Feature Selection using Graph based Clustering methods - A Review

Madhuri Gokhale[1]

[1] *Assistant Professor, Jabalpur Engineering College, Jabalpur (M.P.) India*

---***---

***Abstract* -** Feature selection involves identifying a subset of the most useful features that produces compatible results as the original entire set of features. A feature selection algorithm may be evaluated from both the efficiency and effectiveness points of view. Based on these criteria, clustering-based feature selection algorithms have been given. In which a minimum spanning tree (MST) is used for representation of the graph and solve the cluster identification problem using this representation. Due to their ability to detect clusters with irregular boundaries, minimum spanning tree-based clustering algorithms have been widely used in practice. Here we have study some graph clustering methods.

Index Terms – Feature Selection, Graph base clustering, MST.

I. INTRODUCTION

The mathematics and computer science play more and more important role in data mining [1,2] and gene expression data [3,4], especially in the clustering analysis. Clustering belongs to the unsupervised pattern recognition. Data (object) clustering represents one of the most often encountered problems in data analyses. The basic problem is to partition a data set into "clusters" of data points that are "close" to each other but relatively "far from" other data points. A more general problem is the so-called cluster identification problem [17], which is to identify "dense" clusters in a possibly noisy background.[2paper]. Many methods with different application objectives have been developed to solve the clustering problems, including K-Means [5], the Single Linkage Algorithm (SLA) [3] and other hierarchical clustering methods [21], a self-organizing map [5], the Markov Cluster Algorithm [4], and an unsupervised clustering algorithm for graphs based on flows in graphs [10].[14]. Due to their ability to detect clusters with irregular boundaries, minimum spanning tree-based clustering algorithms have been widely used in practice [1].

The MST method is a graphical analysis of an arbitrary set of data points. In such a graph, two points or vertices can be connected either by a direct edge, or by a sequence of edges called a path. The length of a path is the number of edges on it. The degree of link of a vertex is the number of edges that link to this vertex. A loop in a graph is a closed path. A connected graph has one or more paths between every pair of points. A tree is a connected graph with no closed loops. A spanning tree is a tree that contains every point in the data set. If a value is assigned to each edge in the tree, the tree is called a weighted tree. For example, the weight for each edge can be the distance between its two end points. The weight of a tree is the total sum of the edge weights in the tree. The minimum spanning trees are the spanning trees that have the minimal total weight. Two properties used to identify edges provably in an MST are the cut property and the cycle property [1]. The cut property states that the edge with the smallest weight crossing any two partitions of the vertex set must belong to the MST. The cycle property states that the edge with the largest weight in any cycle in a graph cannot be in the MST. As a result, when the weight associated with each edge denotes a distance between the two end points, any edge in the minimum spanning tree will be the shortest distance between the two sub trees that are connected by that edge. Therefore, removing the longest edge will theoretically result in a two-cluster grouping. Removing the next longest edge will result in a three-cluster grouping, and so on. This corresponds to choosing the breaks where the maximum weights occur in the sorted edges.

Usually, MST-based clustering algorithms consist of three steps: 1) a minimum spanning tree is constructed (typically in quadratic time) using either the Prim's algorithm [16] or the Kruskal's [17] algorithm; 2) the inconsistent edges are removed to get a set of connected components (clusters); and 3) step 2 is repeated until some terminating condition is satisfied.[15].

In next section some graph based clustering algorithms will be studied.

II. FEATURE SUBSET SELECTION

Feature as a group for suitability is evaluated by a subset selection a subset of features. Feature subset selection methods are divided into Wrappers, Filters, Embedded and Hybrid methods. Embedded techniques are embedded in and specific to a model. Wrappers use a search algorithm to search through the space of possible features and evaluate every subset by running a model on the subsets. Wrappers are computationally expensive and they have a risk of over fitting to the model. Filters are like Wrappers in the search approach, but instead of evaluating a filter against a model, a simpler filter is evaluated. Two popular filter metrics for classification problems correlation and mutual information, although both are not true metrics. There are, however, true metrics that are functions of the mutual information. Other available filter metrics are: Correlation-based feature selection, Consistency-based feature selection, and Class separability, which include Error probability, Inter class distance, probabilistic distance, and Entropy.

