Granulation using Clustering and Rough Set Theory & its Tree Representation

Girish Kumar Singh, and Sonajharia Minz

***Abstract*—**Granular computing deals with representation of information in the form of some aggregates and related methods for transformation and analysis for problem solving. A granulation scheme based on clustering and Rough Set Theory is presented with focus on structured conceptualization of information has been presented in this paper. Experiments for the proposed method on four labeled data exhibit good result with reference to classification problem. The proposed granulation technique is semi-supervised imbibing global as well as local information granulation. To represent the results of the attribute oriented granulation a tree structure is proposed in this paper.

***Keywords*—**Granular computing, clustering, Rough sets, data mining.

I. Introduction

THE concept of Granular computing first appeared in 1979 under the name of information granularity in L.A. Zadeh's pioneer paper [19]. The term "granular computing" came to life with a suggestion of T. Y. Lin in the discussion of BISC Special Interest Group on Granular Computing [23]. Granular computing (GrC) is an umbrella term to cover many theories, methodologies, techniques, and tools based on core of granules for problem solving [22]. The ideas of granular computing have been investigated in artificial intelligence through the notions of granularity and abstraction. Hobbs proposed a theory of granularity [6], which is similar to the formulation of theory of rough sets. The theory perceives and represents the universe of problem under various grain sizes, only to abstract those things that serve our present interests. The ability to conceptualize the world at different levels of granularities and to switch among these granularities is fundamental to human intelligence and flexibility. This enables us to map the complexities of real world into computationally tractable simpler theories. Since then, the concept of Granular computing is rapidly developing with growing interest in the topic. A number of methods and models of granular computing have been proposed and studied. Basic concepts of granular computing are granules such as subsets, classes, and clusters of a universe. GrC is a new term in problem solving in computer science and may be viewed more on the conceptual rather than technical level. Although the term is relatively new, the basic notions and principles of granular computing occur under various guises in a wide variety of fields [16, 9]. Belief functions, artificial intelligence, cluster analysis, chunking, data compression, databases, decision trees, divide and conquer, interval computing, machine learning from examples, structure programming, quantization, quotient space theory, and rough set theory are some example fields. Much research has been conducted recently in various aspects of granular computing [18]. Following are three approaches for granulation in the literature [15],

- Zadeh's formulation: A general framework for granular computing based on fuzzy set theory was introduced by Zadeh [22]. Here the granules are constructed and defined based on the concept of generalized constraints. The Relationships between granules are represented in terms of either the fuzzy graphs or the fuzzy if-then rules. The associated computation method has also been known as computing with words (CW) [14, 20].
- Powlak's Rough Set formulation: With granulation of the universe, one considers the elements within a granule as a whole rather than as individuals [19]. The loss of information due to granulation implies that some subsets of the universe can only be approximately described. The theory of rough sets mainly deals with the approximation aspect of information granulation [11].
- Set theoretic: Set theoretic model [15] was proposed using a binary relation over the power set of the objects to represent granulation. Each granule represents the concept, as each element of the granule is an instance of the concept. Yao has also presented a Partition Model for Granular Computing [17] which is basically an extensive model of set-theoretic framework.

In all the literature surveyed by the authors granular computing in general has been presented as a model similar to the ability of humans to perceive the world at different granularity and to change granularities in problem solving. However, there appear to be a need to implement the general models. It has also been suggested that *each field may develop its theories and methodologies in isolation* [17]. Therefore, implementations of other existing models are not considered for comparison. The literature also presents various aspects, methods and frameworks for granular computing without separating the process of granulation from its end use. This paper presents a novel method for granulation based on

Manuscript received November 30, 2006.

Girish Kumar Singh is with the School of Computer & Systems Sciences of Jawaharlal Nehru University New Delhi-11067, USA (phone: 91-9871050492; e-mail: gkrsingh@gmail.com).

