# Approximate Boolean Reasoning Approach to Rough Sets and Data Mining

Hung Son Nguyen

Institute of Mathematics, Warsaw University,
Banacha 2, 02-097 Warsaw, Poland
son@mimuw.edu.pl

**Abstract.** Many problems in rough set theory have been successfully solved by boolean reasoning (BR) approach. The disadvantage of this elegant methodology is based on its high space and time complexity. In this paper we present a modified BR approach that can overcome those difficulties. This methodology is called the approximate boolean reasoning (ABR) approach. We summarize some most recent applications of ABR approach in development of new efficient algorithms in rough sets and data mining.

**Keywords:** Rough sets, data mining, boolean reasoning.

## 1 Introduction

Concept approximation problem is one of most important issues in machine learning and data mining. Classification, clustering, association analysis or regression are examples of well known problems in data mining that can be formulated as concept approximation problems. A great effort of many researchers has been done to design newer, faster and more efficient methods for solving concept approximation problem.

Rough set theory has been introduced by [14] as a tool for concept approximation under uncertainty. The idea is to approximate the concept by two descriptive sets called *lower and upper approximations*. The lower and upper approximations must be extracted from available training data. The main philosophy of rough set approach to concept approximation problem is based on minimizing the difference between upper and lower approximations (also called the *boundary region*). This simple, but brilliant idea, leads to many efficient applications of rough sets in machine learning and data mining like feature selection, rule induction, discretization or classifier construction [4].

As boolean algebra has a fundamental role in computer science, the boolean reasoning approach is also an ideological method in Artificial Intelligence. In recent years, boolean reasoning approach shows to be a powerful tool for designing effective and accurate solutions for many problems in rough set theory. This paper presents a more generalized approach to modern problems in rough set theory as well as their applications in data mining. This generalized method is called the *approximate boolean reasoning* (ABR) approach.

D. Ślęzak et al. (Eds.): RSFDGrC 2005, LNAI 3642, pp. 12–22, 2005.

## 2 Boolean Reasoning Approach

Boolean reasoning approach is a general framework for solving decision and optimization problems. This method comes from the great idea of George Boole to whom we owe a possibility of using symbolic notation in mathematics. He proposed to solve a problem by (1) converting it to a boolean formula, (2) solving a corresponding problem for boolean formula and (3) decoding the solution for boolean function to obtain the solution of the original problem.

By boolean function we denote any function $f : \{0,1\}^n \rightarrow \{0,1\}$. Boolean functions can be described by boolean formulas, i.e., expressions constructed by boolean variables from a set $VAR = \{x_1, ..., x_k\}$, and boolean operators like conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$).

The most famous problem related to boolean functions is the satisfiability problem (SAT). It is based on checking, for a given boolean function, whether there exists such an evaluation of variables that the function becomes satisfied. In other words, the problem is to solve the equation $f(x_1, ..., x_n) = 1$. SAT is the first problem which has been proved to be NP-complete (the Cook's theorem). This important result is used to prove the NP-hardness of many other problems by showing the polynomial transformation of SAT to the studied problem. From practical point of view, any SAT-solver (heuristical algorithm for SAT) can be used to design heuristic solutions for all problems in the class NP. Therefore, instead of solving a couple of hard problems, the main effort may be limited to create efficient heuristics for the SAT problem.

One of possible solutions for scheduling problem is based on SAT-solver. In this method, the specification of scheduling problem is formulated by a boolean function, where each variable encodes one possible assignment of tasks, resources, time slots, etc. The encoding function is satisfiable if and only if there exists a correct schedule for the given specification [17].

The following steps should be taken into account when applying boolean reasoning approach:

– **Encoding:** this is the most important step in BR scheme. It begins with determining the set of boolean variables and their meanings in the original problem. Later, the specification of the studied problem and input data are encoded by boolean expressions over selected variables.
– **Solving the corresponding problem for boolean function:** this step is independent with the original problem. The problem is to select the relevant solution for the encoding boolean function. Selection criteria may be related to the complexity and efficiency of existing solutions for the problem over boolean function.
– **Decoding:** in this step, the solution for the problem over boolean function is converted into the solution of the original problem.

SAT is more useful for solving decision problems. In this paper we consider another problem for boolean functions called *minimal prime implicant problem* that is more suitable for optimization problems. Let us briefly describe this problem in more details.

