Proceedings of the Fifth International Conference of Cognitive Informatics (ICCI'05), 250-259, 2005.

Interactive Classification Using a Granule Network *

Yan Zhao and Yiyu Yao
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{*yanzhao, yyao*}*@cs.uregina.ca*

Abstract

Classification is one of the main tasks in machine learning, data mining and pattern recognition. Compared with the extensively studied data-driven approaches, the interactively user-driven approaches are less explored. A granular computing model is suggested for re-examining the classification problems. An interactive classification method using the granule network is proposed, which allows multi-strategies for granule tree construction and enhances the understanding and interpretation of the classification process. This method is complementary to the existing classification methods.

1. Introduction

Classification is a problem that has been extensively studied in many disciplines, such as machine learning, data mining and pattern recognition. Generally, classification can be achieved by two approaches: one is data-driven, automatically executed by machines, the other is user-driven, executed interactively by both machines and users.

Data-driven classification systems reveal the internal structure of the data by programmed algorithms, and do not allow users, or limit the user to contact, to participate in the discovery process. A typical data-driven approach is a batch processing where all the input is prepared before the program runs. The problem of a data-driven classification system is that a user often cannot relate to the answers, and is left wondering about the meaning and value of the so called discovered knowledge [15]. For a data-driven system, a fixed algorithm may not satisfy the diverse requirements of users.

On the contrary, user-driven classification systems allow users to suggest preferred classifiers and structures, and use machines to assist the calculation and analysis during the discovery process. User-driven approach is in an interactive manner between users and machines. Its input and output are interleaved, like a conversation. A user can freely explore the dataset according to his/her preference and priority, ensure that each classification stage and the corresponding results are all understandable and comprehensible. The constructed classifier is not necessarily the most efficient one comparing to the existing data-driven classifiers. It is close to human thinking by its nature.

It is important to note that users process various skills, intelligence, cognitive styles, frustration tolerances and other mental abilities. They come to a classification problem with various preferences, requirements and background knowledge. Given a set of data, every user may try to make sense of data by seeing it from different angles, in different aspects, and under different views. Based on these differences, there does not exist a universally applicable theory or method to serve the needs of all users. This motivates and justifies the co-existence of many theories and methods for data-driven classification systems, as well as the exploration of new theories and methods. The existing classification algorithms simply represent various heuristics that can be applied as rule searching strategies.

The philosophy behind interactive user-driven classification is cognitive informatics (CI) [16, 17, 18]. As Wang stated in [18] that CI attempts to solve problems in two connected areas: "One, CI uses computing techniques to investigate cognitive science problems, such as memory, learning, and thinking; two, CI uses cognitive theories to investigate informatics, computing, and software engineering." Wang emphasized the relations between object-object, object-attribute and attribute-attribute [17]. In particular, the relational metaphor is suggested, which assumes that *relations* and *connections* of neurons represent information and knowledge in the brain, rather than the neurons [17]. Following the same way of thinking, we believe that for an interactive system, the most critical thing is not only how intelligent the user is, or how efficient the system is, but also how well these two parts connect and communicate. Through interaction and communication, computers and users can divide the labour in order to achieve a good balance of automation and human control. Moreover, an interactive computer system can encourage users's

*The work is partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Regina.

learning, improve insights and understandings of the domain, and stimulate users to explore creative possibilities. Users' feedback can be used to improve the system for abundant, novel and unique features and capabilities. The interaction is bi-beneficial.

The interaction can be conducted through a text-based interface, a graphical user interface, or other kinds of interface, such as speech recognition and/or speech synthesis. In fact, the interactive classification has evolved rapidly in recent years [1, 6, 7, 8]. Most of the existing interactive classifiers add the visual functionality into the process, which enables users to invigilate the classification process at some stages. Han *et al.* pointed out that, "most visualization systems concentrate on the raw data visualization and/or the final results visualization, but lack the ability to visualize the entire process of knowledge discovery." [7] They suggested that the knowledge discovery process include: data preparation, data selection and reduction, data preprocessing and transformation, pattern discovery, pattern evaluation and pattern visualization. In an interactive system, these phases can be carried out as follows:

- Data preparation is to visualize the raw data with specific format. Data distribution and some relationships between attributes can be easily observed.
- Data selection and reduction involves the reduction of the number of attributes and/or the number of tuples. The user can specify the interested attributes and data area, and remove other data that are outside the interesting area.
- Data preprocessing and transformation determines the number of intervals as well as cut-points for continuous datasets, and transforms the dataset into a workable dataset.
- Pattern discovery is to visualize the preprocessed data and interactively construct the patterns under users guidance, monitoring and supervision.
- Pattern evaluation is to evaluate the discovered patterns whenever the user is willing to. The usefulness is subject to the user's judgement.
- Pattern visualization is to visualize the patterns that are perceived in the pattern discovery phase.

Our proposed approach focuses on the interactive pattern discovery phase, and is designed in this regard as to construct a reasonable and meaningful classifier to an individual user. To a novice, the constructive operation is the psychological paradigm in which one constructs his/her own mental model of the given domain; to an expert, the constructive operation is an experienced practice contains anticipation, estimation, understanding and management of the domain.

