

Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing

Andrew Kusiak, *Member, IEEE*

Abstract—The growing volume of information poses interesting challenges and calls for tools that discover properties of data. Data mining has emerged as a discipline that contributes tools for data analysis, discovery of new knowledge, and autonomous decision-making. In this paper, the basic concepts of rough set theory and other aspects of data mining are introduced. The rough set theory offers a viable approach for extraction of decision rules from data sets. The extracted rules can be used for making predictions in the semiconductor industry and other applications. This contrasts other approaches such as regression analysis and neural networks where a single model is built. One of the goals of data mining is to extract meaningful knowledge. The power, generality, accuracy, and longevity of decision rules can be increased by the application of concepts from systems engineering and evolutionary computation introduced in this paper. A new rule-structuring algorithm is proposed. The concepts presented in the paper are illustrated with examples.

Index Terms—Data mining, decision making, evolutionary computation, knowledge discovery, knowledge structuring, rough set theory, semiconductor manufacturing.

I. INTRODUCTION

DATA mining is an emerging area of computational intelligence that offers new theories, techniques, and tools for processing large volumes of data. It has gained considerable attention among practitioners and researchers as evidenced by the number of publications, conferences, and application reports. The growing volume of data that is available in a digital form has accelerated this interest. Data mining relates to other areas, including machine learning, cluster analysis, regression analysis, and neural networks. However, there are fundamental differences between the two approaches and data mining. Both neural network and regression approaches create one model based on a training data set. This model normally uses a predetermined set of features. A machine learning algorithm of data mining generates a number of models (usually in the form of decision rules) capturing relationships between the input features and the decision. In an extreme case, the set of features included in each rule could be independent from all other rules, which is similar to the result produced by cluster analysis. Neural network and regression models can be viewed as “population based” as a single model is formed for the entire population (training data set), while the data mining approach follows an “individual (data object) based” paradigm. The “population based” tools determine features that are common to a population (training data set). The models (rules) created by data mining are explicit.

Manuscript received October 31, 2000; revised March 23, 2001.

The author is with the Department of Industrial Engineering, Intelligent Systems Laboratory, University of Iowa, Iowa City, IA 52242-1527 USA.

Publisher Item Identifier S 1521-334X(01)04697-3.

No.	Features				D
	F1	F2	F3	F4	
1	0	1	0	2	Low
2	1	1	0	2	High
3	0	0	0	1	Low
4	0	1	1	0	Medium
5	0	0	1	3	Low

Fig. 1. Data set.

One of the new data mining theories is the rough set theory [1] that can be used for

- reduction of data sets;
- finding hidden data patterns;
- generation of decision rules.

The rough set theory algorithms used in this paper fall into a broad area of machine learning. Langley and Simon [2] grouped machine learning research into the following categories:

- neural networks;
- genetic algorithms;
- case-based learning;
- rule induction;
- analytical learning.

The major developments in learning and data mining are summarized in the edited volumes [2], [4], [5], and the book by Mitchell [6]. For a survey of important applications of machine learning see [2]. Other related topics include nonparametric regression, which is discussed in [7] and [8] and reinforcement learning covered in [9].

To date, numerous data mining software products have been developed, e.g., Clementine, CHAID, DataLogic, DataQuest, DataScope, GOLDMINE, JMP, and PolyAnalyst. The rule extraction concept, being the basis of this paper, is illustrated in the next section with an example.

II. RULE EXTRACTION

The content of large-scale data sets containing numerical and categorical information can not be easily interpreted unless the information is transformed into a form that can be understood by human users. The rule extraction algorithms are designed to identify patterns in such data sets and express them as decision rules. The rule extraction concept is illustrated in Example 1.

A. Example 1

Consider the data set in Fig. 1 with five objects, four features F1–F4, and the decision (outcome) *D*.

Rule 1. IF (F2 = 0) THEN (D = Low); [2, 66.67%, 100%] [3, 5]
 Rule 2. IF (F1 = 0) AND (F4 = High) THEN (D = 0); [1, 33.33%, 100.00%] [1]
 Rule 3. IF (F4 = 0) THEN (D = Medium); [1, 100%, 100%] [4]
 Rule 4. IF (F1 = 1) THEN (D = High); [1, 100%, 100%] [2]

Fig. 2. Decision rules extracted with a rough set algorithm.

