducts of samples of the original decision table are known as dynamic reducts. The thresholds for "sufficiently open" are normally determined based on the results of empirical experiments. Øhrn, Komorowski, Skowron, and Synak successfully applied the dynamic reduction approach in the ROSETTA system [19,33,69].

3. Rule extraction capabilities of rough sets, ID3, and neural network

In this section, we compare the rule extraction capabilities of rough sets, ID3, and neural networks. We review the contribution measures and derive a heuristic measure assessing the contribution of input variables (condition attributes) to output variables (decision attributes) for rough sets.

3.1. Rule extraction and measures of input contribution in rough sets

Unlike neural networks, rule extraction in rough sets is relatively simple and straightforward, and no extra computational procedures are required before rules can be extracted. Rough set analysis involves diagnosis of equivalence relations and partitions of common knowledge events in order to extract the minimal set of condition attributes (the reduct) that are required for the decision [45]. The process of rough set analysis generates a decision table made up of essential attributes with rules that can be readily applied to guide decision-making. The decision table generated based on the attributes in the reduct can be used to help the decision maker to concentrate on the most essential factors and assist in solving multi-attribute decision problems [39]. For example, based on the information from Table 1, for the condition attributes $\{G, F, L\}$, we can obtain the reduct $\{G, F\}$. We can also derive the following decision rules: (1) If demand growth rate is high, and financial strength is strong, then GO. (2) If demand growth rate is low, then NOGO.

Rough set analysis has been successfully applied in medical diagnosis [63], industrial control [28], and marketing analysis [70]. The method has been used in credit card application analysis [39], prediction of company acquisition [54] and business failures [6], and evaluation of firm bankruptcy risk [53]. It has also been used to analyze the essential and distinctive factors in voting [31]. The method can be extended to deal with choice and ranking problems through the development of pairwise comparison table to represent preference binary relations [13].

3.2. Measures of contribution of condition attributes in rough sets

Various measures can be defined to represent how much B, a set of decision attributes, depends on A, a set of condition attributes. One of the most common measure is the dependency [26,36,56]. The *dependency* of B on A, denoted as $\gamma_A(B)$, is a

plausible measure of how much B depends on A and is defined as follows:

$$\gamma_A(B) = |\mathrm{POS}_S(B^*)|/\mid U \mid,$$

where $(S = (U, \tilde{A}))$ is the approximation space, and $|\ |$ denotes the cardinality (i.e., the number of elements) of a set. Note that $0 \leqslant \gamma_A(B) \leqslant 1$. In particular, three situations arise:

1. $\gamma_A(B) = 1$: B is *totally dependent* on A. i.e., A functionally determines B.
2. $\gamma_A(B) = 0$: A and B are *totally independent* of each other.
3. $0 < \gamma_A(B) < 1$: B is *roughly dependent* on A.

In general, the dependency of B on A can be denoted by $A \rightarrow_\gamma B$. For example, $A \rightarrow_1 B$ if B is totally dependent on A. An alternative to the dependency measure is the *discriminant index*, $\beta_A(B)$, which measures the degree of certainty in determining whether elements in U are elements of B or not, or the amount of uncertainty removed by selecting S [56]:

$$\begin{aligned}\beta_A(B) &= |\mathrm{POS}_S(B^*) \cup \mathrm{NEG}_S(B^*)|/|U| \\ &= |U - \mathrm{BND}_S(B^*)|/|U|.\end{aligned}$$

The significance of B on a is the difference between the dependency of B on the set of all condition attributes C and the dependency of B on the set of all condition attributes *without* the specific attribute a [14,15]. That is, the significance measures the importance level of the attribute by considering how a *deletion* of the attribute a from the entire set of condition attributes affects the dependency. The *significance* of B on a specific condition attribute a, $\sigma_A(B)$, can be defined by using the dependencies as follows:

$$\sigma_{\{a\}}(B) = \gamma_C(B) - \gamma_{C-\{a\}}(B),$$

where $\gamma_{C-\{a\}}$ (B) is the "complement dependency" of $\{a\}$ with respect to C. We can further extend the significance of $\{a\}$ to A, a set of any number of condition attributes, which indicates the significance of the set of condition attributes A:

$$\sigma_A(B) = \gamma_C(B) - \gamma_{C-A}(B).$$

The dependency and discriminant index are direct measures focusing on the direct contribution of one or more condition attributes. On the other hand, the significance is a complementary measure that assesses the importance of the condition attribute based on backward elimination of the attribute from the entire set of condition attributes. In certain situations, such as the case we have in Table 1, we may not be able to assess the dependency and discriminant index of F, Financial strength, and L, Cost of development. This is because the data are incomplete and have few elements in the positive regions of F and L. Under such circumstances, only the significance measure can be computed for F and L based on the backward elimination of variables. To help assess the contribution of condition attributes to data like these, we derive a heuristic measure for the contribution index.

3.3. Deriving a contribution index for rough sets

In this section, we introduce a new measure called a contribution index, denoted by t. It is a heuristic, gross approximation measure derived to assess the contribution of attributes in sparse data, where there are many condition attributes and the values of condition attributes are incomplete. As a result, very few elements are found in the positive regions of these condition attributes. Coarse and sparse data are often found in the real world. Examples include business case data and expert heuristic data. This type of data is incomplete, coarse, and sparse, with many decision variables affecting the decision outcome, and has few elements in positive regions. Mining of expert knowledge presents difficulties when the data collected from experts are coarse and sparse. Expert knowledge, however, is useful information to be stored. To better extract rules and enhance understanding on this type of data, we derive the contribution index, which is a simple and computationally efficient measure used to assess the contribution of condition attributes to the decision attribute in sparse data. The algorithm for deriving this heuristic measure consists of two phases. First we compute a measure based on the row-wise and table-wise goodness of each attribute. Second, we re-examine the entire database and perform adjustment if necessary.

Fig. 1 shows the algorithm we employ to derive this contribution index. Let condition attributes be v_i, $i=1$ to I, and $j_i = 1$ to J_i be the domain of possible values that attribute v_i can take. We assume one decision attribute for the product entry strategy; possible values of the entry strategy are $k = 1, \ldots, K$. Our target database consists of many *cases*, where each case has the form of "if $v_1 = j_1, v_2 = j_2, \ldots, v_I = j_I$, then choose k". Let n_{ijk} be the number of cases in the database for attribute v_i assuming a value of j ($v_i = j$) choosing strategy k. We can view n_{ijk} as the jth row, kth column element of matrix (table) M_i representing v_i. (We dropped the subscript i of j_i to avoid subscript i of subscript j in the following. This does not cause any problem since j for v_i always means j_i.) Let N be the total number of cases; then $N = \sum_j \sum_k n_{ijk}$ for every $i = 1$ to I.