Sonajharia Minz is with the School of Computer & Systems Sciences of Jawaharlal Nehru University New Delhi-11067, USA (phone: 91-9868807594; e-mail: sonaminz@mail.jnu.ac.in).

clustering and Rough Set Theory (RST) which is a set theoretic approach. To present the result of the granulation process this paper proposed *Granulation Tree*. Rest of the paper is organized as follows: section 2 gives the preliminaries. Section 3 gives the proposed method, section 4 presents the proposed tree structure, section 5 presents the result and analysis and section 6 conclude the paper.

II. Preliminaries

A. Information Granule

A granule may be interpreted as one of the numerous small particles forming a larger unit. Collectively, they provide a representation of the unit with respect to a particular level of granularity. Thus a granule may be considered as a localized view or a specific aspect of a large unit. Information Granulation involves partitioning a class of objects into granules i.e. clumps of objects, which are drawn together by indistinguishability, similarity or functionality [21, 8]. The phrase "drawn together by indistinguishability, similarity, or functionality" has been developed as the concept of binary granulation [2] as defined below.

Binary Granulation: Binary granulation is the association of an object $p \in U$ with the granule $B(p) \subseteq U$ (neighborhood), where p varies through all objects of the universe U. This association is mapping $U \rightarrow 2^U$, called as basic or binary granulation (BG).

Binary Granulation: A relation R defined by $R = \{(p, v) \mid p \in B(p), v \in U\}$ is the Binary Relation (BR) defined by the binary granulation (BG).

Binary Neighborhood System: The collection $\{B(p) \mid p$ varies through $U\}$, where B(p) the granule of p also refer as neighborhood of p, is called the Basic Neighborhood System or Binary Neighborhood System (BNS).
The geometric and algebraic views of binary granulation are represented by BNS and BR respectively.

B. Clustering

Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have higher similarity in comparison to objects in other clusters [4]. Cluster analysis has been widely used in various disciplines such as pattern recognition, computer vision, and data mining [7]. There are numerous types of clustering algorithms based on partitioning, hierarchical, nearest-neighbours, fuzzy, density-based, grid-based approaches etc. In this study the concept of density-based clustering method [3] and specifically DBSCAN [3] is used, to obtain the natural intervals of attribute values.

Density Based Clustering: The general idea density based clustering approach is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, that is, for each data point within a given cluster, the neighborhood of given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outlier) and also to discover clusters of arbitrary shape. Some examples of the density based method are DBSCAN [3], DENCLUE [5], GDBSCAN [12] and OPTICS [1].

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that is designed to discover clusters and noise in a spatial data set. The algorithm uses two parameters: *Eps* and *MinPts*. Two objects are said to be *neighbours* of each other if the distance between them is less than or equal to some fixed measure. The neighbourhood of an object within a radius *Eps* is said to be *Eps-neighborhood* of the object. If the Eps-neighborhood of an object contains at least *MinPts* objects, then this object is called a *core object*. A point *p* is *directly density-reachable* from a point *q* w.r.t. Eps and MinPts if 1) *p* is in *Eps-neighborhood* of *q* and 2) *q* is a core point. A point p is *density-reachable* from a point q with respect to *Eps* and *MinPts* if there is a chain of points $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

To find a cluster, DBSCAN starts with an arbitrary object *o* in the data set. If o is a core object with respect to some values of *Eps* and *MinPts*, a cluster with *o* as the core object is created. The algorithm continues growing the cluster by adding to the cluster all the density reachable objects from the core object. The technique to determine the parameter *Eps* for a given *MinPts* is also presented in [3]. However the value of *MinPts* is generally 4.

C. Rough Set Theory

Rough Set theory [10], introduced by Pawlak in early 1980's, is a technique for dealing with uncertainty and for identifying cause-effect relationship in databases as a form of data mining and database learning. It has also been used for improved information retrieval and for uncertainty management in relational databases.

A pair S = (U, A) is called *information system,* where U is a non-empty finite set of objects called the universe and A is a non-empty finite set of attributes and the set V_α is called the value set of the attribute α, such that $\forall\ \alpha \in A$, $\alpha: U \rightarrow V_\alpha$. A *Decision system* is any information system of the form $S = (U, A \cup D)$ where $A \cap D = \phi$, D is set of decision attributes and A the set of conditional attributes.