		Features				
No.	F1	F2	F3	F4	D	
3	0	0	0	1	Low	Rule 1
5	0	0	1	3	Low	Rule 1
1	0	1	0	2	Low	Rule 2
4	0	1	1	0	Medium	Rule 3
2	1	1	0	2	High	Rule 4

Fig. 3. Patterns corresponding to the rules of Fig. 2.

The features denote process parameters (e.g., temperature, pressure) and the decision is the component performance, $D = \{\text{high, medium, low}\}$.

A rule extraction algorithm transforms the data set of Fig. 1 into the decision rules of Fig. 2. The two sets of numbers in square brackets behind each rule describe its properties and are defined in the Appendix.

The decision rules of Fig. 2 correspond to the patterns indicated by shaded cells in the matrix in Fig. 3.

The matrix in Fig. 3 indicates that the number of features used to describe all objects in the data set is three, F1, F2, and F4. The algorithm used to generate rules in Fig. 2 minimized the number of decision rules. The rule extraction algorithms may consider other criteria, including the generation of all possible rules. An algorithm minimizing the number of features included in decision rules generated the result in Fig. 4. These rules are represented with the matrix in Fig. 5.

Only two features F1 and F4 are used in Fig. 5 to represent the five objects. In this case each rule describes one object.

As illustrated in Figs. 3 and 5, the patterns corresponding to the rules extracted by the learning algorithms vary in shape. It is easy to imagine that for large data sets such patterns can be complex. The idea behind the research reported in this paper is that understanding the patterns might add value to the extracted knowledge or even enhance its quality, e.g., performance and robustness of decisions can be improved.

III. RULE STRUCTURING

The goal of structuring decision rules is to enhance the decision-making capability of the knowledge generated with learning algorithms. The need for knowledge structuring is supported by the notion of cognitive maps and mental models discussed in [10] and [11]. By structuring decision rules an evaluation perspective is incorporated into the knowledge extracted from data. The idea of structured knowledge is introduced by two examples of structured matrices in Figs. 6 and 7, where rules and features have been grouped in blocks.

In the matrix of Fig. 6 the decisions $A-D$ are differentiated on feature sets, as a unique feature set and their values are associated with one decision. Each of the four decisions is made based on the values of three to four different features. The feature sets

associated with the four rules and decisions are mutually exclusive.

The structure in the matrix of Fig. 7 is more complex as the same features and their values are associated with different decisions. A decision-maker prefers to deal with the structure in Fig. 6 rather than the structure in Fig. 7 due to the association of a unique set of features with a decision value. The term support used in Figs. 6 and 7 is defined under rule support in the Appendix.

The structure in the matrix of Fig. 7 is more complex as the same features and their values are associated with different decisions. A decision-maker prefers to deal with the structure in Fig. 6 rather than the structure in Fig. 7 due to the association of a unique set of features with a decision value. The term support used in Figs. 6 and 7 is defined under rule support in the Appendix.

Besides the two cases illustrated in Figs. 6 and 7, other shapes of the knowledge structure are possible, e.g., L shape, C shape. Exploring different knowledge structures is helpful in decision making by

- decision processes becoming transparent to the user and computing environment;
- supporting data evolution;
- increased decision accuracy;
- exposing missing features, which is useful in data farming discussed in [14].

The knowledge structure is largely determined by the type of a learning algorithm (e.g., entropy based) and learning criteria (e.g., minimization of the number of rules). A learning algorithm may produce more than one rule for the same decision value. In addition, more than one learning algorithm may be used at a time, which results in multiple rules per decision value. This complicates the knowledge structure (rule-feature matrix) and increases computational complexity of the rule-structuring problem. However, the new patterns may enhance the utility of the extracted knowledge.

The structures embedded in the rule-feature matrices can be enhanced with visualization tools, including virtual reality. These tools would greatly impact the quality and transparency of decision making. The matrices play an important role in the knowledge discovery process by fusing information from diverse sources.

The machine learning algorithms generate rules that are in turn structured by the knowledge-structuring algorithm presented in Section V. This algorithm uses the data engineering principles proposed in the next section.

IV. DATA ENGINEERING

The term data engineering introduced in this paper is analogous to the term genetic engineering and it has some relationship with genetic programming [12] and evolutionary computation

- Rule 1. IF (F4 = 1) THEN (D = Low); [1, 33.33%, 100%] [3]
 Rule 2. IF (F4 = 3) THEN (D = Low); [1, 33.33%, 100%] [5]
 Rule 3. IF (F4 = 2) AND (F1 = Low) THEN (D = 0); [1, 33.33%, 100%] [1]
 Rule 4. IF (F4 = 0) THEN (D = Medium); [1, 100%, 100%] [4]
 Rule 5. IF (F1 = 1) THEN (D = High); [1, 100%, 100%] [2]

Fig. 4. Decision rules with minimum number of features.

