

A Granular Computing Paradigm for Concept Learning*

Yiyu Yao and Xiaofei Deng

Abstract The problem of concept formation and learning is examined from the viewpoint of granular computing. Correspondences are drawn between granules and concepts, between granulations and classifications, and between relations over granules and relations over concepts. Two learning strategies are investigated. A global attribute-oriented strategy searches for a good partition of a universe of objects and a local attribute-value-oriented strategy searches for a good covering. The proposed granular computing paradigm for concept learning offers twofold benefits. Results from concept formulation and learning enrich granular computing and a granular computing viewpoint sheds new light on concept formulation and learning.

1 Introduction

Concepts are basic units of human thought and have been considered in different disciplines, including philosophy, cognitive science, inductive learning, cluster analysis and machine learning. Many views of concepts and categories have been proposed and studied, such as the classical view, the prototype view, the exemplar view, the frame view, and the theory view [8, 15, 21]. Each view captures a particular perspective and emphasizes a specific aspect, with different intended applications. A comprehensive understanding of concepts is based on an integration of those different views. While some views are suitable for human concept formation and learning, other views are appropriate for machine-oriented approaches. The classical view of

Yiyu Yao and Xiaofei Deng

University of Regina, Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2, e-mail: {yyao, deng200x}@cs.uregina.ca

* **Information about this paper:** Yao, Y.Y., Deng, X.F.: A Granular Computing Paradigm for Concept Learning, in: Ramanna, S., Jain, L., Howlett, R.J. (Eds.), *Emerging Paradigms in Machine Learning*, Springer, London, pp. 307-326, 2012.

concepts is perhaps one of the most used ones in machine-oriented concept learning [32].

In the classical view, a concept is understood as a pair of intension and extension. The intension of a concept consists of all properties or attributes (more generally, some formulas of a language) 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. The simple classical view of concepts can be easily related to the notion of granules in an emerging field of study known as granular computing. Specifically, the extension of a concept may be considered as a granule and the intension of the concept as the description of the granule. Consequently, concept learning may be reconsidered in the paradigm of granular computing.

There are two crucial tasks in concept learning. One is to find a good description of a concept based on its extension; the other is to derive relations between concepts based on their corresponding extensions. We typically employ a rule to describe a relationship between two concepts, in which the intensions of concepts are used. To make a concept learning algorithm effective and its results meaningful, we must consider several issues. One issue is the selection of a set of meaningful basic concepts, from which target concepts can be expressed and interpreted. Another issue is to design strategies for learning. Different strategies may lead to different descriptions of the target concepts.

Based on results from the rough set theory [17], a granular computing paradigm of concept learning is introduced and examined in this chapter. Two learning strategies will be studied in an information table, in which a finite set of objects is described by using a finite set of attributes. An attribute-oriented strategy constructs a partition of the universe by using a subset of attributes. The basic granules are equivalence classes of the partition. Concept learning is formulated as a divide and conquer method in search of a good subset of attributes. The results are a decision tree, which can be translated into a family of disjoint rules. An attribute-value-oriented strategy constructs a covering of the universe. The basic granules are subsets of objects defined by a family of attribute-value pairs. Concept learning is formulated as a chip and conquer method in search of a good family of sets of attribute-value pairs. The results are a family of decision trees, from which a set of overlapping rules can be derived.

The rest of this chapter is organized as follows. Section 2 presents an overview of a triarchic theory of granular computing. Section 3 discusses connections between granular computing and concept learning. Section 4 proposes a model of concept learning based on granular computing, involving an investigation of two classes of strategies for concept learning.

2 A Triarchic Theory of Granular Computing

The notions of categorization, abstraction, formulation and approximation at multiple levels of granularity play a crucial role in human perception, cognition, understanding and problem solving [12, 24, 28, 31]. One would also expect that they play equal important role in intelligent systems. For a system to be called intelligent, it must have similar built-in mechanisms that support human intelligence. Granular computing, as an emerging field of study, aims at a systematic investigation of granule based theories and methodologies for supporting human problem solving on one hand and machine problem solving on the other [27, 33].

The triarchic theory of granular computing [25, 26, 27, 29, 30] provides a conceptual model by adopting useful structures called granular structures and weaving together three powerful ideas: structured thinking, structured problem solving, and structured information processing. It emphasizes the exploitation of useful structures that properly reflect multiple levels of granularity or a granularity pyramid. A single hierarchical granular structure called a hierarchy provides a multilevel understanding and representation of a problem or a system. But it typically captures one particular aspect and therefore offers one view. By constructing a family of hierarchies, it is possible to obtain multiple different views. Granular structures are a family of complementary hierarchies working together for a complete and comprehensive multiview understanding and representation. The use of granular structures, characterized by multilevel and multiview, establishes a solid basis on which sit structured thinking, structured problem solving and structured information processing.

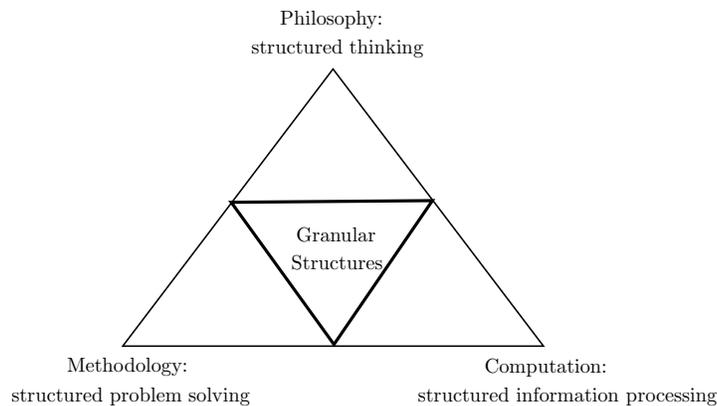


Fig. 1 The Granular Computing Triangle

The main ideas of the triarchic theory of granular computing are summarized by the granular computing triangle of Fig. 1. The center of the triangle is granular struc-

tures, the three angles represent three perspectives of granular computing centered around granular structures.

