A GRANULAR COMPUTING APPROACH TO MACHINE LEARNING

J.T. Yao *Y.Y. Yao*

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {jtyao, yyao}@cs.uregina.ca

ABSTRACT

The key to granular computing is to make use of granules in problem solving. Classification is one of the well studied problems in machine learning and data mining as it involves of discovery knowledge from large databases. We presents a granular computing view to classification problems and propose a granular computing approach to classification in this paper. The ID3 [8] and PRISM [2] algorithma are studied and extended to granular computing algorithm.

1. INTRODUCTION

Knowledge discovering and data mining are frequently refereed in the literature as a process extracting interesting information or patterns from large databases. It is actually a technique or program doing automatic inductive reasoning. Learning, identification and searching for knowledge, patterns, and regularities from data are some of the major tasks of data mining. Knowledge is represented in rules or black-box systems such as neural networks. Extensive studies in the field have been focused on algorithms and methodologies for mining different types of rules [5], as well as speeding up of existing algorithms [4]. Although plenty of experimental and algorithmic studies have been reported in the literature, there is little attention paid to the formal, general and mathematical modeling of data mining [13].

Taken logicians' view point, data mining, especially rule mining, can be molded in two steps, i.e., formation of concepts and identification of relationship between concepts. There are two aspects of a concept, the intension and extension of the concept [3, 10]. The intension of a concept consists of all properties or attributes that are valid for all 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. The extension of a concept, being a subset of the universe, is also called a granule in granular computing.

Granular computing (GrC) is an umbrella term which covers any theories, methodologies, techniques, and tools that make use of granules (i.e., subsets of a universe) in problem solving [12, 16, 17]. A subset of the universe is called a granule in granular computing. Basic ingredients of granular computing are subsets, classes, and clusters of a universe. It deals with the characterization of a concept by a unit of thoughts consisting of two parts, the intension and extension of the concept.

One of the example of the granular computing data mining models combines results from formal concept analysis and granular computing [13]. Each granule is viewed as the extension of a certain concept and a description of the granule is an intension of the concept. This paper adapts a similar point of views and discusses two special granulation cases namely covering and partition in the process of data mining. With granular computing view, granulation oriented ID3 [8] can be extended to covering, and granule oriented PRISM [2] can be extended to multi-class.

The organization of this paper is as follows. We introduce a granular computing view of data mining, the granulation of covering and partition as well as information tables in the next section. Classification problems are formalized in Section 3 with the concepts of partition and covering. We discuss classification algorithms in Section 4. A section concludes this paper is followed.

2. A GRANULAR COMPUTING VIEW OF DATA MINING

Granulation of a universe involves dividing the universe into subsets or grouping individual objects into clusters. A granule is a subset of the universe. A family of granules that contains every object in the universe is called a granulation of the universe.

Partitions and coverings are two simple and commonly used granulations of universe [15]. A partition consists of disjoint subsets of the universe, and a covering consists of possibly overlap subsets. Partitions are a special type of coverings. In granular computing, we treat each element of a partition or covering as a granule. Each granule can also be further divided through partition or covering. For example, a universe $\{a, b, c, d\}$ can be divided into two granules $\{a,b\}$ and $\{c,d\}$ with partition $\{\{a,b\}, \{c,d\}\}$. It can also be divided into another two granules $\{a,b,c\}$ and $\{c,d,e\}$ with covering $\{\{a,b,c\}, \{c,d,e\}\}$.

Classification deals with grouping or clustering of objects based on certain criteria. It is one of the basic learning tasks and is related to concept formation and concept relationship identification. While concept formation involves the construction of classes and description of classes, concept relationship identification involves the connections between classes. These two related issues can be studied formally in a framework that combines formal concept analysis and granular computing [13].

There are two aspects of a concept, the intension and extension of the concept [3, 10]. The intension of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The intension of a concept is its meaning, or its complete definition. The extension of a concept is the set of objects or entities which are instances of the concept. The extension of a concept is the collection, or set, of things to which the concept applies. A concept is thus described jointly by its intension and extension, i.e., a set of properties and a set of objects. The intension of a concept can be expressed by a formula, or an expression, of a certain language, while the extension of a concept is presented as a set of objects satisfy the formula. This formulation enables us to study formal concepts in a logic setting in terms of intension and also in a set-theoretic setting in terms of extensions.