No.	Features				D	
	F1	F2	F3	F4		
1	0	1	0	2	Low	Rule 1
2	1	1	0	2	High	Rule 2
4	0	1	1	0	Medium	Rule 3
3	0	0	0	1	Low	Rule 4
5	0	0	1	3	Low	Rule 5

Fig. 5. Patterns corresponding to the rules from Fig. 4.

Rule No.	Features											Decision	Support
	F1	F7	F10	F4	F6	F9	F3	F5	F8	F12	F2		
Rule 1	bf 6 [7.9-9.9]											A	100 objects
Rule 2	[1.3-1.7] bb 4											B	85 objects
Rule 3	di 7 yes no											C	106 objects
Rule 4	no 5 [42-77]											D	92 objects

Fig. 6. Mutually exclusive rule-feature blocks.

in general [13]. Similar to the genetic engineering approach that may use simple methods such as selective breeding to complicated ones such as gene cloning, data engineering methods may vary in type and scope. We have found that experimenting with data sets from the engineering and medical domains leads to the development of new data engineering methods. Three data engineering principles introduced in this paper aim to improve the quality of decisions generated from rule-feature matrices.

- 1) Merge clusters: Cluster rules with equivalent feature values.
- 2) Feature value replacement: Replace feature values with a value range and merge the corresponding rules with the same decisions.
- 3) Column removal: Remove columns with entries that are not used by the rules.

The three data engineering principles are incorporated into the rule-structuring algorithm that is illustrated in Example 2 of the next section.

V. RULE-STRUCTURING ALGORITHM

The rule-structuring algorithm groups shaded entries of the rule-feature matrix. The features corresponding to the shaded entries are called *marked features*.

The steps of the rule-structuring algorithm are outlined next.

- Step 1) Select from the rule-feature matrix a marked feature with the maximum equivalence class. Break a tie arbitrarily.
- Step 2) Cluster the rules within each equivalence class (for review of clustering algorithms see [14]).

Rule No.	Features											Decision	Support
Rule 1	ax 2 [7.7-9.1] >7] 3 no											A	89 objects
Rule 2	[1.3-1.7] bb 2											B	37 objects
Rule 3	bz 3 di 7 yes no <7]											C	81 objects
Rule 4	.1 yes 4 [52-56]											D	45 objects

Fig. 7. Overlapping rule patterns.

No.	Source 1							Source 2		
	F1	F2	F3	F4	F5	F6	F7	D		
1	1.02	Red	2.98	High	1	Yes	7.2	2		
2	2.03	Black	1.04	Low	3	No	6.1	1		
3	0.99	Blue	3.04	High	1	Yes	6.9	2		
4	2.03	Blue	3.11	High	1	yes	11	2		
5	0.03	Orange	0.96	Low	2	Yes	6.4	1		
6	0.04	Blue	1.04	Medium	2	No	4.4	1		
7	0.99	Orange	1.04	Medium	3	Yes	5.9	2		
8	1.02	Red	0.94	Low	2	No	2.1	1		

Fig. 8. Data set generated from two different sources.

- Step 3) Re-engineer the marked entries of the rule-feature matrix according to the principles of Section IV.
- Step 4) Stop, if a satisfactory matrix structure has been obtained.

The rule-structuring algorithm is illustrated in Example 2, which includes data from two different sources. In fact, one of the greatest advantages of data mining algorithms and the algorithm discussed in this paper are their ability to consider data of different types as well as data created at different sources.

A. Example 2

Consider the data in Fig. 8 for eight objects, seven features, and the decision D .

When the data from each source is considered independently, the rough set algorithm generates the rule sets in Figs. 9 and 10.

The patterns resulting from the rule sets of Figs. 9 and 10 are shown in Fig. 11.

The steps of the rule-structuring algorithm are illustrated next with the matrix of Fig. 11.

- Step 1) The two features F4 and F5 are the most frequently used by the rules. The feature F4 is arbitrarily selected among the two.
- Step 2) The rules 1, 3, and 4 included in the equivalence class $F1 = \text{High}$ are clustered (see Fig. 12).
The rules 2, 5, and 8 in the equivalence class $F4 = \text{Low}$ are clustered (see Fig. 12).
- Step 3) Applying the data engineering principles of Section IV to the matrix of Fig. 12 results in the matrix in Fig. 13.