2.1 Multilevel, Multiview Granular Structures

A key to the success of granular computing is the use of properly constructed granular structures. In the triarchic theory, a hierarchy is used to achieve one particular multilevel view and a family of hierarchies is used to achieve multiview. Granular structures are a family of hierarchies. Fig. 2 illustrates a multilevel hierarchical granular structure. Its main ingredients are explained as follows.

2.1.1 Granules and Granularity

Granules are a primitive notion of granular computing. They are the basic vocabularies of granular computing. Granules are the basic elements in forming granular structures. Using the terminology of systems theory, a granule represents a part of a whole. Granular computing studies the whole through an integration of its parts in terms of their interrelations and their connections to the whole. In Fig. 2, granules are small circle or dots on a plane and each represents a part of the whole given by the entire plane. The physical interpretations and construction of granules are left to particular applications.

The granularity of a granule may be interpreted as an intrinsic or inherent property of the granule. Intuitively, granularity may be interpreted as the degree of abstraction, generalization, complexity or details. A granule with higher degree of granularity is more abstract. It is reasonable to assumed that granularity can be at least partially ordered, so it is possible to form a granularity pyramid. The concept of granularity is essential for constructing and interpreting granules.

2.1.2 Granulations and Levels

Each granule provides a local description of a specific part of a problem or a system. By collecting a family of granules of similar nature or similar granularity, we obtain a complete description of a problem or a system. Such a family of granules is called a granulation of the problem at a particular level. Granules in the family are called focal elements of discussion at the level. In Fig. 2, each level is represented by a plane. While granules at the same level are of similar nature, granules at different levels may be very different. Consequently, we may use different vocabularies and languages for descriptions at different levels.

The processing methods at different levels may also be different. Intitutively, each level of granulation may be considered to be a particular viewpoint of the problem. Thus, a hierarchy provides a family of multiple levels of viewpoints. The

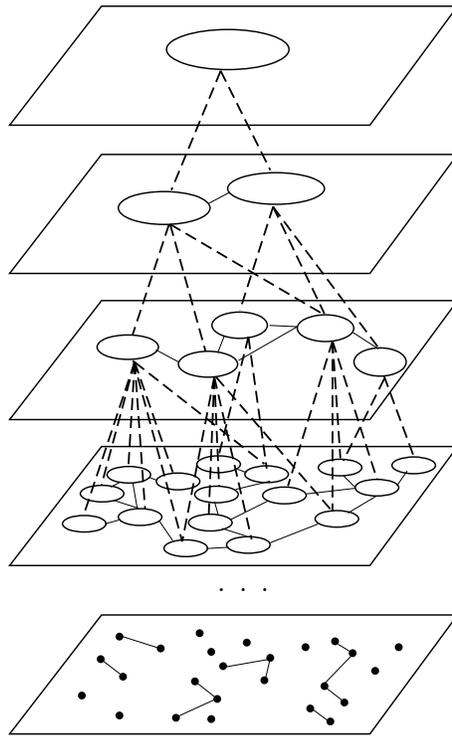


Fig. 2 Multilevel Hierarchical Granular Structure

concept of levels is a very versatile and universal notion that appears in a wide range of disciplines [1, 5, 7, 11, 31]. An interpretation of a granulation hierarchy in terms of levels makes a granular structure to be flexible and useful [31]. The properties of a particular level, namely, a granulation, is determined collectively by a family of granules through their connections and interactions.

2.1.3 Partial Orderings of Granules and Levels

Granules can be ordered based on their relationship or granularity. This leads to a refinement-coarsening relationship between granules. Under this relationship, one can identify dual-roles of a granule. A granule may be considered to be a part that is to be coarsened with other granules into the whole of a coarser or more abstract granule. It may be considered to be a whole that is to be refined into a family of finer or more specific granules. In Fig. 2, the ordering of granules are given by dotted lines between granules in two adjacent levels. It is reasonable to assume that the ordering relation is transitive. In Fig. 2, we omit the connections between no-adjacent levels, as they can be obtained by transitivity.

The ordering of granules leads naturally to the ordering of levels, the result is a multilevel granulation hierarchy as shown in Fig. 2. Although multilevel structures may be rather restrictive to reflect accurately about the reality, it does lead to simplification of computational issues and practical and reasonable approximate solutions to many real world problems.

2.1.4 Three Properties of Granules

A granule has at least three properties: internal properties, external properties, and contextual properties. The individual elements of a granule and their interactions determine internal properties of the granule; the external properties characterize the granule as an inseparable whole; a granule is only meaningful in a certain context. The individual elements of a granule are parts of the granule and are themselves granules. Actually, when forming a granule, we have to ignore the subtle differences between its elements, as well as the connections between each other. In other words, a group of elements are put into the same granule by considering their similar properties and connections and ignoring the subtle differences. We may focus on different issues and form granules with different granularity based on different situations in problem solving.

2.1.5 Relations and Operations on Granules

In a hierarchy, we consider three types of relations on granules. The in-level relations are relationships between granules within the same level. They show the interrelationships and interactions between granules at a particular level. In Fig. 2, they are depicted by solid lines connecting granules on the same plane. In fact, they can be used to study the internal properties of granules in the next upper level. Downward refinement relations show how a coarser granule can be refined into a family of finer granules in the next lower level. They are related to downward operations on granules, such as refinement operation, decomposition operation, or zoom-in operation on granules. Upward coarsening relations show how a family of finer granules is coarsened into a coarser granule in the next upper level. They are related to upward operations on granules, such as coarsening operation, composition operation or zoom-out operation on granules. In Fig. 2, the last two classes of relations are represented by dotted lines between granules in two adjacent levels.