Particularly, the system enables users to be involved in two main issues of the classification. First, it allows the user to visually select and state *where to classify*. A user can express his/her own interest or priority to execute the classification. The interactive system facilitates the selection of the candidate nodes, based on the complete knowledge of the search space provided by a granule network. Second, the system allows the user to visually select and state *how to classify*. Many measures are involved in the classification process. They represent different aspects of the data. The interactive system assists the user to observe all these aspects at the same time, and helps the user to select the measure that is preferred. Compared with the existing classification systems, the interactive approach provides more information and insights into the data, and integrates domain knowledge and user preference into all classification phases.

In the rest of this paper, we first revisit the granular computing model and formally analyze a granule network for classification tasks in Section 2. After a brief review of the existing classification algorithms in Section 3, the interactive granule classification is introduced and demonstrated in Section 4. The experiment is presented in Section 5, followed by conclusion in Section 6.

2. A Granular Computing Model for Classification

This section provides an overall setting of classification problems within a granular computing model [20, 22].

2.1. Granular computing

We need to answer why we have chosen granular computing as an appropriate mathematical model for classification problems.

Ever since the introduction of the term of Granular Computing (GrC) [9], a rapid development of, and a fast growing interest in, this topic have been observed. A granule, a subset of the universe, is regarded as the primitive notion of granular computing. We can refer to a level consists of a family of granules as a granulated view. Granules in different levels are linked by order relations in a hierarchy. A granule in a higher level can be decomposed into many granules in a lower level, and, conversely, many granules in a lower level can be combined into granules in a higher level. A granule in a lower level provides detailed description of the granule in a higher level, and a granule in a higher level has a more abstract description than the granules in a lower level.

From the standpoint of granular computing, a concept may be exemplified by a granule, and be described or labelled by a formula. Once concepts are constructed and described, one can develop computational methods for the granule and the formula, such as the sub- and the superconcepts, and the disjoint and overlapped concepts [21]. These relationships can be conveniently expressed in the form of rules, with some associated quantitative measures indicating the strength.

By combining the results from granular computing and formal concept analysis, 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 granules and formulas. Specifically, classification deals with grouping or clustering of objects based on certain criteria. It is directly related to concept formation and concept relationship identification [22]. While concept formation involves the construction of classes and description of classes, concept relationship identification involves the connections between classes. The classification problem is then properly modelled by the granular computing theory.

2.2. 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. 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:*

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

where U *is a finite nonempty set of objects,*
At *is a finite nonempty set of attributes,*
$\mathscr{L}$ *is a language defined by attributes in* At,
V_a *is a nonempty set of values of* $a \in At$,
$I_a : U \rightarrow V_a$ *is an information function.*

The mapping $I_a(x) = v$ means that the value of object x on attribute a is v, where $v \in V_a$. We can easily extend the information function $I_a(x)$ to an information function on a subset of attributes, or an information function of a subset of objects.

Definition 2 *In the language* $\mathscr{L}$*, an atomic formula is given by* $a = v$*, where* $a \in At$ *and* $v \in V_a$*. If* ϕ *and* ψ *are formulas, then so are* $\neg\phi$*,* $\phi \wedge \psi$*, and* $\phi \vee \psi$*.*

Definition 3 *Given a formula* ϕ*, if an object* s *satisfies* ϕ*, we write* $x \models \phi$*. The set* $m_S(\phi)$*, defined by*

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

Table 1. An information table

	A	B	C	D	*class*
o_1	a_1	b_1	c_1	d_1	+
o_2	a_1	b_1	c_1	d_2	-
o_3	a_1	b_1	c_2	d_1	+
o_4	a_1	b_1	c_2	d_2	-
o_5	a_1	b_2	c_1	d_1	+
o_6	a_1	b_2	c_1	d_2	-
o_7	a_1	b_2	c_2	d_1	+
o_8	a_1	b_2	c_2	d_2	-
o_9	a_2	b_1	c_1	d_1	+
o_{10}	a_2	b_1	c_1	d_2	-
o_{11}	a_2	b_1	c_2	d_1	-
o_{12}	a_2	b_2	c_1	d_1	+
o_{13}	a_2	b_2	c_1	d_2	-
o_{14}	a_2	b_2	c_2	d_1	+
o_{15}	a_2	b_2	c_2	d_2	+
o_{16}	a_3	b_1	c_1	d_1	+
o_{17}	a_3	b_1	c_1	d_2	+
o_{18}	a_3	b_1	c_2	d_1	+
o_{19}	a_3	b_1	c_2	d_2	-
o_{20}	a_3	b_2	c_1	d_1	+
o_{21}	a_3	b_2	c_1	d_2	-
o_{22}	a_3	b_2	c_2	d_1	+

is called the meaning of the formula ϕ *in* S*. If* S *is understood, we simply write* $m(\phi)$*.*

In the theory of granular computing, a granule is a subset of the universe, i.e., a set of objects. A granule is definable if it is associated with at least one formula. A formula ϕ can be viewed as the description of a granule $m(\phi)$; a granule $m(\phi)$ contains the set of objects having the property expressed by ϕ. A connection between formulas of $\mathscr{L}$ and subsets of U is thus established. This formulation enables us to study formal concepts in a logic setting in terms of formulas, and also in a set-theoretic setting in terms of granules.