Rule S1-1. IF (F4 = Low) THEN (D = 1); [3, 75%, 100%] [2, 5, 8]
 Rule S1-2. IF (F1 in [0.035, 0.515]) THEN (D = 1); [1, 25%, 100%] [6]
 Rule S1-3. IF (F4 = High) THEN (D = 2); [3, 75%, 100%] [1, 3, 4]
 Rule S1-4. IF (F1 in [0.515, 1.005]) THEN (D = 2); [2, 50%, 100%] [3,7]

Fig. 9. Decision rules derived from Source 1 data.

Rule S2-1. IF (F5 in [1, 2]) THEN (D = 1); [3, 75%, 100%] [5, 6, 8]
 Rule S2-2. IF (F7 in [6, 6.25]) THEN (D = 1); [1, 25%, 100%] [2]
 Rule S2-3. IF (F5 in [1, 1]) THEN (D = 2); [3, 75%, 100%] [1, 3, 4]
 Rule S2-4. IF (F7 in [5.15, 6]) THEN (D = 2); [1, 25%, 100%] [7]

Fig. 10. Decision rules derived from Source 2 data.

1	1.02	Red	2.98	High	1	Yes	7.2	2	Rule S1-3	Rule S2-3
2	2.03	Black	1.04	Low	3	No	6.1	1	Rule S1-1	Rule S2-2
3	0.99	Blue	3.04	High	1	Yes	6.9	2	Rule S1-3	Rule S2-3
4	2.03	Blue	3.11	High	1	Yes	11.0	2	Rule S1-3 Rule S1-4	Rule S2-3
5	0.03	Orange	0.96	Low	2	Yes	6.4	1	Rule S1-1	Rule S2-1
6	0.04	Blue	1.04	Medium	2	No	4.4	1	Rule S1-2	Rule S2-1
7	0.99	Orange	1.04	Medium	3	Yes	5.9	2	Rule S1-4	Rule S2-4
8	1.02	Red	0.94	Low	2	No	2.1	1	Rule S1-1	Rule S2-1

Fig. 11. Patterns of rules from Figs. 9 and 10.

1	1.02	Red	2.98	High	1	Yes	7.2	2	Rule S1-3	Rule S2-3
3	0.99	Blue	3.04	High	1	Yes	6.9	2	Rule S1-3	Rule S2-3
4	2.03	Blue	3.11	High	1	Yes	11	2	Rule S1-3 Rule S1-4	Rule S2-3
2	2.03	Black	1.04	Low	3	No	6.1	1	Rule S1-1	Rule S2-2
5	0.03	Orange	0.96	Low	2	Yes	6.4	1	Rule S1-1	Rule S2-1
8	1.02	Red	0.94	Low	2	No	2.1	1	Rule S1-1	Rule S2-1
6	0.04	Blue	1.04	Medium	2	No	4.4	1	Rule S1-2	Rule S2-1
7	0.99	Orange	1.04	Medium	3	Yes	5.9	2	Rule S1-4	Rule S2-4

Fig. 12. Clustered matrix.

F1	F4	F5	F7	D	DSM*	DRF*
.99-2.03	High	1	6.9-11.1	2	5	2
0.99	Medium	3	5.9	2	2	2
.03-1.02	Low	2	2.1-6.6	1	4	2
2.03	Low	3	6.1	1	2	2
0.04	Medium	2	4.4	1	2	2

*DSM and DRF are defined in the Appendix.

Fig. 13. Matrix transformed by the feature value replacement and column removal principles.

Based on the feature value replacement data engineering principle (#2), Rule 7 is attached to equivalence class with F4 = High, rules 2, 4 and 7 are merged, rules 5 and 8 are also merged, and the value of the corresponding features are replaced with the value ranges. The column removal principle (#3) has led to the removal of columns with unmarked features.

Fig. 13 contains very useful information for decision making. It is visible that the decisions are made with a decision support measure in the range DSM = 2–5 and with the decision redundancy factor DRF = 2, both defined in [14] and the Appendix.

The analysis of the result in Fig. 13 indicates that feature F7 is not dependable as only a small differentiation in its value

(F7 = 5.9 versus F7 = 6.1) would result in a different decision value ($D = 1$ versus $D = 2$). The range of the feature F7 values associated with the decision $D = 2$ is [5.9–11.0] and it overlaps with the range [2.1–6.6] of F7 associated with the decision $D = 1$. On the other hand, feature F1 is dependable as the value differentiation is relatively large (F1 = .99 versus F1 = 0.04).

