International Journal of Advance Research in Science and Engineering
Vol. No.4, Special Issue (01), August 2015
www.ijarse.com
IJARSE
ISSN 2319 - 8354

A FAST CLUSTERING-BASED FEATURE SUBSET SELECTION ALGORITHM

Akshay S. Agrawal[1], Prof. Sachin Bojewar[2]

[1]P.G. Scholar, Department of Computer Engg., ARMIET, Sapgaon, (India)
[2]Associate Professor, VIT, Wadala.

ABSTRACT

The paper aims at proposing the fast clustering algorithm for eliminating irrelevant and redundant data. Feature selection is applied to reduce the number of features in many applications where data has hundreds or thousands of features. Existing feature selection methods mainly focus on finding relevant features. In this paper, we show that feature relevance alone is insufficient for efficient feature selection of high-dimensional data. We define feature redundancy and propose to perform explicit redundancy analysis in feature selection. A new hypothesis is introduced that dissociate relevance analysis and redundancy analysis. A clustering based method for relevance and redundancy analysis for feature selection is developed and searching based on the selected features will be performed. While the efficiency concerns the time required to find a subset of features, the effectiveness determines the quality of the subset of features. A fast clustering-based feature selection algorithm, FAST, has been selected to be used in the proposed paper. The clustering-based strategy has a higher probability of producing a subset of useful as well as independent features. To ensure the efficiency of FAST, efficient minimum-spanning tree clustering method has been adopted. When compared with FCBF, ReliefF, with respect to the classifier, namely, the tree-based C4.5, FAST not only produces smaller subsets of features but also improves the performances by reducing the time complexity.

Keyterms: Clustering, Feature subset selection, Minimum Spanning Tree, T-Relevance, F-Correlation.

I. INTRODUCTION

Data mining uses a variety of techniques to identify lump of information or decision-making knowledge in bodies of data, and extracting them in such a manner that they can be directly use in the areas such as decision support, estimation prediction and forecasting. The data is often huge, but as it is important to have large amount of data because low value data cannot be of direct use; it is the hidden information in the data that is useful. Data mine tools have to infer a model from the database, and in the case of supervised learning this requires the user to define one or more classes. The database contains various attributes that denote a class of tuple and these are known as predicted attributes. Whereas the remaining attributes present in the data sets are called as predicting attributes. A combination of values of these predicted attributes and predicting attributes defines a class. While learning classification rules the system has to find the rules that predict the class from the predicting attributes so initially the user has to define conditions for each class, the data mine system then constructs descriptions for the classes. Basically the system should given a case or tuple with certain known attribute values so that it is able to predict what class this case belongs to, once classes are defined the system should infer rules that govern the classification therefore the system should be able to find the description of each class [2]. 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 is basically evaluated from the efficiency and effectiveness points of view. The time required to find a subset of features is concerned with the efficiency while the effectiveness is related to the quality of the subset of features. Some feature subset selection algorithms can effectively eliminate irrelevant features but fail to handle redundant features yet some of others can remove the irrelevant while taking care of the redundant features. A Fast clustering-based feature selection algorithm (FAST) is proposed which is based on above criterion handling redundancy and irrelevancy. [1] The Minimum Spanning tree (Kruskal’s algorithm) is constructed from the F-Correlation value which is used to find correlation between any pair of features. Kruskal's algorithm is a greedy algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. It finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.

II. EXISTING SYSTEM

Feature subset selection generally focused on searching relevant features while neglecting the redundant features. A good example of such feature selection is Relief, which weighs each feature according to its ability to discriminate instances under different targets based on distance-based criteria function.[9] But, Relief is ineffective in removing redundant features as the two predictive but highly correlated features are likely to be highly weighted. Relief-F [6] is an extension of the traditional Relief. This method enables working with noisy and incomplete data sets and to deal with multi-class problems, but is still ineffective in identifying redundant features. However, along with irrelevant features, redundant features also do affect the speed and accuracy of all the probable learning algorithms, and thus should are also important to be eliminated. FCBF is a fast filter method which can identify relevant features as well as redundancy among relevant features without pair wise correlation analysis. Different from these algorithms, our proposed FAST algorithm employs clustering based method to choose features.