In a hierarchy, we assume that operations on granules are only defined for granules in two immediately adjacent levels. By repeated applications of operations, it is possible to connect granules in all levels. This assumption, although restrictive, makes computation with a granulation hierarchy tractable.

2.2 Philosophy: Structured Thinking

Granular computing promotes structured approaches based on multilevel hierarchical granular structures. From existing studies, it may be observed that the philosophy of granular computing is not entirely new but draws extensively from reductionist thinking, levelism [3, 6, 11] and systems thinking [23].

Reductionist thinking focuses on breaking a complex problems into relatively simpler parts and inferring properties of the whole by a summary of properties of its parts. It offers an effective technique of analytic thinking. In a top-down fashion, a complex system or problem can be decomposed into many parts and these parts can be further divided if needed. When the top-down decomposition is applied to granular computing, a multilevel hierarchical granular structure may be derived. In a bottom-up manner, one may explain entities (i.e., granules) in an upper level based on entities in a lower level.

Levelism makes use of levels of abstraction and takes a hierarchical view that a complex problem may be divided and stratified into levels. The notion of levels may be interpreted in many different ways, including, for example, objective levels of reality, epistemological levels of understanding and explanation, methodological levels of study [3, 6, 11]. To a large extent, levelism is more relevant to granular computing, where the notion of levels of granularity plays a crucial role. Granular structures are stratified levels and levelist thinking leads to level-wise granular processing.

Systems thinking stresses on systemic properties of a whole system that are emerged from the composition, organization and interaction of its parts and cannot be reduced to the properties of its parts. It offers a technique of synthetic thinking for obtaining a holistic view of a system. Moreover, systems thinking also considers different system levels and has the ability to shift attention between system levels. In the context of granular computing, ideas from systems thinking can be used to study emergent properties at different levels of granularity.

Structured thinking of granular computing emphasizes the importance of useful multilevel structures that are common in reductionist thinking, levelist thinking and systems thinking. It takes advantage of those existing philosophical views, in order to develop its own philosophical standpoint.

2.3 Methodology: Structured Problem Solving

Methodology of granular computing is structured problem solving guided by structured thinking, involving the construction and utilization of multilevel, multiview granular structures. Granular computing relies on a set of practical heuristics and systematic approaches that make effective use of multiple levels of granularity. Like the classical 3R (reading, writing, and arithmetic), granular computing may be viewed as the fourth R, representing a set of problem solve skills that can be used by everyone.

The working principles of structured problem solving have been studied in many disciplines. In some sense, granular computing attempts to extract these principles and make them discipline independent and, hence, more accessible. As examples, we examine a few of such principles.

The first two principles concern mainly multilevel and multiview understanding and a plea for the use of granular structures. They can be stated as follows:

- the principle of multilevel view;
- the principle of multiview understanding.

While the multilevel principle emphasizes multiple levels, i.e., viewpoints within a particular view, the multiview principle states the needs for a holistic understanding from many different angles. By the complementing nature of a family of hierarchies, the limitation of a particular hierarchy may be avoided, and a best hierarchy may be chosen.

The advantages of methodology of granular computing are derived from exploring granular structures. The next three principles deal with the exploration of granular structures. They are stated as:

- the principle of focused efforts;
- the principle of view switching;
- the principle of granularity conversion.

The principle of focused efforts is a kind of divide and conquer strategy and can be applied to both a family of hierarchies and a single hierarchy. With respect to a family of hierarchies, it requires that one concentrates on a particular hierarchy relatively independent of other hierarchies; with respect to a single hierarchy, it requires that one pays attention to a particular level without too much interference from other levels.

The principle of view switching demands a comparative study of the same problem under different views. It is necessary to switch to a different view when needed. The principle of granularity conversion guides top-down and bottom-up conversion with respect to a particular hierarchy. Through refinements and coarsening, one is able to work on different levels of granularity.

2.4 Computation: Structured Information Processing

Granular computing is a paradigm of structured information processing. It can be described by an information processing pyramid [2]. Granulations at multiple levels of differing granularity and their conversion are a base of granular information processing.

Representation and process are two basic tasks of computing with granular structures. A representation usually associates with a formal system that explicitly describes entities. Representations in granular computing must reflect multilevel and

multiview structures. A process can be interpreted as actions that carry out information processing tasks. Downward refinement operations, upward coarsening operations and the zoom-in, zoom-out operations are examples of process.

With a multilevel granular structure, we have at least three modes of structured information processing. Starting with the top level, one can progressively compute in a top-down manner towards lower levels through refinement operations. An approximate solution obtained at a higher level is refined into a more accurate solution at a lower level. The process can be terminated whenever a required approximate solution is obtained. In contrast, one may start with the bottom levels and move in a bottom-up manner towards upper levels through coarsening operations. The processes can be terminate when the required level of abstraction is obtained. A third mode is a middle-out approach that starts with a particular level and moves up or down the hierarchy. In general, granular information processing may be an interactive process involving a mixture of the three modes. Moreover, a granular structure is not given prior to the processing, but is constructed during the processing.

3 Granular Computing and Concept Learning

By drawing correspondences between granules and concepts, categorization and granulations, granule construction and concept formation, we demonstrate that granule computing is relevant to concept learning.

3.1 Granules and Concepts

A concept is a mental representation or a mental symbol of human thought [13, 14]. There are many views of concept and concept learning, such as the classical view, prototype view, exemplar view and explanation-based view. Each view explains concepts and concept learning in a unique way, with its advantages and disadvantages. In this chapter, we adopt the classical view of concepts by establishing a correspondence between concepts and granules [32].