In our product entry problem, $J_1 = J_2 = \cdots J_I = J$, and all the condition attributes have binary values. In general, when J_i are different for different i, we must replace J in the following with J_i. For each pair of i and j, that is, for each row j of matrix M_i, we determine $\max_{ij}$ and $\min_{ij}$ defined as follows:

$$\max_{ij} = \text{the maximum of } n_{ij1}, n_{ij2}, n_{ij3}, \ldots, n_{ijK} \text{ among the } K \text{ strategies},$$

$$\min_{ij} = \text{the minimum of } n_{ij1}, n_{ij2}, n_{ij3}, \ldots, n_{ijK} \text{ among the } K \text{ strategies}.$$

Define $r_{ijk} (0 \leqslant r_{ijk} \leqslant 1)$ as

$$r_{ijk} = \begin{cases} \dfrac{(\max_{ij} - \min_{ij})}{\sum_{k'} n_{ijk'}} & \text{for } k \text{ that corresponds to } \max_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

The term r_{ijk} is a measure to discriminate among K strategy outcomes for specific values of i and j (row-wise goodness). The higher the r_{ijk}, the better the ability of a specific row j of variable v_i (with value j) in discriminating among the possible values of the decision attribute. Next, consider the positions of r_{ijk} for different values of j in the

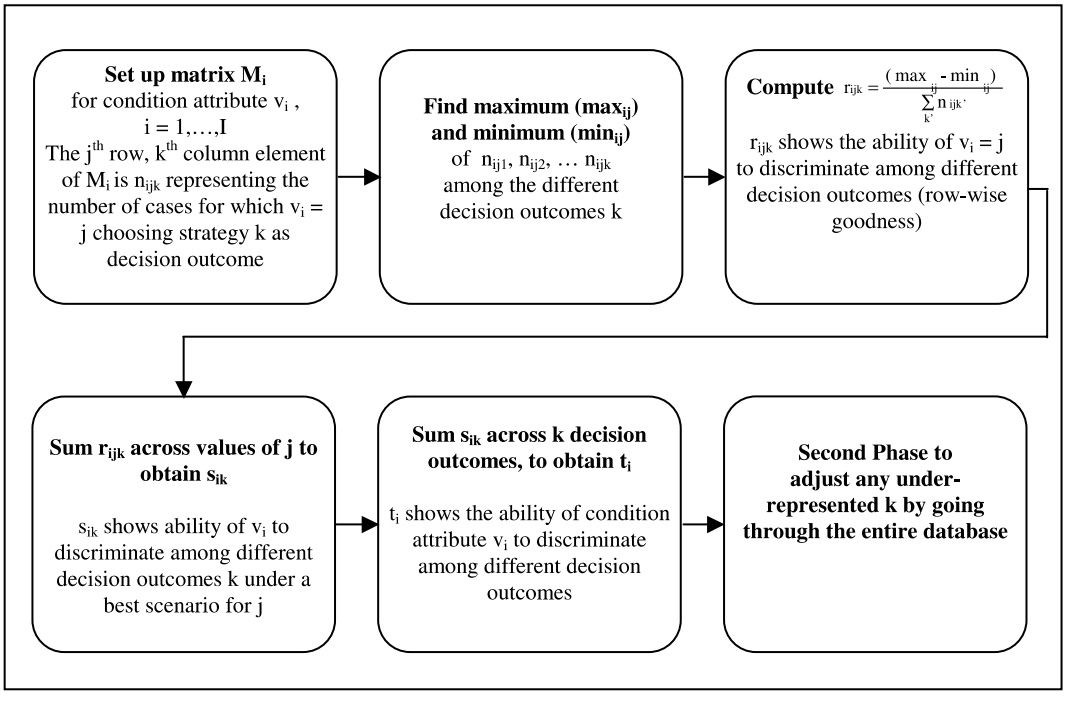

Fig. 1. Algorithms for deriving the contribution index in decision tables.

matrix; more diverse matrix positions over different values of k would indicate a better discriminatory ability of variable v_i. For example, the more diverse matrix in Table 2 below indicates a better discriminatory ability than the one in Table 3.

Define $s_{ik} = \max_j r_{ijk}$ for each strategy k. If there is a tie, arbitrarily choose one. Here s_{ik} can be thought of representing how well strategy k may be discriminated under the best scenario for j. Finally, we define the *contribution index*, t_i, as $t_i = \sum_k s_{ik}$, $k = 1, \ldots, K$. t_i indicates the overall contribution of the different values of variable v_i in distinguishing among all the k strategies (table-wise goodness). The computation of t_i is illustrated using the following example. Suppose the values of n_{ijk} are as given in Table 4, then $t_i = \sum_k s_{ik} = (3/5) + (1/2) = 1.1$.

Based on the values of r_{ijk} shown in Table 5, we can see that the strategy $k = 2$ is under represented. The following additional algorithm is employed to adjust for these under represented strategies.

1. Find k such that $s_{ik} = 0$ for all $i = 1$ to I in the entire database.
2. For those k's found in Step 1, check if n_{ijk} (or n_{ijk}/N) is zero or very small. If so, such k's are insignificant, and they can be discarded. If not, these k's can be re-examined. Go to step 3.