The indiscernibility relation is fundamental to rough sets. Indiscernibility relation partitions the domain into equivalence classes. For any $B \subseteq A$ there is associated an *equivalence relation* $IND_S(B)$:

$$IND_S(B) = \{(x, x') \in UXU \mid \alpha \in B\ \alpha(x) = \alpha(x')\} \qquad (1)$$

If $(x, x') \in IND_S(B)$, then objects x and x′ are indiscernible from each other by attributes from B. $IND_S(B)$ is called the *B-indiscernibility* relation. The equivalence classes of the B-indiscernibility relation are denoted by $[x]_{B.}$

Lower and upper approximation regions of a concept help to distinguish between certain and possible (or partial) inclusions of objects from the universe U, respectively. This partitioning may increase or decrease the granularity of a domain. This may help in grouping together the objects that are considered indiscernible for a given application. For any $X \subseteq U$ a description of X can be obtained using only the information contained in some $B \subseteq A$ by constructing the B-lower approximation and B-upper approximation of X represented as $\underline{B}X$ and $\overline{B}X$ respectively. Where $\underline{B}=\{x:[x]_B \subseteq X\}$ *and* $\overline{B}=\{x:[x]_B \cap X \neq \phi\}$.

The approximation regions defined using the equivalence classes of the indiscernibility relation. The objects in $\underline{B}X$ with certainty are classified as members of X on the basis of the knowledge in B, while the objects in $\overline{B}X$ are classified as possible members of X on the basis of the knowledge in B. The set $BN_B = \bar{B}X - \underline{B}X$ is called the *boundary region* of X, which consists of those objects that cannot decisively be classified into X on the basis of knowledge in B. The set $U - \bar{B}X$ is called the *B-outside region* of X and consist of those objects, which are with certainty classified as not belonging to X on the basis of knowledge in B. A set is said to be *rough* (crisp) if the boundary region is non-empty (empty).

Rough Set Theory determines degree of attribute dependency and significance of attributes. For a given set of conditional attributes B, we can define the *B-positive region* $POS_B(D)$ in the relation IND(D) is defined as,

$$POS_B(D) = \bigcup\{\underline{B}X : X \in D^*\} \qquad (2)$$

where, D* is the partition obtained by the equivalence class relation IND(D). The positive $POS_B(D)$ region contains all the objects in U that can be classified without any error into distinct classes defined by IND(D), based only on information in the relation IND(B). Greater the cardinality of $POS_B(D)$ higher the significance of an attribute with respect to D.

The Rough membership function expresses how strongly an element x belongs to the rough set X in view of the information about the element expressed by the set of attributes B. The Rough membership function can measure the significance of an attribute and is defined by,

$$\mu_X^B(x) = \frac{card\ (X \cap [x]_{IND(B)})}{card\ ([x]_{IND(B)})} \qquad (3)$$

III. Proposed Method

The proposed method comprises of two phases. In the first phase the natural partition of attribute values are obtained and in the second phase granules are formed using the partitioned obtained in first phase.

Phase I

The basic neighborhood system is implemented using the clustering technique as attribute level in the first phase. The first phase consists of two steps: first step obtain partition the attribute values into the natural partition using the density based clustering approach and in the second step the partitions are refined using the Rough Set Theory. The term natural partition means, partition with clear interval. The partition also includes blocks of arbitrary number of attribute values depending on the density. Three threshold values *MaxPoint, MinPoint,* and *MaxLength* are used to categorize the outcomes of the partitioning of attribute values using DBSCAN algorithm. *MaxPoint* and *MinPoints* are used for setting the upper and lower limits respectively to the number of attribute values in an interval, and *MaxLength* is a limit for the purpose of controlling the size of an interval. *MaxPoint* is used to ensure that an interval does not have too many values so that it would not loss the natural view of the partitions. *Minpoint* is used to ensure that an interval must have a minimum number of attribute values to control the number of intervals. Some time it is also required to check the length of the interval even if it is less dense to control the loss of information. *MaxLength* is used to fix the maximum length of attribute intervals. The resultant intervals of the attribute values may be of the following three types based on the number of object in the partition and the length of the partitions,