VI. EVOLUTIONARY COMPUTATION IN RULE ENGINEERING

It is important in any development effort that users have a sufficient degree of trust in a computer-generated solution. The experimental data sets are often small and incomplete thus adding another dimension to the trust development. The decision rules extracted from small data sets may be simple, often involving one feature, and therefore may not be fully trusted by a user. To capture the essence of a user trust in the knowledge extracted from a data set the term rule acceptance measure is introduced (see the Appendix). The rule acceptance measure is a rather subjective assessment that reflects the user confidence in the extracted rules. This is an important property of the rules extracted from a data set that may determine the success or failure of the data mining effort.

The rule acceptance measure may depend on the type of the data set, user’s background, culture, and so on. The rule ac-

Rule 1. IF F1 in [.99 - 2.03] AND F4 in [Medium - High] AND F5 \leq 1
AND F7 about 5.9 THEN D = 2
Rule 2. IF F1 \leq .04 AND F4 in [Low - Medium] AND F5 in [2 - 3] AND
F7 about 6.1 THEN D = 1

Fig. 14. Compound decision rules.

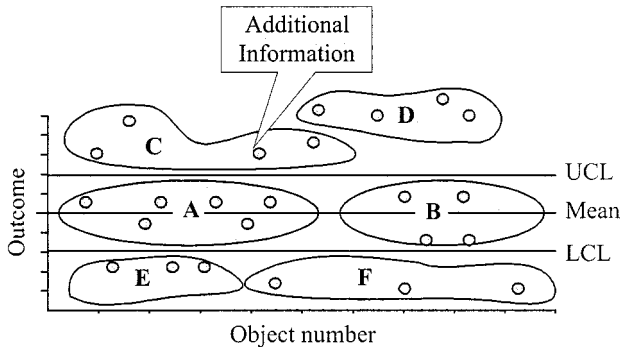


Fig. 15. Process control chart.

ceptance measure taken together with the decision redundancy factor (DRF) and the decision support measure (DSM) all make a comprehensive set of metrics for measuring the quality of the extracted knowledge and decision making.

Based on the results shown in Fig. 13 the following two compound decision rules with a higher value of rule acceptance measure are generated (Fig. 14).

These two rules have the same classification power as the eight rules in Figs. 9 and 10. Each of the two rules was obtained by generalizing attribute values and increasing the decision redundancy factor.

VII. DATA PROCESSING

Data and rule engineering discussed in the earlier sections of this paper aim at improving predictive quality of the extracted knowledge. Another way of impacting the quality of rules is by preprocessing the data set. Here, statistical process control methods are suggested. Statistical control methods are useful in analyzing outcome trends and improving processes (see, for example [15] and [16]). The essence of the relationship between statistical control and data mining is captured in Fig. 15.

The process control chart in Fig. 15 divides the outcome population into three regions, U (upper), N (nominal), and L (lower). The information analyzed with the SPC approach is typically numerical and no direct relationship is captured between these three regions and the features. The rules extracted from a data set decompose the outcomes into numerous regions, for example *A–F* in Fig. 15. Treating the data in regions *A* and *B* differently those of the regions *C–F* might be beneficial, e.g., large-scale training data set could be naturally broken into smaller sets or extracting rules from the areas *A–B*, *C–D*, and *E–F* might enhance their predictive power [17].

VIII. TESTING KNOWLEDGE ACCURACY

Users of the knowledge extracted from a training data set are interested in the accuracy of predictions that can be made based on this knowledge. The methods used to determine classification quality of a rule set are divided in the following three categories [18]:

- a) partitioning;
- b) bootstrapping;
- c) cross validation.

The partitioning method is based on splitting the data into a test set and a training set. The two separate data sets could be created at the data collection phase (*a priori* partitioning) or after the data has been collected (*a posteriori* partitioning) [19]. The classification quality derived from a single test set could be questioned, therefore the posteriori partitioning is repeated numerous times. Rather than arbitrarily determining the size of the test data set, the bootstrapping method suggests splitting the data set according to the following ratios, .632 for the training set and .368 for the test set. The cross-validation method discussed in [20] suggests dividing the set of all objects into k disjoint groups, usually of equal size. One group of k becomes a test set and $k - 1$ groups remain in the training set. This process is repeated k times until all k groups have been tested. When the size of a group becomes one, the method becomes leave-one-out cross-validation method.