In order to formalize the problem, an information table was introduced in [13, 14]. An information table can be formulated as a tuple:

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

where U is a finite nonempty set of objects, At is a finite nonempty set of attributes, $\mathcal{L}$ is a language defined using attributes in At, V_a is a nonempty set of values for a $\in At$, and $I_a : U \rightarrow V_a$ is an information function. 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 [7, 16]. That is, objects are only perceived, observed, or measured by using a finite number of properties. The information function can be easily extended to a subset of attributes. Given any subset $A \subseteq At$, the value of an object x on A is denoted by $I_A(x)$. An information table as shown in Table 1 is adopted from Quinlan [8].

Table 1: An information table

Object	**height**	**hair**	**eyes**	**class**
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

In the language $\mathcal{L}$, an atomic formula is given by $a = v$, where $a \in At$ and $v \in V_a$. Formulas can be formed by logical operators negation, conjunction and disjunction. If ϕ and ψ are formulas, then so are $\neg\phi$, $\phi \wedge \psi$, and $\phi \vee \psi$. 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 defined by the following conditions:

(1) $\quad x \models a = v \text{ iff } I_a(x) = v,$

(2) $\quad x \models \neg\phi \text{ iff } \text{ not } x \models \phi,$

(3) $\quad x \models \phi \wedge \psi \text{ iff } x \models \phi \text{ and } x \models \psi,$

(4) $\quad x \models \phi \vee \psi \text{ iff } x \models \phi \text{ or } x \models \psi.$

If ϕ is a formula, the set $m_S(\phi) = \{x \in U \mid x \models \phi\}$, is called the meaning of the formula ϕ in S. If S is understood, we simply write $m(\phi)$. The meaning of a formula ϕ is 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.

We can have a formal description of concepts with the introduction of language $\mathcal{L}$. 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 $(\phi, m(\phi))$.

By using the language $\mathcal{L}$, we can construct various granules. For an atomic formula $a = v$, we obtain a granule $m(a = v)$. If $m(\phi)$ and $m(\psi)$ are granules corresponding to formulas ϕ and ψ, we obtain granules $m(\phi) \cap m(\psi) = m(\phi \wedge \psi)$ and $m(\phi) \cup m(\psi) = m(\phi \vee \psi)$. In an information table, we are only interested in granules, partitions and coverings that can be described by the language $\mathcal{L}$.

In particular, we are only interested in the following granules, i.e., definable granules, conjunctively definable granules [14]. A subset $X \subseteq U$ is called a definable granule in an information table S if there exists a formula ϕ such that $m(\phi) = X$. A subset $X \subseteq U$ is a conjunctively definable granule in an information table S if there exists a formula ϕ such that ϕ is a conjunction of atomic formulas and $m(\phi) = X$.

With these two special type of granules, we have two special granulations. A partition π is called a conjunctively definable partition if every equivalence class of π is a conjunctively definable granule. A covering τ is called a conjunctively definable covering if every granule of τ is a conjunctively definable granule. With granulation, one can obtain finer partition by further dividing an equivalence class of a partition. Similarly, one can obtain a finer covering by further decomposing a granule of a covering.

3. A TYPICAL MACHINE LEARNING PROBLEM

Classification is a typical machine learning problem. In supervised classification, each object is associated with a unique and predefined class label. Objects are divided into disjoint classes which form a partition of the universe. Suppose an information table is used to describe a set of objects. Without loss of generality, we assume that there is a unique attribute **class** taking class labels as its value. The set of attributes is expressed as $At = F \cup \{\mathbf{class}\}$, where F is the set of attributes used to describe the objects. The goal is to find classification rules of the form, $\phi \Longrightarrow \mathbf{class} = c_i$, where ϕ is a formula over F and c_i is a class label.

Let $\pi_{\mathbf{class}} \in \Pi(U)$ denote the partition induced by the attribute **class**. An information table with a set of attributes $At = F \cup \{\mathbf{class}\}$ is said to provide a consistent classification if all objects with the same description over F have the same class label, namely, if $I_F(x) = I_F(y)$, then $I_{\mathbf{class}}(x) = I_{\mathbf{class}}(y)$.