The classical view treats a concept as basic unit of thought that consists of two parts, namely, the intension and extension. The intension is a subset of attributes or properties, which are valid for instances (objects) of the concept; the extension is the set of instances of the concept. That is, the intension provides an abstract description of common features shared by a set of instances, while extension employs the set of instances to interpret the concept. All objects or concrete examples in the extension have the same set of common properties described by the intension.

The meaning triangle proposed by Ogden and Richards [16] illustrates the classical view of a concept with an added node to represent natural language coding of a concept. As depicted in Fig. 3, the meaning triangle makes a distinction between *Word*, *Concept* and *Referent* in concept formation. The *Concept*, also called

the intension, thought, or idea, represents an abstract entity usually associated with human thinking. The *Word*, also called symbol or name, is the concrete linguistic entity embodied in speech or written text. The *Referent*, also called the extension or object, represents a physical object in the external world. An arrow from *Word* to *Concept* shows that concepts are coded by a language. Another arrow from *Concept* to *Referent* describes that the extensional objects are mapped into intensional concepts through perception. The dashed line from *Word* to *Referent* depicts an indirect mapping through coding and perception [20, 22].

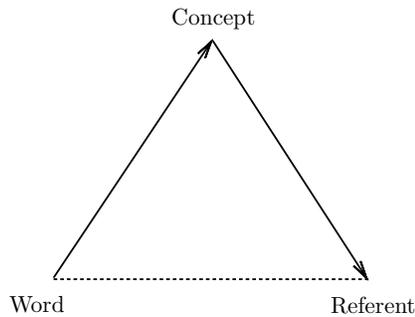


Fig. 3 The meaning triangle of a concept

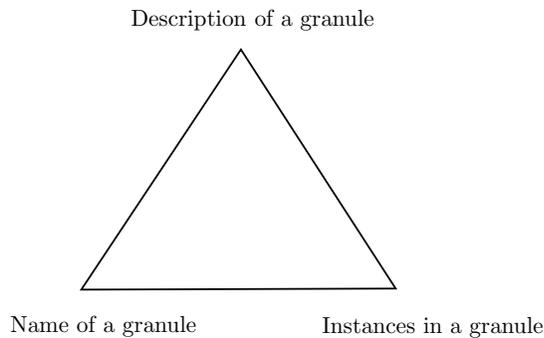


Fig. 4 The meaning triangle of a granule

In the context of concept formation and learning, one may view a granule in terms of the meaning triangle of a concept. As showed in Fig. 4, a granule can be jointly characterized by three elements: *Name of a granule*, *Description of a granule* and *Instances in a granule*. The *Description of a granule* is a representation of a granule, which is a logic formula in this study, the *Instances in a granule* are the family of

objects forming the granule, and the *Name of a granule* is a label assigned to the granule so that the granule can be conveniently referred to.

We may employ a logical language to study description of granules and a set-theoretic language to study instances of granules. According to the connection between concepts and granules, we have a formal definition of a granule in the context of concept learning.

Definition 1. A granule is defined as a triplet:

$$(g, i(g), e(g)),$$

where g is the name assigned to the granule, $i(g)$ is a representation of a granule, and $e(g)$ is a set of objects that are instances of the granule.

In many cases, $i(g)$ and g are the same. Suppose L is a logic language, U is a set of objects, and W is a set of words or labels. Then, $i(g)$ is a formula of language L , $e(g)$ is a subset of U , and g is an element from W .

3.2 Granulation and Classification

A granulation of a universe is a family of granules, with each granule being a subset of the universe. Two types of granulations, called partitions and coverings, can be formed. For simplicity, we only consider a finite nonempty universe.

Definition 2. A family of nonempty subsets of a finite universe U , $\pi = \{b_1, b_2, \dots, b_k\}$, is called a partition of U . If it satisfies the following properties:

$$(P_1) \quad b_i \cap b_j = \emptyset, \text{ for } i \neq j, 1 \leq i, j \leq k, \quad (1)$$

$$(P_2) \quad \bigcup_{i=1}^k b_i = U, \quad (2)$$

each subset $b \in \pi$ is called a block of π .

Definition 3. Suppose Π is the family of all partitions on a finite universe U , a partial order (i.e., a reflexive, antisymmetric and transitive relation) can be defined on Π as follows, for $\pi_1, \pi_2 \in \Pi$,

$$\pi_1 \preceq \pi_2 \iff \forall b \in \pi_1 \exists b' \in \pi_2 (b \subseteq b').$$

The partial ordering provides a refinement-coarsening relation over partitions. When $\pi_1 \preceq \pi_2$, π_1 is called a refinement of π_2 and π_2 a coarsening of π_1 . The relation can be used to build multilevel granulations of a universe, which is a granular structure.

A partition of universe is often called a classification of the universe, where each block represents the extension of a concept. With respect to a partition, granules are pair-wise disjoint. As a generalization, one may consider a covering of a universe.

Definition 4. A family of nonempty subsets of a finite universe U , $\theta = \{c_1, c_2, \dots, c_m\}$, is called a covering of U if it satisfies the condition,

$$(C_1) \quad \bigcup_{i=1}^m c_i = U. \quad (3)$$

It is called a non-redundant covering if it satisfies the condition:

$$(C_2) \quad \bigcup(\theta - \{c\}) \neq U, \quad \forall c \in \theta. \quad (4)$$

That is, each of the subsets in θ is necessary.

Similar to partial ordering on partitions, one can introduce a relation on the family of all coverings.

