CONSTRUCTING MINIMAL SPANNING TREE BASED ON ROUGH SET THEORY FOR GENE SELECTION

Soumen Kumar Pati and Asit Kumar Das

[1]Department of Computer Science/Information Technology, St. Thomas'College of Engineering and Technology, 4, D.H. Road, Kolkata-23
soumen_pati@rediffmail.com
[2]Department of Computer Science and Technology, Bengal Engineering and Science University, Shibpur, Howrah-03
asitdas72@rediffmail.com

ABSTRACT

Microarray gene dataset often contains high dimensionalities which cause difficulty in clustering and classification. Datasets containing huge number of genes lead to increased complexity and therefore, degradation of dataset handling performance. Often, all the measured features of these high-dimensional datasets are not relevant for understanding the underlying phenomena of interest. Dimensionality reduction by reduct generation is hence performed as an important step before clustering and classification. The reduced attribute set has the same characteristics as the entire set of attributes in the information system. In this paper, a new attribute reduction technique, based on directed minimal spanning tree and rough set theory is done, for unsupervised learning. The method, firstly, computes a similarity factor between each pair of attributes using indiscernibility relation, a concept of rough set theory. Based on the similarity factors, an attribute similarity set is formed from which a directed weighted graph with vertices as attributes and edge weights as the inverse of the similarity factor is constructed. Then, all possible minimal spanning trees of the graph are generated. From each tree, iteratively, the most important vertex is included in the reduct set and all its out-going edges are removed. The process stops when the edge set is empty, thus producing multiple reducts. The proposed method and some well-known attribute reduction techniques have been applied on several microarray gene datasets for gene selection. The results obtained show the effectiveness of the method.

KEYWORDS

Gene selection, Reduct Generation, Rooted Directed Minimal Spanning Tree, Rough Set Theory, Unsupervised Learning.

1. INTRODUCTION

In scientific databases like gene microarray dataset, it is common to encounter large sets of observations, represented by hundreds of coordinates. The performance of data analysis such as clustering, classification, etc. degrades in such high dimensional spaces. Gene microarray high dimensional data provides the opportunity to measure the expression level of thousands of genes simultaneously and this kind of high-throughput data has a wide application in bioinformatics research. In DNA microarray data analysis generally biologists measure the expression levels of genes in the tissue samples from patients and find explanations about how the genes of patients relate to the types of cancers they had. Many genes could strongly be correlated to a particular type of cancer, however, biologists prefer to focal point on a small subset of genes that dominates the outcomes before performing in-depth analysis and expensive experiments with a high dimensional dataset. Therefore, automated selection of the minimal set of attributes (i.e., reduct), is highly advantageous.

DOI : 10.5121/ijaia.2013.4112

Feature selection and reduct generation are frequently used as a pre-processing step to data mining and knowledge discovery. It selects an optimal subset of features from the feature space according to a certain evaluation criterion. It has been a fertile field of research and shown very effective in removing irrelevant and redundant features, increasing efficiency in data analysis like clustering, classification, etc. All the measured variables of high-dimensional datasets are not relevant for understanding the underlying phenomena of interest. This enormity in datasets may cause serious problems to many machine learning algorithms with respect to scalability and learning performance. Therefore, feature selection and reduct generation become necessary for data analysis when facing high dimensional data. However, this trend of enormity on both size and dimensionality also poses severe challenges to reduct generation algorithms too. Rough Set Theory (RST) [1, 2], a new mathematical approach to imperfect knowledge, is popularly employed to evaluate significance of attributes and helps to find the reduct.

Hu et al. [3] developed two new algorithms to calculate core attributes and reducts for feature selection. These algorithms can be extensively applied to a wide range of real-life applications with very large data sets. Jensen et al. [4] developed the Quickreduct algorithm to compute a minimal reduct without exhaustively generating all possible subsets and also developed Fuzzy-Rough attribute reduction with application to web categorization. Zhong et al. [5] applies Rough Sets with Heuristics (RSH) and Rough Sets with Boolean Reasoning (RSBR) for attribute reduction and discretization of real-valued attributes. Komorowski et al. [6] studies an application of rough sets to modeling prognostic power of cardiac tests. Bazan [7] compares rough set-based methods, in particular dynamic reducts, with statistical methods, neural networks, decision trees and decision rules. Carlin et al. [8] presents an application of rough sets for diagnosing suspected acute appendicitis.

The main advantage of rough set theory in data analysis is that it does not need any preliminary or additional information about data like probability in statistics [9], basic probability assignment in Dempster-Shafer theory [10], grade of membership or the value of possibility in fuzzy set theory [11] and so on. But finding reduct by exhaustive search of all possible combinations of attributes is an NP-Complete [12] problem and so some heuristic approach should be applied.