In feature selection technique high dimensional data contains many irrelevant and redundant features. Irrelevant features make available no useful information in any context, and redundant features provide no more information than the selected features. Irrelevant features do not contribute to the predetermined accuracy and redundant features do not redound to getting a good predictor. Therefore feature selection is the process of identifying as many irrelevant and redundant features and removing them. The feature subset selection algorithms can eliminate irrelevant features but do not handle redundant features [15], [17], [18], [19], [21]. Some other algorithms can eliminate irrelevant features as well as handles redundant features [16], [20], [10].

A. Feature Selection Definitions

Let X be the original set of features, with cardinality |X| = n. The continuous feature selection problem refers to the assignment of weights wi to each feature xi ϵ X in such a way that the order corresponding to its theoretical relevance is preserved. The feature selection problem can be seen as a search in a hypothesis space (set of possible solutions). In the case of the binary problem, the number of potential subsets to evaluate is 2n. Definition (Feature Selection) Let J(X') be an evaluation measure to be optimized defined as J : X' ⊂ X ⍰R. The feature subset selection is viewed as: Set |X| = m<n. Find X' ⊂ X, such that J(X') is maximum. Set a value J0, this is, the maximum J that is going to be tolerated. Find the X' ⊂ X with smaller |X'|, such that J(X) > J0. Find the compromise among minimizing |X'| and maximizing J(X') Note that, optimal subset of feature is not unique always.

B. Characteristics of Feature Selection Algorithms: The feature selection algorithms have following important characteristics:

1) Search Organization: A search algorithm is useful for driving the feature selection process using a specific strategy. In general, a search procedure examines only a part of the search space. When a specific state has to be visited, the algorithm uses the information of the previously visited states and eventually heuristic knowledge about non-visited ones [33].

2) Generation of Successor: Mechanism by which possible variants (successor candidates) of the current hypothesis are proposed. Up to five different operators can be considered to generate a successor for each state: Forward, Backward, Compound, Weighting, and Random [33].

3) Evaluation measure: Function by which successor candidates are evaluated, allowing comparing different hypothesis to guide the search process [33].

III. GRAPH BASED CLUSTERING ALGORITHM

Irrelevant features as well as redundant features largely affect the learning machines accuracy. Thus, to identify and remove as much of the irrelevant and redundant information as possible, graph based feature subset selection should be generally adopted by researchers.

A. IMST clustering algorithm

In paper [11] Zhiqiang Xie et. al. presented a clustering algorithm which is based on MST (minimum spanning tree) so that it is called IMST (improved minimum spanning tree) clustering algorithm.

i. The primary division project of data set

For constructing MST, this algorithm puts forward primary division project for the data set.

Definition 1 upper bound data set: For a set of sample in a database *Q* = {*R1, R2… Rn*, every sample *Ri* has a set of attributes *pi*＝｛*Ri1, Ri2… Rin*｝,To a certain *Ri,* if it exits, it satisfies *Rj1≥Ri1, Rj2≥Ri2, …, Rjn≥Rin, R* is called the upper bound data set's element, these elements form a set which is called the upper bound data set α.

Definition 2 lower bound data set: For a set of sample in a database *Q* = {*R1, R2, …, Rn*｝， every sample *Ri* has a set of attributes *pi*＝｛*Ri1, Ri2, …, Rin*｝, to a certain *Ri,,* if it exits, it satisfies，*Rj1≤Ri1, Rj2≤Ri2, …,Rjn ≤Rin, Rj* is called the upper bound data set's element, these elements form a set which is called the upper bound data set β.

Definition 3 middle set: if an element *pi*∉α and *pi*∉β, these elements form a set, which is called the middle data set η.

ii. Algorithm description

1） An element is chosen as a MST's root from the sample database, according to the attribute value of the root, all nodes are divided into three parts namely the upper bound data set, lower bound data set and middle set.

2) The elements are selected from upper bound data set and lower bound data set which has the shortest distance (the Euclidean Distance) to root, and they are respectively made the roots of right sub-tree and left sub-tree, the left sub-tree is divided continuatively according to the attribute value of its root. For the two dimensional data, all other left subtrees' elements are divided into the lower bound data set and the node's middle set relative to the node of left sub-tree. Similarly, all other right sub-trees' elements are divided into the upper bound data set and the node's middle set relative to the node of right sub-tree. But, for multidimensional data, the sub-tree's nodes are divided into the upper data set, the lower data set and the node's middle set still. According to the method above, the division is carried out by layer until it is finished.

