# Foundations of Classification

J. T. Yao  Y. Y. Yao and Y. Zhao

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

**Summary.** Classification is one of the main tasks in machine learning, data mining, and pattern recognition. A granular computing model is suggested for learning two basic issues of concept formation and concept relationship identification. A classification problem can be considered as a search for suitable granules organized under a partial order. The structures of search space, solutions to a consistent classification problem, and the structures of solution space are discussed. A classification rule induction method is proposed. Instead of searching for a suitable partition, we concentrate on the search for a suitable covering of the given universe. This method is more general than partition-based methods. For the design of covering granule selection heuristics, several measures on granules are suggested.

## 1 Introduction

Classification is one of the main tasks in machine learning, data mining, and pattern recognition [3, 10, 12]. It deals with classifying labelled objects. Knowledge for classification can be expressed in different forms, such as classification rules, discriminant functions, and decision trees. Extensive research has been done on the construction of classification models.

Mainstream research in classification focus on classification algorithms and their experimental evaluations. By comparison, less attention has been paid to the study of fundamental concepts such as structures of search space, solution to a consistent classification problem, as well as the structures of a solution space. For this reason, we present a granular computing based framework for a systematic study of these fundamental issues.

Granular computing is an umbrella term to cover any theories, methodologies, techniques, and tools that make use of granules in problem solving [25, 27, 33, 34]. 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. The granulation of a given universe involves dividing the universe into subsets or grouping individual objects into clusters. There are many fundamental issues in granular computing, such as the granulation of a given universe, the descriptions of granules, the relationships between granules, and the computation of granules.

Data mining, especially rule-based mining, can be molded in two steps, namely, the formation of concepts and the identification of relationship between concepts. Formal concept analysis may be considered as a concrete model of granular computing. It deals with the characterization of a concept by a unit of thoughts consisting the intension and the extension of the concept [4, 23]. From the standing point of granular computing, the concept of a granule may be exemplified by a set of instances, i.e., the extension; the concept of a granule may be described or labelled by a name, i.e., the intension. Once concepts are constructed and described, one can develop computational methods using granules [27]. In particular, one may study relationships between concepts in terms of their intensions and extensions, such as sub-concepts and super-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. By combining the results from formal concept analysis and granular computing, knowledge discovery and data mining, especially rule mining, can be viewed as a process of forming concepts and finding relationships between concepts in terms of intensions and extensions [28, 30, 32].

The organization of this chapter is as follows. In Section 2, we first present the fundamental concepts of granular computing which serve as the basis of classification problems. Measures associated with granules for classification will be studied in Section 3. In Section 4, we will examine the search spaces of classification rules. In Section 5, we remodel the ID3 and PRISM classification algorithms from the viewpoint of granular computing. We also propose the $k$LR algorithm and a granule network algorithm to complete the study of the methodology in granular computing model.

## 2 Fundamentals of a Granular Computing Model for Classification

This section provides an overview of the granular computing model [28, 30].

### 2.1 Information tables

Information tables are used in granular computing models. An information table provides a convenient way to describe a finite set of objects called a universe by a finite set of attributes [14, 33]. It represents all available information and knowledge. That is, objects are only perceived, observed, or measured by using a finite number of properties.

**Definition 1.** *An information table is the following 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 by using attributes in $At$,*
*$V_a$ is a nonempty set of values of $a \in At$,*
*$I_a : U \to V_a$ is an information function that maps an object of $U$ to exactly one possible value of attribute $a$ in $V_a$.*

We can easily extend the information function $I_a$ to an information function on a subset of attributes. For a subset $A \subseteq At$, the values of an object $x$ on $A$ is denoted by $I_A(x)$, where $I_A(x) = \bigwedge_{a \in A} I_a(x)$.

**Definition 2.** *In the language $\mathcal{L}$, an atomic formula is given by $a = v$, where $a \in At$ and $v \in V_a$. If $\phi$ and $\psi$ 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 the satisfiability of the formulas.

**Definition 3.** *Given the model as an information table $S$, the satisfiability of a formula $\phi$ 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:*

$$\begin{aligned}
&(1)\ \ x \models a = v \text{ iff } I_a(x) = v,\\
&(2)\ \ x \models \neg\phi \text{ iff } \text{ not } x \models \phi,\\
&(3)\ \ x \models \phi \wedge \psi \text{ iff } x \models \phi \text{ and } x \models \psi,\\
&(4)\ \ x \models \phi \vee \psi \text{ iff } x \models \phi \text{ or } x \models \psi.
\end{aligned}$$

**Definition 4.** *Given a formula $\pi$, 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 $\phi$ in $S$. If $S$ is understood, we simply write $m(\phi)$.*

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

### 2.2 Concept formulation

To formalize data mining, we have to analyze the concepts first. There are two aspects of a concept, the intension and the extension [4, 23]. The intension of a concept consists of all properties or attributes that are valid for all 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 a collection, or a 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 that satisfies the formula. 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.

With the introduction of language $\mathcal{L}$, we have a formal description of concepts. A concept which is definable in an information table is a pair of $(\phi, m(\phi))$, where $\phi \in \mathcal{L}$. More specifically, $\phi$ is a description of $m(\phi)$ in $S$, i.e. the intension of concept $(\phi, m(\phi))$, and $m(\phi)$ is the set of objects satisfying $\phi$, i.e. the extension of concept $(\phi, m(\phi))$. We say a formula has meaning if it has an associated subset of objects; we also say a subset of objects is definable if it is associated with at least one formula.

**Definition 5.** *A subset $X \subseteq U$ is called a definable granule in an information table $S$ if there exists at least one formula $\phi$ such that $m(\phi) = X$.*

By using the language $\mathcal{L}$, we can define 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 $\phi$ and $\psi$, we obtain granules $m(\phi) \cap m(\psi) = m(\phi \wedge \psi)$ and $m(\phi) \cup m(\psi) = m(\phi \vee \psi)$.

| 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 | - |

**Table 1.** An information table

To illustrate the idea developed so far, consider an information table given by Table 1, which is adopted from Quinlan [15]. The following expressions are some of the formulas of the language $\mathcal{L}$:

$$\begin{aligned}&\textbf{height} = \text{tall},\\&\textbf{hair} = \text{dark},\\&\textbf{height} = \text{tall} \wedge \textbf{hair} = \text{dark},\\&\textbf{height} = \text{tall} \vee \textbf{hair} = \text{dark}.\end{aligned}$$

The meanings of the above formulas are given by:

$$\begin{aligned}&m(\textbf{height} = \text{tall}) = \{o_3, o_4, o_5, o_6, o_7\},\\&m(\textbf{hair} = \text{dark}) = \{o_4, o_5, o_7\},\\&m(\textbf{height} = \text{tall} \wedge \textbf{hair} = \text{dark}) = \{o_4, o_5, o_7\},\\&m(\textbf{height} = \text{tall} \vee \textbf{hair} = \text{dark}) = \{o_3, o_4, o_5, o_6, o_7\}.\end{aligned}$$

By pairing intensions and extensions, we can obtain formal concepts, such as $(\textbf{height} = \text{tall}, \{o_3, o_4, o_5, o_6, o_7\})$, $(\textbf{hair} = \text{dark}, \{o_4, o_5, o_7\})$, $(\textbf{height} = \text{tall} \wedge \textbf{hair} = \text{dark}, \{o_4, o_5, o_7\})$ and $(\textbf{height} = \text{tall} \vee \textbf{hair} = \text{dark}, \{o_3, o_4, o_5, o_6, o_7\})$. The involved granules such as $\{o_3, o_4, o_5, o_6, o_7\}$, $\{o_4, o_5, o_7\}$ and $\{o_3, o_4, o_5, o_6, o_7\}$) are definable granules.

