On Modeling Data Mining with Granular Computing

Y.Y. Yao
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: yyao@cs.uregina.ca
URL: http://www.cs.uregina.ca/∼yyao

Abstract

The main objective of this paper is to advocate for formal and mathematical modeling of data mining, which unfortunately has not received much attention. A framework is proposed for rule mining based on granular computing. It is developed in the Tarski's style through the notions of a model and satisfiability. The model is a database consisting of a finite set of objects described by a finite set of attributes. Within this framework, a concept is defined as a pair consisting of the intension, an expression in a certain language over the set of attributes, and the extension, a subset of the universe, of the concept. An object satisfies the expression of a concept if the object has the properties as specified by the expression, and the object belongs to the extension of the concepts. Rules are used to describe relationships between concepts. A rule is expressed in terms of the intensions of the two concepts and is interpreted in terms of the extensions of the concepts. Two interpretations of rules are examined in detail, one is based on logical implication and the other on conditional probability.

1. Introduction

One of the tasks of knowledge discovery and data mining is to search for knowledge, patterns, and regularities derivable from data stored in a database. Typically, rules are used to represent such knowledge [3]. Extensive studies in the field have been focused on algorithms and methodologies for mining different types of rules, as well as speeding up of existing algorithms [3]. Many measures have also been proposed and studied to quantify various aspects of rules, such as confidence, uncertainty, applicability, quality, accuracy, usefulness and interestingness [10]. The diversity observed from studies on the interpretations of rules and algorithms for mining rules, on the one hand, shows the richness of the field, and, on the other hand, suggests the need for a unified framework in which different algorithms and methodologies can be examined and analyzed.

Compared with the vast experimental and algorithmic studies, there is very little attention paid to the formal and mathematical modeling of data mining. While studying different algorithms for rule mining is important, it is equally, if not more, important to study formal aspects of data mining independent of any particular algorithm. A formal model would provide a common ground on which various methods can be studied and compared. Hopefully, a well accepted model can provide a common interpretation for many basic concepts that have been either defined or named differently by researchers. It can be argued that such a model is necessary especially for studying non-algorithmic aspects of data mining. Within a well established model, many fundamental issues can be revisited. For example, one may take a close look at the meanings and interpretations of rules. In doing so, it is hoped that one can classify rules into classes so that they represent specific types of knowledge. With each class, one may explicitly state the conditions testable on a database in order to decide if the database contains the specific type of knowledge, and to design algorithms most suitable for discovering such knowledge. Without satisfactory solutions to those issues, data mining may be mainly an experiment oriented study based on try and error.

While it is easy to argue for the needs and necessity of a formal model for data mining, it may be extremely difficult to decide what make up the model. Roughly speaking, data mining, especially rule mining, deals with concept formation and concept relationship identification. Starting with this observation, we will focus on two aspects of a concept, the intension and extension of the concept [2, 8]. The intension (comprehension) of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The extension of a concept is the set of objects or entities which are instances of the concept. A concept is thus described jointly by its intension and extension, i.e., a set of properties and a set of objects. We express the intension of a concept by a formula, or an expression, of a certain language, and extension as the set of objects satisfying the formula. Our framework is built in the Tarski's style through the notions of a model and satisfiability. The model is a database consisting of a finite set of objects. An object satisfies an expression if the object has the properties as specified by the expression. This formulation enables us to study concepts in a set-theoretic setting. The relationships between concepts are studied based on their corresponding extensions.

A subset of the universe is called a granule in granular computing (GrC). Granular computing is a label of theories, methodologies, techniques, and tools that make use of granules in the process of problem solving [9, 11]. As the proposed framework is mainly based on granules defined by the extensions of concepts, we called it a *Granular Computing Model* for data mining. The proposed model may be considered as an initial step towards formal and mathematical modeling of data mining. Although the model may not immediately offer any new data mining algorithms, the insights brought by the model may have a significant impact.

2 Granular Computing as a Basis for Data Analysis and Mining

In this section, we demonstrate that granular computing is suitable for modeling rule mining and propose a granular computing model for data mining.