IX. CONCLUSION

In the paper, basic concepts of data mining and the rough set theory were discussed. The rough set theory is a viable approach for extraction of meaningful knowledge and making predictions for an individual data object (e.g., fault occurrence) rather than a population of objects. A rule extracted from a data set and the corresponding features can be considered as one of many models describing a data set. This property contrasts other approaches such as regression analysis and neural networks where essentially one model with a fixed set of features is constructed for the entire population. The existing concepts of data mining were expanded with rule structuring and data engineering. All these concepts follow the evolutionary computation approach extending longevity of the knowledge.

The patterns formed by the rules extracted with rough set algorithms differ from the patterns generated by algorithms of other types, e.g., decision tree algorithms. The limited overlap among features included in the rough set rules make them suitable for forming meta-structures of interest to semiconductor applications. A new rule-structuring algorithm introduced in the paper forms these meta-structures. This algorithm enhances the utility of the extracted knowledge, allows for transparent knowledge analysis, and leads to informed decision making. The algorithm was illustrated with a numerical example. The structures derived by the rule-structuring algorithm can be further enhanced with visualization tools.

APPENDIX

ROUGH SET THEORY BACKGROUND AND DEFINITIONS

The *rough set theory* is based on the assumption that data and information is associated with every object of the universe of

discourse [21]. Objects described by the same properly selected information (referred in this paper as features) are indiscernible.

The adjective “properly” is of key importance, as the algorithms presented in this paper will select a subset of all features necessary to characterize a category of objects.

A *reduct* is a minimal sufficient subset of features $RED \subseteq A$ such that (Shan *et al.* [22] after Pawlak [21]):

- a) $R(RED) = R(A)$, i.e., RED produces the same classification of objects as the collection A of all features;
- b) for any feature $f \in RED$, $R(RED - \{f\}) \neq R(A)$, i.e., a reduct is a minimal subset with respect to the property a);

Core is the collection of features appearing in all reducts and is computed as the product of all reducts.

Pawlak [1] introduced the concept of lower and upper approximations, which are useful for measuring of the quality and accuracy of classification. Denote U a finite set of objects, Q as a finite set of features, and let $P \subseteq Q$ and $Y \subseteq U$.

The *P-lower approximation* of Y , denoted as $\underline{P}Y$, is the set of all elements of U , which can be certainly classified as elements of Y based on the set of features P .

The *P-upper approximation* of Y , denoted as $\overline{P}Y$, is the set of elements of U , which can be possibly classified as elements of Y based on the set of features P .

The two definitions are expressed formally as

$$\underline{P}Y = \bigcup X \{X \in P^* \text{ and } X \subseteq Y\}$$

$$\overline{P}Y = \bigcup X \{X \in P^* \text{ and } X \cap Y \neq \emptyset\}$$

where P^* is the family of all equivalence classes of indiscernibility relation P^r on the set U . Two objects x and y are indiscernible on the set of features P (xP^ry) if $r(x, q) = r(y, q)$ for every $q \in P$.

Equivalence classes of P^r are called *P-elementary sets* in the set of objects (data set).

Atoms are the Q -elementary sets of objects.

Approximation accuracy (AA) of a data set is the ratio of the total lower approximation for all decision classes and the total upper approximation for all decision classes.

Boundary approximation is the difference between the upper and lower approximation.

Classification accuracy (CA) of a rule set is the ratio of the number of correctly classified objects from the test set and all objects in the test set [23].

Classification quality (CQ) of a feature set is the ratio of the number of objects in the lower approximation and the total number of objects in the data set.

In some areas, e.g., medicine, a broader definition of accuracy is used [24]. Accuracy is defined as the total number of true positives added to the total number of true negatives divided by the total number of patients studied [25], i.e., accuracy = $(A + D)/(A + B + C + D)$ (see Fig. 16).

Based on the quadrant in Fig. 16 the following metrics are defined in addition to accuracy [25]:

- Sensitivity* (true positive rate) = $A/(A + C)$.
- Specificity* (true negative rate) = $D/(B + D)$.
- Positive predicted value* = $A/(A + B)$.
- Negative predicted value* = $D/(C + D)$.

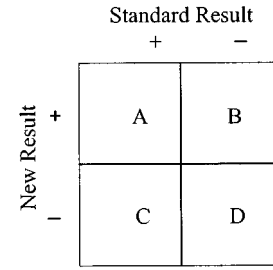


Fig. 16. Classification quadrant.