i) **Normal:** An interval I is said to be normal if
MinPoint ≤ Card(I) ≤ *MaxPoint,* AND Range(I) ≤ *MaxLength.*
i.e. interval I is of allowable length for an interval and the number of attribute values are within limits.

ii) **Large:** An interval I is said to be large if
Card(I) > *MaxPoint* OR Range(I) > *MaxLength* OR both.
i.e. the interval is big in three respect, the number of attribute values is greater than specified limit, or length of the interval is greater than then the allowable length or both.

iii) **Small:** An interval I is said to be small if
Card(I) < *MinPoint.*
i.e. the interval contains few attribute values.

In the refinement step the optimized number of intervals of partition is achieved by repeated splitting of large intervals and merging of small intervals. One of the objectives of refining the partitions is either to maintain the significance of the attribute or possibly enhance it. The RST concept $POSa_i(D)$ provides a measure of significance of an attribute a_i. The positive region includes the set of all objects which can be correctly classified with respect to the decision attribute D based on the values of the attribute a_i. By merging a small intervals to the nearest suitable interval and by splitting a large interval, the aim is to maximize the cardinality of $POSa_i(D)$. This helps maximize the indistinguishability between the objects in binary neighborhood system. By considering all the relevant attribute POS(D) helps to obtain good granule.

To maximize the cardinality of $POSa_i(D)$ we have to refine the partitions in such a way that the maximum number of objects are classified by the attribute a_i as classified by D. To achieve this, a rough membership function corresponding to the partitions with respect to class labels is defined. Let the dataset U includes objects of m classes, say $\{c_1, c_2, c_3 \ldots c_m,\}$. And let the k distinct values of an attribute a_i, in ascending order be $\{v_{i1}, v_{i2}, v_{i3} \ldots v_{ik}\}$ i.e. partition $[v_{i1}, v_{ik}]$. The rough membership function of any partition $I= [v_{i1}, v_{ij}]$ of attribute a_i for a class c_p is given by

$$f(a_i, c_p, I) = \frac{Card\ (\underline{a_{i,I}} X_{cp})}{Card\ (X_{a_i,I})} \qquad (4)$$

Where $X_{a_i,I} = \{x \mid a_i(x) \in I\}$ and $\underline{a_{i,I}}X_{cp} = \{x \mid a_i(x) \in I$ and $D(x)=c_p\}$

From equation (2)

$$POS_{a_i}(D) = \underline{a_{i,I}}X_{c_1} \cup \underline{a_{i,I}}X_{c_2} \cup ...\underline{a_{i,I}}X_{c_p} ... \cup \underline{a_{i,I}}X_{c_m}$$

$$\text{and } \underline{\mathrm{a}_{\mathrm{i}}}\mathrm{X}_{\mathrm{c_p}} = \cup \underline{\mathrm{a}_{\mathrm{i},\mathrm{I}_1}}\mathrm{X}_{\mathrm{c_p}} \cup \underline{\mathrm{a}_{\mathrm{i},\mathrm{I}_2}}\mathrm{X}_{\mathrm{c_p}} ....$$

Therefore by maximizing $f(a_i, c_p, I)$ each of the $\underline{\mathrm{a}_{\mathrm{i},\mathrm{I}_1}}\mathrm{X}_{\mathrm{c_p}}, \underline{\mathrm{a}_{\mathrm{i},\mathrm{I}_2}}\mathrm{X}_{\mathrm{c_p}}$ ...... is maximized and hence $\mathrm{POS}_{\mathrm{a_i}}(\mathrm{D})$.