2.1 Overview of granular computing

Basic ingredients of granular computing are subsets, classes, and clusters of a universe [9, 12]. There are many fundamental issues in granular computing, such as granulation of the universe, description of granules, relationships between granules, and computing with granules. They have been considered either explicitly or implicitly in many fields, such as data and cluster analysis, concept formation, machine learning, and data mining.

Issues of granular computing may be studied from two related aspects, the construction of granules and computing with granules. The former deals with the formation, representation, and interpretation of granules, while the latter deals with the utilization of granules in problem solving.

Granulation of a universe involves the decomposition of the universe into parts, or the grouping of individual elements into classes, based on available information and knowledge. Elements in a granule are drawn together by indistinguishability, similarity, proximity or functionality [12]. The interpretation of granules focuses on the semantics side of granule constructions. It addresses the question of *why* two objects are put into the same granule. Furthermore, information granulation depends on the available knowledge. In the construction of granules, it is necessary to study criteria for deciding if two elements should be put into the same granule, based on available information. In other words, one must provide necessary semantics interpretations for notions such as indistinguishability, similarity, and proximity. It is also necessary to study granulation structures derivable from various granulations of the universe [11]. The formation and representation of granules deal with algorithmic issues of granule construction. They address the problem of *how* to put two objects into the same granule. Algorithms need to be developed for constructing granules efficiently.

Computing with granules can be similarly studied from both the semantic and algorithmic perspectives. On the one hand, one needs to interpret various relationships between granules, such as closeness, dependency, and association, and to define and interpret operations on granules. On the other hand, one needs to design methodologies and tools for computing with granules, such as approximation, reasoning, and inference.

The relevance of granular computing to data mining can be seen from the view point of concept formation and concept relationship identification, if they are considered as basic functions of data mining.

In the study of formal concepts, every concept is understood as a unit of thoughts that consists of two parts, the intension and extension of the concept [8, 2]. The intension (comprehension) of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The extension of a concept is the set of objects or entities which are instances of the concept. All objects in the extension have the same properties that characterize the concept. In other words, the intension of a concept is an abstract description of common features or properties shared by elements in the extension, and the extension consists of concrete examples of the concept. A concept is thus described jointly by its intension and extension. This formulation enables us to study formal concepts in a logic setting in terms of intensions and also in a set-theoretic setting in terms of extensions.

From the standing point of granular computing, each granule may be interpreted as instances of a concept, i.e., the extension. A name can be associated with a granule to describe or label the concept, i.e., the intension. Once concepts are constructed and described, one can develop computational methods using granules [6, 9]. In particular, one may study relationships between concepts in terms of their intensions and extensions, such as sub-concepts, disjoint and overlap concepts, and partial sub-concepts. These relationships can be conveniently expressed in the form of rules and associated quantitative measures indicating the strength of rules. In summary, one can easily establish connections between tasks of granular computing, concept formation, and rule mining [11].

2.2 A granular computing model for data mining

An information table provides a convenient way to describe a finite set of objects called the universe by a finite set of attributes [5, 11]. Formally, an information table can be expressed as:

$$S = (U, At, \mathcal{L}, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}),$$

where

$$\begin{aligned}
&U \text{ is a finite nonempty set of objects,}\\
&At \text{ is a finite nonempty set of attributes,}\\
&\mathcal{L} \text{ a language defined using attributes in } At,\\
&V_a \text{ is a nonempty set of values for } a \in At,\\
&I_a : U \to V_a \text{ is an information function.}
\end{aligned}$$

Each information function I_a is a total function that maps an object of U to exactly one value in V_a. An information table represents all available information and knowledge. That is, objects are only perceived, observed, or measured by using a finite number of properties.

In an information table, we define a language $\mathcal{L}$ for describing objects of the universe. We adopt the decision logic language (DL-language) studied by Pawlak [5]. Similar languages have been studied by many authors (see [2] and references there).