For classification tasks, it is assumed that information about objects is given by an information table, and each object is associated with a unique class label. Objects can be divided into classes which form a granulation of the universe. 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 $At = \mathscr{D} \cup \{class\}$, where $\mathscr{D}$ is the set of attributes used to describe the objects, also called the set of descriptive attributes.

Table 1 is a sample information table with $U = \{o_1, o_2, \ldots o_{22}\}$, and $At = \{A, B, C, D, class\}$. The attribute A has three possible values $V_A = \{a_1, a_2, a_3\}$. Other at-

tributes have two possible values, namely, $V_B = \{b_1, b_2\}$, $V_C = \{c_1, c_2\}$, $V_D = \{d_1, d_2\}$, and $V_{class} = \{+, -\}$, respectively. The attributes A, B, C and D define an object, and the label $+$ or $-$ represents which class an object belongs to.

2.3. Granule networks for classification

Definition 4 *A granule $m(\phi)$ (where ϕ is defined using attributes in the set $\mathscr{D}$ of descriptive attributes) is consistently classified into a class c_i, if $I_{class}(x) = c_i$ for all $x \in m(\phi)$. The consistent classification rules are of the form $\phi \Longrightarrow class = c_i$.*

In many classification applications, one is only interested in formulas of a certain form. Suppose we restrict the connectives of language $\mathscr{L}$ to only the conjunction connective $\wedge$, which means that each formula is a conjunction of atomic formulas. Such a formula is referred to as a conjunctor. A conjunctor that contains only one atomic formula defines a 1-conjunctor granule; a conjunctor that conjuncts n atomic formulas defines a n-conjunctor granule. The most general granule is 0-conjunctor, indicating the whole universe. With respect to the descriptive attribute set $\mathscr{D}$, the most specific granules in an information table are the $|\mathscr{D}|$-conjunctor granules, where $|.|$ denotes the cardinality of the set.

A granule network systematically organizes all the granules and formulas with respect to the given universe. A granule network has $|At|$ levels at most. Each node consists of a granule, and each arc leading from a granule to its child granule is labelled by an atomic formula. A path from a coarse granule to a fine granule indicates a conjunctive relation. According to a hierarchical structure, the root node is the universe. The second level contains all the 1-conjunctor granules, the third level contains all the 2-conjunctor granules, and so on, till the $|At|$-th level contains all $|\mathscr{D}|$-conjunctor granules. To create a granule network, we need to understand the number of conjunctors (conjunctive formulas) and the number of conjunctively definable granules. They determine the size of the granule network.

The number of conjunctors: There is only one 0-conjunctor, $\emptyset$, and $\sum_{a\in\mathscr{D}} |V_a|$ 1-conjunctors. The number of 2-conjunctors is $\sum_{a_i,a_j\in\mathscr{D}} |V_{a_i}| * |V_{a_j}|$. For n-conjunctors, the number is $|V_{a_1}| * \ldots * |V_{a_n}|$, where $a_1 \ldots a_n \in \mathscr{D}$, and they are not the same.

The number of conjunctively definable granules: Because of the associativity of conjunction, two conjunctors $\phi \wedge \psi$ and $\psi \wedge \phi$ are associated with the same granule. In other words, a 2-conjunctor granule can be defined by 2! conjunctors, a 3-conjunctor granule can be defined by 3! conjunctors, and a n-conjunctor granule can be defined by $n!$ conjunctors, etc. It is easy to verify that, the total number of conjunctively definable granules is the product of cardinality of possible values of each attribute plus one, defined by $\prod_{a\in\mathscr{D}}(|V_a| + 1)$.