In the paper, a novel reduct generation method is proposed based on indiscernibility relation. Indiscernibility relation induces partitions of objects from which degree of similarity or similarity factor between two attributes is measured and an attribute similarity (AS) set is obtained. Now, the attribute similarities of AS with similarity factor less than that of average are removed and a directed weighted graph is constructed based on the reduced AS set, where attributes are vertices and weight of an edge is the inverse of the similarity factor of corresponding attribute similarity in AS. All possible directed minimal spanning trees are obtained from the directed graph. Each tree represents all important similarities of attributes by its edges which help to find out the information-rich attributes (i.e., vertices) that form the reduct of the data set. For generating a reduct the vertex having maximum out-degree is selected and included in reduct. Then all its out-going edges are removed. This process continues until the edge set of the tree becomes empty and thus all the selected vertices form a reduct. This is applied to all the trees, multiple reducts are obtained and stored in the set RED.

Finally, the proposed method was applied on several microarray gene datasets for gene selection. Some well-known attribute reduction techniques like PCA [20], SVD [21], etc. were also applied. The reduced datasets were then clustered and the results obtained are compared to show the effectiveness of the method.

The rest of the paper is organized as follows: necessary concepts of indiscernibility relation, reduct, core and algorithm for minimal spanning tree generation for directed graphs are described in section 2. Section 3 discusses the proposed reduct generation method. Section 4 shows the

experimental results and finally conclusion of the paper and the areas for further research are stated in section 5.

2. RELEVANCE ANALYSIS OF BACKGROUND

Conventionally Before proceeding to explain the proposed method, a review of necessary concepts is done below.

2.1. Indiscernibility Relation

Let Rough set theory is a mathematical technique to deal with incomplete, imprecise or uncertain information. The main idea is based on the indiscernibility relation generated by information about objects of interest that are indistinguishable from each other.

Let $I= (U, A)$ be an information system where U is the finite, non-empty set of objects (called the *universe*) and A is a finite, non-empty set of *attributes*. Each attribute $a \in A$ can be defined mathematically, as a function described in Eq. (1).

$$f_a: U \rightarrow V_a \ \forall a \in A \tag{1}$$

Where, V_a, is the set of values of attribute a, called the *domain* of a.

For any $P \subseteq A$, there exists a binary relation IND (P), called indiscernibility relation as defined in Eq. (2).

$$IND(P) = \{(x, y) \in U^2 | f_a(x) = f_a(y) \ \forall a \in P\} \tag{2}$$

Where, $f_a(x)$ denotes the value of attribute a for object x in U.

Obviously, IND (P) is an equivalence relation which induces equivalence classes. The family of all equivalence classes of $IND(P)$, i.e., partition determined by P, is denoted by $U/IND(P)$ or simply U/P and an equivalence class of U/P, i.e., block of the partition U/P, containing x is denoted by $[x]_P$. If object pair (x, y) belongs to IND (P), x and y are called *P-indiscernible*. Equivalence classes IND (P) (or blocks of the partition U/P) are referred to as *P-elementary sets*. The indiscernibility relation is used to define the upper and lower approximations in rough set theory. For each set of attributes P, an indiscernibility relation $IND(P)$ partitions the set of objects into m number of equivalence classes, defined as partition $U/IND(P)$ or U/P, equal to $\{[x]_p\}$, where $|U/P|=m$.

2.2. Reduct and Core Identification

Elements belonging to the same equivalence class are indiscernible; otherwise elements are discernible with respect to P. If one considers a non-empty attributes subset, $R \subset P$ and $IND(R) = IND(P)$, then $P-R$ is dispensable. Any minimal R such that $IND(R)=IND(P)$, is a minimal set of attributes that preserves the indiscernibility relation computed on the set of attributes P and is called reduct of P, denoted as $R=RED(P)$. Attribute set R is minimal in the sense that $[x]_{R-a} \neq [x]_P$, $\forall a \in R$. In other words, no attribute can be removed from set R without changing the equivalence classes $[x]_P$. The reduct of a decision system is not unique. There may be many subsets of attributes which are common to all the reducts and are hence, most important to the information system. The core of P is the intersection of reducts, defined as $CORE$ $(P) = \cap RED$ (P).

2.3. Algorithm for Rooted Directed Minimal Spanning Tree

Generally, Prim's [13] or Kruskal's [14] algorithm is used to find the minimal spanning tree (MST) of an undirected graph. But they do not give the optimal result when applied to directed graphs. Fig. 1 exhibits that the tree, constructed by taking iterative greedy decision of Prim's algorithm, is not a minimal spanning tree of the directed graph.

Chu and Liu [15], Edmonds [16] and Bock [17] have independently given efficient algorithms for finding the MST on a directed graph. The Chu-Liu and Edmonds algorithms are virtually identical; the Bock algorithm is similar but stated on matrices instead of on graphs. Furthermore, a distributed algorithm is given by Humblet [18].