For a subset $A \subseteq At$, it defines a partition π_A of the universe. The consistent classification problem can be formally defined [14]. An information table with a set of attributes $At = F \cup \{\mathbf{class}\}$ is a consistent classification problem if and only if $\pi_F \preceq \pi_{\mathbf{class}}$. For the induction of classification rules, the partition π_F is not very interesting. In fact, one is interested in finding a subset of attributes from F that also produces the correct classification. It can be easily verified that a problem is a consistent classification problem if and only if there exists a conjunctively definable partition π such that $\pi \preceq \pi_{\mathbf{class}}$. Likewise, the problem is a consistent classification problem if and only if there exists a non-redundant conjunctively definable covering τ such that $\tau \preceq \pi_{\mathbf{class}}$. This leads to kinds of solutions to the classification problem.

Formally, a partition solution to a consistent classification problem is a conjunctively definable partition π such that $\pi \preceq \pi_{\mathbf{class}}$. A covering solution to a consistent classification problem is a conjunctively definable covering τ such that $\tau \preceq \pi_{\mathbf{class}}$.

Let X denote a granule in a partition or a covering of the universe, and let $des(X)$ denote its description using language $\mathcal{L}$.

If $X \subseteq m(\textbf{class} = c_i)$, we can construct a classification rule: $des(X) \Rightarrow \textbf{class} = c_i$. For a partition or a covering, we can construct a family of classification rules. The main difference between a partition solution and a covering solution is that an object is only classified by one rule in a partition based solution, while an object may be classified by more than one rule in a covering based solution.

Consider the consistent classification problem of Table 1, we have the partition by **class**, a conjunctively defined partition π, and a conjunctively non-redundant covering τ:

$$\begin{aligned} \pi_{\textbf{class}}: &\quad \{\{o_1, o_3, o_6\}, \{o_2, o_4, o_5, o_7, o_8\}\}, \\ \pi: &\quad \{\{o_1, o_6\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\}, \\ \tau: &\quad \{\{o_1, o_6\}, \{o_2, o_7, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\}. \end{aligned}$$

Clearly, $\pi \preceq \pi_{\textbf{class}}$ and $\tau \preceq \pi_{\textbf{class}}$. A set of classification rules of π is:

$$\begin{aligned} &(\mathrm{r1}) &\quad \textbf{hair} = \text{blond} \wedge \textbf{eyes} = \text{blue} \Longrightarrow \textbf{class} = +, \\ &(\mathrm{r2}) &\quad \textbf{hair} = \text{blond} \wedge \textbf{eyes} = \text{brown} \Longrightarrow \textbf{class} = -, \\ &(\mathrm{r3}) &\quad \textbf{hair} = \text{red} \Longrightarrow \textbf{class} = +, \\ &(\mathrm{r4}) &\quad \textbf{hair} = \text{dark} \Longrightarrow \textbf{class} = -. \end{aligned}$$

A set of classification rules of τ is:

$$\begin{aligned} &(\mathrm{r1}') &\quad \textbf{hair} = \text{red} \Longrightarrow \textbf{class} = +, \\ &(\mathrm{r2}') &\quad \textbf{eyes} = \text{blue} \wedge \textbf{hair} = \text{blond} \Longrightarrow \textbf{class} = +, \\ &(\mathrm{r3}') &\quad \textbf{eyes} = \text{brown} \Longrightarrow \textbf{class} = -. \\ &(\mathrm{r4}') &\quad \textbf{hair} = \text{dark} \Longrightarrow \textbf{class} = -. \end{aligned}$$

In fact, the first set of rules is obtained by the ID3 learning algorithm [8], and the second set by the PRISM algorithm [2]. Obviously, $(\mathrm{r1}')$, $(\mathrm{r2}')$, and $(\mathrm{r4}')$ are the same as (r3), (r1) and (r4) respectively. $(\mathrm{r3}')$ is a part of (r2).

4. CLASSIFICATION ALGORITHMS

With the concepts introduced so far, we can remodel some popular classification algorithms. We study ID3 and PRISM from granular computing view and propose a more general and flexible granulation algorithm.