In the case where we can precisely describe a subset of objects $X$, the description may not be unique. That is, there may exist two formulas such that $m(\phi) = m(\psi) = X$. For example,

$$\begin{aligned}&\textbf{hair} = \mathit{dark},\\&\textbf{height} = \mathit{tall} \wedge \textbf{hair} = \mathit{dark},\end{aligned}$$

have the same meaning set $\{o_4, o_5, o_7\}$. Another two formulas

$$\begin{aligned}&\textbf{class} = +,\\&\textbf{hair} = \text{red} \vee (\textbf{hair} = \text{blond} \wedge \textbf{eyes} = \text{blue}),\end{aligned}$$

have the same meaning set $\{o_1, o_3, o_6\}$.

In many classification algorithms, one is only interested in formulas of a certain form. Suppose we restrict the connectives of language $\mathcal{L}$ to only the conjunction connective $\wedge$. Each formula is a conjunction of atomic formulas and such a formula is referred to as a conjunctor.

**Definition 6.** *A subset $X \subseteq U$ is a conjunctively definable granule in an information table $S$ if there exists a conjunctor $\phi$ such that $m(\phi) = X$.*

The notion of definability of subsets in an information table is essential to data analysis. In fact, definable granules are the basic logic units that can be described and discussed, upon which other notions can be developed.

### 2.3 Granulations as partitions and coverings

Partitions and coverings are two simple and commonly used granulations of the universe.

**Definition 7.** *A partition of a finite universe $U$ is a collection of non-empty, and pairwise disjoint subsets of $U$ whose union is $U$. Each subset in a partition is also called a block or an equivalence class.*

When $U$ is a finite set, a partition $\pi = \{X_i \mid 1 \leq i \leq m\}$ of $U$ consists of a finite number $m$ of blocks. In this case, the conditions for a partition can be simply stated by:

$$\begin{aligned}
&\text{(i).} && \text{for all } i,\ X_i \neq \emptyset,\\
&\text{(ii).} && \text{for all } i \neq j,\ X_i \cap X_j = \emptyset,\\
&\text{(iii).} && \bigcup\{X_i \mid 1 \leq i \leq m\} = U.
\end{aligned}$$

There is a one-to-one correspondence between the partitions of $U$ and the equivalence relations (i.e., reflexive, symmetric, and transitive relations) on $U$. Each equivalence class of the equivalence relation is a block of the corresponding partition. In this paper, we use partitions and equivalence relations, and blocks and equivalence classes interchangeably.

**Definition 8.** *A covering of a finite universe $U$ is a collection of non-empty subsets of $U$ whose union is $U$. The subsets in a covering are called covering granules.*

When $U$ is a finite set, a covering $\tau = \{X_i \mid 1 \leq i \leq m\}$ of $U$ consists of a finite number $m$ of covering granules. In this case, the conditions for a covering can be simply stated by:

$$\begin{aligned}
&\text{(i).} && \text{for all } i, X_i \neq \emptyset,\\
&\text{(ii).} && \bigcup\{X_i \mid 1 \leq i \leq m\} = U.
\end{aligned}$$

According to the definition, a partition consists of disjoint subsets of the universe, and a covering consists of possibly overlapping subsets. Partitions are a special case of coverings.

**Definition 9.** *A covering $\tau$ of $U$ is said to be a non-redundant covering if the collection of subsets derived by deleting any one of the granules from $\tau$ is not a covering.*

One can obtain a finer partition by further dividing the equivalence classes of a partition. Similarly, one can obtain a finer covering by further decomposing the granules of the covering.

**Definition 10.** *A partition $\pi_1$ is a refinement of another partition $\pi_2$, or equivalently, $\pi_2$ is a coarsening of $\pi_1$, denoted by $\pi_1 \preceq \pi_2$, if every block of $\pi_1$ is contained in some block of $\pi_2$. A covering $\tau_1$ is a refinement of another covering $\tau_2$, or equivalently, $\tau_2$ is a coarsening of $\tau_1$, denoted by $\tau_1 \preceq \tau_2$, if every granule of $\tau_1$ is contained in some granule of $\tau_2$.*

The refinement relation is a partial ordering of the set of all partitions, namely, it is reflexive, antisymmetric and transitive. This naturally defines a refinement order on the set of all partitions, and thus form a partition lattice, denoted as $\Pi(U)$. Likewise, a refinement order on the set of all covering forms a covering lattice, denoted as $\mathcal{T}(U)$.

Based on the refinement relation, we can construct multi-level granulations of the universe [29]. Given two partitions $\pi_1$ and $\pi_2$, their meet, $\pi_1 \wedge \pi_2$, is the finest partition of $\pi_1$ and $\pi_2$, their join, $\pi_1 \vee \pi_2$, is the coarsest partition of $\pi_1$ and $\pi_2$. The equivalence classes of a meet are all nonempty intersections of an equivalence class from $\pi_1$ and an equivalence class from $\pi_2$. The equivalence classes of a join are all nonempty unions of an equivalence class from $\pi_1$ and an equivalence class from $\pi_2$.

Since a partition is a covering, we use the same symbol to denote the refinement relation on partitions and refinement relation on covering. For a covering $\tau$ and a partition $\pi$, if $\tau \preceq \pi$, we say that $\tau$ is a refinement of $\pi$, which indicates that every granule of $\tau$ is contained in some granule of $\pi$.

**Definition 11.** *A partition is called a definable partition ($\pi_D$) in an information table $S$ if every equivalence class is a definable granule. A covering is called a definable covering ($\tau_D$) in an information table $S$ if every covering granule is a definable granule.*

For example, in information Table 1 $\{\{o_1, o_2, o_6, o_8\}, \{o_3, o_4, o_5, o_7\}\}$ is a definable partition/covering, since the granule $\{o_1, o_2, o_6, o_8\}$ can be defined by the formula **hair**=blond, and the granule $\{o_3, o_4, o_5, o_7\}$ can be defined by the formula $\neg$**hair**=blond. We can also justify that another partition $\{\{o_1, o_2, o_3, o_4\}, \{o_5, o_6, o_7, o_8\}\}$ is not a definable partition.