In the language $\mathcal{L}$, an atomic formula is given by (a, v), where $a \in At$ and $v \in V_a$. If ϕ and ψ are formulas, then so are $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \to \psi$, and $\phi \equiv \psi$. The semantics of the language $\mathcal{L}$ can be defined in the Tarski's style through the notions of a model and satisfiability. The model is an information table S, which provides interpretation for symbols and formulas of $\mathcal{L}$. The satisfiability of a formula ϕ by an object x, written $x \models_S \phi$ or in short $x \models \phi$ if S is understood, is given by the following conditions:

(1) $x \models (a, v)$ iff $I_a(x) = v$,

(2) $x \models \neg\phi$ iff not $x \models \phi$,

(3) $x \models \phi \wedge \psi$ iff $x \models \phi$ and $x \models \psi$,

(4) $x \models \phi \vee \psi$ iff $x \models \phi$ or $x \models \psi$,

(5) $x \models \phi \to \psi$ iff $x \models \neg\phi \vee \psi$,

(6) $x \models \phi \equiv \psi$ iff $x \models \phi \to \psi$ and $x \models \psi \to \phi$.

If ϕ is a formula, the set $m_S(\phi)$ defined by:

$$m_S(\phi) = \{x \in U \mid x \models \phi\}, \tag{1}$$

is called the meaning of the formula ϕ in S. If S is understood, we simply write $m(\phi)$. Obviously, the following properties hold [5]:

(a) $m(a, v) = \{x \in U \mid I_a(x) = v\}$,

(b) $m(\neg\phi) = -m(\phi)$,

(c) $m(\phi \wedge \psi) = m(\phi) \cap m(\psi)$,

(d) $m(\phi \vee \psi) = m(\phi) \cup m(\psi)$,

(e) $m(\phi \to \psi) = -m(\phi) \cup m(\psi)$,

(f) $m(\phi \equiv \psi) = (m(\phi) \cap m(\psi)) \cup (-m(\phi) \cap -m(\psi))$.

The meaning of a formula ϕ is therefore the set of all objects having the property expressed by the formula ϕ. In other words, ϕ can be viewed as the description of the set of objects $m(\phi)$. Thus, a connection between formulas of $\mathcal{L}$ and subsets of U is established.

A formula ϕ is said to be true in an information table S, written $\models_S \phi$, if and only if $m(\phi) = U$, namely, ϕ is satisfied by all objects in the universe. Two formulas ϕ and ψ are equivalent in S if and only if $m(\phi) = m(\psi)$. By definition, the following properties hold [5]:

(i) $\models_S \phi$ iff $m(\phi) = U$,

(ii) $\models_S \neg\phi$ iff $m(\phi) = \emptyset$,

(iii) $\models_S \phi \to \psi$ iff $m(\phi) \subseteq m(\psi)$,

(iv) $\models_S \phi \equiv \psi$ iff $m(\phi) = m(\psi)$.

Thus, we can study the relationships between concepts described by formulas of $\mathcal{L}$ based on the relationships between their corresponding sets of objects.

With the introduction of language $\mathcal{L}$, we have a formal description of concepts. A concept definable in an information table is a pair $(\phi, m(\phi))$, where $\phi \in \mathcal{L}$. More specifically, ϕ is a description of $m(\phi)$ in S, the intension of concept $(\phi, m(\phi))$, and $m(\phi)$ is the set of objects satisfying ϕ, the extension of concept $(\phi, m(\phi))$. A concept $(\phi, m(\phi))$ is said to be a sub-concept of another concept $(\psi, m(\psi))$, or $(\psi, m(\psi))$ a super-concept of $(\phi, m(\phi))$, if $\models_S \phi \to \psi$ or $m(\phi) \subseteq m(\psi)$. A concept $(\phi, m(\phi))$ is said to be a smallest non-empty concept in S if there does not exist another proper sub-concept of $(\phi, m(\phi))$. It can be easily verified that a smallest non-empty concept must be defined by a formula $\bigwedge_{a \in At} \delta(a)$, where $\delta(a)$ is an atomic formula on the attribute a. It consists of objects with exactly the same description in an information table. Two concepts $(\phi, m(\phi))$ and $(\psi, m(\psi))$ are disjoint if $m(\phi) \cap m(\psi) = \emptyset$. If $m(\phi) \cap m(\psi) \neq \emptyset$, we say that the two concepts have a non-empty overlap and hence are related.