No.	Features				
	F1	F2	F3	F4	D
1	0	1	Yes	2	0
2	0	0	Yes	3	1
3	1	1	No	2	2
4	0	0	No	1	0
5	0	1	Yes	0	0
6	0	0	No	1	2

Fig. 17. Six object data set.

Rule length is the number of elementary condition elements in the rule.

Rule strength is the number of objects in the data set that have the property described by the conditions and the decision of the rule.

Exact rule = an outcome corresponds to one or more different conditions.

Approximate rule = the same condition corresponds to more than one outcome. Note that exact rules are generated for the set of objects in the lower approximation, while approximate rules are generated for the boundary.

Rule support is the number of all objects in the data set that have the property described by the conditions of the rule.

Rule coverage is the proportion of the objects in the training set that are identifiable by this rule.

Both rule coverage and rule support are estimators of the conditional probability.

Rule acceptance is a subjective measure that reflects the user confidence in the extracted rules. It is more general than the rule support and rule coverage. The rule acceptance measure can be expressed as the number of condition terms of a rule.

Discrimination level measures the level of precision with which a rule represent the corresponding objects.

The most basic definitions introduced above are illustrated with the data set in Fig. 17 containing six objects, four features, and the decision D .

The classification quality of each single feature is as follows: $CQ(F1) = .167$, $CQ(F2) = 0$, $CQ(F3) = 0$, $CQ(F4) = .333$. For example, for feature F1 object 3 can be uniquely identified, therefore for $F1 = 1$, $CQ(F1) = 1/6 = .167$.

The classification quality of selected pairs of features is as follows: $CQ(F1, F2) = .5$, $CQ(F2, F3) = .667$. For example, for the feature set $\{F1, F2\}$ three objects 1, 2, and 5 can be uniquely identified, therefore $CQ(F1, F2) = 3/6 = .5$.

The classification quality of selected triple features is as follows: $CQ(F1, F2, F3) = .667$, $CQ(F2, F3, F4) = .667$. For example, for the feature set $\{F1, F2, F3\}$ four objects 1, 2, 3, and

Exact rules

- Rule 1. IF (F2 = 1) AND (F3 = Yes) THEN (D = 0); [2, 66.67%, 100.00%] [1, 5]
 Rule 2. IF (F4 = 3) THEN (D = 1); [1, 100%, 100%] [2]
 Rule 3. IF (F1 = 1) THEN (D = 2); [1, 50%, 100%] [3]

Approximate rule

- Rule 4. IF (F4 = 1) THEN (D = 0) OR (D = 2); [2, 100%, 100%] [4, 6]

Fig. 18. Exact and approximate rules derived from the data set in Fig. 17.

5 can be uniquely identified, therefore $CQ(F1, F2, F3) = 4/6 = .667$.

The classification quality of all features is $CQ(F1, F2, F3, F4) = .667$.

Reducts: {F1, F4}, {F3, F4}, {F3, F4}

Core: { \emptyset }

The classification quality of the core: 0

Number of decision classes: 3 (D = 0, 1, 2 in Fig. 17)

Number of atoms: 5

Class $D = 0$

Number of objects: 3

Lower approximation: 2

Upper approximation: 4

Approximation accuracy: 0.5

Class $D = 1$

Number of objects: 1

Lower approximation: 1

Upper approximation: 1

Approximation accuracy: 1

Class $D = 2$

Number of objects: 2

Lower approximation: 1

Upper approximation: 3

Approximation accuracy: 0.333

Based on the above values for the data set in Fig. 17

- The classification quality of all features is $(2 + 1 + 1)/6 = .667$, and
- The approximation accuracy is $(2+1+1)/(4+1+3) = .5$.

The knowledge extracted from a data set may follow different formats, with the most typical being decision tree, structured matrix, and decision rules. A typical format of a rule extracted from a data set is as follows:

IF (Condition) THEN (Outcome) [Rule support, Rule coverage, Discrimination level] [List of supporting objects]

The following four exact and approximate rules have extracted from the set in Fig. 17.

Decision support measure (DSM) is the total number of rules supporting a decision. It can be also expressed with the number of objects from the training set that support the decision.

Decision redundancy factor (DRF) is the number of mutually exclusive feature sets associated with the same decision (see Fig. 18).

REFERENCES

- [1] Z. Pawlak, "Rough sets," *Int. J. Inform. Comput. Sci.*, vol. 11, no. 5, pp. 341–356, 1982.
- [2] P. Langley and H. A. Simon, "Applications of machine learning and rule induction," *Commun. ACM*, vol. 38, no. 11, pp. 55–64, 1995.
- [3] T. Y. Lin and N. Cercone, *Rough Sets and Data Mining*, T. Y. Lin and N. Cercone, Eds. Boston, MA: Kluwer, 2000.