Definition 5. Suppose Θ is the family of all coverings on a finite universe U . One can define a refinement-coarsening relation on Θ as,

$$\theta_1 \preceq \theta_2 \iff \forall c \in \theta_1 \exists c' \in \theta_2 (c \subseteq c'). \quad (5)$$

The ordering relation \preceq is reflexive and transitive, but not necessarily antisymmetric. When the relation is restricted to the family of all non-redundant coverings, it is also antisymmetric.

3.3 Concept Learning as Searching

An interesting result in cognitive science claims that a concept can be represented by a family of rules that distinguish objects and determine the categorization to which they might belong [8]. This immediately leads to rule-based concept learning. In concept formation and learning, once a concept is formed, it can be used to define new concepts. That is, a crucial task of concept formation and learning is to define new concepts by using existing known concepts.

3.3.1 Learning a single concept

Consider first the task of learning a single concept. Suppose K is a family of subsets of a finite universe U , representing extensions of a family of known concepts. Let $e(g) \in K$ be the extension of a known concept, and $C = e(c) \subseteq U$ the extension of a target concept named c that is to be learned. When $e(g) \subseteq C$, we can formulate a rule,

$$i(g) \rightarrow i(c),$$

That is, if an object is described by $i(g)$ then it is an instance of concept c . In this way, we partially define an unknown concept c by a known concept g . In general,

one can form a family of rules $i(g_i) \rightarrow i(c), i = 1, 2, \dots, p$, to represent c by several known concepts based on the condition $e(g_1) \cup e(g_2) \cup \dots \cup e(g_p) \subseteq C$.

When defining an unknown concept, one may expect that number of rules are as small as possible and each rule is as general as possible.

Definition 6. Suppose C is the extension of an unknown concept and $e(g_1), e(g_2) \in K$ are extensions of two known concepts, c is the name of the unknown concept. If $e(g_1) \subseteq C, e(g_2) \subseteq C$ and $e(g_1) \subseteq e(g_2)$, we say that $e(g_2)$ is more general than $e(g_1)$ and $e(g_1)$ is more specific than $e(g_2)$ in specifying C .

Based on the set-inclusion relation on extensions of concepts, one can design algorithms to search for the best description of unknown concepts by some known concepts.

3.3.2 Learning a classification

For a classification problem, we have a target classification and a family of known concepts, the task is to represent the target classification through known concepts. A solution to a classification problem can be modeled as a search in the space of partitions or coverings whose elements are known concepts.

Definition 7. Let $\pi_D = \{D_1, D_2, \dots, D_t\}$ denotes a target classification. A partition π is called a partition solution to the classification problem if

$$\pi \preceq \pi_D.$$

Similarly, a covering θ is called a covering solution if

$$\theta \preceq \pi_D.$$

If $\pi \preceq \pi_D$ is a solution, then each $b \in \pi$ is included in a class $D_j \in \pi_D$, namely, $b \subseteq D_j$. Thus, we can form a classification rule,

$$i(b) \rightarrow i(D_j),$$

A family of classification rules can be formulated based on all blocks in π .

In practical situations, an arbitrary solution may not be satisfactory. One needs to impose certain conditions. For example, one may require that blocks in π are as large as possible, in other words, their corresponding concepts are as general as possible. This requirement can be precisely stated based on the refinement-coarsening relation \preceq on Π .

Definition 8. Suppose $\pi \preceq \pi_D$ and $\pi' \preceq \pi_D$ are two solutions to the classification π_D . If $\pi \preceq \pi'$, we say that π' is a more general solution than π .

Definition 9. A solution $\pi \preceq \pi_D$ is called a maximal general solution if there do not exists another solution $\pi' \preceq \pi_D$ such that $\pi' \neq \pi$ and $\pi \preceq \pi'$.

A maximal general solution may not be unique and there may exist many maximal general solutions. Based on the notations introduced so far, the problem of learning classification rules may be modeled as a search for a partition such that $\pi \preceq \pi_D$. In particular, one may search for a maximal general solution. The refinement-coarsening relation \preceq provides search direction. One way is to start from a more specific solution, i.e., a finer partition, to move towards a more general solution, i.e., a coarser partition. Alternatively, one may start from a coarser partition that is not a solution and then refine it until a solution is obtained.

The same framework can be easily applied to search for a covering solution. In this case, one needs to consider additional properties on a covering. For example, a covering must be non-redundant so that no redundant rules would be produced. A covering must contain subsets that have least overlap so that a set of less overlapping classification rules would be generated.

4 A Model for Learning a Classification

Based on the discussions of the previous sections, we present a model for learning a classification. The model explicitly considers the following issues raised in Section 3:

- Define a logic language for representing intensions of concepts;
- Construct a family of known concepts;
- Construct a space of partitions for the purpose of searching for a partition solution to a classification;
- Construct a space of coverings for the purpose of searching for a covering solution to a classification.

These issues are investigated by drawing results from rough set theory. A partition-based learning strategy, also called an attribute-oriented strategy, is obtained from algorithms such as ID3 [18], C4.5 [19] and reduct construction in rough set theory [17]. A covering-based strategy, also called an attribute-value-oriented strategy, is obtained from the class of sequential covering algorithms such as the PRISM algorithm [4] and LERS algorithms [9].

4.1 A Decision Logic Language in an Information Table

For precisely defining intension and extension of a concept, rough set theory uses an information table, in which a set of objects is described by a set of attributes.

Definition 10. An information table is a system $S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, where U is a finite nonempty universe of objects, At is a finite nonempty set of attributes, V_a is the domain of attribute $a \in At$, and $I_a : U \rightarrow V_a$ is an information function that maps an object $x \in U$ to a value $v \in V_a$.

We consider a logic language in an information table defined by using attribute-value pairs or logic conjunctive operator \wedge .