### 2.1 Prime Implicant Problems

The boolean function $\phi : \{0,1\}^n \rightarrow \{0,1\}$ is called "*monotone*" if

$$\forall_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} (\mathbf{x} \leqslant \mathbf{y}) \Rightarrow (\phi(\mathbf{x}) \leqslant \phi(\mathbf{y}))$$

It has been shown that monotone functions can be represented by a boolean expression without negations.

Let $\phi$ be a monotone boolean function which can be expressed as a boolean formula over the set of boolean variables $VAR = \{x_1, ..., x_n\}$. The term $\mathbf{T} = x_{i_1} \wedge ... \wedge x_{i_k}$ is called *implicant* of $\phi$ if $\mathbf{T}(\mathbf{x}) \leq \phi(\mathbf{x})$ for any $\mathbf{x} \in \{0,1\}^n$. The term $\mathbf{T}$ is called *prime implicant* of $\phi$ if (1) $\mathbf{T}$ is an implicant and (2) any term $\mathbf{T}'$, which is obtained from $\mathbf{T}$ by removing some variables, is not implicant of $\phi$. If the set of all prime implicants of $\phi$ is denoted by $PI(\phi)$, then $f(\mathbf{x}) = \bigvee_{\mathbf{T}\in PI(\phi)} \mathbf{T}$.
Let us consider the following problem:

**MINIMAL PRIME IMPLICANT PROBLEM:**
**Input:** Monotone boolean function $f$ of $n$ variables.
**Output:** A prime implicant of $f$ with the minimal length.

It has been shown that the minimal prime implicant problem is NP-hard and the corresponding decision problem, e.g., checking the existence of prime implicant of a given length, is NP-complete [3].

### 2.2 Boolean Reasoning Approach to Optimization Problems

Most problems in data mining are formulated as optimization problems. We will show in the next Section that prime implicant problem is very useful for application of boolean reasoning approach to optimization problem. The general boolean reasoning scheme (BR-scheme) for optimization problems is presented in Figure 1.

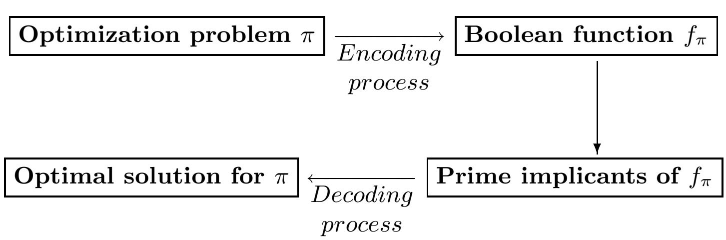


**Fig. 1.** The boolean reasoning scheme for solving optimization problems

Since the minimal prime implicant problem is NP-hard, it cannot be solved (in general case) by exact methods only. It is necessary to create some heuristics to search for short prime implicants of large and complicated boolean functions.

Usually, the input boolean function is given in the CNF form, i.e., it is presented as a conjunction of clauses, and the minimal prime implicant problem is equivalent to the problem of searching for minimal set of variables that has nonempty intersection with each clause of the given function. Let us mention some well known heuristics that have been proposed for prime implicant problem:

1. **Greedy algorithm:** the prime implicant can be treated as a set covering problem, where a set of variables $X$ is said to cover a clause $C$ if $X$ contains at least one variable of $C$. Therefore, in each step, greedy method selects the variable that most frequently occurs within clauses of the given function and removes all those clauses which contain the selected variable.
2. **Linear programming:** the minimal prime implicant can also be resolved by converting the given function into a system of linear inequations and applying the Integer Linear Programming (ILP) approach to this system. More details are described in [15].
3. **Simulated annealing:** many optimization problems are resolved by a Monte-Carlo search method called simulated annealing. In case of minimal prime implicant problem, the search space consists of subsets of variables and the cost function for a given subset $X$ of variables is defined by the size of $X$ and the number of clauses that are uncovered by $X$, see [16].