- [4] J. G. Carbonell, *Machine Learning: Paradigms and Methods*, J. G. Carbonell, Ed. Cambridge, MA: MIT Press, 1990.
- [5] R. S. Michalski, I. Bratko, and M. Kubat, *Machine Learning and Data Mining*, R. S. Michalski, I. Bratko, and M. Kubat, Eds. New York: Wiley, 1998.
- [6] T. Mitchell, *Machine Learning*. New York: MacGraw Hill, 1997.
- [7] R. Reiter, "A theory of diagnosis from first principles," *Artif. Intell.*, vol. 35, pp. 57–95, 1987.
- [8] I. Uysal and H. A. Guvenir, "An overview of regression techniques for knowledge discovery," *Knowl. Eng. Rev.*, vol. 14, no. 4, pp. 319–340, 1999.
- [9] A. Barto and R. S. Sutton, *Reinforcement Learning*. Cambridge, MA: MIT Press, 1998.
- [10] J. M. Carroll and J. Olson, *Mental Models in Human-Computer Interaction: Research Issues About the User of Software Knows*. Washington, DC: Nat. Acad. Press, 1987.
- [11] G. Wickens, S. E. Gordon, and Y. Liu, *An Introduction to Human Factors Engineering*. New York: Harper Collins, 1998.
- [12] J. Koza, *Genetic Programming*. Cambridge, MA: MIT Press, 1992.
- [13] P. J. Bentley, *Evolutionary Design by Computers*, P. J. Bentley, Ed. San Francisco, CA: Morgan Kaufmann, 1999.
- [14] A. Kusiak, *Computational Intelligence in Design and Manufacturing*. New York: Wiley, 2000.
- [15] K. Kapur, *Concurrent Engineering: Automation, Tools, and Techniques*, A. Kusiak, Ed. New York: Wiley, 1993.
- [16] D. H. Besterfield, *Quality Control*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [17] R. J. Bayardo, Jr. and R. Agrawal, "Mining the most interesting rules," in *Proc. 5th Int. ACM SIGKDD Conf. Knowledge Discovery Data Mining*, 1999, pp. 145–154.
- [18] U. S. Carlin, J. Komorowski, and A. Ohrn, "Rough set analysis of patients with suspected of acute appendicitis," in *Proc. IPMU'98*, Paris, France, 1998, pp. 1528–1533.
- [19] A. Kusiak, "Decomposition in data mining: An industrial case study," *IEEE Trans. Electron. Packag. Manufact.*, vol. 23, pp. 345–353, Oct. 2000.
- [20] M. Stone, "Cross-validators choice and assessment of statistical predictions," *J. R. Stat. Soc.*, vol. 36, pp. 111–147, 1974.
- [21] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA: Kluwer, 1991.
- [22] N. Shan, W. Ziarko, H. J. Hamilton, and N. Cercone, "Using rough sets as tools for knowledge discovery," in *Proc. 1st Int. Conf. Knowledge Discovery Data Mining*, U. M. Fayyad and R. Uthurusamy, Eds. Menlo Park, CA, 1995, pp. 263–268.
- [23] Z. Pawlak, K. Slowinski, and R. Slowinski, "Rough classification of patients after highly selective vagotomy for duodenal ulcer," *Int. J. Man-Mach. Stud.*, vol. 24, pp. 413–433, 1998.
- [24] A. Kusiak, J. A. Kern, K. H. Kernstine, and T. L. Tseng, "Autonomous decision-making: A data mining approach," *IEEE Trans. Inform. Technol. Biomed.*, vol. 4, pp. 274–284, Dec. 2000.
- [25] B. Rosner, *Fundamentals of Biostatistics*. Boston, MA: PWS, 1982.



Andrew Kusiak is a Professor of industrial engineering at the University of Iowa, Iowa City. He is interested in theory and applications of computational intelligence, data mining, and optimization in product development, manufacturing, and healthcare. He has published research papers in journals sponsored by AAAI, IEEE, IIE, INFORMS, ESOR, IFIP, IFAC, IPE, ISPE, and SME. He speaks frequently on international meetings, conducts professional seminars, and consults for industrial corporations. He serves on the editorial boards of sixteen journals and edits book series. He is the Editor-in-Chief of the *Journal of Intelligent Manufacturing*.