# Reduct Generation by Formation of Directed Minimal Spanning Tree Using Rough Set Theory

Asit Kumar Das[1], Shampa Sengupta[2], and Saikat Chakrabarty[3]

[1, 3] Dept. of Computer Sc. & Tech., Bengal Engineering & Science University, Shibpur, Howrah – 711 103, West Bengal, India
[2] Dept. of Information Technology, MCKV Institute of Engineering, Liluah, Howrah – 711 204, West Bengal, India
akdas@cs.becs.ac.in, shampa2512@yahoo.co.in, saikatchakrabarty187@gmail.com

**Abstract.** In recent years, dimension of datasets has increased rapidly in many applications which bring great difficulty to data mining and pattern recognition. Also, all the measured variables of these high-dimensional datasets are not relevant for understanding the underlying phenomena of interest. In this paper, firstly, similarities among the attributes are measured by computing similarity factors based on relative indiscernibility relation, a concept of rough set theory. Based on the similarity factors, attribute similarity set AS = $\{(A \xrightarrow{k} B)$ / A, B are attributes and B similar to A with similarity factor $k\}$ is formed which helps to construct a directed weighted graph with weights as the inverse of similarity factor *k*. Then a minimal spanning tree of the graph is generated, from which iteratively most important vertex is selected in reduct set. The iteration completes when the edge set is empty. Thus the selected attributes, from which edges emanate, are the most relevant attributes and are known as reduct. The proposed method has been applied on some benchmark datasets and the classification accuracy is calculated by various classifiers to demonstrate the effectiveness of the method.

## 1 Introduction

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 and classification techniques. In recent years, dimension of datasets has increased rapidly in many applications which bring great difficulty to data mining and pattern recognition. Also, all the measured variables of these high-dimensional datasets are not relevant for understanding the underlying phenomena of interest. This enormity may cause serious problems to many machine learning algorithms with respect to scalability and learning performance. Therefore, feature selection and reduct generation become very necessary for data analysis when facing high dimensional data nowadays. However,

S.C. Satapathy et al. (Eds.): Proceedings of the InConINDIA 2012, AISC 132, pp. 127–135.
springerlink.com 

this trend of enormity on both size and dimensionality also poses severe challenges to reduct generation algorithms. Rough Set Theory (RST) [1, 2] is popularly employed to evaluate significance of attributes and helps to find the reduct. 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 [7], basic probability assignment in Dempster-Shafer theory [8], grade of membership or the value of possibility in fuzzy set theory [9] and so on. But finding reduct by exhaustive search of all possible combinations of attributes is an NP-Complete problem and so many researchers [3-6] applied some heuristic approach for discretization and attribute reduction of real-valued attributes in feature selection.

In the paper, a novel reduct generation method is proposed combining the concept of relative indiscernibility relation [1] of RST and Minimal Spanning Tree (MST) [10]. Relative 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 average similarity value are removed and a directed weighted graph is constructed based on the reduced AS set, where weight of an edge is the inverse of the corresponding similarity factor. A minimal spanning tree is obtained from the directed graph using [11]. The tree represents all important similarities of attributes by its edges which help to find out all the information-rich attributes (i.e., vertices) that form the reduct of the data set. To generate reduct, a root (which has no incoming edge) of the spanning tree is selected first and all its outgoing edges are removed. Then another vertex of the maximum out-degree is selected and associated outgoing edges are removed. This process continues until the edge set of the tree becomes empty and all the selected vertices form a reduct.

The rest of the paper is organized as follows: Similarity measurements of attributes and subsequently reduct generation are demonstrated in section 2. Section 3 shows the experimental result of the proposed method and finally conclusion of the paper and the areas for further research are stated in section 4.

## 2 Proposed Work

The proposed method computes relative indiscernibility of the conditional attributes relative to the decision attribute which helps to measure the degree of similarity among the condition attributes. Based on the similarity of attributes a weighted directed graph is formed and a minimal spanning tree of the graph is obtained which finally generates the reduct.

### 2.1 Relative Indiscernibility and Dependency of Attributes

Let DS = ($U$, $A$, $C$, $D$) be a decision system where $U$ is the finite, non-empty set of objects and $A=C \cup D$ such that $C$ and $D$ are set of condition and decision attributes respectively. Each attribute $a \in A$ can be defined as a function, described in (1).

