The State of Rough Sets for Database Mining Applications

Vijay V. Raghavan
The Center for Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, LA 70504, USA

Hayri Sever
The Department of Computer Science
University of Southwestern Louisiana
Lafayette, LA 70504, USA

Abstract

The database mining problem is often cited as one of the most promising research topics in the fields of database systems and machine learning. Although many available machine learning algorithms are potentially applicable, real-world databases pose additional difficulties partly due to the nature of their contents. In this article, we describe the characteristic features of the database mining problem, a subset of data mining queries, and the approaches for designing a database mining environment. Then, in that context, we summarize the state of rough sets and present future directions.

1 Introduction

It is estimated that the amount of information in the world doubles every 20 months[10]; that is, some scientific, government and corporate information systems are being overwhelmed by a flood of data that are generated and stored, routinely. These massive amounts of data are beyond human experts' ability to be analyzed, though they contain potential gold mine of valuable information. Unfortunately, the database technology of today offers little functionality to explore such data. At the same time, knowledge discovery [1] techniques for intelligent data analysis are not yet mature for large data sets[23, 2, 49, 10, 25]. Furthermore, the fact that data has been organized and collected around the needs of organizational activities may pose a real difficulty in locating relevant data for knowledge discovery techniques from diverse sources. The

[1]Knowledge discovery can be defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from data.

database mining[2] problem is defined to emphasize the challenges of knowledge discovery in large databases and to motivate researchers and application developers for meeting that challenge. It comes from the idea that *large databases* can be viewed as data mines which can be discovered by *efficient* knowledge discovery techniques.

Database mining is a promising area of interest shared by database systems and AI research communities [10, 39]. Recently, due to strong demand for participation and the growing demand for formal proceedings, the American Association for Artificial Intelligence (AAAI) in cooperation with International Joint Conference on Artificial Intelligence (IJCAI) has elevated the format of the previous Knowledge Discovery and Data mining (KDD) workshops to that of an international conference on KDD. It may however be worth pointing out that the connection of the data mining problem to a database is loosely defined because of the terminological gap between AI and database communities on perceiving what a database is; that is, the researchers in database systems community think of a database as a data base within a database management system, while the researchers in AI community consider it as a simple file structure or an off-line data collection. Therefore, the nature of the problem depends on the context that one intends to target. If the knowledge model is integrated/related to a data base within a DBMS, then it should also address issues related to the management of data such as data security, view levels of data, transaction management, and the use of general database functions/facilities. In this article, we consider a relational DBMS since we limit ourselves to knowledge discovery in business databases.

Even though it has been more than a decade since the introduction of the rough set theory, there is still a need for further development of rough functions and for extending rough set model to new applications[32]. We believe that the investigation of the rough set methodology for database mining in relational DBMSs is a challenging research area with promise of high payoffs in many business and scientific domains. Additionally, such investigations will lead to the integration of the rough set methodology with other knowledge discovery methodologies, under the umbrella of database mining applications. In this article, we asses the current status of and trends in the database mining problem from the point of the rough set theory.

This article is organized as follows. In next section, we present broad characteristics of the database mining problem and an interesting subset of data mining queries. In section 3, we focus on the state of rough set methodology in the context of database mining and discuss research directions in rough set theory to make the rough set model suitable for database mining applications.

2 Database mining issues

We discuss database mining issues under two subsections. First, we present various characteristics of the data that should be addressed by a KDD system. Then we consider the case where a KDD system has a DBMS interface. Second, we present a taxonomy of database mining queries, which is not exhaustive; however, it constitutes an interesting subset of the ones cited in the literature[1, 4, 19, 25, 36].

[2] In the literature, the database mining problem is also known as *data mining* or the knowledge discovery in databases.

2.1 Characteristic features of the database mining problem

1. Ultra Large Data: The volume of data in real world database systems has already reached to the level of giga (or tera) bytes and continues to grow rapidly. Therefore, it is impossible to apply knowledge discovery techniques involving exhaustive search over this massive data. A plausible approach to tackle ultra large data is to reduce the data set horizontally by merging identical tuples following the substitution of an attribute value by its higher level value in a generalization hierarchy of categorical attributes [13, 49, 37] or vertically by applying some feature selection methods[17, 3]. One could also use random sampling methods along with the horizontal/vertical pruning methods, but this method has not received much attention yet.