Phase II:

In the second phase the construction of granule is carried out. First we choose an attribute and partition the space corresponding to the partition of the attribute values. This step is repeated for all relevant attribute. Partitioning of a (sub)space is continued only when there is an improvement in the quality of the granules. To measure the quality of the granules we propose parameters two parameters namely-precision of a granule and ratio of the size of partitions. Precision of a granule G is given by

$$\mathrm{Pr}\,es(G) = \frac{MAX\{|X_{c_p}|,\quad X_{c_p} \subseteq G\ AND\ \forall y \in X_{c_p}\ D(y) = c_p\}}{|G|}$$

and, ratio of the partition is given by

$$R(G) = \frac{MIN\ \{|G_i|\}}{MAX\ \{|G_i|\}}, G_i\text{'}s\ are\ the\ partitions\ of\ G$$

, G will be split only if the value of R(G) is less than a threshold value.

Algorithm:

```
Step 1.  for each continuous attribute $A_i \in A$
         1.1  Select distinct values of $A_i$ in an array Distinct
         1.2  Sort(Distinct)
         1.3  Call DBSCAN(Distinct, Eps, MinPts)
         1.4  Refine($I_1$, $I_2$, …. )
Step 2.  for each attribute $A_i \in A$
         partition the each (sub)space based on the partition of the
         $A_i$ if partition gives the fine granule than previous one.
  Step 3.  Stop
```

Refine (I_1, I_2, ….)

```
Step 1.  while (no change in no. of partitions) do
         For each partition $I_j$
           If | $I_j$ | > MaxPoint or Range ($I_j$) > MaxLength
           then
               Temp= Cut_Point ({$v_{j1}$, $v_{j2}$, $v_{j3}$ … $v_{jk}$})
               Replace the partition $I_j$ with two partition $I_{j1}$ =[ $v_{j1}$,
                   Temp] and $I_{j2}$ =[Temp, $v_{jk}$]
           elseif | $I_j$ | < MinPoint then
               If for $I_{k'}$ either neighbour of $I_j$ MR_C($I_j$, $I_{k'}$,
                   MaxPoint, MaxLength ) = True  then  merge $I_j$ to
                   an interval $I_{k'}$
               End if
           End if
         End for
         End while
Step 2.  Stop
```

Cut_Point ({v_{i1}, v_{i2}, v_{i3} … v_{ik}})

```
1.  I = [$v_{i1}$, $v_{ik/2}$] // $v_{ik/2}$ is the middle term of {$v_{i1}$, $v_{i2}$, … $v_{ik}$}
2.  MAXRMV= Max ({ $f(a_i, c_p, I)$ }) ∀ $c_p$,
3.  for each $v_{ij}$ , j=k/2 to 1
      2.1.  I = [$v_{i1}$, $v_{ij}$]
      2.2.  Temp = Max ({ $f(a_i, c_p, I)$ }) ∀ $c_p$;
      2.3.  if Temp> MAXRMV then
                MAXRMV=Temp;
            else
                break;
4.  if (j < k/2) then return $v_{ij}$ as cut point for the partition
    else
       for each $v_{ij}$, j=k/2 to k-1
         4.1.  I = [$v_{i1}$, $v_{ij}$]
         4.2.  Temp = Max ({ $f(a_i, c_p, I)$ }) ∀ $c_p$;
         4.3.  if Temp> MAXRMV then
                   MAXRMV=Temp;
               else
                   Return $v_{ij}$ as cut point for the cluster
               End if
      5.  Stop
```

Given a labeled dataset of size *N*, with *m* classes C_1, C_2, …. C_m, and *n* attributes say *A*= {A_1, A_2 … A_n}. The algorithm employs the function *Refine* to refine the set of intervals obtained by the DBSCAN. The function *Cut_Point* is applied to a large interval which is eligible for split to find the best split point for the large partition.

The Refine ( ) applied attribute wise to the results of partition by DBSCAN. In this function if an interval is large then it will be break in two intervals and if an interval is small then it will be merged to nearest suitable interval.