Definition 11. A decision logic language (DL) is recursively defined as:

1. Atomic formulas: for any $a \in At, v \in V_a$, $(a = v)$ is an atomic formula.
2. Composite formulas: if ϕ and ψ are formulas, $\phi \wedge \psi$ is a formula.

The logic language defined in this study is only a sublanguage of a decision logic language used in rough set theory [17]. In particular, we are only interested in concepts that are conjunctively defined by a set of atomic formulas.

The semantics of formulas in DL is defined through a meaning assignment that associates each formula with a subset of objects in U , based on the satisfiability of a formula by an object [17, 34].

Definition 12. An object $x \in U$ satisfies an atomic formula $a = v$, written $x \models (a = v)$, if $I_a(x) = v$; it satisfies a formula $\phi \wedge \psi$ if it satisfies both ϕ and ψ , namely, $x \models \phi \wedge \psi$ if and only if $x \models \phi$ and $x \models \psi$.

Definition 13. Suppose DL is the set of all formulas of the language DL. The meaning of a formula $\phi \in DL$ is a subset of objects defined by:

$$m(\phi) = \{x \in U \mid x \models \phi\}. \quad (6)$$

It is possible that two distinct formulas may produce the same meaning set in an information table. This suggests that a concept may have more than one description. The meaning assignment satisfies the following two properties:

$$\begin{aligned} (m_1) \quad m(a = v) &= \{x \in U \mid I_a(x) = v\}, \\ (m_2) \quad m(\phi \wedge \psi) &= m(\phi) \cap m(\psi). \end{aligned}$$

We can compute the meaning of a composite formula from its atomic formulas. By the construction of formulas in DL, one can more conveniently represent a formula by a set of attribute-value pairs as follows. Let $A(\phi)$ denote the set of all attribute-value pairs corresponding to atomic formulas in ϕ . It follows that $A(a = v) = \{(a, v)\}$ and $A(\phi \wedge \psi) = A(\phi) \cup A(\psi)$. In the following discussion, we will use ϕ and $A(\phi)$ interchangeably.

With the introduced logic language, we have a precise representation of a concept or a granule. The intension of a concept is given by a logic formula of DL and the extension is the meaning of the formula.

4.2 Conjunctively Definable Concepts

The language DL enables us to precisely define a concept by a logic formula. However, due to the expressive power and the limitation of the set of attributes, we may

not find a formula for an arbitrary subset of objects. This implies that the language DL only allows us to define a sub-family of the power set 2^U of U .

Definition 14. Suppose that a subset of objects $C \subseteq U$ represents the extension of a concept. We say that C is a conjunctively definable set or concept under DL if and only if there exists a formula $\phi \in DL$ such that

$$C = m(\phi). \quad (7)$$

Otherwise, it is conjunctively undefinable. The set of all conjunctively definable concepts is given by:

$$K = \{m(\phi) \mid \phi \in DL\} \subseteq 2^U. \quad (8)$$

The family of conjunctively definable concepts consists of the building blocks, namely, known concepts, from which new unknown concepts may be learned. That is, we use known concepts to express other concepts.

4.3 Attribute-oriented Search Strategies in a Space of Partitions Defined by Subsets of Attributes

A crucial step in learning a classification is to construct a space of partitions based on the family of conjunctively definable sets K . Theoretically speaking, one can obtain such a space by collecting all partitions whose blocks are definable concepts from K . However, constructing and searching the space of all such partitions is practically difficult. Instead, one may use a subspace by considering partitions with additional properties [36]. Rough set theory uses a space of partitions defined by subsets of attributes.

In an information table, each subset of attributes defines an equivalence relation on U .

Definition 15. Suppose $P \subseteq At$ is a subset of attributes, an equivalence relation defined by P is given by: for $x, y \in U$,

$$xE_Py \iff \forall a \in P (I_a(x) = I_a(y)). \quad (9)$$

That is, x and y are equivalent if they have the same values on all attributes in P . The induced partition is given by $U/E_P = \{[x]_{E_P} \mid x \in U\}$, where $[x]_{E_P} = \{y \in U \mid xE_Py\}$ is the equivalence class containing x .

It can be verified that $[x]_{E_P}$ is defined by a logic formula $\bigwedge_{a \in P} a = I_a(x)$, that is, $m(\bigwedge_{a \in P} a = I_a(x)) = [x]_{E_P}$. Thus, $U/E_P \subseteq K$ is a partition whose blocks are conjunctively definable sets. More importantly, the set-inclusion relation on subsets of attributes leads to a refinement-coarsening relation on their induced partitions. For two subsets of attributes, $P_1, P_2 \subseteq At$ with $P_1 \subseteq P_2$, the following implication holds:

$$P_1 \subseteq P_2 \implies U/E_{P_2} \preceq U/E_{P_1}. \quad (10)$$

Algorithm 1: Construction of a Relative Reduct by Deletion

Input: A classification table $S = (U, At = C \cup D, \{V_a\}, \{I_a\})$;
 A fitness function δ ;
Output: A relative reduct R

```

1 begin
2    $R = C$ ;
3    $unchecked = C$ ;
4   while  $unchecked \neq \emptyset$  do
5     Compute the fitness of all the attributes in  $unchecked$  using a fitness function  $\delta$ ;
6     Select an attribute  $a \in unchecked$  with minimum fitness;
7      $unchecked = unchecked - \{a\}$ ;
8     if  $\pi_{R-\{a\}} \preceq \pi_D$  then  $R = R - \{a\}$ ;
9   end
10 end

```

Fig. 5 Attribute-oriented Relative Reduct Construction

Consequently, searching in the space $\Pi_{At} = \{U/E_P \mid P \subseteq At\}$ may be viewed as searching in the space 2^{At} .