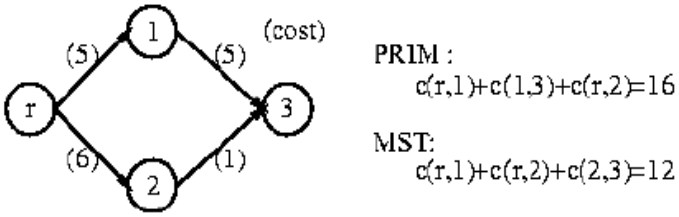

Figure 1. MST construction for directed graph using Prim's algorithm

The rooted directed spanning tree is defined as a graph which connects, without any cycle, all vertices with *n-1* edges, i.e., each vertex, except the root, has one and only one incoming edge. Consider a directed graph, *G=(V, E)*, where *V* and *E* are the set of vertices and edges, respectively. A cost *c(i, j)* is associated with each edge *(i, j)* in *E*. Let *|V|=n* and *|E|=m*. The algorithm is described and explained briefly by the following steps, which computes a rooted directed minimal spanning tree *MST*(*V*, *S*) of the graph *G(V, E)* where *S* is a subset of *E* such that $\sum c(i, j), \forall (i , j)$ in *S* is minimized.

Chu-Liu/Edmond's (CLE) Algorithm

1. Discard the edges entering the root if any; for each vertex other than the root, select the entering edge with the smallest cost. Let the selected *n-1* edges be the set *S*.
2. If no cycle formed, *MST* (*V*, *S*) is a MST. Otherwise, go to step 3.
3. For each cycle formed, contract the vertex in the cycle into a pseudo-vertex*(k)* and modify the cost of each edge which enters a vertex*(j)* in the cycle from some vertex*(i)* outside the cycle, according to the Eq. (3).

$$c(i,k) = c(i,j) - (c(x(j),j) - \min_{\{j\}}(c(x(j),j) \quad (3)$$

   Where $c(x(j),j)$ is the cost of the edge in the cycle which enters *j*.
4. For each pseudo-vertex, select the entering edge which has the smallest modified cost; replace the edge which enters the same real vertex in *S* by the new selected edge.
5. Go to step 2 with the contracted graph.

The key idea of the algorithm is to find the replacing edge(s) which has the minimum extra cost to eliminate cycle(s), if any. The Eq. (3) exhibits the associated extra cost. Figure 2. illustrates that the contraction technique finds the minimum extra cost replacing edge (2, 3) for edge (4, 3) and hence the cycle is eliminated.

3. MULTIPLE REDUCT GENERATION METHOD

The proposed method first computes the equivalence classes by $IND(A_i)$ for each attribute A_i. Then, it calculates the degree of similarity among each pair of attributes with the help of a similarity factor. Based on the similarity of attribute pairs a weighted directed graph is formed and all possible minimal spanning trees of the graph are obtained which finally generate the multiple reducts for gene selection.

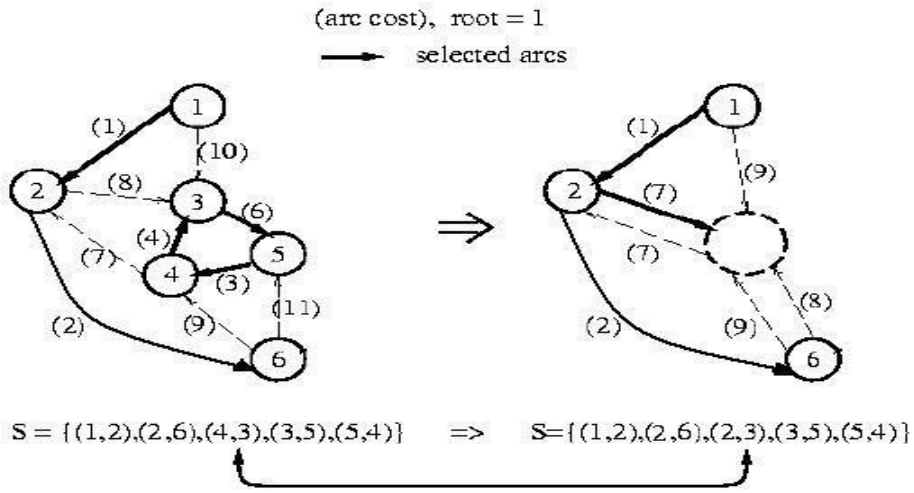

Figure 2. MST construction using Chu-Liu / Edmond's algorithm

To illustrate the method, a sample dataset, shown in Table 1, with eight objects and four attributes is considered.