If partitions $\pi_1$ and $\pi_2$ are definable, $\pi_1 \wedge \pi_2$ and $\pi_1 \vee \pi_2$ are definable partitions. The family of all definable partitions forms a partition lattice $\Pi_D(U)$, which is a sub-lattice of $\Pi(U)$. Likewise, if two coverings $\tau_1$ and $\tau_2$ are definable, $\tau_1 \wedge \tau_2$ and $\tau_1 \vee \tau_2$ are definable coverings. The family of all definable coverings forms a covering lattice $\mathcal{T}_D(U)$, which is a sub-lattice of $\mathcal{T}(U)$.

**Definition 12.** *A partition is called a conjunctively definable partition ($\pi_{CD}$) if every equivalence class is a conjunctively definable granule. A covering is called a conjunctively definable covering ($\tau_{CD}$) if every covering granule is a conjunctively definable granule.*

For example, in Table 1 $\{\{o_1, o_2, o_8\}, \{o_3, o_4, o_5, o_6\}, \{o_7\}\}$ is a conjunctively definable partition or covering since the granule $\{o_1, o_2, o_8\}$ can be defined by the conjunctor **height**=short, the granule $\{o_3, o_4, o_5, o_6\}$ can be defined by the conjunctor **height**=tall$\wedge$**eyes**=blue, and the granule $\{o_7\}$ can be defined by the conjunctor **hair**=dark$\wedge$**eyes**=brown. Note, the join of these three formulas cannot form a tree structure.

The family of conjunctively definable partitions forms a definable partition lattice $\Pi_{CD}(U)$, which is a sub-lattice of $\Pi_D(U)$. The family of conjunctively definable coverings forms a definable covering lattice $\mathcal{T}_{CD}(U)$, which is a sub-lattice of $\mathcal{T}_D(U)$.

**Definition 13.** *A partition is called a tree definable partition ($\pi_{AD}$) if every equivalence class is a conjunctively definable granule, and all the equivalence classes form a tree structure.*

A partition $\Pi_{AD}(U)$ is defined by $At$, or a subset of $At$. For a subset $A$ of attributes, we can define an equivalence relation $E_A$ as follows:

$$\begin{aligned} xE_Ay &\iff \bigwedge a \in A, I_a(x) = I_a(y) \\ &\iff I_A(x) = I_A(y). \end{aligned} \tag{2}$$

For the empty set, we obtain the coarsest partition $\{U\}$. For a nonempty subset of attributes, the induced partition is conjunctively definable. The family of partitions defined by all subsets of attributes forms a definable partition lattice $\Pi_{\mathrm{AD}}(U)$, which is a sub-lattice of $\Pi_{\mathrm{CD}}(U)$.

For the information in Table 1, we obtain the following partitions with respect to subsets of the attributes:

$$\begin{aligned}
\pi_{\emptyset} &= \{U\}, \\
\pi_{\mathbf{height}} &= \{\{o_1, o_2, o_8\}, \{o_3, o_4, o_5, o_6, o_7\}\}, \\
\pi_{\mathbf{hair}} &= \{\{o_1, o_2, o_6, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\}, \\
\pi_{\mathbf{eyes}} &= \{\{o_1, o_3, o_4, o_5, o_6\}, \{o_2, o_7, o_8\}\}, \\
\pi_{\mathbf{height} \wedge \mathbf{hair}} &= \{\{o_1, o_2, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}, \{o_6\}\}, \\
\pi_{\mathbf{height} \wedge \mathbf{eyes}} &= \{\{o_1\}, \{o_2, o_8\}, \{o_3, o_4, o_5, o_6\}, \{o_7\}\}, \\
\pi_{\mathbf{hair} \wedge \mathbf{eyes}} &= \{\{o_1, o_6\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_7\}\}, \\
\pi_{\mathbf{height} \wedge \mathbf{hair} \wedge \mathbf{eyes}} &= \{\{o_1\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_6\}, \{o_7\}\}.
\end{aligned}$$

Since each subset defines a different partition, the partition lattice has the same structure as the lattice defined by the power set of the three attributes **height**, **hair**, and **eyes**.

All the notions developed in this section can be defined relative to a particular subset $A \subseteq At$ of attributes. A subset $X \subseteq U$ is called a definable granule with respect to a subset of attributes $A \subseteq At$ if there exists at least one formula $\phi$ over $A$ such that $m(\phi) = X$. A partition $\pi$ is called a definable partition with respect to a subset of attributes $A$ if every equivalence class is a definable granule with respect to $A$. Let $\Pi_{D(A)}(U)$, $\Pi_{CD(A)}(U)$, and $\Pi_{AD(A)}(U)$ denote the partition (semi-) lattices with respect to a subset of attributes $A \subseteq At$, respectively. We have the following connection between partition (semi-) partition lattices and they provide a formal framework of classification problems.

$$\begin{aligned}
\Pi_{AD}(U) &\subseteq \Pi_{CD}(U) \subseteq \Pi_{D}(U) \subseteq \Pi(U),\\
\Pi_{AD(A)}(U) &\subseteq \Pi_{CD(A)}(U) \subseteq \Pi_{D(A)}(U) \subseteq \Pi(U),\\
\mathcal{T}_{CD}(U) &\subseteq \mathcal{T}_{D}(U) \subseteq \mathcal{T}(U),\\
\mathcal{T}_{CD(A)}(U) &\subseteq \mathcal{T}_{D(A)}(U) \subseteq \mathcal{T}(U).
\end{aligned}$$

## 3 Measures associated with granules

We introduce and review three types of quantitative measures associated with granules, measures of a single granule, measures of relationships between a pair of granules [28, 32], and measures of relationships between a granule and a family of granules, as well as a pair of family of granules.

The only measure of a single granule $m(\phi)$ of a formula $\phi$ is the *generality*, defined as:

$$G(\phi) = \frac{|m(\phi)|}{|U|}, \tag{3}$$

which indicates the relative size of the granule $m(\phi)$. A concept defined by the formula $\phi$ is more general if it covers more instances of the universe. The quantity may be viewed as the probability of a randomly selected object satisfying $\phi$.

Given two formulas $\phi$ and $\psi$, we introduce a symbol $\Rightarrow$ to connect $\phi$ and $\psi$ in the form of $\phi \Rightarrow \psi$. It may be intuitively interpreted as a rule which enables us to infer information about $\psi$ from $\phi$. The strength of $\phi \Rightarrow \psi$ can be quantified by two related measures [20, 28].

The *confidence* or *absolute support* of $\psi$ provided by $\phi$ is the quantity:

$$AS(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|m(\phi)|} = \frac{|m(\phi) \cap m(\psi)|}{|m(\phi)|}. \tag{4}$$

It may be viewed as the conditional probability of a randomly selected object satisfying $\psi$ given that the object satisfies $\phi$. In set-theoretic terms, it is the degree to which $m(\phi)$ is included in $m(\psi)$. Thus, $AS$ is a measure of the correctness or the precision of the inference. A rule with the maximum absolute support 1 is a certain rule. The *coverage* $\psi$ provided by $\phi$ is the quantity:

$$CV(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|m(\psi)|} = \frac{|m(\phi) \cap m(\psi)|}{|m(\psi)|}. \tag{5}$$

It may be viewed as the conditional probability of a randomly selected object satisfying $\phi$ given that the object satisfies $\psi$. Thus, $CV$ is a measure of the applicability or recall of the inference. Obviously, we can infer more information about $\psi$ from $\phi$ if we have both a high absolute support and a high coverage. In general, there is a trade-off between support and coverage.