## 3 Boolean Reasoning Approach to Rough Set Problems

As we have introduced before, searching for approximation of a concept is a fundamental problem in machine learning and data mining. Classification, clustering, association analysis, and many other tasks in data mining can be formulated as concept approximation problems. Let $\mathcal{X}$ be a given universe of objects, and let $\mathcal{L}$ be a predefined descriptive language consisting of such formulas that are interpretable as subsets of $\mathcal{X}$. Concept approximation problem can be understood as a problem of searching for a description $\psi$ of a given concept $C \subset \mathcal{X}$ such that (i) $\psi$ expressible in $\mathcal{L}$ and (ii) the interpretation of $\psi$ should be as close to the original concept as possible. Usually, the concept to be approximated is given on a *finite set of examples* $U \subset \mathcal{X}$, called the training set, only.

The main idea of rough set theory is based on approximating the unknown concept by a pair sets called lower and upper approximations. The lower approximation contains those objects which certainly – according to the actual knowledge of the learner – belong to the concept, the upper approximation contains those objects which possibly belong to the concept.

Let $C \subseteq \mathcal{X}$ be a concept and let $U \subseteq \mathcal{X}$ be a training set. Any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ is called *rough approximation of* $C$ (see [2]) if it satisfies the following conditions:

1. $\mathbf{L} \subseteq \mathbf{U} \subseteq \mathcal{X}$;
2. $\mathbf{L}, \mathbf{U}$ are expressible in the language $\mathcal{L}$;
3. $\mathbf{L} \cap U \subseteq C \cap U \subseteq \mathbf{U} \cap U$;
4. $\mathbf{L}$ is maximal and $\mathbf{U}$ is minimal among those $\mathcal{L}$-definable sets satisfying 3.

The sets $\mathbf{L}$ and $\mathbf{U}$ are called the *lower approximation* and the *upper approximation* of the concept $C$, respectively. The set $\mathbf{BN} = \mathbf{U} - \mathbf{L}$ is called the *boundary region of approximation* of $C$. For objects $x \in \mathbf{U}$, we say that "probably, $x$ is in $C$". The concept $C$ is called *rough* with respect to its approximations $(\mathbf{L}, \mathbf{U})$ if $\mathbf{L} \neq \mathbf{U}$, otherwise $C$ is called *crisp* in $\mathcal{X}$.

The input data for concept approximation problem is given by *decision table* which is a tuple $\mathbb{S} = (U, A, dec)$, where $U$ is a non-empty, finite set of *training objects*, $A$ is a non-empty, finite set of *attributes* and $dec \notin A$ is a distinguished attribute called *decision*. Each attribute $a \in A$ corresponds to the function $a : \mathcal{X} \to V_a$ where $V_a$ is called the *domain* of $a$. For any non-empty set of attributes $B \subseteq A$ and any object $x \in \mathcal{X}$, we define the $B$*-information vector* of $x$ by: $inf_B(x) = \{(a, a(x)) : a \in B\}$. The language $\mathcal{L}$, which is used to describe approximations of concepts, consists of boolean expressions over descriptors of the form $(attribute = value)$ or $(attribute \in set\_of\_values)$. If $C \subset \mathcal{X}$ is a concept to be approximated, then the decision attribute $dec$ is a characteristic function of concept $C$, i.e., if $x \in C$ we have $dec(x) = yes$, otherwise $dec(x) = no$. In general, the decision attribute $dec$ can describe several disjoint concepts.

The first definition of rough approximation was introduced by Pawlak in his pioneering book on rough set theory [14]. For any subset of attributes $B \subset A$, the set of objects $U$ is divided into *equivalence classes* by the *indiscernibility relation* and the upper and lower approximations are defined as unions of corresponding equivalence classes. This definition can be called *the attribute-based rough approximation*. A great effort of many researchers in RS Society has been investigated to modify and to improve this classical approach. One can find many interesting methods for rough approximation like Variable RS Model [24], Tolerance-based Rough Approximation [22], Approximation Space [21], or Classifier-based Rough Approximations [2].

The condition (4) in the above list can be substituted by inclusion to a degree to make it possible to induce approximations of higher quality of the concept on the whole universe $\mathcal{X}$. In practical applications, it is hard to fulfill the last condition. Hence, by using some heuristics we construct sub-optimal instead of maximal or minimal sets. This condition is the main inspiration for all applications of rough sets in data mining and decision support systems.