Table1. Sample dataset

Object	Diploma(i)	Experience (e)	French (f)	Reference(r)
x_1	MBA	Medium	Yes	Excellent
x_2	MBA	Low	Yes	Neutral
x_3	MCE	Low	Yes	Good
x_4	MSc	High	Yes	Neutral
x_5	MSc	Medium	Yes	Neutral
x_6	MSc	High	Yes	Excellent
x_7	MBA	High	No	Good
x_8	MCE	Low	No	Excellent

Here, equivalence classes by $IND(P)$ are formed using Eq. (2) and listed below:

U/i = ({x_1, x_2, x_7}, {x_3, x_8}, {x_4, x_5, x_6})
U/e = ({x_1, x_5}, {x_2, x_3, x_8}, {x_4, x_6, x_7})
U/f = ({x_1, x_2, x_3, x_4, x_5, x_6}, {x_7, x_8})
U/r = ({x_1, x_6, x_8}, {x_2, x_4, x_5}, {x_3, x_7})

3.1. Attribute Similarity Measurement

Elements An attribute A_i is similar to another attribute A_j in context of indiscernibility if they induce the same equivalence classes of objects under their respective indiscernibility relations. But in real situation, it rarely occurs. So similarity of attributes is measured by introducing a similarity factor, based on indiscernibility relation, which indicates the degree of similarity of one attribute to another attribute. Here, an attribute A_i is said to be similar to an attribute A_j with degree of similarity (or similarity factor) $\delta_f^{i,j}$ and is denoted by $A_i \xrightarrow{\delta_f^{i,j}} A_j$ if the probability of inducing the same equivalence classes of objects under their respective indiscernibility relations is $(\delta_f^{i,j} \times 100)\%$, where $\delta_f^{i,j}$ is computed by Eq. (4).

$$\delta_f^{i,j} = \frac{1}{\left|U/_{A_i}\right|} \sum_{[x]_{A_i} \epsilon U/_{A_i}} \frac{1}{|[x]_{A_i}|} max_{[x]_{A_j} \epsilon U/_{A_j}} \left([x]_{A_i} \cap [x]_{A_j}\right) \quad (4)$$

It is quite obvious that $\delta_f^{i,j}$ would have value 1 if A_i and A_j have exactly similar classification pattern. For each pair of conditional attributes (A_i, A_j), similarity factor is computed by Eq. (4). The method of computation of similarity measurement for the attribute similarity $A_i \xrightarrow{\delta_f^{i,j}} A_j$ ($A_i \neq A_j$) is described in algorithm "SIM_FAC" below.

Algorithm: SIM_FAC(A_i, A_j)

/* Similarity factor computation for $A_i \rightarrow A_j$ */

Input: Attributes A_i and A_j

Output: Similarity factor $\delta_f^{i,j}$

Begin

Compute indiscernibility $IND(A_i)$ using Eq. (2)

Compute indiscernibility $IND(A_j)$ using Eq. (2)

/* similarity measurement of A_i to A_j */

$\delta_f^{i,j} = 0$

For each $[x]_{A_i} \in U/A_i$ {

max_overlap = 0

For each $[x]_{A_j} \in U/A_j$ {

Overlap = $|[x]_{A_i} \cap [x]_{A_j}|$

If (overlap > max_overlap) then

max_overlap = overlap

}

$\delta_f^{i,j} = \delta_f^{i,j} + \frac{\max\_overlap}{|[x]_{A_i}|}$

}

$\delta_f^{i,j} = \frac{\delta_f^{i,j}}{|U/A_i|}$

End.

3.2. Formation of Attribute Similarity Set

For each pair of conditional attributes (A_i, A_j), similarity factor is computed by "SIM_FAC" algorithm, described in the previous sub-section. High value of similarity factor of $A_i \rightarrow A_j$ means that the indiscernibility relations $IND(A_i)$ and $IND(A_j)$ produce highly similar equivalence classes. This implies that both the attributes A_i and A_j have almost similar classification power and so $A_i \rightarrow A_j$ is considered as strong similarity of A_i to A_j. Since, for any two attributes A_i and A_j, two similarities $A_i \rightarrow A_j$ and $A_j \rightarrow A_i$ are obtained, only the one with higher similarity factor is included in the attribute similarity set AS. In case, both have equal values of similarity factors, any one is chosen randomly. Thus, for n attributes, $\frac{n(n-1)}{2}$ similarities are selected, of which some are strong and some are not. Out of these, the similarities with $\delta_f^{i,j}$ value less than the average value δ_f of all the similarity factors, are discarded and the rest are considered as the set of attribute similarity AS. So, each element x in AS is of the form x: $A_i \rightarrow A_j$ such that $Left(x)=A_i$ and $Right(x)=A_j$. The algorithm "AS_GEN" described below, computes the attribute similarity set AS.

Algorithm: AS_GEN (A, δ_f)
/* Computes attribute similarity set $\{A_i \rightarrow A_j\}$ */
Input: A = set of attributes and δ_f = 2-D matrix containing similarity factors between each pair of conditional attributes, obtained using Eq. (4).
Output: Attribute Similarity Set AS
Begin

```
AS = { }, sum_δf = 0
/* add n(n – 1)/2 elements to AS */
For i = 1 to (|C| - 1) {
   For j = i+1 to |C| {
      If (δ_f^{i,j} > δ_f^{j,i}) {
         sum_δf = sum_δf + δ_f^{i,j}
         AS = AS ∪ {Ai→Aj}
      }
      Else {
         sum_δf = sum_δf + δ_f^{j,i}
         AS = AS ∪ {Aj→ Ai}
      }
   }
}
/*modify AS to store only {Ai→Aj}for which δ_f^{i,j} > avg_δf */
avg_δf = 2×sum_δf / (|C|(|C|−1))
For each {Ai→Aj}∈AS
   If (δ_f^{i,j} < avg_δf)
```

$AS = AS - \{A_i \rightarrow A_j\}$

End.