Consider now a family of formulas $\Psi = \{\psi_1, \ldots, \psi_n\}$ which induces a partition $\pi(\Psi) = \{m(\psi_1), \ldots, m(\psi_n)\}$ of the universe. Let $\phi \Rightarrow \Psi$ denote the inference relation between $\phi$ and $\Psi$. In this case, we obtain the following probability distribution in terms of $\phi \Rightarrow \psi_i$'s:

$$P(\Psi \mid \phi) = \left( P(\psi_1 \mid \phi) = \frac{|m(\phi) \cap m(\psi_1)|}{|m(\phi)|}, \ldots, P(\psi_n \mid \phi) = \frac{|m(\phi) \cap m(\psi_n)|}{|m(\phi)|} \right).$$

The conditional entropy $H(\Psi \mid \phi)$ defined by:

$$H(\Psi \mid \phi) = -\sum_{i=1}^{n} P(\psi_i \mid \phi) \log P(\psi_i \mid \phi), \tag{6}$$

provides a measure that is inversely related to the strength of the inference $\phi \Rightarrow \Psi$. If $P(\psi_{i_0} \mid \phi) = 1$ for one formula $\psi_{i_0}$ and $P(\psi_i \mid \phi) = 0$ for all $i \neq i_0$, the entropy reaches the minimum value 0. In this case, if an object satisfies $\phi$, one can identify one equivalence class of $\pi(\Psi)$ to which the object belongs without uncertainty. When $P(\psi_1 \mid \phi) = \ldots = (\psi_n \mid \phi) = 1/n$, the entropy reaches the maximum value $\log n$. In this case, we are in a state of total uncertainty. Knowing that an object satisfies the formula $\phi$ does not help in identifying an equivalence class of $\pi(\Psi)$ to which the object belongs.

Suppose another family of formulas $\Phi = \{\phi_1, \ldots, \phi_m\}$ define a partition $\pi(\Phi) = \{m(\phi_1), \ldots, m(\phi_m)\}$. The same symbol $\Rightarrow$ is also used to connect two families of formulas that define two partitions of the universe, namely, $\Phi \Rightarrow \Psi$. The strength of this connection can be measured by the conditional entropy:

$$\begin{aligned} H(\Psi \mid \Phi) &= \sum_{j=1}^{m} P(\phi_j) H(\Psi \mid \phi_j) \\ &= -\sum_{j=1}^{m} \sum_{i=1}^{n} P(\psi_i \wedge \phi_j) \log P(\psi_i \mid \phi_j), \end{aligned} \tag{7}$$

where $P(\phi_j) = G(\phi_j)$. In fact, this is the most commonly used measure for selecting attributes in the construction of decision tree for classification [15].

The measures discussed so far quantified two levels of relationships, i.e., granule level and granulation level. As we will show in the following section, by focusing on different levels, one may obtain different methods for the induction of classification rules.

## 4 Induction of Classification Rules by Searching Granules

For classification tasks, it is assumed that each object is associated with a unique class label. Objects can be divided into classes which form a granulation of the universe. We further assume that information about objects are given by an information table as defined in Section 2. 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 $At = C \cup \{\mathbf{class}\}$, where $C$ 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 $\phi$ is a formula over $C$ 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 $At = C \cup \{\mathbf{class}\}$ is said to provide a consistent classification if all objects with the same description over $C$ have the same class label, namely, if $I_C(x) = I_C(y)$, then $I_{\mathbf{class}}(x) = I_{\mathbf{class}}(y)$. Using the concept of a partition lattice, we define the consistent classification problem as follows.

**Definition 14.** *An information table with a set of attributes $At = C \cup \{\mathbf{class}\}$ is a consistent classification problem if and only if there exists a partition $\pi \in \Pi_{D(C)}(U)$ such that $\pi \preceq \pi_{\mathbf{class}}$, or a covering $\tau \in \mathcal{T}_{D(C)}(U)$ such that $\tau \preceq \pi_{\mathbf{class}}$.*

It can be easily verified that a consistent classification problem can be considered as a search for a definable partition $\pi \in \Pi_D(U)$, or more generally, a conjunctively definable partition $\pi \in \Pi_{CD}(U)$, or a tree definable partition $\pi \in \Pi_{AD}(U)$ such that $\pi \preceq \pi_{\mathbf{class}}$. For the induction of classification rules, the partition $\pi_{AD(C)}(U)$ is not very interesting. In fact, one is more interested in finding a subset $A \subset C$ of attributes such that $\pi_{AD(A)}(U)$ that also produces the correct classification. Similarly, a consistent classification problem can also be considered as a search for a conjunctively definable covering $\tau$ such that $\tau \preceq \pi_{\mathbf{class}}$. This leads to different kinds of solutions to the classification problem.

**Definition 15.** *A partition solution to a consistent classification problem is a conjunctively definable partition $\pi$ such that $\pi \preceq \pi_{\mathbf{class}}$. A covering solution to a consistent classification problem is a conjunctively definable covering $\tau$ such that $\tau \preceq \pi_{\mathbf{class}}$.*

Let $X$ denote a block in a partition or a covering granule of the universe, and let $des(X)$ denote its description using language $\mathcal{L}$. If $X \subseteq m(\mathbf{class} = c_i)$, we can construct a classification rule: $des(X) \Rightarrow \mathbf{class} = c_i$. 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 $\pi$, and a conjunctively defined covering $\tau$:

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

Clearly, $\pi \preceq \pi_{\textbf{class}}$ and $\tau \preceq \pi_{\textbf{class}}$. So $\pi \preceq \pi_{\textbf{class}}$ and $\tau \preceq \pi_{\textbf{class}}$ are solutions of consistent classification problem of Table 1. A set of classification rules of $\pi$ is:

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

A set of classification rules of $\tau$ consists of (r1), (r3), (r4) and part of (r2):

$$(\text{r2}') \qquad \textbf{eyes} = \text{brown} \Longrightarrow \textbf{class} = -.$$

The first set of rules is in fact obtained by the ID3 learning algorithm [16], and the second set is by the PRISM algorithm [2]. In comparison, rule (r2′) is shorter than (r2). Object $o_7$ is classified by (r4) in the partition solution, while it is classified by two rules (r2′) and (r4) in the covering solution.

The left hand side of a rule is a formula whose meaning is a block of the solution. For example, for the first rule, we have $m(\textbf{hair} = \text{blond} \wedge \textbf{eyes} = \text{blue}) = \{o_1, o_6\}$.

We can re-express many fundamental notions of classification in terms of partitions.