2. Noisy Data: Non-systematic errors, which can occur during data-entry or collection of data, are usually referred as *noise*. Unfortunately there is little support by commercial DBMSs to eliminate/reduce errors that occur during data entry, though the potential exists for providing the capability, in relational data models, to force consistency among attribute values with respect to predefined functional dependencies[6]. There is virtually nothing that a DBMS can do to catch the errors introduced by the error-prone collection of data. Hence, erroneous data can be a significant problem in real-world databases. This implies that a knowledge discovery method should be less sensitive to noise in the data set. This problem has been extensively investigated for variations of inductive decision trees, depending on where and how much the noise occurs[33, 5].

3. Null Values: In DBMSs, a null value (also known as missing value) may appear as the value of any attribute that is not a part of the primary key and is treated as a symbol distinct from any other symbol, including other occurrences of null values[42]. The null value does not only mean an *unknown* value, but also can mean *inapplicable*. In relational databases, this problem occurs frequently because the relational model dictates that all tuples in a relation must have the same number of attributes, even if values of some attributes are inapplicable for some tuples. For example, in the list of personal computers, the attribute that contains the model type of the sound cards would be null for some model of computers. We have not come across any work that deals with null values, though there are some recent studies on unknown values[24, 11, 34].

4. Incomplete or Redundant Data: The fact that data has been organized and collected around the needs of organizational activities causes incomplete or redundant data from the view point of the knowledge discovery task. The situation of incomplete data arises when the available information based on a given set of attributes is not precise enough for arriving at certain conclusions. Under such circumstances, the knowledge discovery model should have the capability of providing approximate decisions with some confidence level. Alternatively, the given data set may contain redundant or insignificant attributes with respect to the problem at the hand. This case might arise in several situations. For example, combining relational tables to gather relevant data set may result in redundant attributes that the user is not aware of, since unnormalized relational tables may involve redundant features in their contents. Fortunately, there exist many near-optimal solutions,

or optimal solutions in special cases, with reasonable time complexity that eliminate redundant (or insignificant) attributes from a given attribute set by using weights of either individual attributes or combination of some attributes.

If the database is on-line then we should acknowledge that a fundamental characteristic of most on-line databases is that they are **dynamic**; that is, their contents are ever changing. This situation has several important implications for the Knowledge Discovery (KD) method. First, if a knowledge discovery model is implemented as a database application then the running time efficiency of a knowledge discovery method within the KD model and its use of retrieval functions of the DBMS become important factors for the performance evaluation of the KD method, because the KD methods are strictly read-only, long-running transactions. Second, if we regard the resulting knowledge obtained from dynamic data by a knowledge discovery method persistent in the knowledge discovery system, then we do not have the luxury of having snapshots of data available to us, which is a common practice in machine learning discipline, due to the fact that it may lead to erroneous discoveries. Instead, the knowledge discovery method should have the capability of evolving derived knowledge incrementally as the data changes over time. Active database systems have already provided trigger facilities (or *if-then* action rules) that can be used for implementing incremental knowledge discovery methods.

2.2 Database Mining Queries

1. Data Dependency Query: Data dependencies (also known as functional dependencies) in DBMSs, which are intensional data, are defined during the design of conceptual schema, whereas in machine learning they are induced from given data. Depending on how data dependencies are perceived, their use in these two disciplines is different. For example, data dependencies in DBMSs are used for normalizing relations and indexing relations, whereas in machine learning they are used for reducing the number of attributes given in a data set or constructing a data dependency graph. It is sometimes useful to determine associations among values of certain attributes. This kind of data dependency queries is called *hypothesis testing.* It may also be interesting to determine *associations* among values of an attribute. For example, planning department at a supermarket may like to know if the customer who purchase 'bread' and 'butter' also tends to purchase 'milk', where 'butter', 'bread', and 'milk' are usually stored within the same attribute of a sales transaction. Note that we use attribute in its narrow sense, database mining queries can, in broader sense, be classified as an application or variation of data dependency analysis.

2. Classification Query: This kind of query involves inducing a classification function (also known as inducing a classifier, supervised learning, concept learning or discriminating description of classes) that partitions a given set of tuples into meaningful disjoint subclasses with respect to user defined labels or the values of some decision attributes. This subject has extensively been investigated in the literature [9, 17, 33, 38, 29] and is the primary task of inductive learning.