Let $\mathbb{S} = (U, A \cup \{dec\})$ be a given decision table, where $U = \{u_1, ..., u_n\}$, and $A = \{a_1, ..., a_m\}$. The following rough set methods have been successfully solved by boolean reasoning approach:

**Attribute Reduction:** *Reducts* are subsets of attributes that preserve the same amount of information. In rough set theory a subset of attributes $B \subset A$ is called a decision reduct, if $B$ preserves the same rough approximation of a concept likes $A$. It has been shown in [20] that the problem of searching for minimal reduct of a decision system is equivalent to the minimal prime implicant problem. BR approach has been applied to minimal reduct problem as follows:

– **Boolean Variables:** We associate with each attribute $a_i \in A$ a boolean variable $a_i^*$ for $i = 1, ..., m$.

– **Encoding:** for any pair of objects $u_i, u_j \in U$, where $i, j = 1, ..., n$ we define a discernibility function between $u_i, u_j$ by

$$\psi_{i,j} = \bigvee_{a \in A: a(u_i) \neq a(u_j)} a^*$$

A *discernibility function* $f_{\mathbb{S}}$ for $\mathbb{S}$ is defined by

$$f_{\mathbb{S}}(a_1^*, ..., a_m^*) = \bigwedge_{dec(u_i) \neq dec(u_j)} \psi_{i,j} \quad (1)$$

– **Heuristics:** in the greedy algorithm for reduct problem, quality of a subset of attributes $B$ is measured by the number of pairs of objects that are discerned by $B$. More efficient algorithm based on genetic algorithm was presented in [23].

**Decision Rule Induction:** decision rules are logical formulas that indicate the relationship between conditional and decision attributes. Let us consider those decision rules **r** whose the premise is a boolean monomial of descriptors, i.e.,

$$\mathbf{r} \equiv (a_{i_1} = v_1) \wedge ... \wedge (a_{i_m} = v_m) \Rightarrow (dec = k) \quad (2)$$

In the rough set approach to concept approximation, decision rules are used to define finer rough approximation comparing to attribute-base rough approximation. Each decision rule is supported by some objects and, inversely, the information vector of each object can be reduced to obtain a minimal consistent decision rule. The boolean reasoning approach to decision rule construction from a given decision table $\mathbb{S} = (U, A \cup \{dec\})$ is very similar to the minimal reduct problem. The only difference occurs in the encoding step, i.e.:

– **Encoding:** For any object $u \in U$ in , we define a function $f_u(a_1^*, ..., a_m^*)$, called *discernibility function for* $u$ by

$$f_u(a_1^*, ..., a_m^*) = \bigwedge_{v: dec(v) \neq dec(u)} \psi_{u,v}(x_1, ..., x_k) \quad (3)$$

– **Heuristics:** all heuristics for minimal prime implicant problem can be applied to boolean functions in Equation 3. Because there are $n$ such functions, where $n$ is a number of objects in the decision table, the well known heuristics may show to be time consuming.

**Discretization:** In [6], boolean reasoning approach to real value attribute discretization problem was presented. The problem is to search for a minimal set of cuts on real value attributes that preserve the discernibility between objects. Given a decision table $\mathbb{S} = (U, A \cup \{dec\})$ and a set of candidate cuts $\mathbf{C}$ the discretization problem is encoded as follows:

– **Variables:** Each cut $(a, c) \in \mathbf{C}$ is associated with a boolean variable $x_{(a,c)}$
– **Encoding:** similarly to the reduct problem, a discernibility function between $u_i, u_j \in U$, where $i, j = 1, ..., n$, is defined by

$$\phi_{i,j} = \bigvee_{(a,c) discerns u_i and u_j} x_{(a,c)}$$

and the discretization problem is encoded by the following boolean function

$$\phi = \bigwedge_{dec(u_i) \neq dec(u_j)} \phi_{i,j} \quad (4)$$

– **Heuristics:** again all mentioned heuristics for prime implicant problem can be applied to optimal discretization problem, but we have to take under our attention their computational complexity.

## 4 Approximate Boolean Reasoning Approach

In [6], [18], [9] we have presented few more feature extraction methods based on rough sets and BR approach. Let us mention the following ones:

– **Symbolic value grouping problem:** the idea is to create new features by partition of attribute domains into as less as possible groups of attribute values. This method leads to construction of generalized decision rule of form

$$(a_{i_1} \subset S_1) \wedge ... \wedge (a_{i_m} = S_m) \Rightarrow (dec = k)$$

Each boolean variable encodes a a group of symbolic values in the domain of an attribute.