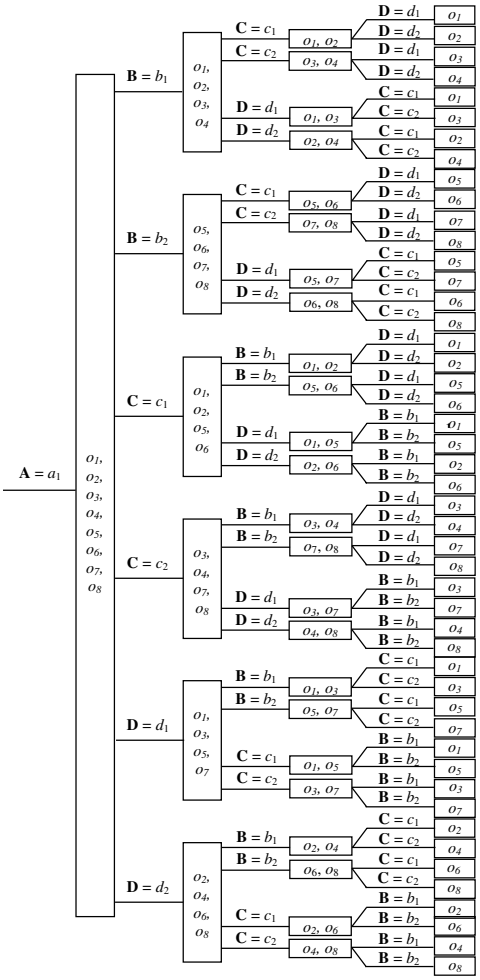


Figure 1. A portion of the granule network of Table 1

Regarding to the classification tasks, the order of atomic formulas forming a conjunctor does affect the efficiency and effectiveness of search and retrieval. For example, one may find that the process of first retrieving the granule $m(\phi)$, then further retrieving the finer granule $m(\phi \wedge \psi)$, is faster, more reasonable, or more feasible, than the process of first retrieving the granule $m(\psi)$, and then further retrieving the finer granule $m(\psi \wedge \phi)$. The other fact is that if a granule is consistently classified, then all its finer granules are also consistently classified, and thus the finer solutions are trivial for the classification purpose. Suppose one does not find this general granule first, but instead, some finer granules are obtained. As a result, one has many finer granules that do not need to be kept. There is a set of formulas associated with a single granule, and the expressive powers of these formulas are not equal.

For our running example, Figure 1 magnifies one portion of the granule network. It shows that while the portion $A = a_1$ is investigated, 27 conjunctively definable granules and 79 formulas are examined. The exact numbers of the conjunctively definable granules and conjunctors are shown in Table 2.

Table 2. The numbers of granules and conjunctors of the granule network corresponding to Table 1

level	# of conjunctively definable granules	# of conjunctors
0-conjunctor	1	1
1-conjunctor	3+2+2+2=9	9*1!=9
2-conjunctor	3(3*2)+3(2*2)=30	30*2!=60
3-conjunctor	3(3*2*2)+2*2*2=44	44*3!=264
4-conjunctor	3*2*2*2=24	24*4!=576
Total	108	910

The immediate result is that a consistent classification task can be understood as a search of the distribution of classes in a granule network defined by the descriptive attribute set $\mathscr{D}$. The analysis shows that the complexity of the search space of a consistent classification task is not a polynomial bound at all. This can be extremely complex especially when the number of possible values of attributes are large, let alone continuous. This forces us to use heuristic algorithms to quickly find solutions in a constrained space. Indeed, the existing heuristic algorithms perform very well. Each of them can be understood as a particular heuristical search within the granule network.

3. The Studies on Classification Algorithms

Partition-based algorithms look for the most promising attributes to split the examined granules at each level, and each is labelled by one of the possible values of the selected attribute. The child granules naturally cover their parent granule, and pairwisely disjoint with each other. Various measures are applied in order to find the most promising attributes. For example, ID3 [11], ASSISTANT [4] and C4.5 [12] use information entropy measurements, and CART [2] uses *Gini* index.

Covering-based algorithms look for the most promising attribute-value pairs at each level that best classify a particular class. It is possible that the granules being searched overlap. It is easy to understand that covering-based algorithms search a bigger space than partition-based algorithms, and therefore covering-based rules tend to be more in quantity and more general in quality than partition-based rules. Many measures are applied when looking for the most promising attribute-value pairs. For example, PRISM [3] uses the *confidence* measure, and CN2 [5] uses the information entropy of an attribute-value pair to all classes. We also may think of the *coverage* measure of an attribute-value pair given a certain class.

Some measures that used for partition-based and covering-based algorithms are listed in Table 3.

Top-down algorithms start the search from the 0-conjunctor granule, then heuristically search down for the most promising attributes (partition-based) or attribute-value pairs (covering-based) to restrict the granule, until the consistent classification solutions are found. Some top-down criterion are based on local optimization. An important feature of ID3-like algorithms is that when splitting a node, an attribute is chosen based on only information about this node, but not on any other nodes at the same level. The consequence is that in the decision tree, different nodes in the same level may use different attributes, and moreover, the same attribute may be used at different levels. Some top-down criteria are based on global optimization, which chooses the attribute in favour of all nodes at the same level. One example is the *k*LR algorithm [23], which can construct a decision tree and evaluate the accuracy level by level.

Bottom-up algorithms start the search from the $|\mathscr{D}|$-conjunctor granules, i.e., the individual objects in the information table. The idea of bottom-up algorithms is that if all the finer granules are consistent solutions, then they are trivial solutions and can be described by their coarsened granule inductively. The AQ algorithm [10] is a classic example of bottom-up algorithms.

Another heuristic for efficient search is that instead of searching for the consistent solutions, one can search for the satisfactory solutions. **Pre-pruning methods** are used by top-down algorithms, that prematurely halt the search by meeting a predefined threshold. For example, ASSISTANT [4] and *k*LR [23] use an *accuracy* threshold as a cutoff criterion. CN2 [5] uses the Laplacian function. One can also use a *generality* threshold, which indicates the size of the granule. When the generality of a certain granule is too small to be significant as a good solution, one may ignore it. **Post-pruning methods** consist of three parts. First, grow a decision tree or decision rules for the data. Second, prune from the tree/rules a sequence of subtrees/sub-rules. Finally, try to select from the sequence of subtree/sub-rules which estimates the true regression function as best as possible. The examples of post-pruning algorithms are CART [2] and C4.5 [12]. According to the three steps, the space complexity for constructing the tree is not saved, but only the rule presentation is simplified and coarsened.

Some measures that used for different pruning methods are listed in Table 4.