3. Clustering Query: We call unsupervised partitioning of tuples of a relational table a clustering query (also known as unsupervised learning in the context of inductive learning).

There are numerous clustering algorithms ranging from the traditional methods of pattern recognition [9, 16] to clustering techniques in machine learning[35]. User-defined parameters such as the maximum number of tuples within a cluster or the number of clusters can influence the result of a clustering query. Clustering queries may be helpful for the following two cases. First, the user may not know the nature or structure of the data. Second, even if the user have some domain knowledge, labeling a large set of tuples can be surprisingly costly and time consuming. Instead, a classifier may be designed on a small, labeled set of samples, and then *tuned up* by allowing it to run without supervision on a large and unlabeled set of tuples. Alternatively, interactive cluster techniques may be applied, which combine the computer's computational power with a human's knowledge.

4. Characterization Query: A classification query emphasizes the finding of features that distinguish different classes. On the other hand, the characterization query describes common features of a class regardless of the characteristics of other classes. Typical examples of characterization methods can be found in[19, 13].

3 The state of rough set computation

> The important question is not "What are approximation spaces ?"; the important question is "How approximation spaces are used ?", R. E. Kent[20].

The rough set theory[3], based on either algebraic or probabilistic approximation spaces, is used to reason about data that may contain uncertain information for a particular knowledge discovery task. In the following, we review the state of rough sets with respect to the database mining problem.

The theory of rough sets is based on the premise that the universe of discourse (or the set of objects) is *finite*; that is, it considers a snapshot of a database, which may not be a valid assumption if the background knowledge is indeed *dynamic*. A plausible remedy for this problem is to design an incremental method and separate the summary and the result of a method from one to another. For example, in the context of rough classification, the strength of a possible decision rule[47] is a part of the summary of the decision algorithm. Similarly, a further refinement of antecedent parts of rules in a decision algorithm is a part of the summary if the decision algorithm is *persistent* in the system and the background knowledge from which the decision algorithm has been induced is dynamic. In Deogun et al. [7], we presented an upper classification method that we believe would be a good starting point to develop an incremental rough classifier.

In the algebraic space, rough set theory approximates given concept(s) using lower and upper sets of the concept(s). Given that the uncertainty in a data set is caused by *noisy* or *incomplete* data, this approach is not always desirable because it does not exercise opportunities to discover/generalize a valuable pattern that is polluted by noise or that is almost certain. This subject is, however, is not novel to the rough set methodology because there have been numerous works on developing rough approximation methods based on different definitions positive

[3]We assume that the reader is acquainted with the basic notions of the rough set theory[29].

(and boundary) regions[7, 18, 44, 14]. For example, in the *elementary set approximation* of an unknown concept[7], an elementary set is mapped to the positive region of an unknown concept if its degree of membership is bigger than a user defined threshold value. Alternatively, another approach would be to shift the domain of the problem from algebraic space to the probabilistic space, if one can assign prior probabilistic measures to the definable sets [31, 45].

In rough set based classification, inconsistent rough classifiers (or decision algorithms) have not received as much attention as consistent rough classifiers. In the rough set literature, the terms 'inconsistent' and 'nondeterministic' decision algorithms (or rules) are used interchangeably [40, 28, 46], though they are different concepts. The 'inconsistency' is attributed to the result of a classification method while the 'nondetermism' is attributed to the interpretation of that result. As shown in [7], inconsistent decision algorithms, under an appropriate representation structure, can be interpreted deterministically as well as nondeterministically. This is an important result, particularly when the background knowledge is *incomplete and dynamic.*

Redundant data can be eliminated by pruning insignificant attributes with respect to a certain problem at hand. In the rough set terminology, the emphasis is, however, is given to more restrictive version of the redundancy problem that is called reduction of an information system (also known as attribute-value system). It is the process of reducing an information system such that the set of attributes of the reduced information system is independent and no attribute can be eliminated further without losing some information from the system, the result of which is called *reduct*[27, 18]. Given the fact that exhaustive search over the attribute space is exponential in the number of attributes it might not always be computationally feasible to search for the minimum size reduct of attributes. Furthermore, finding just a single reduct of the attributes may be too restrictive for some data analysis problems, which is one of the arguments stated in Kohavi & Frasca's paper[22]. One plausible approach is to utilize the idea of γ-*reduct* given in [30]. Actually, Modrzejewski instead of using the entropy theory applied a variation of γ-*reduct* algorithm to select more significant features for constructing decision trees [26].