– **Oblique hyperplanes extraction:** new features are defined by linear combination of the existing ones.

All mentioned problems can be encoded by boolean functions but the complexity of heuristic solutions are very different. Table 1 compares the complexity of encoding functions for the mentioned above problems.

**Table 1.** Complexity of encoding boolean functions for basic problems in rough sets ($n$, $m$ are numbers of objects and attributes of the given decision table, respectively)

| **Problem** | **Complexity of encoding function** |
|---|---|
| minimal reduct | $O(m)$ variables, $O(n^2)$ clauses |
| decision rules | $O(n)$ functions containing $O(m)$ variables and $O(n)$ clauses each |
| discretiztion | $O(mn)$ variables, $O(n^2)$ clauses |
| grouping | $O(\sum_{a \in A} 2^{\|V_a\|})$ variables, $O(n^2)$ clauses |
| hyperplanes | $O(n^m)$ variables , $O(n^2)$ clauses |

The problem with computational complexity becomes more serious in data mining applications on very large databases. We have proposed a novel solution called *approximate boolean reasoning approach*. Figure 2 presents a general scheme of this method. The idea is to approximate every step in the BR-scheme. Let us discuss some possible techniques that were applied in rough set methods.

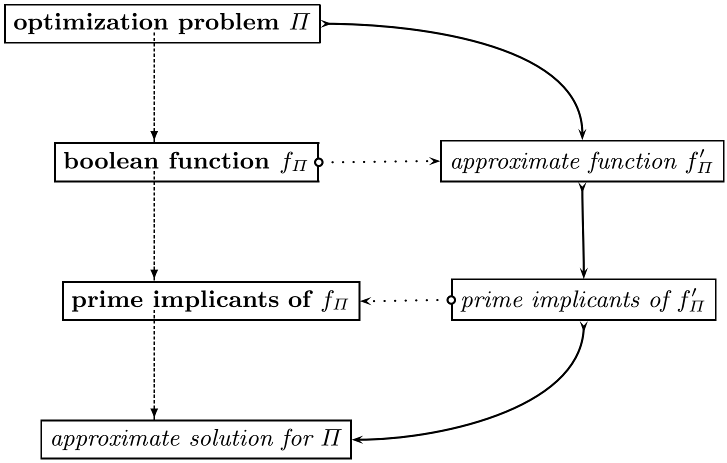


**Fig. 2.** General scheme of approximate boolean reasoning approach

### 4.1 The $\alpha$-Reduct Problem

One method for computation time reduction is based on weakening the requirement of a problem. $\alpha$-reducts are example of this technique, see [7].

The set $B$ of attributes is called $\alpha$-reduct if $B$ has nonempty intersection with at least $\alpha \cdot N$ clauses of the discernibility function (1), where $N$ is the total number of clauses occurring in (1) and $\alpha \in [0, 1]$ is a real parameter.

In some applications (see [19]), e.g., rough classifier construction, $\alpha$-reducts produce much shorter and more accurate classifiers comparing with original reducts. Practical experiments show that in some cases, the 95%-reducts are two times shorter than 100%-reducts.

Let $\mathbb{S} = (U, A)$ be a given information system and let $\mathbf{T} = D_1 \wedge D_2 ... \wedge D_k$ be an extracted pattern (or frequent itemset [1]). Consider the set of descriptors $\mathbf{P} \subset \{D_1, D_2 ..., D_k\}$, the implication

$$\bigwedge_{D_i \in \mathbf{P}} D_i \Rightarrow \bigwedge_{D_j \notin \mathbf{P}} D_j$$

1. is 100%-irreducible association rule from $\mathbf{T}$ if and only if $\mathbf{P}$ is reduct in $\mathbb{S}|_{\mathbf{T}}$.
2. is $c$-irreducible association rule from $\mathbf{T}$ if and only if $\mathbf{P}$ is $\alpha$-reduct in $\mathbb{S}|_{\mathbf{T}}$, where $\alpha = 1 - (\frac{1}{c} - 1)/(\frac{n}{s} - 1)$, $n$ is the total number of objects from $U$ and $s = support(\mathbf{T})$.