In rough set theory, searching for a partition solution to a classification is formulated as finding an attribute reduct.

Definition 16. Suppose $S = (U, At = C \cup D, \{V_a\}, \{I_a\})$ a consistent classification or a decision table, where C is the set of condition attributes, D is the set of classification or decision attribute and $\pi_C \preceq \pi_D$. A subset $R \subseteq C$, is called a relative reduct of C with respect to D , if R satisfies two conditions:

1. $U/E_R \preceq U/E_D$;
2. $\forall a \in R, \neg(U/E_{(R-\{a\})} \preceq U/E_D)$.

Condition (1) suggests that π_R is a solution to the classification π_D . That is, attributes in R are jointly sufficient. Suppose $P \subseteq C$ is a subset of attributes with $\pi_P \preceq \pi_D$. An attribute in P is said to be redundant if $\pi_{P-\{a\}} \preceq \pi_D$. Condition (2) of Definition 16 states that R does not contain any redundant attributes. That is, attributes in R are individually necessary. The two conditions together imply that π_R is a maximal general solution to π_D in Π_C .

There are three searching strategies for finding a reduct, namely, deletion strategy, addition-deletion strategy and addition strategy [35]. A deletion strategy starts from the entire set of conditional attributes and sequentially deletes redundant attributes. An addition-deletion strategy starts with the empty set and sequentially adds attributes until a subset of attributes satisfying condition (1) of Definition 16. is obtained; it then delete redundant attributes. An addition strategy only adds attributes that will form a reduct.

Fig. 5 gives an algorithm that implements a deletion strategy for constructing a reduct. The algorithm starts from the most specific partition solution π_C . At each iteration of the while loop, an attribute may be deleted to generate a more general

solution. When every attribute is checked, the algorithm produces a reduct that provides a maximal general solution. Many authors have introduced and studied fitness functions, including dependency measures, mutual information, conditional entropy and others.

4.4 Attribute-value-oriented Search Strategies in a Space of Coverings Defined by Families of Sets of Attribute-Value Pairs

The partition based model can be modified to formulate a covering based model. Recall that each definable concept in K is defined by a conjunction of a family of atomic formula and it can be eventually expressed by the corresponding set of attribute-value pairs. Thus, a covering $\theta \subseteq K$ of the universe U can be viewed as a family of sets of attribute-value pairs. This leads to attribute-value-oriented search strategy in a space of coverings.

The set of all definable set K is a covering of U . The problem of finding a covering solution can be formulated in terms of a reduct of K relative to π_D .

Definition 17. A covering $\theta_R \subseteq K$ is called a relative reduct of K with respect to π_D if θ_R satisfies the following conditions:

1. $\theta_R \preceq \pi_D$;
2. For any $c \in \theta_R, \forall c' \in K (c \subset c' \implies \neg(((\theta_R - \{c\}) \cup \{c'\}) \preceq \pi_D))$;
3. For any $c \in \theta_R, \neg(\bigcup(\theta_R - \{c\}) = U \wedge ((\theta_R - \{c\}) \preceq \pi_D))$.

Condition (1) states that the covering must be a solution to π_D . The covering $(\theta_R - \{c\}) \cup \{c'\}$ is produced by replacing c in θ_R with one of its proper supersets in K . Condition (2) states that such a replacement will not produce a new solution. This implies that each set in θ_R must be maximal. Condition (3) states that each set in θ_R is necessary; the deletion of any of them will produce a family of sets that is either not a covering or not a covering solution to π_D .

By the definition of a relative attribute-value reduct, we can select a family of sets from K based on conditions (1)-(3) of Definition 17. In order to ensure condition (1), each set in a covering must be a subset of a class in π_D . Thus, we construct a covering solution to π_D as follows:

$$\theta_K = \{c \in K \mid \exists D_j \in \pi_D (c \subseteq D_j)\}$$

According the requirement that each set in a covering must be a maximal set by condition (2), we simplify θ_K by removing the sets in θ_K that are subsets of other sets in θ_K :

$$\theta_K^* = \{c \in \theta_K \mid \neg(\exists c' \in \theta_K (c \subset c'))\}.$$

That is, θ_K^* is a covering such that no set in θ_K^* is a subset of another set in θ_K^* . Finally, we can delete sequentially redundant set in θ_K^* .

Fig. 6 is an algorithm for constructing a covering. It should be pointed out that the proposed algorithm is a straightforward implementation based on the definition of

Algorithm 2: Construction of a relative attribute-value reduct

Input: A decision table $S = (U, At = C \cup D, \{V_a\}, \{I_a\})$;
a fitness function γ ;

Output: A relative attribute-value reduct θ_R

```

1 begin
2    $\theta_K = \{c \in K \mid \exists D_j \in \pi_D(c \subseteq D_j)\}$ ;
3    $\theta_K^* = \{c \in \theta_K \mid \neg(\exists c' \in \theta_K(c \subset c'))\}$ ;
4    $\theta_R = \theta_K^*$ ;
5    $unchecked = \theta_R$ ;
6   while  $unchecked \neq \emptyset$  do
7     Compute the fitness of all the sets in  $unchecked$  using the fitness function  $\gamma$ ;
8     Select a set  $c \in unchecked$  with minimum fitness ;
9      $unchecked = unchecked - \{c\}$ ;
10    if  $\cup(\theta_R - \{c\}) = U \wedge (\theta_R - \{c\}) \preceq \pi_D$  then  $\theta_R = \theta_R - \{c\}$ ;
11  end
12 end

```

Fig. 6 Attribute-value-oriented Relative Reduct Construction

relative reduct. One may consider more efficient implementations based on existing sequential covering algorithm such as PRISM [4] and LERS [9, 10].