For an arbitrary subset of the universe, $A \subseteq U$, it may be impossible to find a concept $(\phi, m(\phi))$ such that $m(\phi) = A$. This means the available information in the information table does not allow us to describe every subset of the universe precisely. Even if we can describe the subset precisely, we may find that the description is not unique. In words, we may find two formulas such that $m(\phi) = m(\psi) = A$. It should also be noted that the definition of concepts is based solely on the information in the information table.

The above formulation of concepts is different from the study of Wille [8] on concept lattice. Instead of using a subset of attributes to represent the intension of a concept, we use a formula from $\mathcal{L}$. In our case, we can also form a concept lattice based on logical implication $\rightarrow$ or set inclusion $\subseteq$. More specifically, for two concepts $(\phi, m(\phi))$ and $(\psi, m(\psi))$, the meet and join are defined by:

$$(\phi, m(\phi)) \sqcap (\psi, m(\psi)) \;=\; (\phi \wedge \psi, m(\phi) \cap m(\psi)),$$
$$(\phi, m(\phi)) \sqcup (\psi, m(\psi)) \;=\; (\phi \vee \psi, m(\phi) \cup m(\psi)). \quad (2)$$

In our formulation, one can easily define the extension based on the intension of a concept. However, the reverse is no longer true as in the case of formal concept lattice suggested by Wille [8]. It may be useful to compare the two formulations of concepts with reference to data mining.

Based on the notions introduced so far, data mining for rules can be viewed as searching for relationship between overlap concepts. A rule can be expressed in the form, $\phi \Rightarrow \psi$, where ϕ and ψ are intensions of two concepts. By expressing rules with intensions of concepts, we may easily explain them in natural language, provided that we can explain formulas of the language $\mathcal{L}$. On the other hand, it may also lead to seductive semantics [1]. A crucial issue is therefore the characterization, classification, and interpretation of rules. It is reasonable to expect that different types of rules represent different kinds of knowledge derivable from a database. Different quantitative measures should be used and different mining algorithms should be designed.

In many studies of machine learning and data mining, a rule is usually paraphrased by an if-then statement, "if an object satisfies ϕ then the object satisfies ψ." The interpretation suggests a kind of cause and effect relationship between ϕ and ψ. However, it is not clear if such a cause and effect relationship does exist. In fact, the interpretation may lead to a common mistake called seductive semantics [1]. The naming and explanation of rules, as being interpreted in their ordinary (non-scientific) usage, convey a far more profound and substantial meaning than can be readily ascertained from the available theoretical and/or empirical evidence. One therefore needs to closely look at the meaning and interpretation of rules.

3 Interpretation of Rules

In the context of fuzzy logic, Zadeh [12] pointed out that although keywords such as IF and THEN are used in describing fuzzy if-then rules, one should not interpret the rules as expressing logical implications. These keywords are used to simply link concepts together. This argument is particularly applicable to rules derived from a database. Following Zadeh, we treat the symbol $\Rightarrow$ in a rule $\phi \Rightarrow \psi$ simply as a connective linking two concepts, as represented by the intensions ϕ and ψ, together. The meanings and interpretations of $\Rightarrow$ are further clarified using the extensions $m(\phi)$ and $m(\psi)$ of the concepts. Rules can be classified based on different interpretations of $\Rightarrow$, depending on the type of knowledge to be represented by $\Rightarrow$.