4.1. ID3

The ID3 [8] algorithm probably is the most popular algorithm in data mining. It uses information gain as a criterion to find a suitable attribute to partition the universe until all granules can be understood or expressed by a formula. Much effort has been made to extend the ID3 algorithm in order to get a better classification result. The C4.5 [9] proposed by Quinlan himself and fuzzy decision tree [6] are among them. Figure 1 show the learning algorithm of ID3.

Figure 1: The learning algorithm of ID3

IF all cases in the training set belong to the same class

THEN Return the value of the class

ELSE

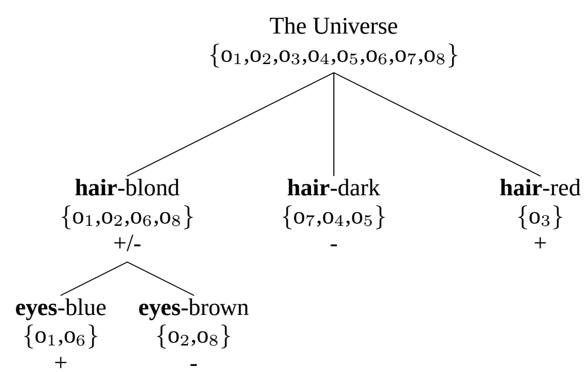
(1) Select an attribute **a** to split the universe.

(3) Divide the training set into non empty subsets, one for each value of attribute **a**.

(3) Return a tree with one branch for each subset, each branch having a descendant subtree or a class value produced by applying the algorithm recursively for each subset in turn.

Following the algorithm we start with the selection of the attribute **hair**. The first step of granulation is to partition the universe with values of **hair** as it is with the largest information gain. Since there are three values for **hair**, now we have three granules for this partition. Elements of (**hair**-dark) and (**hair**-red) granules happened to belong to the same class, we will not conduct any granulation to these two granules. As elements in granule (**hair**-blond) do not belong to same class, we granulate the new universe (**hair**-blond) with attribute **eyes**. We stop granulation when elements in the two new granules (**eyes**-blue) and (**eyes**-brown) are in the same class. The partition tree is shown in Figure 2 which happens to be our familiar decision tree.

Figure 2: An example of partition by ID3



4.2. The extension to ID3 type of algorithms

ID3 is granulation oriented search algorithm. It search a partition of a problem at one time. We can extend this algorithm to another granulation, covering, with modification to its algorithm. The top-down construction of a decision tree for classification searches for a partition solution to a classification problem. The induction process can be briefly described as follows. Based on a measure of connection between two partitions, one selects an attribute to divide the universe into a partition [8]. If an equivalence is not a subset of a user defined class, it is further divided by using another attribute. The process continues until one finds a decision tree that correctly classifies all objects. Each node of the decision tree is labelled by an attribute, and each branch is labelled by a value of the parent attribute.

When we search a covering solution, we can not immediately use a decision tree to represent the results. We modify the decision tree method and introduce the concept of granule network [15]. In a granule network, each node is labelled by a subset of objects. The arc leading from a larger granule to a smaller granule is labelled by an atomic formula. In addition, the smaller granule is obtained by selecting those objects of the larger granule that satisfy the atomic formula. The family of the smallest granules thus forms a conjunctively definable covering of the universe.

4.3. PRISM

PRISM [2] is an algorithm proposed by Jadia Cendrowska in 1987. Instead of using the principle of generating decision trees which can be converted to decision rules, PRISM generates rules from training set directly. Most important, PRISM is a covering based method. The algorithm is described in Figure 3,

Figure 3: The learning algorithm of PRISM

For i=1 to n

repeat until all instances of class i have been removed

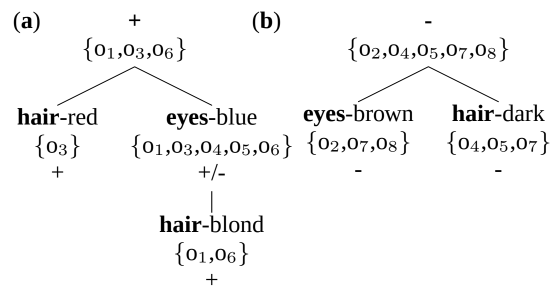
- **(1) Calculate** the probability of occurrence of class i for each attribute-value pair.
- **(2) Select** the attribute-value pair with maximum probability and create a subset of the training set comprising all instances with the selected combination.
- **(3) Repeat** (1) and (2) for this subset until it contains only instances of class i. The induced rule is then the conjunction of all the attribute-value pairs selected in creating this subset.
- **(4) Remove** all instances covered by this rule from training set.