5 Conclusion

This chapter examines a granular computing paradigm for concept learning. To a large degree, it reformulates and reinterprets some existing results in the proposed paradigm with two purposes. One is to provide a concrete model of granular computing and the other to provide a different view for concept learning.

The triarchic theory of granular computing is briefly summarized. The connections between granules in granular computing and concepts in concept learning are established. We argue that a concept may be expressed by a granule based on the classical view of concepts. For classification, the notion of a granulation, i.e., a family of granules, is introduced. Two types of granulation, namely, partitions and coverings, are investigated. The two types of granulation lead to two classes of strategies for solving a classification problem. A model for learning a classification is proposed. Based on the rough set theory and other concept learning algorithms, two strategies are introduced, namely, an attribute-oriented strategy for searching a space of partitions and an attribute-value oriented strategy for search space of coverings. More importantly, the notion of a relative attribute-value reduct is formally expressed, which is complementary to the widely used notion of a relative attribute reduct.

References

1. Ahl V, Allen TFH (1996) *Hierarchy Theory: A Vision, Vocabulary, and Epistemology*. Columbia University Press, New York
2. Bargiela A, Pedrycz W (2002) *Granular Computing: An Introduction*. Kluwer Academic Publishers, Boston
3. Brown HC (1916) Structural levels in the scientist's world. *The Journal of Philosophy, Psychology and Scientific Methods* 13: 337-345
4. Cendrowska J (1987) PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies* 27: 349-370
5. Craik FIM, Lockhart RS (1972) Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior* 11: 671-684
6. Floridi L (2008) The method of levels of abstraction. *Minds and Machines* 18: 303-329
7. Foster CL, Foster C (1992) *Algorithms, Abstraction and Implementation: Levels of Detail in Cognitive Science*. Academic Press, London
8. Goldstone RL, Kersten A (2003) Concepts and categorization. In: Alice FH, Robert WP, Irving BW (eds.) *Handbook of Psychology, Experimental Psychology*, pp. 597-621. Wiley, New Jersey
9. Grzymala-Busse J (1992) LERS - A system for learning from examples based on rough sets. In: Slowinski R (ed.) *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, pp. 3-18. Kluwer Academic Publishers, Boston
10. Grzymala-Busse J, Rzasa, W (2008) Approximation space and LEM2-like algorithms for computing local coverings, *Fundamenta Informaticae* 85: 205-217
11. Heil J (2003) Levels of reality. *Ratio* 16: 205-221.
12. Hobbs JR (1985) Granularity. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*: 432-435
13. Hunt EB (1962) *Concept Learning: An Information Processing Problem*. Wiley, New York
14. Margolis E, Laurence S (2007) The ontology of concepts - abstract objects or mental representations? *Noûs* 41: 561-593
15. Medin DL, Smith EE (1984) Concepts and concept formation. *Annual Review of Psychology* 35: 113-138
16. Ogden CK, Richards IA (1946) *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, 8th edition, Harcourt Brace, New York
17. Pawlak Z (1991) *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Boston
18. Quinlan JR (1986) Induction of decision trees. *Machine learning* 1: 81-106
19. Quinlan J (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco
20. Regoczei S, Hirst G (1990) The meaning triangle as a tool for the acquisition of abstract, conceptual knowledge. *International Journal of Man-machine Studies* 33: 505-520
21. Smith EE, Medin DL (1981) *Categories and Concepts*. Harvard University Press, Cambridge
22. Tzafestas SG (1997) *Knowledge Based Systems: Advanced Concepts, Techniques & Applications*. World Scientific, New Jersey
23. Wuketits FM (1987) Synthetic and analytical thinking. *Fresenius' Journal of Analytical Chemistry* 326: 320-323
24. Yao YY (2001) Information granulation and rough set approximation. *International Journal of Intelligent Systems* 16: 87-104
25. Yao YY (2005) Perspectives of granular computing. In: *Proceedings of the 2005 IEEE International Conference on Granular Computing* 1: 85-90
26. Yao YY (2006) Three perspectives of granular computing. *Journal of Nanchang Institute of Technology* 25: 16-21
27. Yao YY (2006) Granular computing for data mining. In: *Proceedings of SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*. doi:10.1117/12.669023

28. Yao YY (2007) The art of granular computing. In: Proceedings of the International Conference on Rough Sets and Intelligent Systems Paradigms, LNCS(LNAI) 4585: 101-112
29. Yao YY (2008) A unified framework of granular computing. In: Pedrycz W, Skowron A and Kreinovich V (eds.) Handbook of Granular Computing, pp. 401-410. Wiley, New York
30. Yao YY (2008) Granular computing: past, present, and future. In: Proceedings of the 2008 IEEE International Conference on Granular Computing: 80-85
31. Yao YY (2009) Integrative levels of granularity. In: Bargiela A and Pedrycz W (eds.) Human-Centric Information Processing Through Granular Modelling, pp. 31-47. Springer, Berlin
32. Yao YY (2009) Interpreting concept learning in cognitive informatics and granular computing. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39: 855-866
33. Yao YY (2010) Human-inspired granular computing. In: Yao JT (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, pp. 1-15. Information Science Reference, Hershey
34. Yao YY, Zhou B (2007) A logic language of granular computing. In: Proceedings of the 6th IEEE International Conference on Cognitive Informatics: 178-185
35. Yao YY, Zhao Y, Wang J (2006) On reduct construction algorithms. In: Proceedings of the 1st International Conference on Rough Sets and Knowledge Technology, LNCS(LNAI) 6042: 297-304
36. Zhao Y, Yao YY, Yao JT (2006) Level-wise construction of decision trees for classification. International Journal of Software Engineering and Knowledge Engineering 16:103-126