3) The middle set's elements are dealt with by the following algorithm: at first, the lowest layer's middle set is handled with upward method step by step, calculating the distance between an element *x* in the middle set and nodes which includes the parent node of middle set's root and all the node below the parent node. If search continues upgrade, the distance between node *x* in middle set and above node is larger than that between node x in middle set and its parent node. The node e which has the shortest distance to *x* is selected and connected. To other trees' nodes, if the node is connected with x and the distance is *w*, there must be a loop in this tree, if there is edges whose value are larger than *w*, the edge which has the maximal value will be deleted and the edge whose value is *w* is kept, otherwise the node is not connected. Using this method to deal with all the middle set, finally, the MST is gotten. For example in Figure 1 *x* is an element in the middle set, e is the shortest node and is connected with *x* and the distance is *w*1. If there is a loop and $w2 \geq w1 \geq w3$, the edge whose value is *w*2 will be deleted. At last, the root middle set's elements are dealt, the node which owns the shortest distance to the root middle set's element is found, then connected and deleted the loop according to the method above.

4) The MST which has been finished will be divided into *k* sub-trees after it is deleted the *k*-1 longest edges by the clustering algorithm of eliminating the longest edges and IMST clustering algorithm. Matrixes are constructed according to their relevant sub-trees. The node which has maximal degree in the matrix of the sub-tree will be the center of a cluster According to the distance relationship of every node and the medoid ,we can quickly finish classifying of a sub-tree by shortest distance, so this clustering is finished.

B. Parallel Clustering Algorithm

In paper [13] Victor Olman et. al. presents a clustering algorithm in which the basic idea of this clustering algorithm is that it first represents a target data set as a weighted undirected graph, with each data point represented as a vertex and each pair of data points is connected by an edge with its weight being the "distance" between the two data points. Then, it builds a minimum spanning tree (MST) of the graph. A key property of an MST is that it generally preserves the structures of clusters in the sense that each cluster is generally presented as a subtree of an MST, as we have previously established [17], and hence, a clustering problem (or, generally, a cluster identification problem) can be solved on the MST representation of a data set. Such a cluster-preserving representation facilitates efficient algorithms for identifying clusters in a data set. We have previously developed a rigorous and linear algorithm for identifying clusters based on the MST representation [17] of the data set.

i. Cluster Identification

This algorithm represents the idea of cluster identification using MST.
Let S be the set of elements s ϵ S and W(s1; s2) be a distance between any two elements of S. We expand our definition of W to the distance W(S1; S2) between sets S1 and S2 as the shortest distance between elements of sets

$$W(S1; S2)= \min\{W(s1; s2)|s1 \in S1; s2 \in S2\}:$$

According to definitions in [12] and [17], the Necessary Condition for a subset C ⊏ S to be a cluster is that for any partition C = C1 U C2, where C1 ≠θ;, C2 ≠θ ;, and C1∩C2 ≠θ;

$$W(C1; S - C1) = W(C1; C2)$$

In other words, regardless of the partition of a cluster, the two parts of a cluster will still be closer to each other than to any other element of S. A key idea of our MST based clustering algorithm is to represent the given data set using a linear representation (LR) through the construction of an MST as follows: LR is a list of the elements of S whose sequential order is the same as the order that these elements got selected by Prim's algorithm into the MST during its construction. In addition, each element s has a numerical value associated with it, which is the Wð:; sÞ value of the edge that Prim's algorithm used to add s into the MST. A highly useful property of this LR is that data clusters in the given data set, as defined in [30], have a one-to-one correspondence with the "valleys" in this LR if we view it in a 2D coordinate system with the sequential order as the x-axis and the values of individual elements as the y-axis [30]. Hence, data clusters in the given data set can be identified through identifying valleys in this LR. In comparison to SLA, our approach searches for clusters satisfying the special condition of a cluster (NC) that essentially narrows the set of all subtrees of MST, as it is done in SLA.

ii. Parallel Algorithm Of MST Construction

We first present our parallel algorithm for the MST construction of a graph representation G = (E, V) of a given data set S, which has the following key steps:

partitioning G into s subgraphs, {Gj = {Vj, Ej}, j = 1, . . . , s, where the value of s is determined later in this section, Vj is the set of vertices in Gj, and Ej ⊂ E is the set of edges connecting the vertices of Vj; defining bipartite graphs Bij = {Vi∪Vj, Eij}, where Vi and Vj are vertex sets of Gi and Gj, and Eij ⊂ E is a set of edges between the vertices of Vi and the vertices of Vj, i ≠ j, for each such pair of subgraphs from {Gi};

Constructing an MST Tii on each Gi and Tij on each Bij in parallel; Building a new graph G^0 =∪Tij, 1 ≤i ≤j≤ s, by merging all the MSTs from the previous step. A result of the merging operation is a subgraph G^0 of G with a vertex set V and edges from trees Tij, 1 ≤i ≤j ≤ s; and constructing an MST of G^0.