Table 2
Diverse position of r_{ijk} indicates better discriminatory ability

r_{ijk}	$k = 1$	$k = 2$	$k = 3$
$J = 1$	0	r_{i12}	0
$J = 2$	r_{i21}	0	0

Table 3
Less diverse position of r_{ij} indicates worse discriminatory ability

r_{ijk}	$k = 1$	$k = 2$	$k = 3$
$J = 1$	0	r_{i12}	0
$J = 2$	0	r_{i22}	0

Table 4
An example to illustrate the computation of t_i – values of n_{ijk}

n_{ijk}	$k = 1$	$k = 2$	$k = 3$	$\sum_k n_{ijk}$
$j = 1$	3	2	0	5
$j = 2$	1	0	1	2

Table 5
An example to illustrate the computation of t_i – values of r_{ijk} and s_{ik}

r_{ijk}	$k = 1$	$k = 2$	$k = 3$
$j = 1$	3/5	0	0
$j = 2$	1/2	0	1/2
s_{ik}	3/5	0	1/2

3. For each k to be re-examined, find max n_{ijk} for $i = 1$ to I, $j = 1$ to J, going over the entire database. The specific values of *i, j, k* found will be "representative" for decision k. Even though this k may not be dominant in the first round of evaluation, the specific (i,j,k) is considered important for leading to this k (otherwise, no rule may be obtained leading to strategy k).
4. To incorporate the above, one simple way is to insert new $r^*_{ij} = n_{ijk} / \sum_k n_{ijk}$ in the previous matrix, and update s_{ik} and t_i accordingly. Or, the new r^*_{ij} can be multiplied by a constant factor, λ, greater than or smaller than 1, to emphasize or de-emphasize the effect of k.

For example, refer to Table 5, where $s_{i2} = 0$. That is, there is no case that leads to $k = 2$ for this i. Suppose that $s_{i2} = 0$ for all $i = 1, \ldots, I$. This means that no variable leads to strategy $k = 2$ in the entire database. Suppose further that the entry 2 for $j = 1$ and $k = 2$ in Table 4 is the maximum n_{ijk} for $i = 1$ to I and $j = 1$ to J. Then we insert $r^*_{ij} = 2/5$ into $k = 2$ as follows. The new adjusted table is shown in Table 6. The updated t_i becomes $3/5 + 2/5 + 1/2 = 1.5$.

There are other measures to evaluate the importance of each condition attribute toward determining decision attribute values. The entropy in information theory employed by ID3 introduced below is a possible measure, somewhat similar to the contribution index. To illustrate the similarity, we may consider two extreme case scenarios. The first scenario is the worst case for the importance

Table 6
Adjusted table for under-represented $k = 2$

r_{ijk}	$k = 1$	$k = 2$	$k = 3$
$j = 1$	3/5	2/5	0
$j = 2$	1/2	0	1/2
s_{ik}	3/5	2/5	1/2

of a variable. In Table 4, suppose the entries for $j = 1$ are 3, 3 and 3 for $k = 1$, 2 and 3, respectively. This means that experts' opinions are equally divided among which strategy to choose. Similarly, suppose that the entries for $j = 2$ are also equally split as 2, 2, and 2. Such a variable would be considered useless since it does not lead to any definite conclusion no matter what value it assumes. In this case, the entropy gain will be 0. Our r_{ijk} will all be zero, and hence the contribution index will also be zero. The second scenario is the best case. In Table 4, suppose the entries for $j = 1$ are 9, 0, and 0, and for $j = 2$ are 0, 0, and 6. Both entropy gain and contribution index will be maximal.

3.4. Rule extraction and measures of input contribution in ID3

Instead of decision tables, ID3 [35] involves the development of decision trees to classify examples and make predictions for discrete class intervals. Classification is based on recursive partitioning of the data set into categories involving intersection among the variables in various values. At each node of the decision tree, the remaining variables with the highest reduction in entropy, or highest information gain, would be selected for the next stage of partitioning. Entropy values are used to indicate the contribution of the partitioning variables to the output. The entropy values of information of a set of examples K are:

$$E(K) = \sum_{i=1}^{m} P_i \log m(1/P_i) = -\sum_{i=1}^{m} P_i \log mP_i,$$

where P_i is the ratio of class K_i in the set of examples K. When class K_i partitions set K with attribute T_j, value of information $E(T_j)$ is:

$$E(T_j) = \sum_{i=1}^{|X_j|} w_i \times E(K_i'),$$

where K_i' is the lower level of examples with partition based on attribute T_j, and w_i is the ratio of the number of examples in K_i' to the number of examples in K. Information gain obtained by decision tree classifying attribute T_j from examples K is

$$\text{Gain}(T_j) = E(K) - E(T_j).$$

Decision rules are developed based on relationships in decision tree generated with ID3. New cases can be mapped to the decision tree to match to the most similar case for prediction purposes. When the number of alternatives and condition attributes is small, a decision tree may be a better decision aid than a decision table as it provides a better graphical overview of the alternatives available. However, as the number of alternatives gets large, it may be too complicated to represent all the alternatives with a decision tree, and ID3 may focus on extracting important rules based on the entropy criterion.

Quinlan has developed C4.5, an extension of ID3 that deals with missing values, continuous attribute value ranges, pruning of decision trees, and rule derivation [44,61]. Quinlan has further enhanced C4.5 and designed C5.0 to analyze substantial databases containing thousands of records (reference: www.rulequest.com). It is faster and easier to use than C4.5, and allows the user to assign variable misclassification costs. Quinlan has also developed the Cubist model to analyze continuous data with piecewise linear regression models. Regression trees are used in Cubist, with regression equations for variables broken into different intervals for the purpose of predictions. In addition, Quinlan also developed the Magnum Opus regime that finds association rules from the data.

3.5. Rule extraction and measures of input contribution in neural networks

Neural network is one of the most widely used artificial intelligence techniques for pattern recognition and machine learning. However, rule distillation in neural networks involves complicated analysis. Researchers have analyzed the weights connected to an input and weights to elements in the hidden layer in order to understand the contribution of inputs and the impact of hidden units on the output. For example, Omlin and Giles [32] developed a recurrent network and used the weights to analyze the rules guiding the state transition behavior of the hidden units. They an-

alyzed how the recurrent nodes in the hidden layer transit from one state to another.

Setiono and Liu [46] have developed the Neurorule to prune neural networks and RG algorithms to extract rules from neural networks. They assessed the activation values of hidden units in a three-layer feedforward network to establish the relationship between the inputs and outputs of a network. They used an oversized network and removed redundant connections and hidden units by pruning. The backpropagation algorithm of the neural network minimized a function consisting of the entropy function of the output and a penalty term used for weight decay. In this way, the network was pruned to maintain only salient connections. The activation values of the hidden unit were discretized and clustered in order to generate simple rules relating the network outputs to the hidden units. Rules were also generated relating the discretized activation values of hidden units to the inputs. By merging the two sets of rules, the rules that relate the inputs and the outputs were established.

They illustrated the rule extraction mechanism with an Iris species data set containing fifty examples each of the classes Iris setosa, Iris versicolor, and Iris virginica. Their networks consisted of thirty-nine input units (made up of discrete intervals of the attributes sepal-length, sepal-width, petal-length and petal-width), three hidden units, and three output units. An example of a rule relating the hidden units H_1 and H_2 to the outputs is as follows.