There are different approaches available to perform learning. The wrapper methods make use of predictive accuracy of a predetermined learning algorithm to determine the effectiveness of the selected subsets.[7] The accuracy of the learning algorithms [1] is usually high. The however the generality of the selected features is limited and the computational complexity is very large. Thus the wrapper methods are computationally expensive and tend to over fit on small feature training sets. Wrapper uses a search algorithm for searching through the space of possible features and evaluates individual subset by running a model on the subset. The filter methods [3] are independent of the learning algorithms, and also have good generality. Computational complexity is low, but the accuracy of such learning algorithms is not guaranteed. The hybrid method used in our approach is a combination of filter and wrapper methods, filter method reduces search space of computation that will be considered by the subsequent wrapper.

III. PROPOSED SYSTEM

The symmetric uncertainty (SU) is derived from the mutual information by normalizing it to the entropies of feature values or feature values and target classes Therefore; symmetric uncertainty is chosen as the measure of correlation between either two features or a feature and the target concept.[8]

The **symmetric uncertainty (SU)** is defined as follows,

$$SU(X,Y) = \frac{2 \times Gain\left(\frac{X}{Y}\right)}{H(X) + H(Y)}$$

Where, $H(X)$ is the entropy of a discrete random variable X. Let (x) be the prior probabilities for all values of X, then (X) is defined by

$$H(X) = -\sum_{x \in X} p(x) \log 2p(x)$$

Gain (X | Y) determines the amount by which the entropy of Y decreases. It is given by,

$$\begin{aligned} Gain\ (X|Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}$$

Where H(X | Y) is the conditional entropy and is calculated as,

$$H\left(\frac{X}{Y}\right) = -\sum_{y \in Y} p(y) \sum_{x \in X} p(x) \log_2 p(x)$$

Where, X is a Feature and Y is a Class.

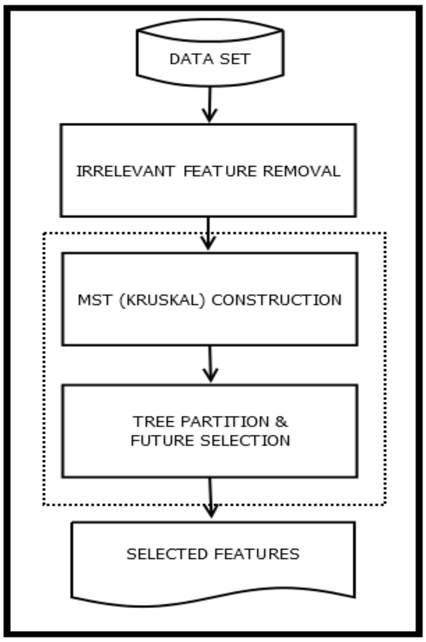


Fig. 3.1: Feature Subset Selection Process.

Given that (X, Y)be 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 redundancy F-Redundancy and the representative feature R- feature of a feature cluster can be defined as follows.

T-Relevance - The relevance between the feature $F_i \in F$ and the target concept is referred to as the T-Relevance of F_i and C, and denoted by SU (F_i,C). If SU (F_i,C) is greater than a predetermined threshold θ, Symmetric Uncertainty of each Feature is greater than the T-Relevance threshold(θ) is checked.

$SU(X,Y) > \theta$ *then X is submittedin Feature set S*

Where, $'S'$ is a set of Relevant Features

we say that F_i is a strong T-Relevance feature.

F-Correlation - The correlation between any pair of features and F_j (F_i, $Fj \in F \wedge i \neq j$) is called the F-Correlation of F_i and, and denoted by SU (F_i, F_j).