Depending on the particular lattice used, one can easily establish properties of the family of solutions. Let $\Pi_\alpha(U)$, where $\alpha = \text{AD}(C), \text{CD}(C), \text{D}(C)$, denote a (semi-) lattice of definable partitions. Let $\Pi^S_\alpha(U)$ be the corresponding set of all solution partitions. We have:

(i) For $\alpha = \text{AD}(C), \text{CD}(C), \text{D}(C)$, if $\pi' \in \Pi_\alpha(U)$, $\pi \in \Pi^S_\alpha(U)$ and $\pi' \preceq \pi$, then $\pi' \in \Pi^S_\alpha(U)$;
(ii) For $\alpha = \text{AD}(C), \text{CD}(C), \text{D}(C)$, if $\pi', \pi \in \Pi^S_\alpha(U)$, then $\pi' \wedge \pi \in \Pi^S_\alpha(U)$;
(iii)For $\alpha = \text{D}(C)$, if $\pi', \pi \in \Pi^S_\alpha(U)$, then $\pi' \vee \pi \in \Pi^S_\alpha(U)$;

It follows that the set of all solution partitions forms a definable lattice, a conjunctively definable lattice, or a tree definable lattice.

Mining classification rules can be formulated as a search for a partition from a partition lattice. A definable lattice provides the search space of potential solutions, and the partial order of the lattice provides the search direction. The standard search methods, such as depth-first search, breadth-first search, bounded depth-first search, and heuristic search, can be used to find a solution to the consistent classification problem. Depending on the required properties of rules, one may use different definable lattices that are introduced earlier. For example, by searching the conjunctively definable partition lattice $\Pi_{\mathrm{CD}(C)}(U)$, we can obtain classification rules whose left hand sides are only conjunctions of atomic formulas. By searching the lattice $\Pi_{\mathrm{AD}(C)}(U)$, one can obtain a similar solution that can form a tree structure. The well-known ID3 learning algorithm in fact searches $\Pi_{AD(C)}(U)$ for classification rules [15].

**Definition 16.** *For two solutions $\pi_1, \pi_2 \in \Pi_\alpha$ of a consistent classification problem, namely, $\pi_1 \preceq \pi_{\mathbf{class}}$ and $\pi_2 \preceq \pi_{\mathbf{class}}$, if $\pi_1 \preceq \pi_2$, we say that $\pi_1$ is a more specific solution than $\pi_2$, or equivalently, $\pi_2$ is a more general solution than $\pi_1$.*

**Definition 17.** *A solution $\pi \in \Pi_\alpha$ of a consistent classification problem is called the most general solution if there does not exist another solution $\pi' \in \Pi_\alpha$, $\pi \neq \pi'$, such that $\pi \preceq \pi' \preceq \pi_{\mathbf{class}}$.*

For a consistent classification problem, the partition defined by all attributes in $C$ is the finest partition in $\Pi_\alpha$. Thus, the most general solution always exists. However, the most general solution may not be unique. There may exist more than one the most general solutions.

In the information Table 1, consider three partitions from the lattice $\Pi_{\mathrm{CD}(C)}(U)$:

$$\begin{aligned}
\pi_1 &: \{\{o_1\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_6\}, \{o_7\}\}, \\
\pi_2 &: \{\{o_1, o_6\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\}, \\
\pi_3 &: \{\{o_1, o_6\}, \{o_2, o_7, o_8\}, \{o_3\}, \{o_4, o_5\}\}.
\end{aligned}$$

We have $\pi_1 \preceq \pi_2 \preceq \pi_{\mathbf{class}}$ and $\pi_1 \preceq \pi_3 \preceq \pi_{\mathbf{class}}$. Thus, $\pi_1$ is the more specific solution than both $\pi_2$ and $\pi_3$. In fact, both $\pi_2$ and $\pi_3$ are the most general solutions.

The roles of attributes are well-studied in the theory of rough sets [14], and can be re-expressed as follows:

**Definition 18.** *An attribute $a \in C$ is called a core attribute if $\pi_{C-\{a\}}$ is not a solution to the consistent classification problem.*

**Definition 19.** *An attribute $a \in C$ is called a superfluous attribute if $\pi_{C-\{a\}}$ is a solution to the consistent classification problem, namely, $\pi_{C-\{a\}} \preceq \pi_{\mathbf{class}}$.*

**Definition 20.** *A subset $A \subseteq C$ is called a reduct if $\pi_A$ is a solution to the consistent classification problem and $\pi_B$ is not a solution for any proper subset $B \subset A$.*

For a given consistent classification problem, there may exist more than one reduct.

In the information Table 1, attributes **hair** and **eyes** are core attributes. Attribute **height** is a superfluous attribute. The only reduct is the set of attributes $\{\mathbf{hair}, \mathbf{eyes}\}$.

## 5 The Studies on Classification Algorithms

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

### 5.1 ID3

The ID3 [15] algorithm is probably the most popular algorithm in data mining. Many efforts have been made to extend the ID3 algorithm in order to get a better classification result. The C4.5 algorithm [17] is proposed by Quinlan himself and generates fuzzy decision tree [8].

The ID3-like learning algorithms can be formulated as a heuristic search of the semi- conjunctively definable lattice $\Pi_{\mathrm{AD}(C)}(U)$. The heuristic used for the ID3-like algorithms is based on an information-theoretic measure of dependency between the partition defined by **class** and another conjunctively definable partition with respect to the set of attributes $C$. Roughly speaking, the measure quantifies the degree to which a partition $\pi \in \Pi_{\mathrm{AD}(C)}(U)$ satisfies the condition $\pi \preceq \pi_{\mathbf{class}}$ of a solution partition.

Specifically, the direction of ID3 search is from the coarsest partitions of $\Pi_{\mathrm{AD}(C)}(U)$ to more refined partitions. The largest partitions in $\Pi_{\mathrm{AD}(C)}(U)$ are the partitions defined by single attributes in $C$. Using the information-theoretic measure, ID3 first selects a partition defined by a single attribute. If an equivalence class in the partition is not a conjunctively definable granule with respect to **class**, the equivalence class is further divided into smaller granules by using an additional attribute. The same information-theoretic measure is used for the selection of the new attribute. The smaller granules are conjunctively definable granules with respect to $C$. The search process continues until a partition $\pi \in \Pi_{\mathrm{AD}(C)}(U)$ is obtained such that $\pi \preceq \pi_{\mathbf{class}}$.

Figure 1 shows the learning algorithm of ID3.