To break the big interval, cut_point() function is used to find the best suitable cut point to split a big interval to maximizing Pos(D) of the attribute.

IV. Tree Representation of Granulation

It is very important to represent, store, access and display the output of any process in such a way that it is interpretable and is of use for further processing. The result of granulation must therefore be represented in such a way that the user can visualize the granules and can use them for further processing. A tree structure is proposed in this paper to represent results of an attribute oriented granulation scheme which on one hand is easy to understand and on the other hand has efficient algorithms for access and processing of the granules. In the proposed tree structure all the nodes of the tree represent information granules corresponding to the given data. The root node represents the universe. The tree is constructed by decomposing the universe based on the discrete values of the attributes, considering them one at a time. The granules get finer while traversing a path from root towards leave and coarser when traversing in the opposite direction. A granule is described by the set of ordered pair <attribute, attribute values> which is unique for each granule.

Granulation Tree Construction

"*Granulation of an object A, leads to a collection of granules of A, with a granule being a clump of points (objects) drawn together by indistinguishability, similarity, proximity or functionality*". Zadeh 1997

The root of the granulation tree represents the universe a single clump consisting of all objects. From the relational algebra for a set of attributes $\{A_1, A_2 \ldots A_m\}$with $n_1, n_2 \ldots n_m$ as number of discretized values respectively, the number of possible unique tuples based on these attribute is equal to $n_1 * n_2 * \ldots * n_m$. Consider one attribute to obtain the partition of the universe into subspaces. Partitioning of the space is carried out on the basis of the discrete values of the attribute. Each subspaces corresponding to an attribute value is represented by a child node of the root node. The descendents at depth-1 represent the information granules of level-1. The process of granulation is continued by considering one unique attribute for each node at a given level, thus recursively partitioning the subspace. The partition of each subspace at depth i is represented by the set of all its child nodes at depth i+1 i.e. the granules at level i+1. Each child node represents a finer granule than its parents. An attribute once used for granulation is not reconsidered. A granule at a node at level-k may be denoted by the granule descriptor, a set of k ordered pairs of <attribute, attribute values> along the path from root to the node. A tree of depth k may represent all granules of level less than or equal to granules of level-k of the universe. Therefore given a set of m attributes, a granulation tree of depth m can represent all granules of level-k of the universe, $1 \leq k \leq m$. The set of granules represented by the all the leaf nodes is unique irrespective of the order in which they are considered.

The growth of a granulation tree can be monitored to control the number of granules. For labeled data a granule containing the objects of same class should not be further partitioned. In case of unlabeled data a threshold value of average dissimilarity between the objects may be used to terminate granulation tree construction.

Theorem: Given a universe with m attributes the number of granules at level m is same whatever may be the order of attribute selection for partitioning the space. Moreover the granule descriptor for each granule is unique.

Proof: Let $n_1, n_2 \ldots n_m$ be the number of discretized values for the attributes $A_1, A_2 \ldots A_m$ respectively. Therefore the number of possible unique tuples based on these attribute is equal to $n_1 * n_2 * \ldots * n_m$ irrespective of the order of attributes considered to represent the tuple.

Now we show that set of granule descriptor is same. Let $\{d_1, d_2 \ldots d_N\}$ be the set of descriptors of N granule when attributes are consider for partition in $A_1, A_2 \ldots A_m$ order. Where $N= n_1 X n_2 X \ldots X n_m$ and $d_i= \{< A_1, V_{i1}>, < A_2, V_{i2}> \ldots < A_m, V_{im}>\}$, $V_{ij} \in \text{dom}(A_j)$.

Now suppose the order of attribute has been change and it is $A_2, A_1 \ldots A_m$ then granule represented by d_i is will be represented by the $\{< A_2, V_{i2}>, < A_1, V_{i1}> \ldots < A_m, V_{im}>\}$. Similarly for every d_i, one can have different sequence of descriptor using all m attribute in a different granulation tree, where the order of selection of attribute is altered. However the set of m descriptor will remain unique for a given granule.