C. MST-Inspired Clustering Algorithm

In paper [12] Xiaochun Wang et. al. present a clustering algorithm which called MST-Inspired Clustering Algorithm. MST-inspired clustering algorithm can be summarized in the following:
1. Start with a spanning tree built by the SI.
2. Calculate the mean and the standard deviation of the edge weights in the current distance array and use their sum as the threshold. Partially refine the spanning tree by running our DHCA multiple times until the percentage threshold difference between two consecutively updated distance arrays is below 10_6.
3. Identify and verify the longest edge candidates by running MDHCA until two consecutive longest edge distances converge to the same value at the same places.
4. Remove this longest edge.
5. If the number of clusters in the data set is preset or if the difference between two consecutively removed longest edges has a percentage decrement larger than 50 percent of the previous one, we stop. Otherwise go to Step 3.

The terminating condition presented in the above MST inspired clustering algorithm is under the assumptions that the clusters are well separated and there are no outstanding outliers. However, in many real-world problems, the clusters are not always well separated and noise in the form of outliers often exists. For these cases, some of the longest edges do not correspond to any cluster separations or breaks but are associated with the outliers. For such cases, we propose terminating conditions that are the adaptation results from the LM algorithm and the MSDR algorithm. Before doing that, we realize, at any step of our algorithm, we have a spanning tree upon which MST-inspired clustering operation can be performed. The advantage of the LM algorithm is the avoidance of unnecessary large number of small clusters. The problem with it is that the number of clusters is not usually known a priori, though, for the more general cases, a loose estimate of the maximum and minimum numbers of data points in each cluster is possible. The advantage of the MSDR algorithm is that it can find the optimal cluster separations, particularly for cases where there exist some unknown hidden structures in the data set. However, this optimization is based on an exhaustive search of the breaking edges in the MST whose cutting will give the maximum standard deviation reduction and, therefore, has asymptotic complexity exponential with respect to the size of the data set and does not perform well when a lot of outliers exist. For MST-based clustering algorithm, our remedies to these problems are two terminating conditions developed for the LM algorithm and the MSDR algorithm to be less sensitive to outliers. The first one is for the LM algorithm when a loose estimate of the maximum and minimum number of data points in each cluster is possible and the other is for the MSDR algorithm when there exist some unknown hidden structures in the data set.

Our adapted LM algorithm is the following:

1. Get a loose estimate of the maximum and minimum number of data points for each cluster.
2. Always cut the largest subcluster and cut an edge only when the sizes of both clusters resulted by cutting that edge are larger than the minimum number of data points.
3. Terminates when the size of the largest cluster becomes smaller than the estimated maximum number of data points.

Our adapted MSDR algorithm is the following:

1. Calculate the mean and the standard deviation of the edge weights in the distance array and use their sum as the threshold.
2. Remove the longest edge that is larger than the threshold and that links either a single point or a very small number of data points to the MST.
3. Continue Steps 1 and 2 until the edge is reached, by removing which, two large groups will form from the single largest group before that edge is cut.
4. Apply the MSDR algorithm on the denoised MST.
5. Assign the removed data points the same cluster label as their nearest neighbor's.

To summarize, the numerical parameters the algorithm needs from the user include the data set, the loosely estimated minimum and maximum numbers of data points in each cluster, the input k to the DHCA and MDHCA, and number of nearest neighbors to keep for each data item in the auxiliary arrays, while the outputs will be the final distance and index arrays, and a labeling array that remembers the cluster label each data item belongs to.

D. Clustering based Feature Subset selection algorithm

In paper [8] Qinbao Song et. al. presented a feature selection algorithm which perform clustering before feature selection. There are 4 steps of this algorithm:

1. Remove irrelevant features.
2. Construct a MST of remaining features.
3. Perform clustering on MST.
4. Select representative features from each cluster.