Table 3. Measures for partition-based and covering-based algorithms

Some partition–based measures :

$$\begin{aligned} Entropy(a) &= -\sum_{v\in V_a} p(a=v) \sum_{c_i\in V_{class}} p(class=c_i|a=v)\log(class=c_i|a=v), \\ &= -\sum_{v\in V_a}\sum_{c_i\in V_{class}} p(class=c_i,a=v)\log(class=c_i|a=v), \\ Gini(a) &= 1-\sum_{v\in V_a} p(a=v) \sum_{c_i\in V_{class}} p^2(class=c_i|a=v), \end{aligned}$$

Some covering–based measures :

$$\begin{aligned} confidence(class=c_i|a=v) &= p(class=c_i|a=v), \\ coverage(a=v|class=c_i) &= p(a=v|class=c_i), \\ Entropy(a=v) &= -\sum_{c_i\in V_{class}} p(class=c_i|a=v)\log(class=c_i|a=v). \end{aligned}$$

Table 4. Measures for pruning methods

accuracy :	$\frac{k}{k+n}$,
Laplacian function :	$\frac{k+1}{k+n+\lvert V_{class}\rvert}$,
generality :	$\frac{\lvert m(\phi)\rvert}{\lvert U\rvert}$,

where k is the number of objects classified correctly by a formula ϕ, n is the number of objects that classified incorrectly.

4. The Interactive Granule Classification

The interactive granule classification algorithm is proposed as user-driven approach to complement the existing data-driven classification algorithms. The domain knowledge and the user preference can be profitably included in the search phases.

4.1. The interactive granule classification algorithm

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

Definition 5 *A granule X is inactive if it meets one of the following two conditions:*

(i). $X \subseteq m(class = c_i)$, *where* $c_i \in V_{class}$,

(ii). $X = \bigcup Y$, *where each Y is a child node of X and is being searched, or has been searched.*

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

The first condition indicates that a granule is inactive if it is a consistent solution of a class. One can manually set an inconsistent granule inactive if it exceeds a certain threshold that the user is interested in, for example, its generality is too small to be significant, or its confidence is high enough to the user. The second condition indicates that a granule is inactive if its finer granules being searched can cover it. Though, one can still manually set such a granule active, if he/she wants to see the measures of other finer granules.

The induction process of the interactive granule classification is briefly outlined in Figure 2 (refer to [22]), with an information table as the input, and a *granule tree* as the output. The name, a granule tree, is derived from the fact that the classification results are searched from a granule network, and can be arranged into a tree structure.

Two important issues of the algorithm are the evaluation of the fitness of each basic concept, and the evaluation of the activity status of each active granule. The measures discussed in the last section can be used to define fitness/activity functions. The measure does not need to be fixed. As a result, different measures can be used at different levels of construction.

4.2. Implementation

An interactive user-driven classification system has been implemented using the concepts of granule network. The graphical user interface shown in Figure 3 contains the constructed granule tree, shows on the central left in a tree-view format, and several sub-forms:

- The form on the left upper corner allows the user to input a target dataset for classification. User can decide either to use random leave-out method, or 5-cross validation

Set U as the root node of a granule tree at the initial stage.

Set the status of U as active.

While the active granule(s) is available and the user wants to continue, do:

(1) Select the active granule N.

(2) Select the basic concept $bc = (a = v, m(a = v))$ with respect to N.

(3) Modify the granule tree by adding the granule $N \cap m(a = v)$ as a new node, connecting it to N by an arc, and labelling it by $a = v$.

(4) Set the activity status of the new node.

(5) Update the activity status of N.

Figure 2. An algorithm for constructing a granule tree

method to divide the dataset into two parts. One part is for constructing classification rules and trees, the other part is for testing the accuracy of the learnt rules and granule tree.

- The right upper corner is the form called basic-concept-infor, that shows the basic concepts for a selected active node. At the initial stage, it shows the information about all the basic concepts for the root. Users can add the interested node to the granule tree, on the left, according to any measurements listed. The basic concepts and their measures are listed by text-based table that can be sorted by any measure, and the graphic-based chart that can help the visualization. These two representations are interchangeable.

- The right central form is the form called selected-branch-infor, that shows the branch information having the selected node as its leaf. At the initial stage, it shows the evaluations of the root with respect to generality, confidence, coverage and entropy. While an active node in the granule tree, on the left, is highlighted, the information of the branch, from the root to the active node, is shown here. Users can manually set a branch a reasonable label in this form. They can also remove a branch from the granule tree if it seems of no good. The branch information can be visualized by a graph, too.

- The tree-report form that below the selected-branch-infor is to report the updating accuracy of the classification. The value of "Processed" tracks the percentage of the objects covered by any rules. The value of "Accuracy" tracks the percentage of the correctly classified objects that covered by consistent solutions and/or manually-set solutions, while the value of "SureClassified" tracks the percentage of the correctly classified objects that covered by consistent solutions only. The values are updated when a new node is added in the basic-concept-infor form, a label is manually set for a branch, or a branch is deleted in the branch-infor form.

- The left bottom form shows all the generated classification rules, and the right bottom form shows the test results. The classification rules can be evaluated while they are fired to classify the instances in the testing test. One does not need to complete the whole training process before doing the test. The test process can be carried out while the user feels like to do it. And similarly, the classification process can be stopped manually while the accuracy of the testing result is satisfiable.