The reasoning about *ultra large data set* is a novel area for the rough set methodology. As stated earlier, one of the plausible approaches to tackle ultra large data is to reduce the data set horizontally, which is not strange to the rough set community. For example, in KDD-R system, the data preprocessing unit discretizes the numerical attributes either by using user-supplied discretization formula or by applying an automatic discretization algorithm[48]. Alternatively, horizontal reduction of a very large data set table may use a generalization hierarchy of attributes to merge identical tuples, after the substitution of an attribute value, by its higher level concept in the generalization hierarchy. This is one of the strategies used in the *attribute oriented* approach for inductive concept learning[4, 13, 15]. Since an attribute-oriented learning technique operates on relations, its strategies can be easily adapted to rough classifiers to reduce the size of some categorical attributes.

When we inspect *the database mining queries* with respect to the rough set methodology, we see that attribute dependency analysis and classification are well investigated subjects among others. The *hypothesis testing* and *association between values of an attribute* can easily be solved by the rough set methodology[8]. A recent theoretical paper by Kent [20] extends the notions of approximation and rough equality to formal concept analysis. An immediate result of this study, in our database mining context, is to be able to use the rough set methodology for the *characterization of a concept* (or more generally for concept exploration). As a final note, for handling an interesting subset of database mining queries by the rough set methodology, the rough classifiers face a problem when a new object (coming from outside of the data set) is introduced and the description of the object is not found in the corresponding classifier. In other words, the problem is to find the closeness of given object to known concepts at hand. The usual remedy for this problem is to map non-quantitative values into a numerical scale and use a distance function for the evaluation. For example, Kira & Rendell suggested a binary scale and the authors used it in their *Relief* algorithm for feature selection[21]. A similar methodology was adopted for rough classifiers by Slowinski & Stefanowiski[40], in which the authors incorporated a 'closeness' relation defined on all pairs of attribute values into a distance function.

4 Future Directions

As mentioned in the previous section, some aspects of the nature of data (i.e., incomplete, redundant, and uncertain data) have already been investigated in the rough set methodology, but they need to be tested in large databases. Towards this direction, there have already been some reported works on using the rough set methodology based knowledge discovery tools on off-line data; KDD-R, an experimental open tool box[48]; LERS, a machine learning system from examples[12]; and DataLogic/R[41], a commercial product for database mining and decision support. In the following, we present future research directions that are critical for database mining applications.

- Incremental rough approximation: This is a must feature that has to be provided for if the decision algorithm is to be persistent in the rough set model and the background knowledge is dynamic. One of the claims in [7] is that evolving rough classifier schemes can be developed, if the decision table is accommodated with a composite *increment* field that contains frequencies of rows.

- Closeness of two rules: Slowinski & Stefonowski's study on determining the nearest rule, in the case that the description of a given object does not match to those of known concepts, is a key contribution in enhancing the performance of a rough classifier when the data set is poorly designed or sampled from a large data. Even though it is not stated in the paper, such a measure can make the rough set methodology be used for *clustering queries*. This is a very important subject that needs to be studied by the rough set community.

- Null values: As stated before, a null value of an attribute is more general than unknown value of that attribute, and the reasoning about null values remains an open problem in the studies of database mining. A less restrictive version of the problem, which is known as *unknown attribute values*, has been studied by Grzymala-Busse and implemented in the LERS, a machine learning system[11, 12].

- Characterization query: Even though data dependency analysis within the rough set methodology can be applied to characterize concepts, it lacks an explicit structure such as hierarchy of persistent concepts to exploit concept dependencies. This subject has been formally studied by Wille[43] and used for concept modeling. In previous section, we acknowledged Kent's study [20] on extending the rough set methodology to the area of formal concept analysis. We believe that this study can be further extended to capture approximate characterization of concepts.

- Computational aspects of the rough set methodology: In addition to the concern of the decision algorithm accuracy, database mining applications require efficient techniques. Unfortunately, to our best knowledge, there is virtually no comprehensive study on this subject.

Matheus, Chan, & Piatetsky used the tradeoff between 'versatility' and 'autonomy' for evaluating a KDD system[25]. They argued that an ideal KDD system would handle knowledge discovery tasks autonomously while being applicable across many domains. It is a rather rigorous statement, though it rightfully points out the difference between a KDD system and an expert system. In other terms, the database mining is a practical problem that guides theoretical studies toward understanding the reasoning about *large and existing* data.