Relevant features have strong correlation with target concept so are always necessary for a best subset, while redundant features are not because their values are completely correlated with each other. Thus, notions of feature redundancy and feature relevance are normally in terms of feature correlation and feature-target concept correlation.
Mutual information measures how much the distribution of the feature values and target classes differ from statistical independence. This is a nonlinear estimation of correlation between feature values or feature values and target classes. The symmetric uncertainty (SU) [28] is derived from the mutual information by normalizing it to the entropies of feature values or feature values and target classes, and has been used to evaluate the goodness of features for classification by a number of researchers (e.g., Hall [16], Hall and Smith [7], Yu and Liu [10], [25], Zhao and Liu [26], [27]). Therefore, we choose symmetric uncertainty as the measure of correlation between either two features or a feature and the target concept.

The symmetric uncertainty is defined as follows:

$$SU(X, Y) = 2*Gain\ (X|Y) \ / \ (H(X) + H(Y))$$

H(X) is the entropy of a discrete random variable X computed as:

$$H(X) = -\sum p(x) \log_2 p(x)$$

Where, p(x) is the prior probability for all values of X.

$$Gain\ (X|Y)= H(X) - H(X|Y)$$

Information gain Gain (X|Y) is the amount by which the entropy of Y decreases. It reflects the additional information about Y provided by X. Where H(X|Y) is the conditional entropy which quantifies the remaining entropy (i.e., uncertainty) of a random variable X given that the value of another random variable Y is known.

$$H(X|Y)= -\sum p(y)\sum p(x|y)\log_2 p(x|y)$$

Where, p(x|y) is the posterior probabilities of X given the value of Y.
Given SU(X; Y) the symmetric uncertainty of variables X and Y , the relevance T-Relevance between a feature and the target concept C, the correlation F-Correlation between a pair of features, the feature redundance F-Redundancy and the representative feature R-Feature of a feature cluster can be defined as follows.

Definition (T-Relevance). The relevance between the feature Fi ϵF and the target concept C is referred to as the T Relevance of Fi and C, and denoted by SU(Fi; C). If SU(Fi; C) is greater than a predetermined threshold θ, we say that Fi is a strong T-Relevance feature.
Definition (F-Correlation). The correlation between any pair of features Fi and Fj (Fi, Fj ϵ F ∧ i ≠j) is called the Correlation of Fi and Fj, and denoted by SU(Fi, Fj).

Definition (F-Redundancy). Let S = {F1; F2; . . . ; Fi; . . . ; Fk<|F|} be a cluster of features. if Fj ϵ S, SU(Fj; C) ≥SU(Fi; C) ∧ SU(Fi; Fj) > SU(Fi; C) is always corrected for each Fi ϵ S (i ≠ j), then Fi are redundant features with respect to the given Fj (i.e., each Fi is a F-Redundancy).
Definition (R-Feature). A feature Fi ϵ S = {F1; F2; . . . ; Fk}, (k < |F|) is a representative feature of the cluster S (i.e., Fi is a R-Feature) if and only if, Fi = argmaxFiϵSSU(Fj; C).

i. Algorithm

```
inputs: D(F1; F2; . . . ; Fi; . . . ;Fm, C) - the given data set θ - the T-Relevance threshold.
 output: S - selected feature subset.
// Part 1: Irrelevant Feature Removal
    1   for i = 1 to m do
    2   T-Relevance = SU (Fi, Fj )
    3   if T-Relevance > then
    4   S = S ∪ { Fi, Fj };
    5   end if
    6   end for

//Part 2: Minimum Spanning Tree Construction
    7   G = NULL; //G is a complete graph
    8   for each pair of features { ′ Fi, Fj, ′ } ⊂ S do
    9   F-Correlation = SU (Fi, Fj )
    10  / ′ F-Correlation ;
    11  end for 1
    12  minSpanTree = Prim (G); //Using Prim Algorithm to
         generate the minimum spanning tree

 //Part 3: Tree Partition and Representative Feature Selection
    13  Forest = minSpanTree
    14  for each edge ∈ Forest do
    15  if SU(Fi, Fj ) < SU(Fi, Fj ) ∧ SU(Fi, Fj ) < SU(Fi, Fj ) then
         13 Forest = Forest − 14 end if 15 end for
    16  S = ϕ
    17  17 for each tree ∈ Forest do
    18   Fi_R= argmax′ ∈ SU(Fi, Fj)
    19  S = S ∪ { Fi_R };
    end for
return S.
```

IV. EXPERIMENTAL STUDY

A. Experimental Setup

To evaluate the performance of feature subset selection algorithms and compare with other feature selection algorithms the experimental set up as follows.