V. Result & analysis

Proposed scheme for granulation has been implemented in C language. The experiments were carried out on the following four Labeled datasets obtained from the UC Irvine ML repository [13]. To check the feasibility of the approach the data sets considered are relatively small in size. Two out of four datasets, namely iris and pid, are described by continuous attributes where ion and hea have both the continuous and the discrete attributes. A detailed description of the data sets is presented in Table I. The properties Cont/Mix describe the attribute types of the data set. Cont indicates all the attributes are continuous and Mix denotes that some attributes are continuous and some are nominal.

1. Iris Plants dataset (iris),
2. Johns Hopkins University Ionosphere dataset (ion),
3. Statlog Project Heart Disease dataset (hea),
4. Pima Indians Diabetes dataset (pid).

TABLE I
Data Set Description

Properties	Datasets			
	Iris	Ion	Hea	Pid
No. of Examples	150	351	270	768
No. of Classes	3	2	2	2
No. of Attributes	4	34	13	8
No. of Cont. Attributes	4	32	6	8
Cont/Mix	Cont	Mix	Mix	Cont

The result of proposed method has been shown in Table II. The result has been shown under four headings- 1) number of Granules 2) number of Granules having one type of object 3) number of Granules having more than one type of object and 4) number of object which cause for result 3. The number of granule gives how many granules are obtained in the granulation process; minimum number of granule is required provided granules are fine granules. A fine granule can be measured by the type of object it have. If a granule has one type of object then it is a fine granule. Maximum the number of granules containing the one type of object shows good granulation result. Some time it may not be possible to get fine granules or a big granule can't be a fine granule because of some few other type of object in that granule. These two things can be measured by "No. of Granules having more than one type of object" and "No. of object which cause previous one". Result has also shows the percentage of fine granule of total granules. The tree representation of granulation result for Iris data has been shown in the Fig. 1.

TABLE II
GRANULATION RESULT

Result	Datasets			
	Iris	Ion	Hea	Pid
1. No. of Granules	23	89	153	209
2. No. of Granules having one type of object	18	88	151	145
3. No. of Granules having more than one type of object	5	1	2	64
4. No. of objects which cause the result 3	5	1	2	128
5. % of fine granule	78	99	98	70

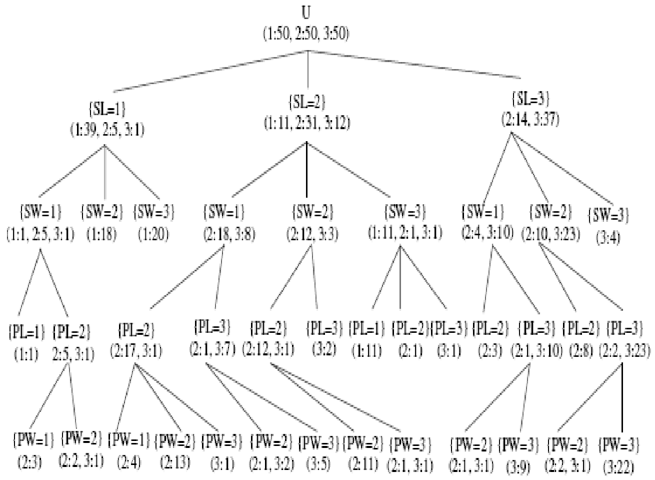

Fig. 1 Tree representation of Granulation of Iris Data

VI. CONCLUSION

The technique proposed in this paper for granular computing is semi-supervised using clustering yet imposing controls during granule formation. In the proposed method the natural intervals of attribute values (local) are obtained which maximize the mutual class-attribute interdependence (a global phenomenon) to generate a possibly minimum number of granules. A high percentage of good granules indicate the goodness of the scheme. This may enable determine thresholds to define the goodness of a granule.