5 Summary

In this paper we reviewed the state of rough set methodology and presented future directions in the context of database mining problem. We believe that database mining is such an application area where the theoretical studies of rough set theory can be tested, in order to help us understand its strong and weak sides in practice.

References

[1] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and B. Swami. An interval classifier for database mining applications. In *Proceedings of the 18th VLDB Conf.*, pages 560–573, Vancouver, British Columbia, Canada, 1992.

[2] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Trans. Knowledge and Data Eng.*, 5(6):914–924, 1993.

[3] H. Almuallim and T. Dietterich. Learning with many irrelevant features. In *Proceedings of AAAI-91*, pages 547–552, Menlo Park, CA, 1991. AAAI Press.

[4] Y. Cai, N. Cercone, and J. Han. Attribute-oriented induction in relational databases. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 213–228. AAAI/MIT, Cambridge, MA, 1991.

[5] B. Cestnik, I. Kononenko, and I. Bratko. ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In I. Bratko and N. Lavrac, editors, *Progress in Machine Learning*, pages 31–45. Wilmslow, Sigma, 1987.

[6] C. J. Date. *Relational Database Writings 1985-1989.* Addison-Wesley Publishing Company, Inc., 1990.

[7] J. S. Deogun, V. V. Raghavan, and H. Sever. Rough set based classification methods and extended decision tables. In *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pages 302–309, San Jose, California, 1994.

[8] J. S. Deogun, V. V. Raghavan, and H. Sever. Rough set model for database mining applications. Technical Report TR-94-6-10, The University of Southwestern Louisiana, The Center for Advanced Computer Studies, 1994.

[9] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[10] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery databases: An overview. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 1–27. AAAI/MIT, Cambridge, MA, 1991.

[11] J. W. Grzymala-Busse. On the unknown attribute values in learning from examples. In Z. W. Ras and M. Zemankowa, editors, *Proceedings of Methodologies for Intelligent Systems*, Lecture Notes in AI, 542, pages 368–377. Springer-Verlag, New York, 1991.

[12] J. W. Grzymala-Busse. The rule induction system LERS Q: a version for personal computers. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, page 509, Banff, Alberta, Canada, 1993.

[13] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In *Proceedings of the 18th VLDB Conference*, pages 547–559, Vancouver, British Columbia, Canada, 1992.

[14] R. R. Hashemi, B. A. Pearce, W. G. Hinson, M. G. Paule, and J. F. Young. IQ estimation of monkeys based on human data using rough sets. In *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pages 400–407, San Jose, California, 1994.

[15] X. Hu, N. Cercone, and J. Han. An attribute-oriented rough set approach for knowledge discovery in databases. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, pages 79–94, Banff, Alberta, Canada, 1993.

[16] M. A. Ismail and M. S. Kamel. Multidimensional data clustering: Utilizing hybrid search strategies. *Pattern Recognition*, 22:75–89, 1989.

[17] M. James. *Classification Algorithms.* John Wiley & Sons, 1985.

[18] J. D. Katzberg and W. Ziarko. Variable precision rough sets with asymmetric bounds. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, pages 163–190, Banff, Alberta, Canada, 1993.

[19] K. A. Kaufman, R. S. Michalski, and L. Kerschberg. Mining for knowledge in databases: Goals and general description of the INLEN system. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 449–461. AAAI/MIT, Cambridge, MA, 1991.

[20] R. E. Kent. Rough concept analysis. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, pages 245–253, Banff, Alberta, Canada, 1993.

[21] K. Kira and L. Rendell. The feature selection problem: Tradational methods and a new algorithm. In *Proocedings of AAAI-92*, pages 129–134. AAAI Press, 1992.

[22] R. Kohavi and B. Frasca. Useful feature subsets and rough set reducts. In *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pages 310–317, San Jose, California, 1994.

[23] R. Krishnamurty and T. Imielinski. Research directions in knowledge discovery. *SIGMOD RECORD*, 20:76–78, 1991.

[24] T. Luba and R. Lasocki. On unknown attribute values in functional dependencies. In *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pages 490–497, San Jose, CA, 1994.