The algorithms are compared with different feature selection algorithms, like (i) FCBF [10], [24], (ii) Relief-F [23], (iii) CFS [16], (iv) FAST [8], respectively. FCBF and Relief-F evaluate features separately. For FCBF, in the experiments, the relevance threshold to be the value of the /log ranked feature for every data set (is the number of features in a given data set). Relief-F searches for nearest neighbors of instances of different classes and weights features according to how well they differentiate instances of different classes. CFS uses best-first search based on the evaluation of a subset that contains features highly predictive of the target concept, yet not predictive of each other. For FAST algorithm, set to be the value of the √∗lg ranked feature for each data set.

Different types of classification algorithms are used to classify data sets prior and after feature selection. Such as (i) the tree-based C4.5, (ii) the probability-based Naive Bayes (NB), (iii) the rule-based RIPPER, (iv) the instance-based lazy learning algorithm IB1, respectively. Naive Bayes employs a probabilistic method for classification by multiplying the individual probabilities of every featurevalue pair. This algorithm assumes independence among the features and even then provides excellent classification results. Decision tree learning algorithm C4.5 is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation etc. The tree comprises of nodes (features) that are selected by information entropy. Instance-based learner IB1 is a single-nearest neighbor algorithm, and it classifies entities taking the class of the closest associated vectors in the training set via distance metrics. It is the simplest among the algorithms used in our study. Inductive rule learner RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is a propositional rule learner that defines a rule based detection model and seeks to improve it iteratively by using different heuristic techniques. The constructed rule set is then used to classify new instances.

When evaluating the performance of the feature subset selection algorithms, different metrics, such as (i) the proportion of selected features (ii) the time to obtain the feature subset, (iii) the classification accuracy, are used. The proportion of selected features is the ratio of the number of features selected by a feature selection algorithm to the original number of features of a data set. B. CFS In order to make the best use of the data and obtain stable results, a (M = 5) × (N = 10)-cross-validation strategy is used. That is, for each data set, each feature subset selection algorithm and each classification algorithm, the 10-fold cross-validation is repeated M = 5 times, with each time the order of the instances of the data set being randomized. This is because many of the algorithms exhibit order effects, in that certain orderings dramatically improve or degrade performance. Randomizing the order of the inputs can help diminish the order effects. In the experiment, for each feature subset selection algorithm, we obtain M×N feature subsets Subset and the corresponding runtime Time with each data set. Average | Subset | and Time, we obtain the number of selected features further the proportion of selected features and the corresponding runtime for each feature selection algorithm on each data set. For each classification algorithm, we obtain M×N classification Accuracy for each feature selection algorithm and each data set. Average these Accuracy, we obtain mean accuracy of each classification algorithm under each feature selection algorithm and each data set. The procedure Experimental Process shows the details.

Procedure: ExKperimental Process

```
M = 5, N = 10
DATA = { 1, 2, ..., 35 }
Learners = {NB, C4.5, IB1, RIPPER}
FeatureSelectors = {FAST, FCBF, ReliefF, CFS}
for each data ∈ DATA do
    for each times ∈ [1, M] do
        randomize instance-order for data
        generate N bins from the randomized data
        for each fold ∈ [1, N] do
            TestData = bin[fold]
            TrainingData = data - TestData
            for each selector ∈ FeatureSelectors do
                (Subset, Time) = selector(TrainingData)
                TrainingData′ = select Subset from TrainingData
                TestData′ = select Subset from TestData
                for each learner ∈ Learners do
                    classifier = learner(TrainingData′ )
                    Accuracy = apply classifier to TestData′
                end for
            end for
        end for
    end for
end for
```

IV. References

[1] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, “Introduction To Data Mining”, third ed. Pearson Education, 2009.

[2] Duda, P.E. Hart, and D.G. Stork, Pattern Classification, second ed. Wiley, 2001.

[3] M. Dash and H. Liu, “Feature Selection for Classification,” Intelligent Data Analysis, Vol. 1, No. 3, pp. 131-156, 1997.

[4] A.K. Jain, R.P.W. Duin, and J.C. Mao, "Statistical pattern recognition: a review", IEEE Transactions Pattern Analysis, Vol. 22, pp.4-37, 2000.