IF $H_2 = 1$, THEN Iris setosa

ELSE IF $H_1 = 0$ AND $H_2 = -0.5$, THEN Iris versicolor

ELSE Iris virginica.

By analyzing the weights to the hidden units to generate rules relating the inputs to the hidden units H_1 and H_2, and merging the two sets of rules together, they obtained the following rules relating the inputs to the outputs.

RULE 1: IF Petal-length $\leqslant 1.9$ THEN Iris setosa.

RULE 2: IF Petal-length $\leqslant 4.9$ AND Pental-width $\leqslant 1.6$, THEN Iris versicolor. DEFAULT RULE: Iris virginica.

These rules were found to be comparable with the decision tree rules generated from C4.5.

Similar to the RG rule extraction mechanism, the Cascade ARTMAP architecture developed by Tan [59] also involved making multi-step inferences with intermediate concepts. Rule extraction proceeded in two phases. The network was first parsed to derive the relationships among inputs, intermediate concepts, and outputs. Using X_1, X_2, and X_3 as input attributes, Y as an intermediate concept, and K as an output attribute, the following rules were generated.

RULE 1: IF X_1 AND X_2 THEN Y

RULE 2: IF Y AND X_3 THEN K.

These relationships would then be used to set up the Cascade ARTMAP in the second phase, with X_1, X_2, X_3, Y, and K as the inputs to Network A, and X_1, X_2, X_3, Y, and K as the outputs to Network B, and Network A was mapped to Network B. Further training of the ARTMAP refined the antecedent and consequent of the rules.

Neural networks have also been used to model relationships in a decision tree. Ivanova and Kubat [17] proposed a method to initialize a feedforward network with prior knowledge of logical relationships among inputs obtained through ID3 to help network training. Instead of using sigmoid activation functions, they used threshold units of logic functions. They developed a network with an input layer, a hidden layer of AND logic threshold relationships, and an output layer of OR logic threshold relationships. They showed that this hybrid network has better performances than network learning alone or C4.5 alone.

Besides the efforts to extract rules from neural networks relating the inputs to the outputs, researchers have also analyzed the contribution of inputs to the outputs of a neural network [47]. For example, indices have been developed by Yoon et al. [66,67] and Garson [11] to assess the contribution of the inputs based on the w_{ji} and v_{jk} weights of a network which has stabilized in training. Yoon et al.'s measure of the relative contribution of input i on output k is

$$\frac{\sum_{j=1}^{J} W_{ji} V_{jk}}{\sum_{i=1}^{I} \left| \sum_{j=1}^{J} W_{ji} V_{jk} \right|},$$

where input $i = 1, \ldots, I$, and hidden unit $j = 1, \ldots, J$, output $k = 1, \ldots, K$, and $|\ \ |$ denotes

taking the absolute values of the summation of the product of the weights w_{ji} and v_{jk}.

Similarly, Garson has developed an index that also uses the weights to assess the contribution. However, Garson's index is based on comparison of the absolute values of the weights and ignores the direction of the influence. Garson's measure of contribution of input i on output k is

$$\frac{\sum_{j=1}^{J} \frac{|W_{ji}||V_{jk}|}{\sum_{i=1}^{I}|W_{ji}|}}{\sum_{i=1}^{I}\sum_{j=1}^{J} \frac{|W_{ji}||V_{jk}|}{\sum_{i=1}^{I}|W_{ji}|}}.$$

Both the measures of Garson and Yoon have considered the effect of change of the input on the hidden unit (indicated by w_{ji}) and the effect of change of the hidden unit on the output (indicated by v_{jk}), but ignored the changes taking place at the hidden layer. To address the rates of change across the hidden layer, Mak and Blanning [22] developed the following index to assess the impact of input i on output k:

$$\frac{\sum_{j=1}^{J} V_{jk}\beta_j W_{ji}}{\sum_{i=1}^{I}\left|\sum_{j=1}^{J} V_{jk}\beta_j W_{ji}\right|},$$

where $\beta_j = 1/T\sum_{t=1}^{T} z_{jt}(1-z_{jt})$ and t is the number of trials of the training set, $t = 1,\ldots,T$. $z_j(1-z_j)$ is the rate of change of output of the jth hidden element with respect to its input, and β_j is the average of $z_j(1-z_j)$ across the trials of the training set. This measure is based on differentiation of the output of the network with respect to the input by considering the transformation in the hidden layer as well. This measure can be disaggregated into J measures, where J is the number of elements in the hidden layer, in order to assess the unique contribution of each of the hidden layer elements.

4. An empirical comparison of ID3, rough sets, and neural networks

In this section, we compare the capability of ID3, rough sets, and neural networks in analyzing expert heuristics on new product entry. We note that each of these three methods has many versions and variations. Here we apply the typical version of the methods. Our focus is to obtain an overall idea of the general performance of the three methods in rule extraction. We select expert heuristic data as our target data, partly because rule extraction is important for this type of data. It is also because heuristic data are incomplete and imprecise, and resemble the properties of a lot of important data that we would like to mine in the business world.

We compare the methods with respect to their classification accuracy and predictive accuracy. We also compare the contribution of the input variables (condition attributes) in affecting the output variable (decision attribute). The decision attribute, or output variable, is the product entry strategy. In addition to the three condition attributes displayed in Table 1, four additional condition attributes were used. The seven condition attributes were used as a framework to elicit knowledge from the subjects. These variables and their values are: the position of the firm (dominant or small), the financial strength of the firm (strong or weak), the expected demand growth (high or low), the product's life cycle (long or short), diffusion across competitors (fast or slow), cannibalization (high or low), and the cost of market development (high or low). The heuristics were obtained from 36 senior MBA students who were experienced in strategic analysis. In total 233 cases were collected, 105 cases were used for training and 128 cases for testing. Details of the knowledge acquisition process are found in [22,23].

The rough sets and ID3 methods were applied to analyze the data, and to develop decision tables and decision trees. A neural network of one hidden layer with three hidden units was constructed to analyze the data. This structure was adopted because it was the most parsimonious neural network architecture used for modeling the data, and addition of additional elements in the hidden layer did not result in increase in accuracy [22].