Suppose we use 5-cross validation method to classify and test the instances in Table 1. For the first run, the top 18 instances (top 80%) are for training, and the rest 20% are for test. The measures of all the basic concepts for the root are calculated. The measures of generality, confidence, coverage and entropy with respect to the covering-based algorithms are listed in Figure 3, and the measures of entropy and Gini with respect to the partition-based algorithms can be viewed and utilized under another tab. The information can be visualized by various charts.

Suppose one uses the information of attribute-value pairs for classification, and the granule $m(D = d_1)$ is selected in favour of its minimum entropy value, maximum generality, or highest confidence and coverage of class '+'. In this case, we have two active granules U and $D = d_1$. Suppose the user is still interested in the universe, he/she may be willing to investigate some other basic granule to start the search. For example, the granule $m(D = d_2)$ is selected in favour of its second smallest entropy, and/or strong confidence and coverage of class '-'. And then, suppose for the granule $m(D = d_1)$, the attribute A is chosen, then all the possible values of this attribute, i.e., granules $m(A = a_1), m(A = a_2)$ and $m(A = a_3)$ will be attached to the granule tree simultaneously (shown in Figure 3).

The granules $m(D = d_1 \wedge A = a_1)$ and $m(D = d_1 \wedge A = a_3)$ are inactive with respect to condition (i). While the active granule $m(D = d_1 \wedge A = a_2)$ is highlighted, the basic-concept-infor form shows all the basic concepts that are possibly conjuncted to it, and the branch-infor form shows the measures of the conjunctor itself. It can be easily verified from the branch-infor that 75% out of 4 instances in this granule belong to class '+', which covers 27% of the positive instances. One may decide to set this granule to be positive. If so, the branch will be labelled as '+', and a semi-rule $D = d_1 \wedge A = a_2 \Rightarrow class = +$ is added to the rule set, the accuracy of the classification will be updated. Then, after setting the granule $m(D = d_1 \wedge A = a_2)$ be positive, a careful user may notice all the three possible values of A are all labelled as positive. In this case, one may decide to delete the finer granules, and only keep the coarse granule $m(D = d_1)$ and set it as positive. By doing this, all

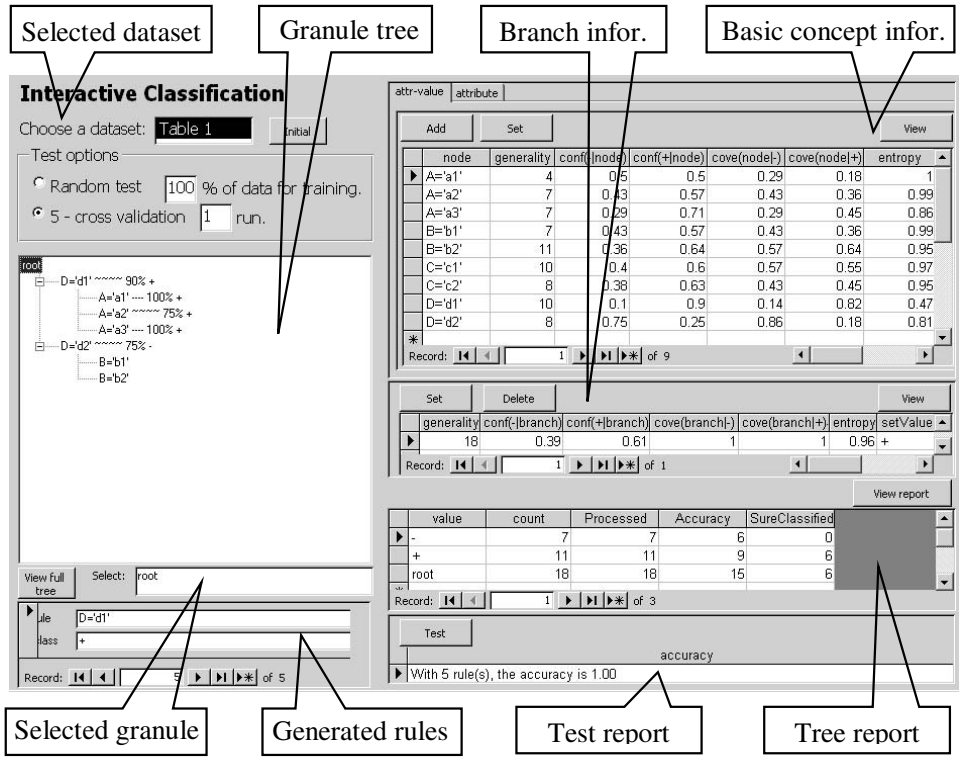


Figure 3. The GUI of interactive user-driven classification system

the information including the tree structure, the rule set, the measurements are updated accordingly.

Suppose the user has set the granule $m(D = d_1)$ be positive, and the granule $m(D = d_2)$ be negative. By firing these two rules, the user notice the accuracy for testing reaches 100%. In this case, the user may decide to stop the training process, or he/she may want to finish training process till there is no active granule available.