From granular computing point of view, PRISM is actually finding covering to the universe (training set). Let's still use the example of Table 1. The starting point of PRISM is the current knowledge, classification + and -. For each class, a granule which covers the current universe is selected. For the given example, we know that the $(\mathbf{class} = +)$ is $\{o_1, o_3, o_6\}$, the $(\mathbf{class} = -)$ is $\{o_2, o_4, o_5, o_7, o_8\}$. The largest probability is $P(+|\mathbf{hair}{-}\mathrm{red})$. We use this attribute-value pair to form a granule $\{o_3\}$. The second largest probability is $P(+|\mathbf{eyes}{-}\mathrm{blue})$. We use this attribute-value pair to form a second granule $\{o_1, o_3, o_4, o_5, o_6\}$. So far these two granules cover $(\mathbf{class} = +)$. We do the same for $(\mathbf{class} = -)$ and find two granules $\{o_2, o_7, o_8\}$ and $\{o_4, o_5, o_7\}$ which cover $(\mathbf{class} = -)$. The current covering of universe has 4 granules. All granules are in the same class except for granule **eyes**-blue. We calculate the new set of probabilities for the second universe. The largest probability with high occurrence is **hair**-blond pair. We form a new granule $\{o_1, o_6\}$. It is in fact the intersection of **eyes**-blue and **hair**-blond. The covering is shown in Figure 4. Please note that o_3 is in granule (**hair**-red) and granule (**eyes**-blue). For this particular example, PRISM can provide short rules than ID3. This is consistent with Cendrowska's results and a recent review [1].

4.4. The extension to PRISM search algorithm

PRISM is granule oriented search algorithm. It deals with one class at one time. We can extend this algorithm to multi-class search approach which is addressed in another paper.

Figure 4: An example of covering by PRISM



4.5. Granular computing approach

With the extension of ID3 and PRISM, a granular computing approach to a particular machine learning problem is proposed. Atomic formulas define *basic* granules, which serve as the basis for the granule network. The pair $(a = v, m(a = v))$ is called a basic concept. Table 2 shows the basic granules for the example of Table 1. Each node in the granule network is a conjunction of some basic granules, and thus a conjunctively definable granule. The granule network for a classification problem can constructed by a top-down search of granules. Figure 5 outline an algorithm for the construction of a granule network. The results of the example by granular computing approach is shown in Figure 6.

Table 2: Basic granules for Table 1 example

Formula	Granule
$\mathbf{height} = \mathrm{short}$	$\{o_1, o_2, o_8\}$
$\mathbf{height} = \mathrm{tall}$	$\{o_3, o_4, o_5, o_6, o_7\}$
$\mathbf{hair} = \mathrm{blond}$	$\{o_1, o_2, o_6, o_8\}$
$\mathbf{hair} = \mathrm{red}$	$\{o_3\}$
$\mathbf{hair} = \mathrm{dark}$	$\{o_4, o_5, o_7\}$
$\mathbf{eyes} = \mathrm{blue}$	$\{o_1, o_3, o_4, o_5, o_6\}$
$\mathbf{eyes} = \mathrm{brown}$	$\{o_2, o_7, o_8\}$

The two importance issues of the algorithm is the evaluation of the fitness of each basic concept and the modification of existing partial granule network [13, 15]. The algorithm is basically a heuristic search algorithm. We can conduct depth first search or breadth first search. The results of depth first search and breadth first search are the same for ID3. PRISM is basically a depth first search algorithm.