3.1 Logical interpretation

A rule, $\phi \Rightarrow \psi$, can be interpreted by logical implication, namely, the symbol $\Rightarrow$ is interpreted as the logical implication $\rightarrow$. It may be used to define the so called certain rules [11]. If $m(\phi) \neq \emptyset$ and $\models_S \phi \rightarrow \psi$, we obtain a certain rule. If an object x satisfies ϕ, i.e., $x \in m(\phi)$, by the certain rule, we can conclude that x must satisfy ψ, i.e., $x \in m(\psi)$. One can see that certain rules can only describe sub-concept relationship. In general, two concepts have a non-empty overlap and one is not a sub-concept of the other. In this case, the expression $\phi \rightarrow \psi$ is not true in the information table. Nevertheless, the ratio of objects satisfying $\phi \rightarrow \psi$ can be used to define a quantitative measure of the strength of the rule:

$$T(\phi \Rightarrow \psi) = \frac{|m(\phi \rightarrow \psi)|}{|U|}, \quad (3)$$

where $|\cdot|$ denotes the cardinality of a set. It may be interpreted as a measure of the degree of truth of the expression $\phi \rightarrow \psi$ in the information table.

A problem with the logic implication interpretation can be seen as follows. For an object, if it does not satisfy ϕ, by definition, it satisfies $\phi \rightarrow \psi$. In the degenerated case where $m(\phi) = \emptyset$, we have $\models_S \phi \rightarrow \psi$. Even if the degree of truth of $\phi \rightarrow \psi$ is very high, we may not conclude too much on the satisfiability of ψ given the object satisfies ϕ. In many situation, we want to know the satisfiability of ψ under the condition that the object satisfies ϕ. Our main concern is the satisfiability of ψ in the subset $m(\phi)$. Obviously, logical implication is inappropriate. For the same reason, the notion of conditional has been proposed and studied in the context of rule based expert systems [4].

3.2 Probabilistic interpretation

In data mining, rules are typically interpreted in terms of conditional probability [10]. For a rule $\phi \Rightarrow \psi$, its characteristics can be summarized by the following contingency table:

	ψ	$\neg\psi$	Totals
ϕ	a	b	$a+b$
$\neg\phi$	c	d	$c+d$
Totals	$a+c$	$b+d$	$a+b+c+d=n$

$$a = |m(\phi \wedge \psi)|, \qquad b = |m(\phi \wedge \neg\psi)|,$$
$$c = |m(\neg\phi \wedge \psi)|, \qquad d = |m(\neg\phi \wedge \neg\psi)|.$$

Different measures can be defined to reflect various aspects of rules.

The *generality* of ϕ is defined by:

$$G(\phi) = \frac{|m(\phi)|}{|U|} = \frac{a+b}{n}, \tag{4}$$

which indicates the relative size of the concept ϕ. Obviously, we have $0 \leq G(\phi) \leq 1$. A concept is more general if it covers more instances of the universe. A sub-concept has a lower generality than its super-concept. The quantity may be viewed as the probability of a randomly selected element satisfying ϕ.

The *absolute support* of ψ provided by ϕ is the quantity:

$$\begin{aligned} AS(\phi \Rightarrow \psi) &= AS(\psi|\phi) \\ &= \frac{|m(\psi) \cap m(\phi)|}{|m(\phi)|} = \frac{a}{a+b}, \end{aligned} \tag{5}$$

The quantity, $0 \leq AS(\psi|\phi) \leq 1$, states the degree to which ϕ supports ψ. It may be viewed as the conditional probability of a randomly selected element satisfying ψ given that the element satisfies ϕ. In set-theoretic terms, it is the degree to which $m(\phi)$ is included in $m(\psi)$. Clearly, $AS(\psi|\phi) = 1$, if and only if $m(\phi) \neq \emptyset$ and $\models_S \phi \rightarrow \psi$. That is, a rule with the maximum absolute support 1 is a certain rule. The *mutual support* of ψ and ϕ is defined by:

$$\begin{aligned} MS(\phi, \psi) &= \frac{|m(\phi) \cap m(\psi)|}{|m(\phi) \cup m(\psi)|} \\ &= \frac{a}{a+b+c}. \end{aligned} \tag{6}$$

One may interpret the mutual support, $0 \leq MS(\phi, \psi) \leq 1$, as a measure of the strength of a pair of rules $\phi \Rightarrow \psi$ and $\psi \Rightarrow \phi$.