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

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

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

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

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*(*B*) or simply *U*/*P* and an equivalence class of *U*/*P*, i.e., block of the partition *U*/*P*, containing *x* is denoted by *P*(*x*).

In the paper, relative indiscernibility relation is introduced based on the concept of conventional indiscernibility relation. It gives indiscernibility of objects for an attribute, relative to another attribute (decision attribute in this case). Every conditional attribute $A_i$ of C determines a relative (to decision attribute) indiscernibility relation (*RIR*) over *U* and is denoted as $RIR_D(A_i)$, which can be defined by equation (3).

$$RIR_D(A_i) = \left\{(x, y) \in \Pi_{A_i}[x]_D \times \Pi_{A_i}[x]_D \;\middle|\; f_{A_i}(x) = f_{A_i}(y) \forall\, [x]_D \in U/D\right\} \tag{3}$$

where, $\Pi_{A_i}[x]_D$ is the projection operation that selects only the conditional attribute $A_i$ for the objects $[x]_D$, $f_{A_i}(x)$ and $f_{A_i}(y)$ are computed using (1). For each conditional attribute $A_i$, a relative indiscernibility relation $RIR_D(A_i)$ partitions the set of objects into *n*-number of equivalence classes, defined as partition $U/RIR_D(A_i)$ or $U_D/A_i$, equal to $\{[x]_{A_{i/D}}\}$, where $|U_D/A_i| = n$. Obviously, each equivalence class $\{[x]_{A_{i/D}}\}$ contains objects with same decision value which are indiscernible by attribute $A_i$.

To illustrate the method, a sample dataset represented by Table 1 is considered with eight objects, four conditional and one decision attribute.

**Table 1.** Sample dataset

| Object | Diploma(i) | Experience(e) | French(f) | Reference(r) | Decision |
|---|---|---|---|---|---|
| $X_1$ | MBA | Medium | Yes | Excellent | Accept |
| $X_2$ | MBA | Low | Yes | Neutral | Reject |
| $X_3$ | MCE | Low | Yes | Good | Reject |
| $X_4$ | MSc | High | Yes | Neutral | Accept |
| $X_5$ | MSc | Medium | Yes | Neutral | Reject |
| $X_6$ | MSc | High | Yes | Excellent | Reject |
| $X_7$ | MBA | High | No | Good | Accept |
| $X_8$ | MCE | Low | No | Excellent | Reject |

Here, equivalence classes by *IND*(*P*) and $RIR_D(A_i)$ are formed using (2) and (3) respectively and listed in Table 2.

**Table 2.** Equivalence classes by two different relations

| Equivalence classes by *IND*(*P*) | Equivalence classes by $RIR_D(A_i)$ |
|---|---|
| U/D = ({$x_1$,$x_4$,$x_7$},{$x_2$,$x_3$,$x_5$,$x_6$,$x_8$}) | $U_D$/i = ({$x_1$,$x_7$},{$x_2$},{$x_3$,$x_8$},{$x_4$},{$x_5$,$x_6$}) |
| U/i = ({$x_1$,$x_2$,$x_7$},{$x_3$,$x_8$},{$x_4$,$x_5$,$x_6$}) | $U_D$/e = ({$x_1$},{$x_5$},{$x_2$,$x_3$,$x_8$},{$x_4$,$x_7$},{$x_6$}) |
| U/e = ({$x_1$,$x_5$},{$x_2$,$x_3$,$x_8$},{$x_4$,$x_6$,$x_7$}) | $U_D$/f = ({$x_1$,$x_4$},{$x_2$,$x_3$,$x_5$,$x_6$},{$x_7$},{$x_8$}) |
| U/f = ({$x_1$,$x_2$,$x_3$,$x_4$,$x_5$,$x_6$},{$x_7$, $x_8$}) | $U_D$/r = ({$x_1$},{$x_6$,$x_8$},{$x_2$,$x_5$},{$x_4$},{$x_3$,$x_7$}) |
| U/r = ({$x_1$,$x_6$,$x_8$},{$x_2$,$x_4$,$x_5$},{$x_3$,$x_7$}) | |