Table 7 shows the robustness and predictive ability of rough sets. Here, robustness refers to the rate of correct prediction for the original training data set, while predictive ability refers to the rate of correct prediction for the test data set. The robustness of rough sets is the percentage of number of elements in the positive regions in the training

Table 7
Robustness and predictive accuracy of rough sets

	Robustness		Predictive accuracy	
	1	0	1	0
1	57	10	70	18
0	10	28	17	23
Hit rate	80.95%		72.66%	

Table 8
Classification accuracy of ID3, rough sets, and neural networks

	ID3 (%)	Rough sets (%)	Neural network (%)
Robustness	83.8	80.95	89.52
Predictive accuracy	72.66	72.66	71.88

data. Out of the 105 training cases, 57 are in the positive region with strategy $k = 2$, while 28 are in the positive region with strategy $k = 1$. Thus the robustness of the rough set method is 85/105 (81%).

Based on the models obtained from the training data, the generalizability of the three methods was analyzed with the test data. In predicting the test data with the rough set method, existing rules are matched to a test case. An exact rule is applied to make the prediction for the test case if a match is found. However, if no rules can be found to match the test case, the variables in the test case are modified based on its contribution index until a match occurs between the modified case and one of the rules. Then the rule is applied to predict this modified test case. The predictive ability is then computed based on the percentage of test cases correctly predicted. It is about 73%.

Table 8 shows the comparison of the classification accuracy of the three methods. The neural network performs best in robustness (90%), and has predictive ability slightly worse than those of rough sets and ID3.

Next we compare the contribution of input measures for ID3, neural networks, and rough sets. The three indices developed by Garson, Yoon, Mak and Blanning, were computed to assess the contribution of the input elements to the output in the neural network. In the rough set analysis, the importance of condition attributes in affecting the decision attribute was assessed using the significance measure and the contribution index we developed in the previous section. No dependency and discriminant index could be computed because the data were incomplete and information was insufficient.

Table 9 shows the contribution index, t, computed for the rough set method, as well as the rank order of the importance of the variables for all the five methods. The measures indicating the importance of the variables are included in parentheses.

As shown in the table, the rough sets and ID3 measures all indicate expected growth demand as the most important variable, followed by position

Table 9
Rank order of the importance of variables

	ID3 (entropy reduction $= \delta$)	Neural network (Garson index $= \gamma$)	Neural network (Yoon index $= \beta$)	Neural network (Mak & Blanning index $= \alpha$)	Rough sets (significance $= \sigma$)	Rough sets (contribution index $= t$)
Position of the firm	2($\delta = 0.082$)	1($\gamma = 0.216$)	1($\beta = 0.356$)	1($\alpha = 0.308$)	1($\sigma = 0.62$)	2($t = 0.64$)
Financial strength of the firm	7($\delta = 0.002$)	2($\gamma = 0.193$)	7($\beta = 0.001$)	6($\alpha = 0.051$)	6($\sigma = 0.51$)	7($t = 0.35$)
Expected demand growth	1($\delta = 0.123$)	3($\gamma = 0.149$)	5($\beta = 0.109$)	5($\alpha = 0.122$)	1($\sigma = 0.62$)	1($t = 0.75$)
The product's life cycle	3($\delta = 0.065$)	6($\gamma = 0.102$)	3($\beta = 0.180$)	3($\alpha = 0.162$)	4($\sigma = 0.53$)	3($t = 0.59$)
Diffusion across competitors	4($\delta = 0.04$)	4($\gamma = 0.137$)	6($\beta = -0.017$)	7($\alpha = -0.041$)	3($\sigma = 0.54$)	4($t = 0.54$)
Cannibalization	6($\delta = 0.016$)	5($\gamma = 0.105$)	4($\beta = -0.152$)	4($\alpha = -0.129$)	4($\sigma = 0.53$)	6($t = 0.44$)
The cost of market development	5($\delta = 0.034$)	7($\gamma = 0.098$)	2($\beta = -0.186$)	2($\alpha = -0.186$)	7($\sigma = 0.46$)	5($t = 0.53$)

of the firm, product's life cycle, and diffusion across competitors. On the other hand, the neural network measures all agree that position of the firm was the most important variable. Garson's index was different than those of Yoon, Mak and Blanning, probably due to the fact that direction of influences is not considered in Garson's measure but in the other two measures. The similarity in ranking between the measures for ID3 and rough sets may be due to the similarity in the type of partitioning conducted in both methods. Non-linear transformation and partitioning are performed in neural networks and this non-linearity may constitute for a different set of contribution indices for the input variables.

5. Discussion

In evaluating the rules extracted from neural networks, Tickle et al. [60] have used the following criteria: (1) expressive power of the extracted rules; (2) the quality of the extracted rules; (3) the translucency of the view taken within the rule extraction techniques; (4) the algorithmic complexity of the rule extraction technique; (5) the extent to which the underlying neural network incorporates specialized training. Tickle et al. [60] further suggested the following measures for assessing rule quality: (a) rule accuracy (how accurate can the rule set classify unseen examples) (b) rule fidelity (how true can the rule set represent the behavior of the neural network) (c) rule consistency (how consistent can the rule set classify unseen examples under differing training conditions) (d) rule comprehensibility.

Based on these criteria, rule extraction in neural networks may suffer from certain inflexibility due to the non-linear and complicated nature of transformation performed in the hidden layers. The fact that neural network models can provide high accuracy in modeling that is related to the non-linear sigmoid transformation in the hidden layers. The adoption of sigmoid transformation allows differentiation of the error function and backpropagation of the network to update the weights to minimize the error. Once rules are extracted using clustering and discretization, these rules may lose their accuracy in modeling as they may over-generalize the relationships modeled in the network. The rules as abstract representation may provide an overall idea about the relationships between inputs and outputs and the intermediate products, but they may not classify and predict as accurately as the neural network.

Overall speaking, the rough set method offers much better explanatory capability than the neural network method and distills the data into a set of simple and usable rules. When the sample size is small or if the underlying distribution of the data deviates significantly from multivariate normal distribution, rough sets may perform better since there is no assumption on the data size and distribution [56]. Rough sets may also perform better when the data are incomplete, imprecise, heterogeneous, or non-numeric. It is particularly useful in distilling complicated data into simple and easy-to-understand rules. Rough sets can be used as a tool in decision analysis to offer intelligent decision support [52,55]. The rules obtained from rough set analysis can be readily applied to predict new cases or new situations (see Appendix A).