F-Redundancy - Let S = {F1, F2, F_i, F_k<|F|} be a cluster of features.

If $\exists$ F_j $\in$S, (F_j) $\geq$ SU(F_i, C) ∧ SU (F_i, F_j) >SU (F_i, C) is always corrected for each $Fi \in S\ (i \neq j)$, then F_i are redundant features with respect to the given F_j (i.e. each F_i is a F-Redundancy).

R-Feature - A feature $F_i \in S$ ={F1, F2, ..., F_k } (k < |F |) is a representative feature of the cluster S (i.e. F_i is a R-Feature) if and only if, F_i = argmax$F_j \in S$ SU (F_j, C).

This means the feature, which has the strongest T Relevance, can act as an R-Feature (Most relevant Feature) for all the features in the cluster.

1) Irrelevant features have no/weak correlation with target concept;

2) Redundant features are assembled in a cluster and a representative feature can be taken out of the cluster. [4]

IV. MST CONSTRUCTION

With the F-Correlation value computed, the Minimum Spanning tree is constructed. Kruskal‟s algorithm is used which forms MST effectively. Kruskal's algorithm is a greedy algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it finds a minimum spanning forest (a minimum spanning tree for each connected component).

Minimum spanning tree using Kruskal’s algorithm is constructed and then a threshold value and step size is set. Those edges from the MST, whose lengths are greater than the threshold value are removed. The ratio between the intra-cluster distance and inter-cluster distance is calculated and the ratio as well as the threshold is recorded. The threshold value is updated by incrementing the step size. Every time the new (updated) threshold value is obtained, the above procedure is repeated. When the threshold value is maximum and as such no MST edges can be removed the above procedure is stopped. In such situation, all the data points belong to a single cluster. Finally the minimum value of the recorded ratio is obtained and the clusters are formed corresponding to the stored threshold value.

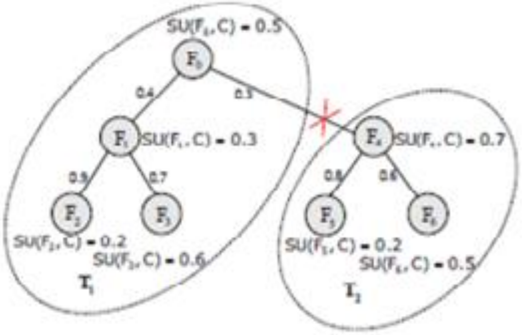


Fig. 3.2: Clustering with MST Construction.

1. Create a forest F (a set of trees), where each vertex in the graph is a separate tree.

2. Create a set S containing all the edges in the graph.

3. While S is nonempty and F is not yet spanning.

Remove an edge with minimum weight from S. If that edge connects two different trees, then add it to the forest, combining two trees into a single tree, otherwise discard that edge. At the termination of the algorithm, the forest forms a minimum spanning forest of the graph. If the graph is connected, the forest has a single component and forms a minimum spanning tree.[1]

V. PROPOSED ALGORITHM

Features in different clusters are relatively independent; the clustering-based strategy of FAST has a high probability of producing a subset of useful and independent features. To ensure the efficiency of FAST, we adopt the efficient minimum-spanning tree (MST) clustering method.

Algorithm:

Inputs: D (F1, F2 … Fm, C) (High Dimensional Dataset).

Output: S-Selected feature subset for searching. [1]

Part 1: Removing irrelevant features:

The features whose SU (Fi,C) values are greater than a predefined threshold(Ø) comprise the target relevant feature subset. Consider feature input dataset (F).