[25] C. J. Matheus, P. K. Chan, and G. Piatetsky-Shapiro. Systems for knowledge discovery in databases. *IEEE Trans. on Knowledge and Data Engineering*, 5(6):903–912, 1993.

[26] M. Modrzejewski. Feature selection using rough sets theory. In P. B. Brazdil, editor, *Machine Learning: Proc. of ECML-93*, pages 213–226. Springer-Verlag, 1993.

[27] Z. Pawlak. Rough classification. *Int. J. of Man-Machine Studies*, 20:469–483, 1984.

[28] Z. Pawlak. On learning- a rough set approach. In *Lecture Notes*, volume 208, pages 197–227. Springer Verlak, 1986.

[29] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Boston, MA, 1991.

[30] Z. Pawlak, K. Slowinski, and R. Slowinski. Rough classification of patients after highly selective vagotomy for duodenal ulcer. *Int. J. of Man-Machine Studies*, 24:413–433, 1986.

[31] Z. Pawlak, S. K. M. Wong, and W. Ziarko. Rough sets: Probabilistic versus deterministic approach. *Int. J. of Man-Machine Studies*, 29:81–95, 1988.

[32] Zdzislaw Pawlak. Rough sets present state and further prospects. In *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pages 72–76, San Jose, California, 1994.

[33] J. R. Quinlan. Induction of decision trees. *Machine Learning*, pages 81–106, 1986.

[34] J. R. Quinlan. Unknown attribute values in induction. In A. M. Segre, editor, *Proceedings of the Sixth International Machine Learning Workshop*, pages 164–168, San Mateo, CA, 1989. Morgan Kaufmann Pub.

[35] V. V. Raghavan. Clustering algorithms for information retrieval: An AI perspective. In *Proc. of 9th Annual Hawaii International Conference on System Sciences*, pages 142–151, Honolulu, HI, 1986.

[36] V. V. Raghavan, H. Sever, and J. S. Deogun. A system architecture for database mining applications. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, pages 73–77, Banff, Alberta, Canada, 1993.

[37] V. V. Raghavan, H. Sever, and J. S. Deogun. A system architecture for database mining applications. In W. P. Ziarko, editor, *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Workshops in Computing Series. Springer-Verlag, 1994.

[38] S. Salzberg. *Learning with Nested Generalized Exemplars.* Kluwer Academic Publishers, Boston, MA, 1990.

[39] A. Silberschatz, M. Stonebraker, and J. Ullman. Database systems: achievements and opportunities. Technical Report TR-90-22, University of Texas at Austin, Department of Computer Science, 1990.

[40] R. Slowinski and J. Stefanowiski. Rough classification with valued closeness relation. In *Submitted to Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, San Jose, CA, 1995.

[41] A. J. Szladow. DataLogic/R: for database mining and decision support. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, page 511, Banff, Alberta, Canada, 1993.

[42] J. D. Ulman. *Principles of Database and Knowledge-Base Systems*, volume 1. Computer Science Press, Rockville, Maryland, 1988.

[43] R. Wille. Restructuring lattice theory: An approach based on hierarchies on concepts. In I. Rival, editor, *Ordered Sets*. Reidel, Dordrecht-Boston, 1982.

[44] Y. .Y Yao and X. Li. Uncertainity reasoning with interval-set algebra. In *Proc. of The Int. Workshop on Rough Sets and Knowledge Discovery*, pages 191–201, Banff, Alberta, Canada, 1993.

[45] Y. Y. Yao and K. M. Wong. A decision theoretic framework for approximating concepts. *Int. J. Man-Machine Studies*, 37:793–809, 1992.

[46] R. Yasdi. Learning classification rules from database in the context of knowledge acquisition and representation. *IEEE Trans. on Knowl. and Data Eng*, 3(3):293–306, 1991.

[47] W. Ziarko. The discovery, analysis, and representation of data dependencies in databases. In G. Piatetsky-Shapiro and W .J. Frawley, editors, *Knowledge Discovery in Databases*. AAAI/MIT, Cambridge, MA, 1991.

[48] W. Ziarko and N. Shan. KDD-R: a comprehensive system for knowledge discovery in databases using rough sets. In *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pages 164–173, San Jose, California, 1994.

[49] J. M. Zytkow and J. Baker. Interactive mining of regularities in databases. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 31–53. AAAI/MIT, Cambridge, MA, 1991.