Besides rule comprehensibility, another important factor is the time involved in the training phases of the three methods. There are fundamental differences in the training methods involved in neural networks, ID3 and rough sets. Training of a neural network requires long computational time before the network stabilizes or converges. Sometimes if the data are inconsistent or incomplete, neural networks may fail to converge. On the other hand, training time for ID3 and rough sets is considerably shorter. As observed in this study, the training time required of ID3 and rough sets is also less than that of neural networks.

Neural networks provide best fit with numeric data, while ID3 and rough sets perform best with non-numeric data. Thus, when the original input is in non-numeric form, it must be converted to numeric before it can be analyzed with neural networks. This process requires computation time and may cause partial loss of data integrity. On the other hand, ID3 and rough sets deal with discrete data. When the original data are continuous, quantization or discretization is typically performed before the data can be analyzed with ID3

or rough sets. Thus each of these methods may be more appropriate than the others depending on the type of data analyzed and the objective of the analysis. If numeric data are involved and the objective is high robustness in modeling training data, neural networks should be the best method. On the other hand, if qualitative case data are involved and if the objective is to obtain an easy-to-use decision table or decision tree, then ID3 and rough sets should be used instead.

Grzymala-Busse [14] has found that the rough set method showed better predictive capability compared with ID3 when applied to refine imperfect data and classify unseen data. Under special circumstances, when the distribution of objects in the boundary region is equally probable [64], the criterion for selecting dominant attributes based on rough sets can be theoretically shown to be a special case of ID3.

Comparisons of these two techniques, rough sets and ID3, in general terms are very difficult. Theoretical analysis of comparisons tends to be limited to special cases satisfying certain conditions or assumptions, while experimental studies may only apply for certain sets of data. Obviously these two methods employ different classification criteria. Rough set theory is typically based on relations between the condition and decision attributes, and on concepts of positive and boundary regions, reducts and cores. ID3 uses entropy for the classification process. In general, ID3 prunes search trees based on entropy but rough sets do not.

One might argue that the rules derived from rough sets are more extensive, while ID3 focuses on important rules based on the entropy criterion. ID3 may be more efficient to deal with excessively large number of rules, but may overlook potentially useful rules. Also, the ways to represent derived knowledge or rules in these two methods are different. Rough set theory is based on information tables while ID3 is based on decision trees. Certain classes of problems are probably best represented by tables, some by trees, and still others by some other types of data structures. Searching for specific rules in a knowledge base of tree form is generally efficient. On the other hand, merging trees for knowledge base restructuring may be harder than merging tables. However, there is no mathematical proof to answer these questions for general cases. Probably a consensus would be that no single approach is the best for all problems.

Instead of applying one single method, a combined approach involving several methods would contribute to the understanding of data mining [34]. Researchers have developed hybrid systems that capitalize on the combined strengths of rough sets and other data mining techniques [32,42,43]. The rough set method has been combined with neural networks, genetic algorithms, and Petri nets. The rough set method has been used to reduce the data for preprocessing data input to neural networks [18,58]. In addition, genetic algorithms have been employed to look for the set of minimal reducts used in rough set analysis [30,65]. Genetic algorithms may also be employed to determine the thresholds used in selecting dynamic reducts [49] or to look for the type of Boolean relationships used in learning tolerance relations [50,51]. Further, rough set methods can be applied to specify which concurrent systems can be used for the generation of Petri nets [41].

Rough sets may also be combined with ID3. An entropy analysis may be performed in data preprocessing and entropy reduction may be used as the selection criteria for determining which features should be selected as condition attributes for rough set analysis. Alternatively, entropy may be computed in rough sets as a measure of certainty for uncertain rules [62], and less important rules derived from rough sets can be pruned based on entropy. Based on these continual research efforts on hybrid systems, a researcher will gain a better understanding on the strengths and limitations of existing methodologies, and discover better and newer ones by drawing on the strengths of existing data mining techniques.

6. Conclusion

In this article we compare the rule extraction capability of ID3, rough set method, and neural networks. We also derive a measure to capture the contribution of the condition attributes to the

decision attribute. On going research has been conducted to develop hybrid systems capitalizing on the combined strengths of neural networks, rough sets and ID3 [2,16], as well as other data mining techniques such as genetic algorithms and Boolean analysis. Future research should address how common standards can be developed to assess the explanatory capability of these new systems in order to improve the rule extraction and knowledge discovery process.

Appendix A. Applications of rules derived from rough sets

Rules on new product entry were obtained from our database. An example of a "no-entry" decision rule is shown in Table 10, and an example of an "entry" decision rule is shown in Table 11.

The rule in Table 10 suggests that a small firm with weak financial resources should not enter the market when the expected demand is low, cannibalization is high, and product life cycle is short, regardless of the cost of market development and the diffusion rate across competitors. When the market attractiveness is low (low demand, short product life cycle and high cannibalization), a company that has weak financial resources should take a less aggressive strategy. An erroneous move in this vulnerable situation may mean disaster.

Alternatively, Table 11 suggests that if the firm is small but has strong financial resources, and if the market demand is expected to be high and product life cycle is expected to be long, the company should enter the market even if there is high cannibalization, regardless of the cost of market development and diffusion across competitors. In other words, a strong company should make full use of opportunities and enter an attractive market, even if it may mean a competitive market with high competitor imitation and high development cost.

These rules can be used to predict new cases. The exact rules are applied for prediction whenever there is an exact match between the values of the scenario variables in the new case and the rules. If no exact match occurs, we find the closest

Table 10
An example of no entry decision

Rule 1: no-entry decision		
If:	Position of firm	= small
	Financial strength of firm	= weak
	Expected demand growth	= low
	Product life cycle	= short
	Cannibalization	= high
Regardless of:	Cost of market development	
	Diffusion rate across competitors	
Recommended strategy		= do not enter the market

Table 11
An example of entry decision

Rule 2: Entry decision		
If:	Position of firm	= small
	Financial strength of firm	= strong
	Expected demand growth	= high
	Product life cycle	= long
	Cannibalization	= high
Regardless of:	Cost of market development	
	Diffusion rate across competitors	
Recommended strategy		= enter the market

Table 12
A new case example

New case		
	Position of firm	= small
	Financial strength of firm	= weak
	Expected demand growth	= low
	Product life cycle	= short
	Cannibalization	= high
	Cost of market development	= high
	Diffusion rate across competitors	= high

match to an existing rule and modify the values of the variables in the new case. The extent of modification is assessed using the contribution index of the variables involved in the modification, and the least amount of modification is preferred.