The *change of support* of ψ provided by ϕ is defined by:

$$\begin{aligned} CS(\phi \Rightarrow \psi) &= CS(\psi|\phi) = AS(\psi|\phi) - G(\psi) \\ &= \frac{an - (a+b)(a+c)}{(a+b)n}. \end{aligned} \tag{7}$$

Unlike the absolute support, the change of support varies from -1 to 1. One may consider $G(\psi)$ to be the prior probability of ψ and $AS(\psi|\phi)$ the posterior probability of ψ after knowing ϕ. The difference of posterior and prior probabilities represents the change of our confidence regarding whether ϕ actually related to ψ. For a positive value, one may say that ϕ positively related to ψ; for a negative value, one may say that ϕ is negatively related to ψ.

The generality $G(\psi)$ is related to the satisfiability of ψ by all objects in the database, and $AS(\phi \Rightarrow \psi)$ is related to the satisfiability of ψ in the subset $m(\phi)$. A high $AS(\phi \Rightarrow \psi)$ does not necessarily suggest a strong association between ϕ and ψ, as a concept ψ with a large $G(\psi)$ value tends to have a large $AS(\phi \Rightarrow \psi)$ value. The change of support $CS(\phi \Rightarrow \psi)$ may be more accurate.

4 Analysis of association rules, exception rules and peculiarity rules

To illustrate the usefulness of the proposed framework, we analyze some existing types of rules.

Association rules were first proposed for mining rules from transaction databases. A transaction database can be equivalently represented by a binary information table in which the set of items corresponds to the set of attributes, the set of transactions corresponds to the set of objects, and the domain of each attribute is limited to the set $\{0, 1\}$. An association rule takes a special kind of relationships in the sense that both the left and right hand sites are conjunction of atomic formulas of the form $(i, 1)$, where i stands for an item. One can easily extend association rules to non-transaction databases and allows more general expressions in the rules.

Two measures, called the support and the confidence, are used for mining association rules. They are indeed the generality and absolute support:

$$\begin{aligned} supp(\phi \Rightarrow \psi) &= G(\phi \wedge \psi), \\ conf(\phi \Rightarrow \psi) &= AS(\phi \Rightarrow \psi). \end{aligned} \tag{8}$$

By specifying threshold values of support and confidence, one can obtain all association rules whose support and confidence are above the thresholds. With an association rule, it is very tempting to relate a large confidence with a strong association between two concepts. However, such a connection may not exist. Suppose we have $conf(\phi \Rightarrow \psi) = 0.90$. If we also have $G(\psi) = 0.95$, we can conclude that ϕ is in fact negatively associated with ψ. This suggests that an association rule may not reflect the true association. Conversely, an association rule with low confidence may have a large change of support. In mining association rules, concepts with low support are not considered in the search for association. On the other hand, two concepts with low supports may have either large confidence or a large change of support. In summary, algorithms for mining association rules may fail to find such useful rules. Other mining algorithms are needed.

Exception rules have been studied as extension of association rules to resolve some of the above problems [7]. For an association rule, $\phi \Rightarrow \psi$, with high confidence, one may associates an exception rule $\phi \wedge \phi' \Rightarrow \neg\psi$. Roughly speaking, ϕ' can be viewed as the condition for exception to rule $\phi \Rightarrow \psi$. To be consistent with the intended interpretation of exception rule, it is reasonable to assume that $\phi \wedge \phi' \Rightarrow \neg\psi$ have a high confidence and low support. More specifically, we would expect a low generality of $\phi \wedge \phi'$. Otherwise, the rule cannot be viewed as describing exceptional situations. Consequently, exception rules cannot be discovered by association rule mining algorithms.

Zhong *et al.* [13] identified and studied a new class of rules called peculiarity rules. In mining peculiarity rules, the distribution of attribute values is taken into consideration. Attention is paid to objects whose attribute values are different from that of other objects. After the isolation of such peculiarity data, rules with low support and high confidence, and consequently a high change of support, are searched. Although a peculiarity rule may share the some properties with an exception rule, as expressed in terms of support, confidence, and change of support, it does not express exception to another rule. Semantically, they are very different.