## 2.2 Formation of Attribute Similarity Set Using Similarity Measurement

An attribute $A_i$ is similar to another attribute $A_j$ in context of classification power if they induce the same equivalence classes of objects under their respective indiscernibility relations. But in real situation, it rarely occurs and so similarity of attributes is measured by introducing the similarity measurement factor 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 relative indiscernible relations is ($\delta_f^{i,j}\times 100$)%, where $\delta_f^{i,j}$ is computed by equation (4).

$$\delta_f^{i,j} = \frac{1}{|U_D/A_i|}\sum_{[x]_{A_{i/D}}\in U_D/A_i}\frac{1}{|[x]_{A_{i/D}}|}\max_{[x]_{A_{j/D}}\in U_D/A_j}\left([x]_{A_{i/D}} \cap [x]_{A_{j/D}}\right) \tag{4}$$

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 (4). High value of similarity factor of $A_i \rightarrow A_j$ means that the relative indiscernibility relations $RIR_D(A_i)$ and $RIR_D(A_j)$ 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 one with higher similarity factor is selected in the list of attribute similarity set AS. Thus, for *n* attributes, $\frac{n(n-1)}{2}$ similarities are selected, out of which some are strong and some are not. Out of these, the similarities with $\delta_f^{i,j}$ value less than the average $\delta_f$ value are discarded and rest is 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 "ASS_GEN" described below, computes the attribute similarity set AS.

```
Algorithm: ASS_GEN(C, δf)
/*   Computes   attribute   similarity   set   {Ai→Aj}   */
Input: C = set of conditional attributes and δf = 2-D matrix
containing similarity factors between each pair of conditional
attributes, obtained using (4).
Output: Attribute Similarity Set AS
Begin
  AS = {}, sum_δf = 0;
  /* compute only n(n - 1)/2 elements in AS */
  for i = 1 to |C| - 1 {
    for j = i+1 to |C| {
      if(δf^{i,j} > δf^{j,i}){sum_δf = sum_δf + δf^{i,j};
         AS = AS∪ {Ai→Aj}}
      else{sum_δf = sum_δf +δf^{j,i}; AS = AS ∪ {Aj→ Ai}}
  }} /* end of i and j loops */
/* modify AS to store only {Ai→Aj} for which δf^{i,j} >avg_δf */
    ASmod = {}; avg_δf = (2×sum_δf)/(|C|(|C|−1)) ;
    for each {Ai→Aj}∈ AS { if(δf^{i,j} > avg_δf){
         ASmod = ASmod∪{Ai→Aj}; AS = AS - {Ai→Aj}}
    }
    AS = ASmod
  End.
```

Initially, algorithm "AS_GEN" selects AS = {i→f, i→ r, e→i, e→f, e→ r, r→ f} and constructs Table 3. As the average similarity factor avg_$\delta_f$ = 0.786 which is less than the similarity factors for attribute similarities i→f, e→i, e→f and r→f, the modified attribute similarity set AS = {i→f, e→i, e→f, r→f }.

**Table 3.** 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→f | $\delta_f^{i,j} = 0.8$ | Yes |
| i→r | $\delta_f^{i,r} = 0.7$ | |
| e→i | $\delta_f^{\theta,i} = 0.83$ | Yes |
| e→f | $\delta_f^{\theta,f} = 0.83$ | Yes |
| e→r | $\delta_f^{\theta,r} = 0.76$ | |
| r→f | $\delta_f^{r,f} = 0.8$ | Yes |
| Average $\delta_f$ | 0.786 | |

### 2.3 Minimal Spanning Tree Generation of Attribute Similarity Graph

The minimized attribute similarity set $AS = \left\{ A_i \xrightarrow{\delta_f^{i,j}} A_j \left( A_i \neq A_j \right) \right\}$ 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 conditional 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. 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*. The ASG, therefore, represents the total similarity structure of the similarity set AS. Some vertices in the ASG may have multiple incoming edges which imply that a particular vertex *v* is similar to more than one other vertex. Without loss of generality, if one of these vertices to which *v* is the most similar can be identified, the other edges incident on *v* may be dropped. To construct the minimal spanning tree, weights associated to each edge of the directed graph ASG are inversed and Chu-Liu/Edmond's Algorithm [11] is applied. In the process, the vertices that have only outgoing edges and no incoming edges are considered as the good candidates for the selection of a 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 considered it as the root of the graph.

```
Algorithm: MST_GEN(AS)
/* generates minimal spanning tree of ASG */
Input: AS = modified attribute similarity set obtained from
ASS_GEN algorithm.
Output: Rooted Directed Minimal Spanning Tree M
Begin
  Construct weighted graph ASG = (V, E) from AS, where
  V = {A_i | A_i ∈ Left(x) ∪ Right(x), ∀x ∈ AS}
```