[5] H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan, "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs" Combinatorica, Vol. 6, pp.109–122,1986.

[6] David R. Karger, Philip N. Klein, and Robert E. Tarjan, "A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees",ACM Journal of the Aswcl.tmn for Computing g Machinery, Vol. 42, No.2, March 1995.

[7] M.A. Hall and L.A. Smith, "Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper," Proc. 12th Int'l Florida Artificial Intelligence Research Soc. Conf. , pp. 235-239,1999.

[8] Qinbao Song, Jingjie Ni, and Guangtao Wang "A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data", IEEE Transactions on knowledge and data engineering, Vol. 25, No. 1, 2013.

[9] X. He, D. Cai, and P. Niyogi, "Laplacian Score for Feature Selection", Proc. Advances in Neural Information Processing Systems, Vol. 18, 2005.

[10] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", Proc. 20th Int'l Conf. Machine Leaning, Vol. 20, No. 2, pp. 856-863, 2003.

[11] Zhiqiang Xie, Liang Yu, and Jing Yang, "A Clustering Algorithm Based on Improved Minimum Spanning Tree", 4th International Conference on Fuzzy Systems and Knowledge Discovery FSKD, pp. 3-5, 2007.

[12] Xiaochun Wang, Xiali Wang, and D. Mitchell Wilkes, "A Divide-and-Conquer Approach for Minimum Spanning Tree-Based Clustering" IEEE Transactions On Knowledge And Data Engineering, Vol. 21, No. 7, pp. 3-6, 2009.

[13] Victor Olman, Fenglou Mao, Hongwei Wu, and Ying Xu, "Parallel Clustering Algorithm for Large Data Sets with Applications in Bioinformatics", IEEE Transactions on Computational Biology and Bioinformatics, Vol. 6, No. 2, 2009.

[14] Dina Elsayad, Amal Khalifa, Mohammed Essam Khalifa, and El-Sayed El-Horbaty, "An Improved Parallel Minimum Spanning Tree Based Clustering Algorithm for Microarrays Data Analysis", International Conference on Informatics and Systems (INFOS 2012) Advances in Data Engineering and Management, pp. 5-7, 2012.

[15] Forman G., An extensive empirical study of feature selection metrics for text classification, Journal of Machine Learning Research, 3, pp 1289-1305, 2003.

[16] [22] Hall M.A., Correlation-Based Feature Subset Selection for Machine Learning, Ph.D. dissertation Waikato, New Zealand: Univ. Waikato, 1999.

[17] [24] Hall M.A., Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning, In Proceedings of 17th International Conference on Machine Learning, pp 359-366, 2000.

[18] [27] Kira K. and Rendell L.A., The feature selection problem: Traditional methods and a new algorithm, In Proceedings of Nineth National Conference on Artificial Intelligence, pp 129-134, 1992.

[19] [29] Kononenko I., Estimating Attributes: Analysis and Extensions of RELIEF, In Proceedings of the 1994 European Conference on Machine Learning, pp 171-182, 1994.

[20] [31] Liu H. and Setiono R., A Probabilistic Approach to Feature Selection: A Filter Solution, in Proceedings of the 13th International Conference on Machine Learning, pp 319-327, 1996.

[21] [32] Modrzejewski M., Feature selection using rough sets theory, In Proceedings of the European Conference on Machine Learning, pp 213-226, 1993.

[22] [33] Molina L.C., Belanche L. and Nebot A., Feature selection algorithms: A survey and experimental evaluation, in Proc. IEEE Int. Conf. Data Mining, pp 306-313, 2002.

[23] [36] Robnik-Sikonja M. and Kononenko I., Theoretical and empirical analysis of Relief and ReliefF, Machine Learning, 53, pp 23-69, 2003.

[24] [45] Yu L. and Liu H., Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research, 10(5), pp 1205-1224, 2004.

[25] [71] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," J. Machine Learning Research, vol. 10, no. 5, pp. 1205-1224, 2004.

[26] [72] Z. Zhao and H. Liu, "Searching for Interacting Features," Proc. 20th Int'l Joint Conf. Artificial Intelligence, 2007.

[27] [73] Z. Zhao and H. Liu, "Searching for Interacting Features in Subset Selection," J. Intelligent Data Analysis, vol. 13, no. 2, pp. 207-228, 2009.

[28] [53] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, Numerical Recipes in C. Cambridge Univ. Press 1988.