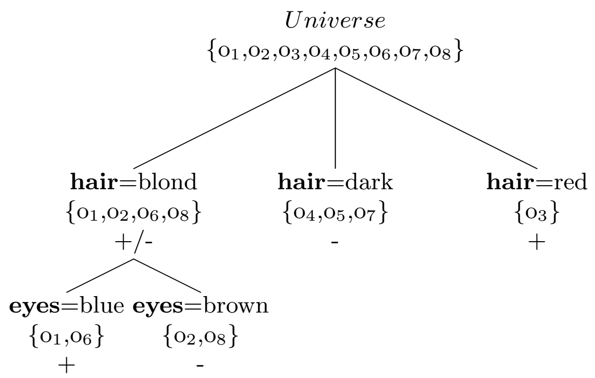
**Fig. 1.** The learning algorithm of ID3

```
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, which is with the
       maximum information gain.
   (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**, we obtain three granules for this partition. Elements of (**hair**=dark) and (**hair**=red) granules belong to the same class, we do not need to further decompose 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 is the familiar ID3 decision tree.

ID3 is a granulation oriented search algorithm. It searches a partition of a problem at one time. The top-down construction of a decision tree for classification searches for a partition solution to a consistent 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 [15]. If an equivalence class 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.

**Fig. 2.** An example partition generated by ID3




### 5.2 *k*LR

Algorithms for finding a reduct in the theory of rough sets can also be viewed as heuristic search of the partition lattice $\Pi_{\mathrm{AD}(C)}(U)$. Two directions of search can be carried, either from coarsening partitions to refinement partitions or from refinement partitions to coarsening partitions.

The smallest partition in $\Pi_{\mathrm{AD}(C)}(U)$ is $\pi_C$. By dropping an attribute $a$ from $C$, one obtains a coarsening partition $\pi_{C-\{a\}}$. Typically, a certain fitness measure is used for the selection of the attribute. The process continues until no further attributes can be dropped. That is, we find a subset $A \subseteq C$ such that $\pi_A \preceq \pi_{\mathbf{class}}$ and $\neg(\pi_B \preceq \pi_{\mathbf{class}})$ for all proper subsets $B \subset A$. The resulting set of attributes $A$ is a reduct.

**Fig. 3.** The learning algorithm of $k$LR

```
Let k = 0.
The k-level, k > 0, of the classification tree is built based on the
   (k-1)^th level described as follows:
if there is a node in (k - 1)^th level that does not consist of only
   elements of the same class then
   (1) Choose an attribute based on a certain criterion γ : At ⟶ ℜ;
   (2) Divide all the inconsistent nodes based on the selected
       attribute and produce the k^th level nodes, which are subsets
       of the inconsistent nodes;
   (3) Label the inconsistent nodes by the attribute name, and label
       the branches coming out from the inconsistent nodes by the
       values of the attribute.
```

The largest partition in $\Pi_{\mathrm{AD}(C)}(U)$ is $\pi_{\emptyset}$. By adding an attribute $a$, one obtains a refined partition $\pi_a$. The process continues until we have a partition satisfying the condition $\pi_A \preceq \pi_{\mathbf{class}}$. The resulting set of attributes $A$ is a reduct.

The $k$LR algorithm [31] is proposed as one of the rough set-type search algorithms to find a reduct, which is a set of individually necessary and jointly sufficient attributes that correctly classify the objects. $k$LR is described in Figure 3.

Comparing the $k$LR algorithm with ID3-like algorithms, we note that an important feature of ID3-like algorithms is that when partitioning an inconsistent granule, a formula is chosen based on only information about the current granule. The criteria used by ID3-like methods is based on local optimization. In the decision tree, different granules at the same level may use different formulas, and moreover the same attribute may be used at different levels. The use of local optimal criteria makes it difficult to judge the overall quality of the partial decision tree during its construction process. The $k$LR algorithm may solve this difficulty by partitioning the inconsistent granules at the same level at the same time using the same formula. One can construct a $k$LR decision tree and evaluate its quality level by level. Normally, a $k$LR decision tree is different from the corresponding ID3 tree. However, the running example in Table 1 is too small to show the different resulting trees generated by ID3 and $k$LR.

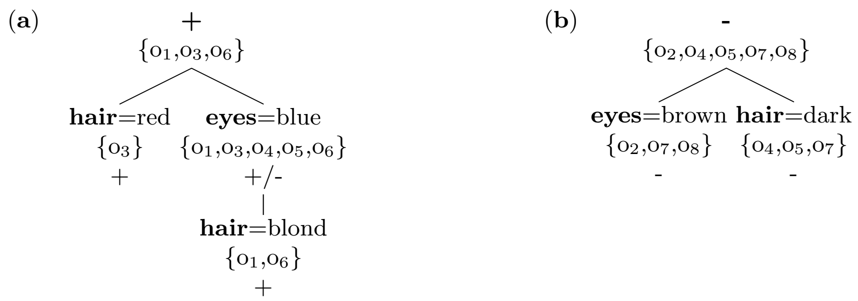
### 5.3 PRISM

PRISM [2] is an algorithm proposed by Cendrowska in 1987. Instead of using the principle of generating decision trees which can be converted to decision rules, PRISM generates rules from the training set directly. More importantly, PRISM is a covering-based method. The PRISM algorithm can be formulated as a heuristic search of the conjunctively definable covering lattice $\mathcal{T}_{CD(C)}(U)$. The heuristic used for PRISM is the conditional probability of a given class value given a formula. The algorithm is described in Figure 4.

**Fig. 4.** 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.
```

**Fig. 5.** An example covering generated by PRISM




From granular computing point of view, PRISM is actually finding a covering of the universe. Let's still use the example of Table 1. There are two classes, + and -. For (**class** = +), the meaning set is $\{o_1, o_3, o_6\}$. The largest conditional probability of class + given all the attribute-value pairs is $P(+|\textbf{hair}=\text{red})$. We use this attribute-value pair to form a granule $\{o_3\}$. The second largest probability is $P(+|\textbf{eyes}=\text{blue})$. We use this attribute-value pair to form a second granule $\{o_1, o_3, o_4, o_5, o_6\}$, and further refine it by combining **eyes**= blue $\wedge$ **hair**= blond. The granule $\{o_1, o_6\}$ contains only (**class** $= +$). So far these two granules cover (**class** $= +$). We do the same for (**class** $= -$) and find two granules $\{o_2, o_7, o_8\}$ and $\{o_4, o_5, o_7\}$ which cover (**class** $= -$). The covering of universe has four granules. The covering is shown in Figure 5. For this particular example, PRISM provide shorter rules than ID3. This is consistent with Cendrowska's results and a recent review [1].

### 5.4 Granular computing approach

A granular computing approach [25, 26] is proposed as a granule network to extend the existing classification algorithms. 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, a 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.

We need to introduce the concepts of inactive and active granules for the implementation of this approach.

**Definition 21.** *A granule $X$ is inactive if it meets one of the following two conditions:*

$$\text{(i).} \quad X \subseteq m(\textbf{class} = c_i) \text{ where } c_i \text{ is one possilbe value of } V_{\textbf{class}},$$
$$\text{(ii).} \quad X = \bigcup Y, \text{where each } Y \text{ is a child node of } X.$$

*A granule $X$ is active if it does not meet any of the above conditions.*

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. Each node in the granule network is a conjunction of some basic granules, and thus a conjunctively definable granule. The induction process of the granule network can be briefly described as follows. The whole universe $U$ is selected as the root node at the initial stage. Evaluate and set the activity status of $U$. If $U$ is active with respect to the conditions (i) and (ii), based on a measure of fitness, one selects a basic concept $bc$ to cover a subset of the universe. Set the status for both the root node $U$ and the new node. Based on a measure of activity, one of the active node is selected for further classification. This iterative process stops when a non-redundant covering solution is found. Figure 6 outlines an algorithm for the construction of a granule network [26].