5. CONCLUSION

We present a granular computing view of data mining in particular to classification problems. With the induce of partition and covering defined by a set of attribute values, one can find a solution by granulation. ID3 and PRISM types of algorithms are examples of partition and coving search algorithms. As suggested by No Free Lunch theorem [11], there is no algorithm which performs better than any other algorithm for all kinds of possible problems. It is useless to judge an algorithm irrespectively of the optimization

Figure 5: An Algorithm for constructing a granule network

(1) Construct the family of basic concept with respect to atomic formulas:

$$BC(U) = \{(a = v, m(a = v)) \mid a \in F, v \in V_a\}.$$

(2) Set the unused basic concepts to the set of basic concepts:

$$UBC(U) = BC(U).$$

(3) Set the granule network to $GN = (\{U\}, \emptyset)$, which is a graph consists of
only one node and no arc.

(4) While the set of smallest granules in GN is not a covering solution of
the classification problem **do** the following:

(4.1) Compute the fitness of each unused basic concept.

(4.2) Select the basic concept $C = (a = v, m(a = v))$ with maximum
value of fitness.

(4.3) Set $UBC(U) = UBC(U) - \{C\}$.

(4.4) Modify the granule network GN by adding new nodes which are
the intersection of $m(a = v)$ and the original nodes of GN; **connect**
the new nodes by arcs labelled by $a = v$.

problem. For some data set, ID3 type may be better than PRISM type and verse versa.

We provide a more general framework for consistent classification problem. The approach discussed in this paper provides more freedom of choice on heuristic and measures according to users needs. We will further examine on other data set and compare the results in the future.

6. REFERENCES

[1] Bramer, M.A.,"Automatic induction of classification rules from examples using N-PRISM", *Research and Development in Intelligent Systems XVI*, Springer-Verlag, pp.99-121, 2000.

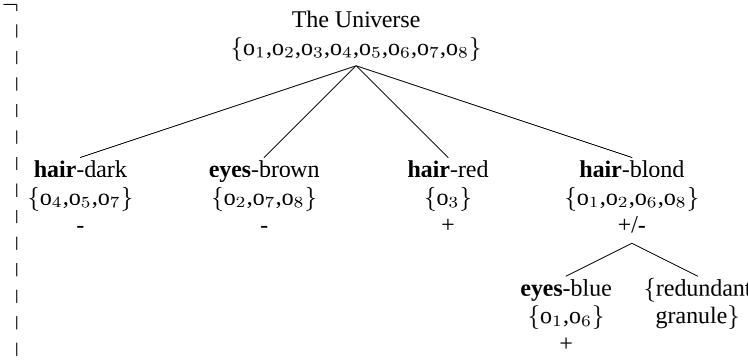
[2] Cendrowska, J., "PRISM: an algorithm for inducing modular rules", *International Journal of Man-Machine Studies*, **27**, 349-370, 1987.

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

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

[5] Halgamuge, S.K., Glesner, M., "FuNe deluxe: a group of fuzzy-neural methods for complex data analysis problems", *Proceedings of the European Congress on Fuzzy and Intelligent Technologies*, 1995.

Figure 6: An example of granular computing approach



[6] Janikow, C., "Fuzzy decision trees: issues and methods", *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, **28**(1), 1–14, 1998.

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

[8] Quinlan, J.R. "Learning efficient classification procedures and their application to chess end-games", in: *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, Michalski, J.S., Carbonell, J.G., and Mirchell, T.M. (Eds.), Morgan Kaufmann, Palo Alto, CA, pp.463-482, 1983.

[9] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

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

[11] Wolpert, D. H., and Macready, W. G, "No free lunch theorems for optimization" *IEEE Transactions on Evolutionary Computation*, **1**(1), 67-82, 1997.

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

[13] Yao, Y.Y. On modeling data mining with granular computing, *Proceedings of COMPSAC 2001*, pp.638-643, 2001.

[14] Yao, Y.Y. and Yao, J. T., "Granular computing as a basis for consistent classification problems", *Communications of Institute of Information and Computing Machinery (special issue of PAKDD'02 Workshop on Toward the Foundation of Data Mining)*, Vol. 5, No.2, pp.101-106, 2002.

[15] Yao, Y.Y. and Yao, J. T., "Induction of classification rules by granular computing",*Proceedings of The Third International Conference on Rough Sets and Current Trends in Computing*, 2002.

[16] 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*, pp.573-580, 1999.

[17] 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.