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

```
/* Merge nodes with in-deg zero to create a new node */
  Root = { }
  for each vertex N_i∈ V {if(in_deg(N_i) = 0){
       Root = Root ∪ {N_i}
       Modify ASG by fusing all vertices in 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 MST of ASG using Chu-Liu/Edmond's algorithm */
```

```
  for each vertex v ϵ V - Root
  select the entering edge with the smallest cost;
  Let S = selected |V - Root| edges;
  Repeat {
    If (no cycle) MST(V, S) is a minimal Spanning Tree;
    Else {for each cycle formed {
    Merge vertices in cycle to a new vertex (k);
    Modify the cost of each edge which enters a  vertex(j)
  in the cycle from some vertex(i) outside the cycle using
  c(i,k)=c(i,j)-(c(x(j),j)- min_{j}(c(x(j),j), where c(x(j),j)
  is the cost of the edge in the cycle which enters j;}
    For each new vertex {
     Select the entering edge with smallest modified cost
     Replace the existing edge by the new selected edge}
  Until(no cycle formed);
  End.
```

The attribute similarity graph (ASG) generated from set AS, modified ASG and corresponding minimal spanning tree (MST) are shown in Fig. 1(a), Fig. 1(b) and Fig. 1(c) respectively.

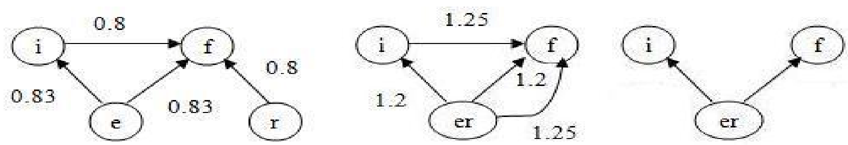


**Fig. 1.** (a) ASG (b) Modified ASG (c) MST of Fig. 1(b)

## 2.4 Reduct Generation

The above generated rooted directed minimal spanning tree would give the highest similarities between the attributes. In the final stage, the maximal 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 is similar. So, this node is added to the initially empty reduct set 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 is obtained.

```
Algorithm: RED_GEN(MST)
/* generates reduct from rooted directed minimal spanning tree
of ASG */
Input: MST(V, S) = Rooted Directed Minimal Spanning Tree
Output: Reduct

Begin

  R = { }
  order[V]= array of vertices of MST sorted in descending order
   of their out-degree
```

```
for i = 1 to |V| {
     Remove outgoing edges from vertex order[i]
     R = R ∪{order[i]}
    if (S = Φ) return (R)}
End
```

Reduct generated from Fig. 1(c) is {e, r} as shown in Fig. 2.

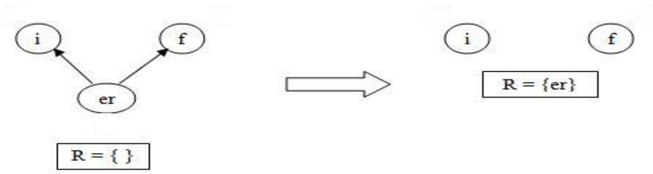


**Fig. 2.** Reduct Generation from Minimal Spanning Tree

## 3 Experimental Results

The proposed method computes a single reduct for datasets collected from UCI machine learning repository [12]. To measure the efficiency of the method, k-fold cross-validations, where k ranges from 1 to 10 have been carried out on the dataset and classified using "Weka" tool [13]. The proposed method (PRP) and well known dimensionality reduction methods, such as, *Cfs Subset Evaluation* (CFS) method [14] and *Consistency Subset Evaluator* (CON) method [15] have been applied on the dataset for dimension reduction and the reduced datasets are classified on various classifiers. Original number of attributes, number of attributes after applying various reduction methods and the accuracies (in %) of the datasets are computed and listed in Table 4, which shows the efficiency of the proposed method.