F´= { F_1´,F_2´,…F_k´} (k<=M)

```
1.  for i = 1 to m do
2.  T-Relevance = SU ($Fi, C$)
3. if T-Relevance > $\theta$ then
4.  S = S ∪ { };
```

Part 2: Removing redundant features:

The F-correlation SU (Fi´,Fj´) value for each pair of features.

```
5.  G = NULL; //G is a complete graph
6.  for each pair of features {$F'_i, F'$} ⊂ S do
7. F-Correlation = SU ($F'_i$ , $F'_j$  )
8. $F'i$ and/or $F'j$ to   with  F-Correlation  as the weight of the corresponding edge;
9. MinSpanTree = Kruskal's (G); //Using Kruskal’s algorithm to generate minimum spanning tree.
```

Part 3 : Feature selection.

```
Forest = minSpanTree
for each edge $E_{i,j} \in$ Forest do
    if SU ($F'_i$ , $F'_j$ ) < SU($F'_i$ ,C) ∧ SU($F'_i$ , $F'_j$ )< SU($F'_j$ , C) then
        Forest = Forest − $E_{ij}$
S = $\phi$
for each tree $T_i \in$ Forest do
    $F_R^j$ = argmax $F_k \in$ SU($F'_k$ , C)
    S = S ∪ { $F_R^j$ };
Return S.
```

The algorithm can be expected to be divided into 3 major parts:

The first part is concerned with removal of irrelevant features;

The second part is used for removing the redundant features and The final part of the algorithm is concerned with feature selection based on the value of the Forest. [1]

5.1 Working

A. First Step:

The data set ‘D’ with ‘m’ features F= (F1,F2,…Fm) and class ‘C’, ‘I’ compute the T-Relevance ‘**SU’** (Fi, C) value for every feature ($1 \leq i \leq m$).

B. Second step:

Here the first step is to calculate the F-Correlation ‘**SU’** (F'_i , F'_j) value for each pair of features F'_i and F'_j Then, seeing features F'_i and F'_j as vertices and ‘**SU’** (F'_i , F'_j) the edge between vertices F'_i and F'_j a weighted complete graph G= (V,E) is constructed which is an undirected graph. The complete graph reflects the correlations among the target-relevant features. [3]

C. Third step:

Here, unnecessary edges can be removed. Each tree $T_j \in$ Forest shows a cluster that is denoted as V (T_j), which is the vertex set of T_j . For each cluster V(T_j), select a representative feature whose T-Relevance SU(F_jR, C) is the highest. All F_jR (j = 1...|*Forest*|) consist of the final feature subset ∪ F_jR .

A clustering tree depending on the domain that the admin selects while uploading the file is created. Proposed system then stores the file in the cluster by using the minimum spanning tree method (MST). While in the searching domain; user passes the query and the results are generated in the required format. i.e. either image result, text result or a file result along with the time complexity. FAST algorithm reduces the run time complexity as compared to the other available Algorithms. It removes the redundant features by calculating the Correlations among the various features. F-correlation is calculated as SU (Fi, Fj).

A threshold value (θ) is defined to calculate the relevance among the selected features. If any feature exceeds a particular threshold value then that feature is treated as irrelevant.

F′= { F1′,F2′,…Fk′} (k<=M) [1]

VI. ADVANTAGES

Table 5.1.: Advantages and Disadvantages [5]

SR. NO.	Techniques (or) Algorithms	Advantages	Disadvantages
1.	FAST Algorithm	Improve the performance of the classifiers. The efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset.	--
2.	Consistency Measure	Fast, Remove noisy and irrelevant data.	Unable to handle large volumes of data.
3.	Wrapper Approach	Accuracy is high.	Computational complexity is large.
4.	Filter Approach	Suitable for very large features.	Accuracy is not guaranteed.
5.	Agglomerative linkage algorithm	Reduce Complexity.	Decrease the Quality when dimensionality becomes high.
6.	INTERACT Algorithm	Improve Accuracy.	Only deal with irrelevant data.
7.	Distributional clustering	Higher classification accuracy.	Difficult to evaluation.
8.	Relief Algorithm	Improve efficiency and Reduce Cost.	Powerless to detect redundancy.
9.	Grid based method	Jobs can automatically restart if a failure occurs.	You may need to have a fast interconnect between compute resources.
10.	Model based method	Clusters can be characterized by a small number of parameters.	Need large data sets. Hard to estimate the number of clusters.