Initially, algorithm "AS_GEN" selects AS= {i→e, i→f, r→i, e→f, r→e, r→f} and constructs Table 2. As the similarity factors for attribute similarities i→e, i→f, e→f and r→f are greater than the average similarity factor avg_δ_f = 0.63, modified attribute similarity set *AS*= {i→e, i→f, e→f, r→f }.

Table 2. Selection of attribute similarities in AS

Attribute Similarity ($A_i \rightarrow A_j$, $i \neq j$ and $\delta_f^{i,j} > \delta_f^{j,i}$)	Similarity Factor of $A_i \rightarrow A_j$ ($\delta_f^{i,j}$)	$\delta_f^{i,j} > \delta_f$
i→e	$\delta_f^{e,i}$ = 0.67	Yes
i→f	$\delta_f^{i,f}$ = 0.72	Yes
r→i	$\delta_f^{i,r}$ = 0.50	
e→f	$\delta_f^{e,f}$ = 0.78	Yes
r→e	$\delta_f^{e,r}$ = 0.39	
r→f	$\delta_f^{r,f}$ = 0.72	Yes
Average	avg_δ_f = 0.63	

3.3. Construction of Attribute Similarity Graph

The minimized attribute similarity set AS= $\{A_i \xrightarrow{\delta_f^{i,j}} A_j\}$ contains the set of pairs of attributes that are most strongly related to each other. To generate a reduct, firstly this set is represented by a directed graph, called *attribute similarity graph (ASG)*. The vertices of *ASG* are the attributes present in the set *AS* and weighted edge exists from attribute A_i to attribute A_j with weight $\delta_f^{i,j}$ if $A_i \xrightarrow{\delta_f^{i,j}} A_j \in$ AS. The weight of an edge between two vertices is the value of the similarity factor between those two attributes of the data set. Thus, attribute similarity $A_i \rightarrow A_j$ with $\delta_f^{i,j} = w$, present in set *AS* is represented by a directed edge from vertex A_i to vertex A_j with weight *w*. Mathematically, ASG is denoted as *G(V, E)* where

$$V = \{A_i | A_i \in (Left(x) \cup Right(x)) \forall x \in AS\} \quad (5)$$

$$E = \left\{ (A_i, A_j) \middle| A_i \xrightarrow{\delta_f^{i,j}} A_j \in AS \right\} \quad (6)$$

The attribute similarity graph generated from Table 2 is shown in Fig. 3.

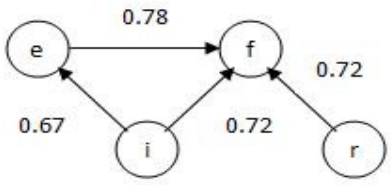

Figure 3. ASG obtained from Table 2.

3.3. Directed Minimal Spanning Tree(s) Construction

The ASG, therefore, represents the total similarity structure of the attribute similarity set *AS*. Some vertices in the ASG may have multiple incoming edges which imply that a particular vertex (attribute) *v* is similar to more than one other vertex (attribute). Now, the vertices of the graph, which have one or more out-going edges, represent the attributes to which some other attributes are similar. The weights of the edges between them denote the strength of their similarity. Therefore, a maximal spanning tree of this graph would give the highest similarities between two attributes. Constructing maximal spanning tree is equivalent to constructing minimal spanning tree with the weights inversed. So, to construct the minimal spanning tree, weights associated to each edge of the directed graph ASG are inversed and Chu-Liu / Edmond's Algorithm is applied. For instance, edge weight *w* is replaced by w^{-1}. In the process, the vertex that has only outgoing edges and no incoming edges is considered as the root. If more than one such vertex exists, then they are fused to form a single vertex. So, before construction of the minimal spanning tree, ASG is modified to merge all the nodes with in-degree zero to a single node and it is considered as the root of the graph. That means the new node *A'* formed by merging other nodes is given by Eq. (7).

$$A' = \bigcup_i A_i \, , deg^-(A_i) = 0 \tag{7}$$

where, deg $^-$ (A_i) denotes in-degree of vertex *A'*, *A'* becomes the tail of all outgoing edges from each A_i and heads of the outgoing edges from each A_i remain the same. Thus, *A'* is the root of the graph as it does not have any incoming edge.

Now, a graph can have multiple minimal spanning trees. This happens when more than one edge, entering the same node, have minimum weights. Keeping track of such edges and using Chu-Liu / Edmond's Algorithm, all possible directed minimal spanning trees of the graph are generated.

Algorithm: MST_GEN (AS)

/* generates minimal spanning trees of ASG */

Input: AS = attribute similarity set obtained from AS_GEN algorithm.