**Table 4.** Accuracy Comparison of Proposed, CFS and CON methods

| Classifier | Wine (13) | | | Heart (13) | | | Glass (9) | | |
|---|---|---|---|---|---|---|---|---|---|
| | PRP (9) | CFS (8) | CON (8) | PRP (9) | CFS (8) | CON (11) | PRP (8) | CFS (7) | CON (9) |
| Naïve Bayes | 93.70 | 94.80 | 95.30 | 83.27 | 84.38 | 85.50 | 67.28 | 43.92 | 47.20 |
| SMO | 94.91 | 94.30 | 93.74 | 83.27 | 84.75 | 80.38 | 64.48 | 57.94 | 57.48 |
| KSTAR | 95.48 | 92.17 | 93.17 | 83.81 | 82.15 | 81.78 | 83.64 | 79.91 | 78.50 |
| Bagging | 92.09 | 90.35 | 90.91 | 82.52 | 82.52 | 83.64 | 76.63 | 73.83 | 71.50 |
| J48 | 92.65 | 92.17 | 92.61 | 82.89 | 80.52 | 81.15 | 70.09 | 68.69 | 64.20 |
| PART | 92.09 | 90.17 | 91.17 | 79.43 | 81.41 | 78.55 | 75.23 | 70.09 | 68.60 |
| Average | 92.64 | 92.30 | 92.80 | 82.53 | 82.60 | 81.80 | 68.50 | 63.20 | 62.10 |

## 4 Conclusion and Future Enhancements

The paper describes a new method of attribute reduction using minimal spanning tree. It does not use any heuristic algorithm which gives good result only if the heuristic is powerful. The results show that the new method is good enough and often gives better accuracy than the existing ones in most of the cases. Future enhancements to this work may include generation of all possible maximal spanning trees to compute multiple reduct sets and finally select the best one for classification.

## References


1. Pawlak, Z.: Rough sets. International Journal of Information and Computer Sciences 11, 341–356 (1982)
2. Pawlak, Z.: Rough set theory and its applications to data analysis. Cybernetics and Systems 29, 661–688 (1998)
3. Hu, X., Lin, T.Y., Jianchao, J.: A New Rough Sets Model Based on Database Systems. Fundamental Informaticae, 1–18 (2004)
4. Jensen, R., Shen, Q.: Fuzzy-Rough Attribute Reduction with Application to Web Categorization. Fuzzy Sets and Systems 141(3), 469–485 (2004)
5. Zhong, N., Skowron, A.: A Rough Set-Based Knowledge Discovery Process. Int. Journal of Applied Mathematics and Computer Science 11(3), 603–619 (2001); BIME Journal 05(1) (2005)
6. Kerber, R.: ChiMerge: Discretization of Numeric Attributes. In: Proceedings of AAAI-1992, Ninth Int'l Conf. Artificial Intelligence, pp. 123–128. AAAI Press (1992)
7. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, Newyork (1996)
8. Gupta, S.C., Kapoor, V.K.: Fundamental of Mathematical Statistics. Sultan Chand & Sons, A.S. Printing Press, India (1994)
9. Pal, S.K., Mitra, S.: Neuro-Fuzzy pattern Recognition: Methods in Soft Computing. Willey, New York (1999)
10. Bang-Jensen, J., Gutin, G.: Digraphs: Theory, Algorithms and Applications. Springer, Heidelberg, ISBN 1-85233-268-9
11. Chu, Y.J., Liu, T.H.: On the shortest arborescence of a directed graph. Science Sinica 14, 1396–1400 (1965)
12. Murphy, P., Aha, W.: UCI repository of machine learning databases (1996), `http://www.ics.uci.edu/mlearn/MLRepository.html`
13. WEKA: Machine Learning Software, `http://www.cs.waikato.ac.nz/~ml/`
14. Hall, M.A.: Correlation-Based Feature Selection for Machine Learning PhD thesis, Dept. of Computer Science, Univ. of Waikato, Hamilton, New Zealand (1998)
15. Liu, Setiono, R.: A Probabilistic Approach to Feature Selection: A Filter Solution. In: Proc. 13th Int'l Conf. Machine