One can show that for a given $\alpha$, the problems of searching for shortest $\alpha$-reducts and for all $\alpha$-reducts are also NP-hard [10].

### 4.2 Discretization Problem

Two solutions based on approximate boolean reasoning approach have been proposed for discretization problem.

Discretization process always effects on a lost of information. The optimal discretization algorithm preserves the discernibility of all attributes only. Therefore, the discretized decision table is irreducible. We have proposed for a discretization method that preserves some more reducts for a given decision table [5]. This method also can be solved by boolean reasoning approach, but the encoding function consists of $O(mn)$ variables and $O(n^2 2^n)$ clauses. In the approximate boolean reasoning approach the encoding function is replaced by approximate encoding function containing $O(n^2)$ clauses only.

Another discretization method was proposed for relational database systems [8]. This method minimizes the number of simple SQL queries necessary to search for the best cuts by using "divide and conquer" search strategy. To make it possible, we develop some novel "approximate measures" which are defined on intervals of attribute values. Proposed measures are necessary to evaluate a chance that a given interval contains the best cut.

### 4.3 Symbolic Value Grouping Problem

This method is the best demonstration of approximate boolean reasoning approach. This problem can be encoded by a boolean function containing $O(n^2)$ clauses and $O(\sum_{a \in A} 2^{|V_a|})$ variables. We have proposed an approximate encoding function containing $O(\sum_{a \in A} |V_a|^2)$ variables (and still $O(n^2)$ clauses) only [18]. In this application, the decoding process was not trivial, since it is equivalent to a well-known graph vertex coloring problem which is NP-hard. One more heuristical algorithm for this graph coloring problem is necessary to construct a whole solution for symbolic value grouping problem.

## 5 Applications in Data Mining

Rough sets and approximate boolean reasoning approach to data mining has been presented in [11], [9], [7]. Both discretization and symbolic value grouping methods can be used to construct accurate decision trees from large databases.

In the decision tree construction method based on rough sets and boolean reasoning approach, the quality of a split (defined either by a cut on a continuous attribute or by a partition symbolic values) is measured by the number of pairs of objects from different classes that are discerned by the split. In case of large data sets, an application of approximate boolean reasoning approach makes a search for semi-optimal cuts very efficient, particularly when the data set is stored in a relational database system [11]. We have proposed a concept of soft cuts and soft decision trees which have many advantages. Comparing with the standard decision tree concept, soft decision tree model maintains the high classification accuracy, but it can be constructed very fast [12].

The latest applications of approximate boolean reasoning approach is related to the concept of layered learning [13]. This method allows improving the accuracy of concept approximation by utilizing the domain knowledge in the learning process. In cases, when the domain knowledge is given in form of concept ontology, we have proposed a layered learning method based on rough sets and boolean reasoning approach [13].

## 6 Conclusions

We have presented a boolean reasoning approach and its extension called approximate boolean reasoning approach as general methods for designing efficient solutions for rough sets and data mining. Recently, we are working on an application of boolean reasoning approach to decision tables with continuous decision. The method is called the differential calculus for pseudo boolean functions, and some first experiments are showing that this method is quite promising. We are also planning to apply the approximate boolean reasoning method in layered learning algorithms that make approximation of concept from data and domain knowledge possible.