Output: Set of Rooted Directed Minimal Spanning Trees M

Begin

/* Represent AS as a graph using Eq. (5) and Eq. (6) */

Construct graph ASG = (V, E) where

$V = \{A_i \mid A_i \in (Left(x) \cup Right(x)), \forall x \in AS\}$

$E = \{(A_i, A_j) \mid A_i \xrightarrow{\delta_f^{i,j}} A_j \in AS\}$

/* Merge nodes with no incoming edges to create a new node using Eq. (7) */

Root = { }

For each node $N_i \in V$
If (deg- (N_i) = 0) then {
Root = Root $\cup$ {N_i}
Modify ASG by fusing all vertices in set Root
}
For each edge $A_i \xrightarrow{\delta_f^{i,j}} A_j \in E$
$\delta_f^{i,j} = (\delta_f^{i,j})^{-1}$
Compute set of all possible Directed Minimal Spanning Trees M of the ASG using CLE_algorithm
End.

The algorithm modifies the attribute similarity graph shown in Fig. 3 to a new graph, as shown in Fig. 4 and constructs all possible directed minimal spanning tree(s), shown in Fig. 5 (in this case only one such tree is possible).

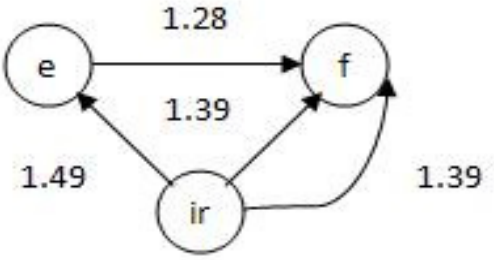

Figure 4. Modified ASG obtained from Figure 3.

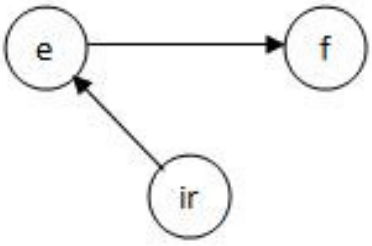

Figure 5. Minimal spanning tree(s) of the graph in Figure 4.

3.5. Multiple Reducts Generation

The above generated rooted directed minimal spanning tree would give the highest similarities between pairs of attributes. Now, our aim is to reduce the number of attributes but also preserve the equivalence class structure of the dataset considering all the attributes. To do this, the attributes, which produce equivalence classes similar to that of some other attribute, may be neglected without affecting the overall equivalence class structure of the dataset. Therefore, on the basis of attribute similarity, those attributes to which some other attributes are similar, are the most important ones and hence should be considered as the reduct.

Now, the tail of every edge $e_i \in E$ of a directed minimal spanning tree $m_j=(V_j, S_j)$, $\forall m_j \in M$ denotes an attribute to which some other attribute is similar. So, every attribute $Tail(e_i)$, $\forall e_i \in E$ should be included in the reduct(s). For each tree, this is done in the following way.

The minimal spanning tree is searched to find the vertex with highest out-degree. The vertex with highest out-degree is an attribute to which most number of other attributes are similar. So, this node is added to the initially empty reduct set R_j and its out-going edges are removed from the tree. This process of trimming the edges of the tree and adding the vertex (attribute) to the reduct set continues till the edge set of the tree becomes empty and thus final reduct R_j is obtained. Basically, it performs vertex covering of the tree.

Repeating this process for all the generated directed minimal spanning trees, multiple reducts R_j are obtained and the set *RED* of all R_j gives the multiple reducts for the dataset.

Algorithm: RED_GEN (M)
/*generates multiple reducts from all possible rooted directed minimal spanning trees M of ASG */
Input: M = Set of All Possible Rooted Directed Minimal Spanning Trees
Output: RED = Set of Multiple Reducts
Begin
RED = { }
For j = 1 to |M| {
R_j = { }
Order [V_j] = array of vertices of minimal spanning tree $m_j \in M$ sorted in descending order of their out-degree
For i = 1 to | V_j | {
Remove outgoing edges in m_j from vertex order[i]
$R_j = R_j \cup \{order[i]\}$
If ($S_j = \Phi$) then
$RED = RED \cup \{R_j\}$
}
}
Return (RED)
End.

Reduct generated from Fig. 5 is {i, r, e} as shown in Figure 6.

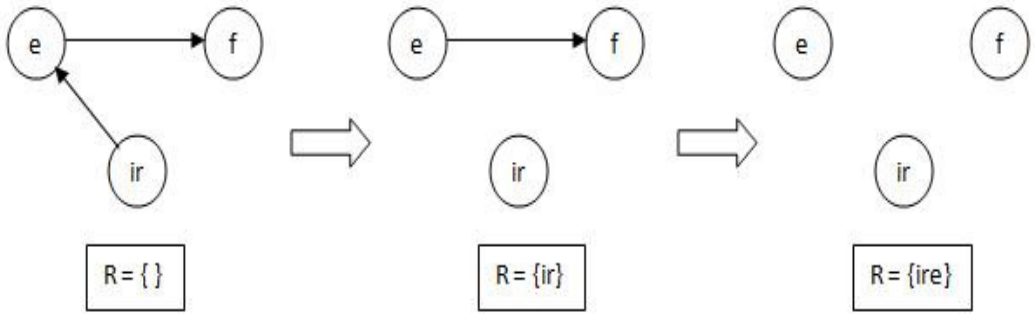

Figure 6. Reduct Generation from directed minimal spanning tree in Figure 5.