We should note that an interactive system can perform differently for different people. For example, Figure 4 presents two different granule trees constructed by different strategies, having all the instances are used for training. The tree on the left applies the covering-liked strategy, namely, the user searches for the rules that describe the predict the positive instances first till all the positive training instances are covered by one or more rules, then the negative instances. The tree on the right applies the partition-liked strategy. In the interactive user-driven classification system, one can construct different trees to meet his/her own requirement. The effectiveness of classification is highly dependent on the skill and background knowledge of the user. An expert may find richer results than a novice. Even different experts using different domain knowledge, or processing different inquiries can introduce different rule sets. Hence, evaluating this interactive system is not easy.

5. Experiment

Many real databases have been tested to verify the effectiveness of the proposed interactively user-driven classification method.

5.1. Partition or covering

It is noticed that to achieve a whole success, namely to describe and classify all the classes, partition-based granule trees are easy to be constructed. In scalable datasets, the number of attributes is a source of increasing difficulty. Whenever an expecting attribute is picked, all of its possible values are added to the granule tree, the corresponding granules are pair-wisely disjointed. This approach ensures no portion will be missed for classification. One can apply the depth-first fashion to explore each branch sequentially, or apply the breath-first fashion to explore the granules at the same level. The mixture method is also allowed.

If one needs to achieve a partial success, namely to describe and classify one particular class, then covering-based granule trees are more suitable for this purpose. In

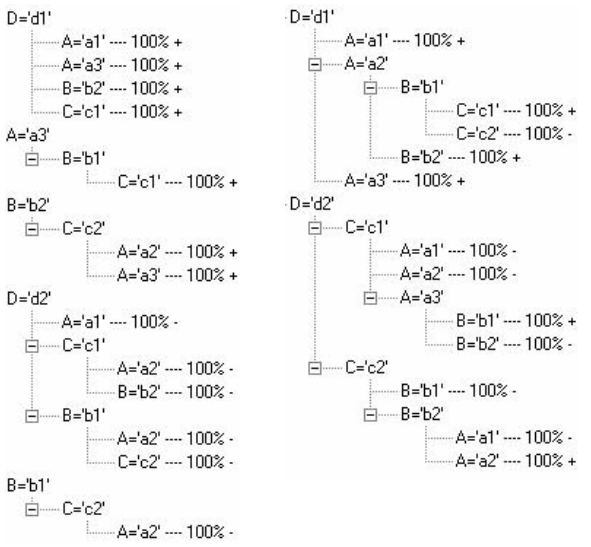
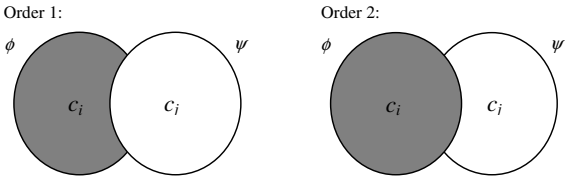


Figure 4. Two different granule trees of Table 1

this case, the active attribute-value pairs and their measurements are more under the concern. Normally, the progression of investigation moves from the most promising branch to the less promising ones.

5.2. Ordering and accuracy

Another observation is that, for covering-based granule trees, the order of adding granules to the tree affects the accuracy of classification. This happens when the inconsistent solutions are manually labelled and added.



real label of $m(\phi \wedge \psi) \setminus$ order	$\phi \wedge \psi$	$\psi \wedge \phi$
c_i	remain	$\uparrow$
c_j	$\uparrow$	remain

real label of $m(\phi \wedge \psi) \setminus$ order	$\phi \wedge \psi - \psi$	$\psi \wedge \phi - \psi$
c_i	remain	$\downarrow$
c_j	$\downarrow$	remain

Figure 5. Different orders cause different accuracies

We use a pair of overlapped granules for a simple illustration. Suppose two granules are sequentially added as two partial solutions: (1) $\phi \Rightarrow class = c_i$, (2) $\psi \Rightarrow class = c_j$. Then the granule $m(\phi \wedge \psi)$ is first labelled as c_i and then labelled as c_j. If these two granules are added in a reverse order, then the granule $m(\psi \wedge \phi)$ is finally labelled as c_i, as shown in Figure 5. Different orders cause the measures of accuracy and sure classification change consequently. If the real labels of the instances in the granule $m(\phi \wedge \psi)$ are mainly in c_i, then the second order can effectively increase the accuracy. If the real labels are mainly in c_j, then the first order is better. Suppose the user wants to remove the unsatisfying granule ψ from the tree. In the case that the real labels of the granule are mainly in c_j, after deleting ψ from the first ordering, (denoted as $\phi \wedge \psi - \psi$), the accuracy decreases. If the real labels are mainly in c_i, then the accuracy decreases when the second ordering has been processed. Unfortunately, one cannot predict the major label of the intersection to decide the ordering. Therefore, it is possible that two trees process the same rule sets, but have different degrees of accuracy.

5.3. Pre-pruning or post-pruning

For real datasets, error data and inconsistency are inevitable. Although the longer granule trees always lead to the higher accuracy, users can hardly understand and manipulate long rules. Pruning and pre-maturing methods should be considered for real dataset classification. The interactive classification supports both pre-pruning and post-pruning, and provides the maximum flexibility to add, delete, stop, continue and undo.