For example, we consider the following new case shown in Table 12. Assuming that Rules 1 and 2 are the closest rules to the new case, this case can be modified in two ways:

1. Modify the case for Rule 1 by (i) changing the value of demand growth rate ($t = 0.75$) from high to low, and (ii) the value of product life cycle ($t = 0.59$) from long to short.
2. Modify the case for Rule 2 by changing the value of financial strength of firm ($t = 0.35$) from weak to strong.

The contribution index helps us to assess the amount of change involved in the two modifications.

The second modification involves changing a variable of lower contribution index, and this means a lesser extent of modification. Therefore the second way of modification is preferred to the first. Rule 2 will be applied for the new case if there is no other rule that matches exactly with the new case.

References

[1] R. Agrawal, T. Imielinski, A. Swami, Database mining: A performance perspective, IEEE Transactions on Knowledge and Data Engineering 5 (6) (1993) 914–925.
[2] M. Banerjee, S. Mitra, S.K. Pal, Rough fuzzy MLP: Knowledge encoding and classification, IEEE Transactions on Neural Networks 9 (6) (1998) 1203–1216.
[3] M. Chen, J. Han, P. Yu, Data mining: An overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 866–883.
[4] S. Curram, P.J. Mingers, Neural networks, decision tree induction and discriminant analysis: An empirical comparison, Journal of the Operational Research Society 45 (4) (1994) 440–450.
[5] V. Dhar, A. Tuzhilin, Abstract-driven pattern discovery in databases, IEEE Transactions on Knowledge and Data Engineering 5 (6) (1993) 926–937.
[6] A.I. Dimitras, R. Slowinski, R. Susmaga, C. Zopounidis, Business failure prediction using rough sets, European Journal of Operational Research 114 (1999) 263–280.
[7] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, The KDD process for extracting useful knowledge from volumes of data, Communications of the ACM 39 (11) (1996) 27–34.
[8] U. Fayyad, G. Piatetsky-Shapiro, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, MA, 1996.
[9] U. Fayyad, R. Uthurusamy, Data mining and knowledge discovery in databases, Communications of the ACM 39 (11) (1996) 24–26.
[10] L.M. Fu, Knowledge discovery based on neural networks, Communications of the ACM 42 (11) (1999) 47–50.
[11] D. Garson, Interpreting neural-network connection weights, AI Expert (1991) 47–51.
[12] C. Glymour, D. Madigan, D. Pregibon, P. Smyth, Statistical inference and data mining, Communications of the ACM 39 (11) (1996) 35–41.
[13] S. Greco, B. Matarazzo, R. Slowinski, Rough approximation of a preference relation by dominance relations, European Journal of Operational Research 117 (1999) 63–83.
[14] D.M. Grzymala-Busse, J.W. Grzymala-Busse, The usefulness of a machine learning approach to knowledge acquisition, Computational Intelligence 11 (2) (1995) 268–279.
[15] J.W. Grzymala-Busse, Rough sets, Advances in Imaging and Electron Physics 94 (1995) 151–195.
[16] R.R. Hashemi, L.A. Le Blanc, C.T. Rucks, A. Rajaratnam, A hybrid intelligent system for predicting bank holding structures, European Journal of Operational Research 109 (1998) 390–402.
[17] I. Ivanova, M. Kubat, Initialization of neural networks by means of decision trees, Knowledge-based Systems 8 (6) (1995) 333–344.

[18] J. Jelonek, K. Krawiec, R. Slowinski, J. Stefanowski, J. Szymas, Rough sets as an intelligent front-end for the neural network, in: Proceedings of the First National Conference on Neural Networks and Applications, 2, Czestochowa, Poland, 1994, pp. 116–122.
[19] J. Komorowski, L. Polkowski, A. Skowron, Rough Sets: A tutorial, lecture materials in the course rough sets: An introduction, in: A. Skowron, J. Komorowski, L. Polkowski (Eds.), in: Proceedings of the 11th European Summer School in Logic, Language and Information, Utrecht University, August 9–20, 1999.
[20] P. Langley, H.A. Simon, Applications of machine learning and rule induction, Communications of the ACM 38 (11) (1995) 54–64.
[21] H. Lu, R. Setiono, H. Liu, Effective data mining using neural networks, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 957–961.
[22] B. Mak, R.W. Blanning, An empirical measure of element contribution in neural networks, IEEE Transactions on Systems, Man, and Cybernetics 28 (4) (1998) 561–564.
[23] B. Mak, T. Bui, Modeling experts consensual judgments for new product entry timing, IEEE Transactions on Systems, Man, and Cybernetics 26 (9) (1996) 659–667.
[24] C.J. Matheus, P.K. Chan, G. Piatetsky-Shapiro, Systems for knowledge discovery in databases, IEEE Transactions on Knowledge and Data Engineering 5 (6) (1993) 903–913.
[25] T. Mitchell, Machine learning and data mining, Communications of the ACM 42 (11) (1999) 30–36.
[26] A. Mrózek, A new method for discovering rules from examples in expert systems, International Journal of Man-Machine Studies 36 (1992) 127–143.
[27] T. Munakata (Ed.), Knowledge discovery, Communications of the ACM 42(11) (1999) 26–29.
[28] T. Munakata, Z. Pawlak, Rough control: Applications of rough set theory to control, in: Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany 1 (1996) 209–217.
[29] T. Munakata, Fundamentals of the new artificial intelligence: Beyond traditional paradigms, Springer, Berlin, 1998.
[30] S.H. Nguyen, A. Skowron, P. Synak, J. Wróblewski, Knowledge discovery in data bases, Rough set approach, in: Mares, Meisar, Novak, Ramik (Eds.), Proceedings of the Seventh International Fuzzy Systems Association World Congress (IFSA'97), June 25–29, Academia, Prague 1, 1997, pp. 1–529.
[31] H. Nurmi, J. Kacprzyk, M. Fedrizzi, Probabilistic, fuzzy, and rough concepts in social choice, European Journal of Operational Research 95 (1996) 264–277.
[32] C.W. Omlin, C.L. Giles, Rule revision with recurrent neural networks, IEEE Transactions on Knowledge and Data Engineering 8 (1) (1996) 183–188.
[33] A. Øhrn, J. Komorowski, A. Skowron, P. Synak, The design and implementation of a knowledge discovery toolkit based on rough sets – The ROSETTA system, in: L. Polkowski, A. Skowron, (Eds.), Rough Sets in Knowledge Discovery 1: Methodology and Applications, Physica, Heidelberg, 1998, 376–399.
[34] S.K. Pal, A. Skowron (Eds.), Rough Fuzzy Hybridization – A New Trend in Decision Making, Springer, Berlin, 1999.
[35] Y. Pao, Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, Reading, 1989.
[36] Z. Pawlak, Rough set approach to knowledge-based decision support, European Journal of Operational Research 99 (1997) 48–57.
[37] Z. Pawlak, Rough Sets: Theoretical Aspects and Reasoning about Data, Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.
[38] Z. Pawlak, Rough sets, International Journal of Information and Computer Sciences 11 (1982) 341–356.
[39] Z. Pawlak, R. Slowinski, Rough set approach to multi-attribute decision analysis, European Journal of Operational Research 72 (1994) 443–459.
[40] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough sets, Communications of the ACM 38 (11) (1995) 89–95.
[41] J. F. Peters III, Time clock information systems: Concepts and roughly fuzzy Petri net models, in: L. Polkowski, A. Skowron (Eds.), Rough sets in knowledge discovery: Applications, case studies and software systems, Physica, Heidelberg, 1998, pp. 387–419.
[42] L. Polkowski, A. Skowron (Eds.), Rough sets in knowledge discovery 1: Methodology and Applications, Physica, Heidelberg, 1998.
[43] L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Physica, Heidelberg, 1998.
[44] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Los Altos, CA, 1993.
[45] H. Salonen, H. Nurmi, A note on rough sets and common knowledge events, European Journal of Operational Research 112 (1999) 692–695.
[46] R. Setiono, H. Liu, Symbolic representation of neural networks, IEEE Computer 29 (3) (1996) 71–77.
[47] Z. Shen, M. Clarke, R.W. Jones, Input contribution analysis in a double input layered neural network, in: Proceedings of the International Conference on Artificial Neural Networks, Italy, 1994, 541–544.
[48] A. Silberschatz, A. Tuzhilin, What makes patterns interesting in knowledge discovery systems, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 970–974.
[49] A. Skowron, L. Polkowski, Rough mereological foundations for design, analysis, synthesis, and control in distributive systems, Information Sciences 104 (1–2) (1998) 129–156.
[50] A. Skowron, L. Polkowski, J. Komorowski, Learning tolerance relations by Boolean descriptors, automatic feature extraction from data tables, in: Tsumoto, Kobayashi, Yokomori, Tanaka, Nakamura (Eds.), in: Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96). The University of Tokyo, November 6–8, 1996, pp. 11–17.