The two importance issues of the algorithm is the evaluation of the fitness of each basic concept, and the the evaluation of the activity status of each active node. The algorithm is basically a heuristic search algorithm.

**Fig. 6.** An Algorithm for constructing a granule network

```
Construct the family of basic concepts with respect to atomic
  formulas:
            BC(U) = {(a = v, m(a = v)) | a ∈ C, v ∈ V_a}.
Set the granule network to GN = ({U}, ∅), which is a graph consisting
  of only one node and no arc.
Set the activity status of U.
While the set of inactive nodes is not a non-redundant covering
  solution of the consistent classification problem, do:
  (1) Select the active node N with the maximum value of activity.
  (2) Select the basic concept bc = (a = v, m(a = v)) with maximum value
      of fitness with respect to N.
  (3) Modify the granule network GN by adding the granule
      N ∩ m(a = v) as a new node, connecting it to N by arc, and
      labelling it by a = v.
  (4) Set the activity status of the new node.
  (5) Update the activity status of N.
```

The measures discussed in the last section can be used to define different fitness/activity functions. User can also use a measure or some measures to choose the basic concept and active node. The measure does not need to be fixed. In other words, in the process of granular computing classification, user can interactively decide what measure to be used. As a result, different measures can be used at the different levels of construction. In the rest of this section, we will use the running example, Table 1, to illustrate the basic ideas.

The initial node $U$ is an active granule with respect to condition (i). Table 2 summarizes the measures of basic concepts with respect to $U$. There are three granules which are a subset of one of class values, i.e., $\{o_3\} \subseteq$ (**class** $= +$), $\{o_4, o_5, o_7\} \subseteq$ (**class** $= -$) and $\{o_2, o_7, o_8\} \subseteq$ (**class** $= -$). The values of entropy of these granules are minimum, i.e., 0. Therefore, these three granules

| Formula | Granule | Generality | Confidence + | Confidence - | Coverage + | Coverage - | Entropy |
|---|---|---|---|---|---|---|---|
| **height** = short | $\{o_1, o_2, o_8\}$ | 3/8 | 1/3 | 2/3 | 1/3 | 2/5 | 0.92 |
| **height** = tall | $\{o_3, o_4, o_5, o_6, o_7\}$ | 5/8 | 2/5 | 3/5 | 2/3 | 3/5 | 0.97 |
| **hair** = blond | $\{o_1, o_2, o_6, o_8\}$ | 4/8 | 2/4 | 2/4 | 2/3 | 2/5 | 1.00 |
| **hair** = red | $\{o_3\}$ | 1/8 | 1/1 | 0/1 | 1/3 | 0/5 | 0.00 |
| **hair** = dark | $\{o_4, o_5, o_7\}$ | 3/8 | 0/3 | 3/3 | 0/3 | 3/5 | 0.00 |
| **eyes** = blue | $\{o_1, o_3, o_4, o_5, o_6\}$ | 5/8 | 3/5 | 2/5 | 3/3 | 2/5 | 0.97 |
| **eyes** = brown | $\{o_2, o_7, o_8\}$ | 3/8 | 0/3 | 3/3 | 0/3 | 3/5 | 0.00 |

**Table 2.** Basic granules and their measures for the selected active node $U$

are inactive. The generality of latter two granules are higher than the first granule. These three granules are added to the granule network one by one.

The union of the inactive granules $\{\{o_4, o_5, o_7\}, \{o_2, o_7, o_8\}, \{o_3\}$ cannot cover the universe. After adding these three granules to the granule network, the universe $U$ is still the only active node (with respect to condition (i) and (ii)), therefore, it is selected. With the consideration of a non-redundant covering, we will not choose a granule that will not cover the universe even if other measures are in favor of this granule. Based on the fitness measures summarized in Table 3, **hair** = blond and **eyes** = blue contain the objects $\{o_1, o_6\}$ that can possibly form a non-redundant covering solution. **height** = tall and **eyes** = blue have the highest generality, confidence and coverage. **height** = short has the smallest entropy. One can make a comprehensive decision based on all these measures. For example, the basic concept (**eyes** = blue, $\{o_1, o_3, o_4, o_5, o_6\}$) is selected. The granule $\{o_1, o_3, o_4, o_5, o_6\}$ is added to the granule network, labelled by **eyes** = blue. The new node is active (with respect to condition (i)). By adding it to the granule network, $U$ is no longer active (with respect to condition (ii)).

| Formula | Granule | Generality | Confidence | | Coverage | | Entropy |
|---|---|---|---|---|---|---|---|
| | | | + | - | + | - | |
| **height** = short | $\{o_1, o_2, o_8\}$ | 3/8 | 1/3 | 2/3 | 1/3 | 2/5 | 0.92 |
| **height** = tall | $\{o_3, o_4, o_5, o_6, o_7\}$ | 5/8 | 2/5 | 3/5 | 2/3 | 3/5 | 0.97 |
| **hair** = blond | $\{o_1, o_2, o_6, o_8\}$ | 4/8 | 2/4 | 2/4 | 2/3 | 2/5 | 1.00 |
| **eyes** = blue | $\{o_1, o_3, o_4, o_5, o_6\}$ | 5/8 | 3/5 | 2/5 | 3/3 | 2/5 | 0.97 |

**Table 3.** Basic granules and their measures for the selected active node $U$

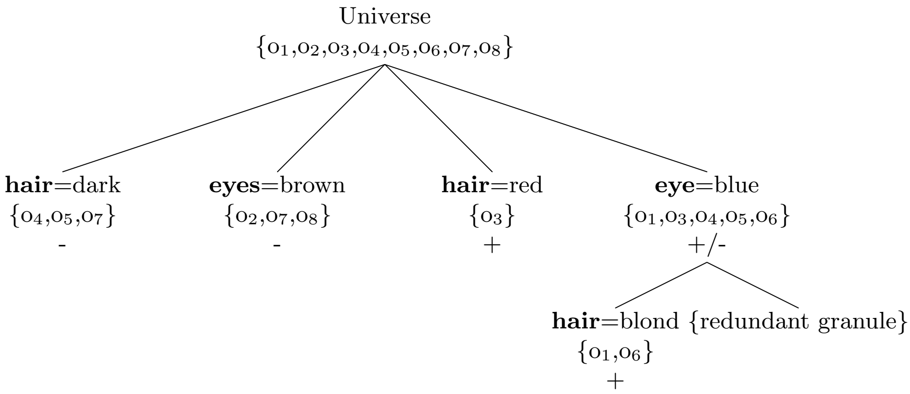
The new node is the only active granule at this stage (with respect to condition (i)). Table 4 summarizes the measures of basic concepts with respect to it. Based on the fitness measures again, **hair** = blond contains all the objects of $\{o_1, o_6\}$ that can possibly form a non-redundant covering solution, it is also in favour of confidence, coverage and entropy. By adding

| Formula | Granule | Generality | Confidence | | Coverage | | Entropy |
|---|---|---|---|---|---|---|---|
| | | | + | - | + | - | |
| $\wedge$**height** = short | $\{o_1\}$ | 1/5 | 1/1 | 0/1 | 1/3 | 0/2 | 0.00 |
| $\wedge$**height** = tall | $\{o_3, o_4, o_5, o_6\}$ | 4/5 | 2/4 | 2/4 | 2/3 | 2/2 | 1.00 |
| $\wedge$**hair** = blond | $\{o_1, o_6\}$ | 2/5 | 2/2 | 0/2 | 2/3 | 0/2 | 0.00 |
| $\wedge$**hair** = red | $\{o_3\}$ | 1/5 | 1/1 | 0/1 | 1/3 | 0/2 | 0.00 |
| $\wedge$**hair** = dark | $\{o_4, o_5\}$ | 2/5 | 0/2 | 2/2 | 0/3 | 2/2 | 0.00 |