We can qualitatively characterize association rules, exception rules and peculiarity rules in the following table:

Rule	G (supp)	AS (conf)	CS
Association rule:			
$\phi \Rightarrow \psi$	High	High	Unknown
Exception rule:			
$\phi \Rightarrow \psi$	High	High	Unknown
$\phi \wedge \phi' \Rightarrow \neg\psi$	Low	High	High
Peculiarity rule:			
$\phi \Rightarrow \psi$	Low	High	High

Both exception rule mining and peculiarity rule mining aim at finding rules missed by association rule mining. While exception rules and peculiarity rules have a high change of support, indicating a strong association between two concepts, association rules do not necessarily have this property. All three types of rules are focused on rules with high level of absolute support. For an exception rule, it is also expected that the generality of $\phi \wedge \phi'$ is low. For peculiarity, the generalities of both ϕ and ψ are expected to be low. In contrast, the generality of right hand of an exception rule does not have to be low.

One may say that rules with high absolute support and high change of support are of interest. The use of generality (support) in association rule mining is mainly for the sake of low computational cost, rather than semantics consideration. Exception rules and peculiarity rules are two subsets of rules with high absolute support and high change of support. It may be interesting to design algorithms to find *all* rules with high absolute support and/or high change of support. All three types of rules are not related to the mutual support measure MS. In some application, it may be necessary to consider rules $\phi \Rightarrow \psi$ and $\psi \Rightarrow \phi$ together.

5 Conclusion

We argue that it is important to study formal and mathematical modeling of data mining. For this purpose, a granular computing model is proposed and analyzed. The model is based on the examination of a concept as characterized by a pair of intension and extension. Information table is used to define precisely the intensions and extensions of concepts. Intensions are defined by formulas of a language and extensions are defined by subsets of objects in the universe. While rules are expressed in terms of the intensions of concepts, they are interpreted by the extensions of the concepts. Within the proposed model, some existing data mining methods are compared and analyzed.

References

[1] Bezdek, J.C. What is computational intelligence? in: *Computational Intelligence: Imitating Life*, Zurada, J.M., Marks II, R.J. and Robinson, C.J. (Eds.), IEEE Press, 1-12, 1994.

[2] Demri, S. and Orlowska, E. Logical analysis of indiscernibility, in: *Incomplete Information: Rough Set Analysis*, Physica-Verlag, Heidelberg, 347-380, 1998.

[3] Fayyad, U.M., Piatetsky-Shapiro, G. (eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996.

[4] Goodman, I.R., Nguyen, H.T. and Walker, E.A. *Conditional Inference and Logic for Intelligent Systems*, North-Holland, Amsterdam, 1991.

[5] Pawlak, Z. *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[6] Skowron, A. Toward intelligent systems: calculi of information granules, *Bulletin of International Rough Set Society*, **5**, 9-30, 2001.

[7] Suzuki, E. Autonomous discovery of reliable exception rules, *Proceedings of KDD'97*, 259-262, 1997.

[8] Wille, R. Concept lattices and conceptual knowledge systems, *Computers Mathematics with Applications*, **23**, 493-515, 1992.

[9] Yao, Y.Y. Granular computing: basic issues and possible solutions, *Proceedings of the 5th Joint Conference on Information Sciences*, 186-189, 2000.

[10] Yao, Y.Y. and Zhong, N. An analysis of quantitative measures associated with rules, *Proceedings of PAKDD'99*, 479-488, 1999.

[11] Yao, Y.Y. and Zhong, N. Potential applications of granular computing in knowledge discovery and data mining, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, 573-580, 1999.

[12] Zadeh, L.A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, **19**, 111-127, 1997.

[13] Zhong, N., Yao, Y.Y. and Ohsuga, S. Peculiarity oriented multi-database mining, *Proceedings of PKDD'99*, 136-146, 1999.