[51] A. Skowron, J. Stepaniuk, Tolerance approximation spaces, Fundamenta Informaticae 27 (1996) 245–253.
[52] R. Slowinski, Rough set approach to decision analysis, AI Expert 10 (3) (1995) 18–25.
[53] R. Slowinski, C. Zopounidis, Application of the rough set approach to evaluation of bankruptcy risk, Intelligent Systems in Accounting, Finance and Management 4 (1995) 27–41.
[54] R. Slowinski, C. Zopounidis, A.I. Dimitras, Prediction of company acquisition in Greece by means of the rough set approach, European Journal of Operational Research 100 (1997) 1–15.
[55] R. Slowinski (Ed.), Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Dordrecht, 1992.
[56] J. Stefanowski, Rough sets theory and discriminant methods as tools for analysis of information systems, a comparative study, Foundations of Computing and Decision Sciences 17 (2) (1992) 81–98.
[57] V. Subramanian, M. Hung, M. Hu, An experimental evaluation of neural networks for classification, Computers and Operations Research 20 (7) (1993) 769–782.
[58] R. Swiniarski, Rough sets Bayesian methods applied to cancer detection, in: L. Polkowski, A. Skowron (Eds.), Proceedings of the First International Conference on Rough Sets and Soft Computing (RSCTC'98). Warszawa, Poland, June 22–27, Springer, LNAI 1424, 1998, pp. 617–624.
[59] A. Tan, Cascade ARTMAP: Integrating neural computation and symbolic knowledge processing, IEEE Transactions on Neural Networks 8 (2) (1997) 237–250.
[60] A. Tickle, R. Andrews, M. Golea, J. Diederich, The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEEE Transactions on Neural Networks 9 (6) (1998) 1057–1068.
[61] H. Tsukimoto, Rule extraction from prediction models, Methodologies for Knowledge Discovery and Data Mining. in: Proceedings of the Third Pacific-Asia Conference, PAKDD-99, Springer, Berlin, Germany, xv+523, 1999, 34–43.
[62] S. Tsumoto, H. Tanaka, Characterization of structure of decision trees based on rough sets and greedoid theory, in: Proceedings of RSSC'94, Third International Workshop on Rough Sets and Soft Computing, San Jose, CA, November 10–12, 1994, 450–460.
[63] S. Tsumoto, Discovery of rules for medical expert systems – Rough set approach, in: Proceedings of ICCIMA'99, Third International Conference on Computational Intelligence and Multimedia Applications 1999, 212–216.
[64] S.K.M. Wong, W. Ziarko, R.L. Ye, Comparison of rough-set and statistical methods in inductive learning, International Journal of Man–Machine Studies 24 (1986) 53–72.
[65] J. Wrblewski, Finding minimal reducts using genetic algorithms, in: P.P. Wang (Ed.), in: Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences (JCIS'95), Wrightsville Beach, North Carolina, 28 September –1 October, 1995, pp. 186–189.
[66] Y. Yoon, G. Swales, T. Margavio, A comparison of discriminant analysis versus artificial neural networks, Journal of the Operational Research Society 44 (1) (1993) 51–60.
[67] Y. Yoon, T. Guimaraes, G. Swales, Integrating artificial neural networks with rule-based expert systems, Decision Support Systems 11 (5) (1994) 497–507.
[68] L.A. Zadeh, Similarity relations and fuzzy orderings, Information Sciences 3 (1971) 177–200.
[69] N. Zhong, A. Skowron, S. Ohsuga (Eds.), New direction in rough sets data mining and granular – Soft computing, in: Proceedings of the Seventh International Workshop, RSFDGrC'99, Yamaguchi, Japan, November 9–11, Springer, 1999.