**Table 4.** Basic granules and their measures for the selected active node $m($**eye** $=$ blue)

the concept (**hair** = blond, $\{o_1, o_6\}$) to the granule network, we can get another inactive granule, and the union of all inactive granules forms a non-redundant covering solution of the consistent classification problem. It is $\tau = \{\{o_4, o_5, o_7\}, \{o_2, o_7, o_8\}, \{o_3\}, \{o_1, o_6\}\}$. The results are also shown as the granule network in Figure 7.

**Fig. 7.** An example of granule network




## 6 Conclusion

A consistent classification problem can be modelled as a search for a partition or a covering defined by a set of attribute values. In this chapter, we apply a granular computing model for solving classification problems. We explore the structures of classification of a universe. The consistent classification problems are expressed as the relationships between granules of the universe. Different classification lattices are introduced. Depending on the properties of classification rules, solutions to a consistent classification problem are definable granules in one of the lattices. Such a solution can be obtained by searching the lattice. The notion of a granule network is used to represent the classification knowledge. Our formulation is similar to the well established version space search method for machine learning [11].

The ID3, $k$LR and PRISM algorithms are examples of partition and covering search algorithms. As suggested by the No Free Lunch theorem [24], there is no algorithm which performs better than any other algorithms for all kinds of possible problems. It is useless to judge an algorithm irrespectively of the optimization problem. For some data sets, partition algorithm may be better than covering algorithm, for some other sets, the situation is vice versa. The new formulation enables us to precisely and concisely define many notions, and to present a more general framework for classification.

The granular computing classification approach discussed in this chapter provides more freedom of choice on heuristic and measures according to the user needs. The process is penetrated with the idea that the classification task can be more useful if it carries with the user preference and user interaction. In the future research, we will study various heuristics defined by the measures suggested in this chapter, and the evaluation of the proposed algorithm using real world data sets.

## References


1. Bramer, M.A., Automatic induction of classification rules from examples using N-PRISM, *Research and Development in Intelligent Systems XVI*, Springer-Verlag, 99-121, 2000.
2. Cendrowska, J., PRISM: an algorithm for inducing modular rules, *International Journal of Man-Machine Studies*, **27**, 349-370, 1987.
3. Duda, R.O. and Hart, P.E., *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
4. Demri, S. and Orlowska, E., Logical analysis of indiscernibility, in: *Incomplete Information: Rough Set Analysis*, Orlowska, E. (Ed.), Physica-Verlag, Heidelberg, 347-380, 1998.
5. Fayyad, U.M. and Piatetsky-Shapiro, G. (Eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996.
6. Ganascia, J.-G., TDIS: an algebraic formalization, *Proceedings of IJCAI'93*, 1008-1015, 1993.
7. Holte, R.C., Very simple classification rules perform well on most commonly used datasets, *Machine Learning*, **11**, 63-91, 1993.
8. Janikow, C., Fuzzy Decision Trees: Issues and Methods, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, **28**(1), 1–14, 1998.
9. Mehta, M., Agrawal, R. and Rissanen, J., SLIQ: A fast scalable classifier for data mining, *Proceedings of International Conference on Extending Database Technology*, 1996.
10. Michalski, J.S., Carbonell, J.G., and Mirchell, T.M. (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Palo Alto, CA, 463-482, 1983.
11. Mitchell, T.M., Generalization as search, *Artificial Intelligence*, **18**, 203-226, 1982.
12. Mitchell, T.M., *Machine Learning*, McGraw-Hill, 1997.
13. Pawlak, Z., Rough sets, *International Journal of Computer and Information Sciences*, **11**(5), 341-356, 1982.
14. Pawlak, Z., *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.
15. 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, 463-482, 1983.

16. Quinlan, J.R., Induction of decision trees, *Machine Learning*, **1**(1), 81–106, 1986.
17. Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
18. Shafer, J.C., Agrawal, R. and Mehta, M., SPRINT: A Scalable Parallel Classifier for Data Mining, *Proceedings of VLDB'96*, 544-555, 1996.
19. Trochim, W., *The Research Methods Knowledge Base*, 2nd Edition. Atomic Dog Publishing, Cincinnati, OH., 2000.
20. Tsumoto, S., Modelling medical diagnostic rules based on rough sets, *Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence, 1424*, Springer-Verlag, Berlin, 475-482, 1998.
21. Wang, G.Y., Algebra view and information view of rough sets theory, *Proceedings of SPIE'01*, **4384**: 200-207, 2001.
22. Wang, G.Y., Yu, H. and Yang, D.C., Decision table reduction based on conditional information entropy, *Chinese Journal of Computers*, **25**(7), 2002.
23. Wille, R., Concept lattices and conceptual knowledge systems, *Computers Mathematics with Applications*, **23**, 493-515, 1992.
24. Wolpert, D. H., and Macready, W.G., No free lunch theorems for optimization *IEEE Transactions on Evolutionary Computation*, **1**(1), 67-82, 1997.
25. Yao, J.T. and Yao, Y.Y., Induction of Classification Rules by Granular Computing,*Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence*, 331-338, 2002.
26. Yao, J.T. and Yao, Y.Y., A granular computing approach to machine learning, *Proceedings of the First International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'02)*, Singapore, pp732-736, 2002.
27. Yao, Y.Y., Granular computing: basic issues and possible solutions, *Proceedings of the Fifth Joint Conference on Information Sciences*, 186-189, 2000.
28. Yao, Y.Y., On Modeling data mining with granular computing, *Proceedings of COMPSAC'01*, 638-643, 2001.
29. Yao, Y.Y., Information granulation and rough set approximation, *International Journal of Intelligent Systems*, **16**, 87-104, 2001.
30. 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)*, **5**(2), 101-106, 2002.
31. Yao, Y.Y., Zhao, Y. and Yao, J.T., Level construction of decision trees in a partition-based framework for classification, *Proceedings of SEKE'04*, 199-205, 2004.
32. Yao, Y.Y. and Zhong, N., An analysis of quantitative measures associated with rules, *Proceedings of PAKDD'99*, 479-488, 1999.
33. 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.
34. 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.