4. EXPERIMENTAL RESULTS

The proposed method computes multiple reducts (gene selection) for different kinds of microarray gene datasets (cancerous data), few of which are summarized below:

- **Leukemia (ALL v.s. AML) dataset:** Training dataset consists of 38 bone marrow samples (27 ALL and 11 AML), over 7129 human genes. The raw data is available at http://www-genome.wi.mit.edu/cgi bin/cancer/datasets. cgi.

- **Lung Cancer dataset:** Training dataset contains 16 samples labeled as "MPM" and 16 samples labeled as "ADCA" with around 12533 genes. The raw data available at http://www.chestsurg.org/microarray.htm.

- **Prostate Cancer dataset:** Training dataset contains 52 samples labeled as "relapse" and 50 patients having remained relapse free labeled as "non-relapse" prostate samples with around 12600 genes. The raw data available at http://www-genome.wi.mit.edu/mpr/prostate.

- **Colon Cancer dataset:** The colon cancer data contains 62 samples collected from colon-cancer patients. Among them, 40 tumor biopsies are from tumors (labeled as "negative") and 22 normal (labeled as "positive") biopsies are from healthy parts of the colons of the same patients. 2000 out of around 6500 genes were selected based on the confidence in the measured expression levels. The raw data available at http://microarray.princeton.edu/oncology/ affydata/index.html.

- **Central Nervous System dataset:** Patients' outcome prediction for central nervous system embryonic tumor. Survivors are patients who are alive after treatment whiles the failures are those who succumbed to their disease. The data set contains 60 patient samples, 21 are survivors (labeled as "Class1") and 39 are failures (labeled as "Class0"). There are 7129 genes in the dataset. The raw data available at http://www-genome.wi.mit.edu/mpr/CNS.

At first, all the numeric attributes are discretized by ChiMerge [19] discretization algorithm. The proposed method (PRP) and some well-known dimensionality reduction methods for unsupervised learning such as, Principle Component Analysis (PCA) [20] and Singular Value Decomposition (SVD) [21] were applied on the datasets. The reduced datasets were then clustered using Simple K-Means, EM and Make Density Based Clusterer using 'Weka' tool [22]. From the multiple reducts which were generated by the proposed method, results of the best reducts are shown. The numbers of genes obtained from the best reducts are 198, 170,129,119 and 137 for leukemia, lung, prostate, colon and central nervous system datasets respectively. Within Cluster Sum of Squared Errors (S) and Log Likelihood (L) have been compared and listed in Table 3, which shows that the proposed method produces better results than PCA and SVD.

Table 3. Comparison of Proposed, PCA and SVD methods

Dataset	Attribute Reduction Technique	Clustering Methods		
		EM(L)	Make Density Based Cluster(L)	Simple K-means(s)
Leukemia (ALL/AML)	PRP	**-208.09**	**-211.40**	**981.61**
	PCA	-231.03	-228.13	1119.47
	SVD	-225.49	-224.63	1013.51
Lung Cancer	PRP	**-135**	**-136.98**	**464.32**
	PCA	-152.44	-149.46	489.86
	SVD	-141.52	-146.71	481.05
Prostate Cancer	PRP	**-61.74**	**-73.31**	**835.98**
	PCA	-102.78	-83.82	917.70
	SVD	-89.59	-79.99	878.97
Colon Cancer	PRP	**-86.33**	**-87.09**	**602.55**
	PCA	-120.65	-118.66	733.76
	SVD	-113.24	-115.06	701.14
Central Nervous System	PRP	**-171.78**	**-173.88**	**856.43**
	PCA	-210.43	-213.44	1163.54
	SVD	-193.19	-191.32	991.81

5. DISCUSSIONS AND CONCLUSIONS

Systematic and unbiased approach to cancer classification is of great importance to cancer treatment and drug discovery. It has been known that gene expression contains the keys to the fundamental problems of cancer diagnosis, cancer treatment and drug discovery. The recent advent of microarray technology has made the production of large amount of gene expression data possible. This has motivated the researchers in proposing different cancer classification algorithms using gene expression data.

This paper describes a new method of attribute reduction using concepts of Rough Set Theory and Graph Theory. In this method multiple reducts are generated. Here, the data mining problem is converted to graph theoretic problem and then solved. Many attribute reduction techniques use heuristic algorithms which often degrade the performance. But this method has a strong mathematical background and hence, produces good results. This method of attribute reduction is applied on microarray gene dataset to select a subset of important genes. Future enhancements to this work may include integration of the multiple reducts generated to form a better quality single reduct using some techniques.