**Acknowledgement.** The research has been partially supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

## References

1. Agrawal R., Imielinski T., Suami A.: Mining Assocation Rules Between Sets of Items in Large Datatabes, ACM SIGMOD. Conference on Management of Data, Washington, D.C., 1993, pp. 207–216.
2. Bazan J., Nguyen H.S., Skowron A., Szczuka M.: A view on rough set concept approximation. In proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003),Chongqing, China. LNAI 2639, Heidelberg, Germany, Springer-Verlag, 2003, pp. 181–188
3. Brown E.M.: *Boolean Reasoning*, Kluwer Academic Publishers, Dordrecht, 1990.
4. Kloesgen W., Żytkow J., eds.: *Handbook of Knowledge Discovery and Data Mining.* Oxford University Press, Oxford, 2002.
5. Nguyen H. S.: Discretization problems for rough set methods. In L. Polkowski and A. Skowron, editors, *New Direction in Rough Sets, Data Mining and Granular-Soft Computing (Proc. of RSCTC'98, Warsaw, Poland)*, LNAI 1424, Springer-Verlag, Berlin Heidelberg, 1998, pp. 545–552.
6. Nguyen H. S., Nguyen S. H.: Discretization methods for data mining. In L. Polkowski and A. Skowron, editors, *Rough Sets in Knowledge Discovery*, Physica-Verlag, Heidelberg New York, 1998, pp. 451–482.
7. Nguyen H. S., Nguyen S. H.: Rough sets and association rule generation. *Fundamenta Informaticae*, 40(4), 1999, pp. 310–318.
8. Nguyen H. S.: Efficient sql-querying method for data mining in large data bases. In *Proc. of Sixteenth International Joint Conference on Artificial Intelligence, IJCAI-99*, Stockholm, Sweden, Morgan Kaufmann, 1999, pp. 806–811.
9. Nguyen H. S.: From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae*, 34(1–2), 1998, pp. 145–174.
10. Nguyen H. S., Slezak D.: Approximate reducts and association rules – correspondence and complexity results. In Proc. of RSFDGrC'99, Yamaguchi, Japan, LNAI 1711, Springer-Verlag, Berlin Heidelberg, 1999, pp. 107–115.

11. Nguyen H. S.: On efficient handling of continuous attributes in large data bases. *Fundamenta Informaticae*, 48(1), 2001, pp. 61–81.
12. Nguyen H. S.: On exploring soft discretization of continuous attributes. In Sankar K. Pal, Lech Polkowski, and Andrzej Skowron, editors, *Rough-Neural Computing Techniques for Computing with Words*, chapter 13, Springer, 2004, pages 333–350.
13. Nguyen S. H., Bazan J., Skowron A., and Nguyen H.S.: Layered learning for concept synthesis. In Peters at al. (eds), *Transactions on Rough Sets I*, volume LNCS 3100, Springer, 2004, pages 187–208.
14. Pawlak Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Volume 9 of System Theory, Knowledge Engineering and Problem Solving. Kluwer Academic Publishers, Dordrecht, The Netherlands (1991)
15. Pizzuti C.: Computing Prime Implicants by Integer Programming. In Proceedings of IEEE International Conference on Tools with Artificial Intelligence, 1996.
16. Sen S.: Minimal cost set covering using probabilistic methods In Proceedings 1993 ACM/SIGAPP Symposium on Applied Computing, 1993, pp. 157–194.
17. Selman B., Kautz H., McAllester D.: Ten Challenges in Propositional Reasoning and Search. In *Proc. IJCAI'97*, Japan, 1997.
18. Skowron A., Nguyen H. S.: Boolean reasoning schema with some applications in data mining. In J. Zytkow and J. Rauch, editors, Principple of Data Mining and Knowledge Discovery (Proc. of PKDD'1999, Praga, Czech), LNAI 1704, Springer-Verlag, Berlin Heidelberg, 1999, pp. 107–115.
19. Skowron A. Synthesis of adaptive decision systems from experimental data. In A. Aamodt, J. Komorowski (eds), Proc. of the $5^{th}$ Scandinavian Conference on AI (SCAI'95), IOS Press, May 1995, Trondheim, Norway, pp. 220–238.
20. Skowron A., Rauszer C.: The discernibility matrices and functions in information systems. In: R. Słowiński (Ed.), *Intelligent decision support: Handbook of applications and advances of the rough sets theory*, Kluwer Academic Publishers, Dordrecht, 1992, pp. 331-362.
21. Skowron A., Stepaniuk J.: Tolerance approximation spaces. Fundamenta Informaticae **27**, 1996, pp. 245–253
22. Słowiński R., Vanderpooten D., 1995. *Similarity Relation as a Basis for Rough Approximations*. In: P. Wang (Ed.): Advances in Machine Intelligence & Soft Computing, Bookwrights, Raleigh NC, 1997, pp. 17–33.
23. Wróblewski J.: Covering with reducts - a fast algorithm for rule generation. In: Polkowski L., Skowron A.(eds.): Proc. of RSCTC'98, Warsaw, Poland. Springer-Verlag, Berlin, 1998, pp. 402–407.
24. Ziarko W.: Variable Precision Rough Set Model. In Journal of Computer and System Sciences, Vol. 46, 1993, pp. 39-59.