VII. RESULT

In the proposed system data set of heart diseases [10] possessing high dimensional features containing 75 categorical, integer and real attributes have been used to eliminate the irrelevant and redundant features by selecting any one feature 'num' from the data set and to form a cluster.

Before proceeding with the actual implementation the files having dataset and features are being uploaded and the specific feature on which the clustering is to be done is inserted ('num' is the feature on which the clustering is done is selected in the proposed algorithm).

Step 1: Removal of irrelevant features:

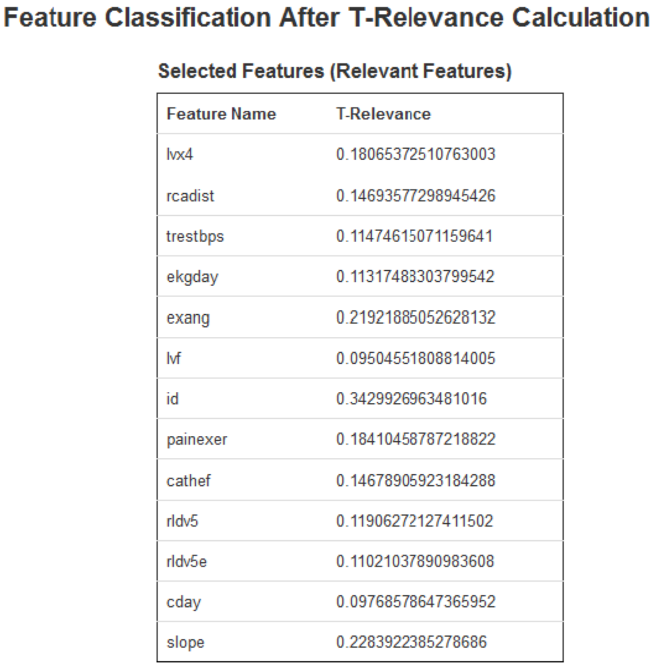
Relevant features have strong correlation with target concept so are always necessary for a best subset, while redundant features are not because their values are completely correlated with each other. T-Relevance is calculated using Symmetric Uncertainty (*SU)* where each attribute/feature (*Fi)* is checked with the class *(C).*

$$T\text{-}Relevance = SU\ (F_i,\ C)$$

$$\text{if } T\text{-}Relevance > \theta \text{ then } S = S\ \cup \{\ \};$$

$$SU(X, Y) = \frac{2 \times \text{Gain}\left(\frac{X}{Y}\right)}{H(X) + H(Y)}$$

The relevance between the feature $F_i \in$ □ and the target concept C is referred to as the T-Relevance of F_i and C, and denoted by (F_i, C). If (F_i, C) is greater than a predetermined threshold θ, we say that F_i is a strong T-Relevance feature.

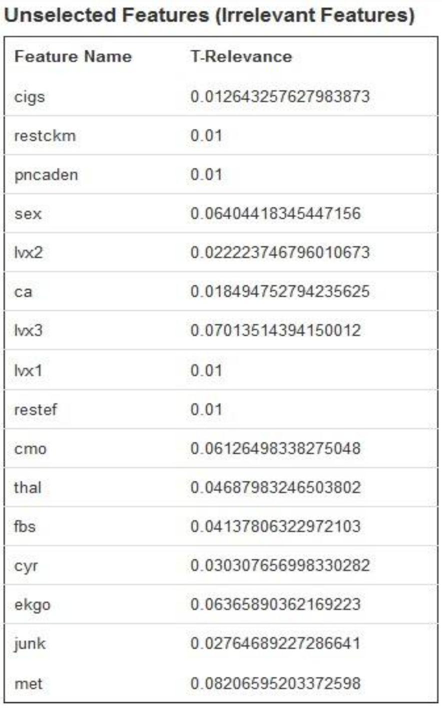


Feature Classification After T-Relevance Calculation

Selected Features (Relevant Features)