REFERENCES

[1] Pawlak, Z., (1982) "Rough sets", *International journal of information and computer sciences*.,Vol. 11, pp. 341-356.

[2] Pawlak, Z., (1998) "Rough set theory and its applications to data analysis", *Cybernetics and systems,* vol. 29, pp. 661-688.

[3] Hu, X., Lin, T.Y. & Jianchao, J., (2004) "A New Rough Sets Model Based on Database Systems", *Fundamental Informaticae*, pp.1-18.

[4] Jensen R. & QiangShen, (2004) "Fuzzy-Rough Attribute Reduction with Application to Web Categorization", *Fuzzy Sets and Systems*, Vol.141, No.3, pp.469-485.

[5] Zhong, N. & Skowron, A., (2005) "A Rough Set-Based Knowledge Discovery Process", *International Journal of Applied Mathematics and Computer Science*. Vol. 11(3), pp. 603-619.

[6] Komorowski, J. & Ohrn, A., (1999) "Modelling Prognostic Power of Cardiac tests using rough sets", *Artificial Intelligence in Medicine*, vol. 15, pp. 167-191.

[7] Bazan, J., (1998) "A Comparison of dynamic and nondynamic rough set methods for extracting laws from decision tables", *Rough Sets in Knowledge Discovery*, PhysicaVerlag.

[8] Carlin, U., Komorowski, J. & Ohrn, A., (1998) "Rough Set Analysis of Patients with Suspected Acute Appendicitis", Proceeding *IPMU*.

[9] Devroye, L., Gyorfi, L., & Lugosi, G., (1996) "A Probabilistic Theory of Pattern Recognition", Newyork: *Springer-Verlag*.

[10] Gupta, S. C. & Kapoor, V. K., (1994) "Fundamental of Mathematical Statistics", Published by: *Sultan Chand & Sons*, A.S. Printing Press, India.

[11] Pal, S., K., & Mitra, S., (1999) "Neuro-Fuzzy pattern Recognition: Methods in Soft Computing", New York: *Willey*.

[12] Garey, M. & Johnson, D., (1979) "Computers and intractability - A guide to the theory of NP-completeness", *Freeman*, New York.

[13] Prim, R. C., (1957) "Shortest connection networks and some generalizations", In: *Bell System Technical Journal*, pp. 1389-1401.

[14] Joseph, Kruskal, B., (1956) "On the shortest Spanning Subtree of a graph and the traveling salesman problem", In: Proceedings of the *American Mathematical Society*, Vol. 7, pp. 48-50.

[15] Chu, Y. J. & Liu, T. H., (1965) "On the shortest arborescence of a directed graph", *Science Sinica*, vol.14, pp.1396-1400.

[16] Edmonds, J., (1967) "Optimum branching", *J. Research of the National Bureau of Standards*, 71B, pp.233-240.

[17] Bock, F., (1971) "An algorithm to construct a minimum spanning tree in a directed network", *Developments in Operations Research*, Gordon and Breach, NY, pp. 29-44.

[18] Humblet, P., (1983) "A distributed algorithm for minimum weighted directed spanning trees", *IEEE Trans. on Communications*, vol. COM-31, no.6, pp.756-762.

[19] Kerber, R. & ChiMerge, (1992) "Discretization of Numeric Attributes", in Proceedings of AAAI-92, *Ninth International Conf. Artificial Intelligence*, AAAI-Press, pp. 123-128.

[20] Mozer, M. C., Jordan, M. I. & Petsche T., (1997) "A principled alternative to the self-organising map", in Advances in *Neural Information Processing Systems*, vol. 9, MIT Press, Cambridge, MA.

[21] Petrou M. & Bosdogianni, P., (2000) "Image Processing: The Fundamentals-an example of SVD", *John Wiley*, pp. 37-44.

[22] WEKA: Machine Learning Software, http://www.cs. waikato.ac.nz/ml/weka.

Authors

Mr. Soumen Kumar Pati is an Assistant Professor of Computer Science/Information Technology at St. Thomas' College of Engineering and Technology, Kidderpore, Kolkata,West Bengal, India. He has received M.Tech degree in Computer Science and Engg from Jadavpur University. He is registered for PhD (Engg) degree at Bengal Engineering and Science University, Shibpur, Howrah. His research interests include Bio-informatics, Data Mining and Pattern Recognition, Roughset Theory, etc.

Dr.Asit Kr. Das is an Assistant Professor of Computer Science and Technology at Bengal Engineering and Science University, Shibpur, Howrah. He has received B.Sc. Honours in Mathematics, B. Tech. and M.Tech degree in Computer Science and Engg from Calcutta University. He obtained PhD (Engg) degree from Bengal Engineering and Science University, Shibpur, Howrah. His research interests include Data Mining and Pattern Recognition, Text Categorization, Rough Set Theory, Bio-informatics etc.