Figure 6 shows the resulting trees of the dataset, discrx, chosen from UCI machine learning repository [14]. SGI's MLC++ utilities 2.0 [13] is used to discretize the continuous attribute sets. The middle granule tree is almost identical to the left C4.5 tree generated by Weka software [19]. The right granule tree uses a user chosen attribute set. These three trees achieve the same accuracy. Many different trees can be constructed. We need to emphasize once again that the complexity of the construction might be high according to the user interaction, revision and thinking. However, this construction process does improve the understanding of the classification subject itself.

6. Conclusion

The contributions of this paper are twofold. First, a framework of using a granule network for classification tasks is developed. A classification problem can be modelled as a search for a partition or a covering defined by a set of attribute-values in a granule network. The entire search space is studied. Second, an interactive user-driven classification method is proposed for the classifier construction. The proposed approach provides more free-

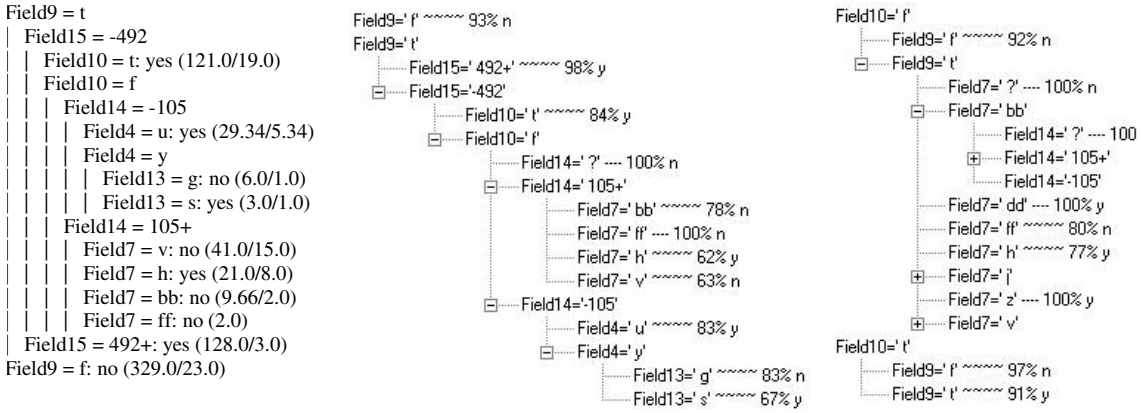


Figure 6. Weka tree and two granule trees of the dis-crx dataset

dom of choices on heuristics and measures according to the user's needs. The process is based on the idea that the classification task can be more useful if it carries with user preference and user interaction. It overcomes the limitation of most of the existing data-driven classification algorithms that fix on one heuristic to decide where to classify and how to classify.

References

[1] Ankerst M., Elsen C., Ester M. and Kriegel H.P., Visual classification: an interactive approach to decision tree construction, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 392-396, 1999.

[2] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., *Classification and Regression Trees*, Wadsworth Int. Group, Belmont, California, USA, 1984.

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

[4] Cestnik, B., Kononenko, I., and Bratko, I., ASSISTANT 86: a knowledge-elicitation tool for sophisticated users. *Proceedings of the 2nd European Working Session on Learning*, Yugoslavia, 31-45, 1987.

[5] Clark, P. and Niblett, T., The CN2 induction algorithm, *Machine Learning*, **3**-4, 261-283, 1989.

[6] Finin, T. and Silverman, D., Interactive classification of conceptual knowledge, *Proceedings of the First International Workshop on Expert Database Systems*, 79-90, 1986.

[7] Han, J., Hu, X. and Cercone, N., A visualization model of interactive knowledge discovery systems and its implementations, *Information Visualization*, 2(2), 105-125, 2003.

[8] Hatano, K., Sano, R., Duan, Y., and Tanaka, K., An Interactive classification of Web documents by self-organizing maps and search engines, *Proceedings of the 6th International Conference on Database Systems for Advanced Applications*, 19-22, 1999.

[9] Lin, T.Y., Granular computing, *Announcement of the BISC Special Interest Group on Granular Computing*, 1997.

[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] 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.

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

[13] SGI's MLC++ utilities 2.0: the discretize utility. http://www.sgi.com/tech/mlc

[14] UCI Machine Learning Repository, http://www1.ics.uci.edu/∼mlearn/MLRepository.html

[15] University of British Columbia, *Towards Interactive Data Mining*, http://www.cs.ubc.ca/r̃ng/mitacs/

[16] Wang, Y.X., On cognitive informatics, *Proceedings of ICCI'02*, 34-42, 2002.

[17] Wang, Y.X. and Liu, D., On information and knowledge representation in the brain, *Proceedings of ICCI'03*, 26-29, 2003.

[18] Wang, Y.X., On autonomous computing and cognitive processes, *Proceedings of ICCI'03*, 3-4, 2004.

[19] Weka 3 - Data Mining with Open Source Machine Learning Software in Java, http://www.cs.waikato.ac.nz/ml/weka

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

[21] Yao, Y.Y., Concept formation and learning: a cognitive informatics perspective, *Proceedings of the 3rd IEEE International Conference on Cognitive Informatics*, 42-51, 2004.

[22] Yao, Y.Y. and Yao, J.T., Granular computing as a basis for consistent classification problems, *Communications of Institute of Information and Computing Machinery*, **5**(2), 101-106, 2002.

[23] 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.