Feature Name	T-Relevance
lvx4	0.18065372510763003
rcadist	0.14693577298945426
trestbps	0.11474615071159641
ekgday	0.11317483303799542
exang	0.21921885052628132
lvf	0.09504551808814005
id	0.3429926963481016
painexer	0.18410458787218822
cathef	0.14678905923184288
rldv5	0.11906272127411502
rldv5e	0.11021037890983608
cday	0.09768578647365952
slope	0.2283922385278686

If $S(F_i, C)$ is lesser than a predetermined threshold θ, we say that F_i is a not an T-Relevance feature.



Unselected Features (Irrelevant Features)

Feature Name	T-Relevance
cigs	0.012643257627983873
restckm	0.01
pncaden	0.01
sex	0.06404418345447156
lvx2	0.022223746796010673
ca	0.018494752794235625
lvx3	0.07013514394150012
lvx1	0.01
restef	0.01
cmo	0.06126498338275048
thal	0.04687983246503802
fbs	0.04137806322972103
cyr	0.030307656998330282
ekgo	0.06365890362169223
junk	0.02764689227286641
met	0.08206595203372598

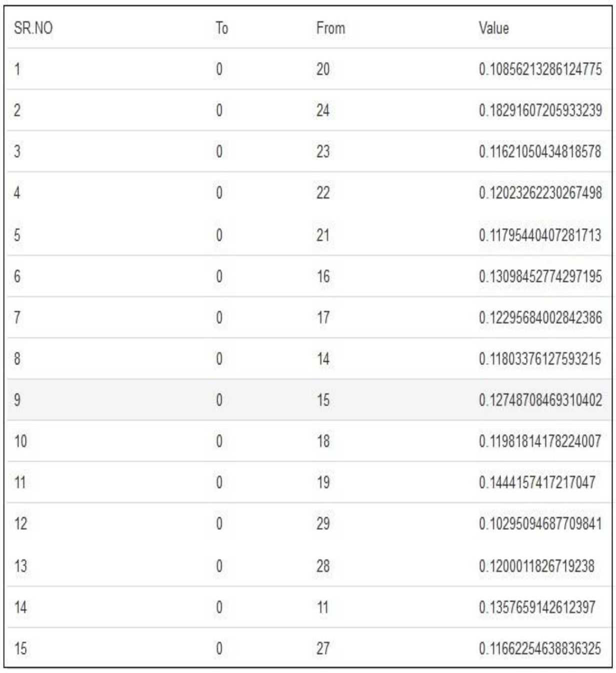
Step 2: Removal of redundant data:

Redundant features are assembled in a cluster and a representative feature can be taken out of the cluster.

Let S = {F1, F2, F_i, F_k<|F|} be a cluster of features.

If $\exists$ F_j $\in$S, (F_j) $\geq$ SU(F_i, C) $\wedge$ SU (F_i, F_j) >SU (F_i, C) is always corrected for each $F_i \in S\ (i \neq j)$, then F_i are redundant features with respect to the given F_j (i.e. each F_i is a F-Redundancy).

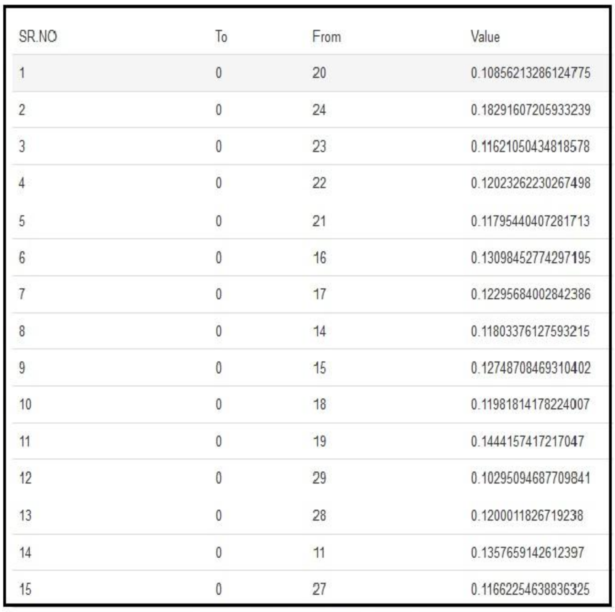
Redundant data are removed using MST.