The scheme does not include measure to test the goodness of granules at any desired point. The attribute relevance analysis is not embedded in the algorithm. Granulation based on only relevant attributes may further improve the performance by reducing the number of granules and also may increase the percentage of good granules. Some heuristics may also be used to obtain good granules of variable size. This paper presents a Granulation Tree, which is used represent, the granulation result. The application of this granulation tree should be seen in the data mining techniques.

REFERENCES

[1] Ankerst M., M. Breunig, Kriegel H.P., and Sander J., "OPTICS: Ordering Points to Identify the Clustering Structure", In Proceeding ACM SIGMOD, International Conference on Management of Data (SIGMOD'99), Philadelphia, PA, pages 49--60, 1999.

[2] Chiang I-Jen, Lin T. Y., and Liu Y., “Table Representations of Granulations Revisited”, Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference, RSFDGrC 2005, Regina, Canada, pp. 728-737, August 31 - September 3, 2005.

[3] Ester M., Kriegel H.-P., Sander J., and Xu X., “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise” In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD’96), Portland: Oregon, pp. 226-231, 1996.

[4] Han, J. & Kamber, M. (2001). Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann.

[5] Hinneburg A., and Keim D. A., “An Efficient Approach to Clustering in Large Multimedia Databases with Noise”, in Proceeding of International Conference on Knowledge Discovery and Data Mining (KDD98), pages 58-65, August 1998.

[6] Hobbs, J.R. “Granularity”, Proceedings of the 9th International Joint Conference on Artificial Intelligence, 432-435, 1985.

[7] Jain A.K. and Dubes R.C. (1988) “Algorithms for Clustering Data” Prentice Hall, Upper Saddle River: New Jersey.

[8] Lin, T. Y., “Granular computing on binary relations I: data mining and neighborhood systems”, Rough Sets In Knowledge Discovery, Springer-Verlag, pp 107-140, 1998.

[9] Lin, T.Y. “Granular computing”, LNCS 2639, Springer, Berlin, 16-24, 2003.

[10] Pawlak, Z. "Rough sets." International Journal of Computer and Information Sciences 11 (1982): 341-356.

[11] Pawlak, Z. Granularity of knowledge, indiscernibility and rough sets, Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 106-110, 1998.

[12] Sander J., Ester M., Kriegel H. P., and X. Xu, “Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications”, Data Mining and Knowledge Discovery, Kluwer Academic Publishers, Vol. 2, No. 2, 1998.

[13] http://www.ics.uci.edu/~mlearn/MLRepository.html.

[14] Yager, R.R. and Filev D., “Operations for granular computing: mixing words with numbers”, Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 123-128, 1998.

[15] Yao Y. Y., “Granular computing: basic issues and possible solutions”, Proceedings of the 5th Joint Conference on Information Sciences, ppl86-189, 2000.

[16] Yao Y. Y., “Granular Computing”, Computer Science, 31(10.A), 1-5, 2004.

[17] Yao Y. Y., “A partition model of granular computing”, LNCS, Transactions on Rough Sets, 1, 232-253, 2004.

[18] Yao Jing Tao, “Information Granulation and Granular Relationships”, Proc. Granular Computing, 2005 IEEE International Conference July 2005 Page(s):326 - 329 Vol. 1.

[19] Zadeh L.A., “Fuzzy sets and information granurity, Advances in Fuzzy Set Theory and Applications”, M. Gupta, R.K. Ragade, R.R. Yager (eds), North-Holland Publishing Company, pp3-18, 1979.

[20] Zadeh, L. A., “Fuzzy logic = computing with words”, IEEE Transactions on Fuzzy Systems, 4, 103-111, 1996.

[21] Zadeh, L. A., “The Key Roles of Information Granulation and Fuzzy Logic in Human Reasoning, Concept Formulation and Computing with Words”, FUZZ-IEEE '96, Fifth IEEE International Conference on Fuzzy Systems, New Orleans, USA, September 8--11, 1996.

[22] Zadeh L.A., “Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic”, Fuzzy Sets and Systems, 90(2), 111-127, 1997.

[23] Zadeh L.A., “Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems”, Soft Computing, 2(1), 23 25, 1998.