SR.NO	To	From	Value
1	0	20	0.10856213286124775
2	0	24	0.18291607205933239
3	0	23	0.11621050434818578
4	0	22	0.12023262230267498
5	0	21	0.11795440407281713
6	0	16	0.13098452774297195
7	0	17	0.12295684002842386
8	0	14	0.11803376127593215
9	0	15	0.12748708469310402
10	0	18	0.11981814178224007
11	0	19	0.1444157417217047
12	0	29	0.10295094687709841
13	0	28	0.1200011826719238
14	0	11	0.1357659142612397
15	0	27	0.11662254638836325

MST BY PRIMS ALGORITHM

In the proposed system the redundant values are removed using both Kruskal’s and Prim’s. The complete graph reflects the correlations among the target-relevant features.

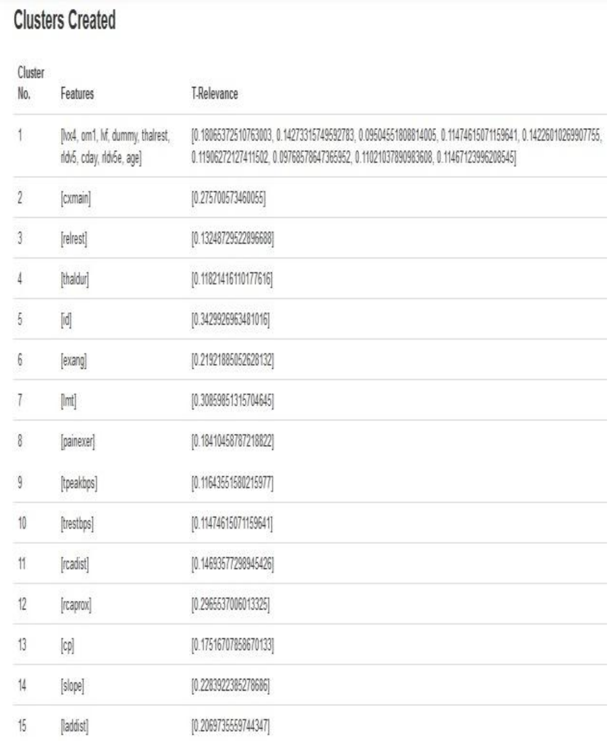


SR.NO	To	From	Value
1	0	20	0.10856213286124775
2	0	24	0.18291607205933239
3	0	23	0.11621050434818578
4	0	22	0.12023262230267498
5	0	21	0.11795440407281713
6	0	16	0.13098452774297195
7	0	17	0.12295684002842386
8	0	14	0.11803376127593215
9	0	15	0.12748708469310402
10	0	18	0.11981814178224007
11	0	19	0.1444157417217047
12	0	29	0.10295094687709841
13	0	28	0.1200011826719238
14	0	11	0.1357659142612397
15	0	27	0.11662254638836325

MST BY KRUSKAL'S ALGORITHM

Step 3: Feature Selection

Here, unnecessary edges can be removed. Each tree $T_j \in Forest$ shows a cluster that is denoted as $V(T_j)$, which is the vertex set of T_j . For each cluster $V(T_j)$, select a representative feature whose *T-Relevance* $SU(F_jR, C)$ is the highest. All F_jR (j = 1...|*Forest*|) consist of the final feature subset ∪ F_jR .



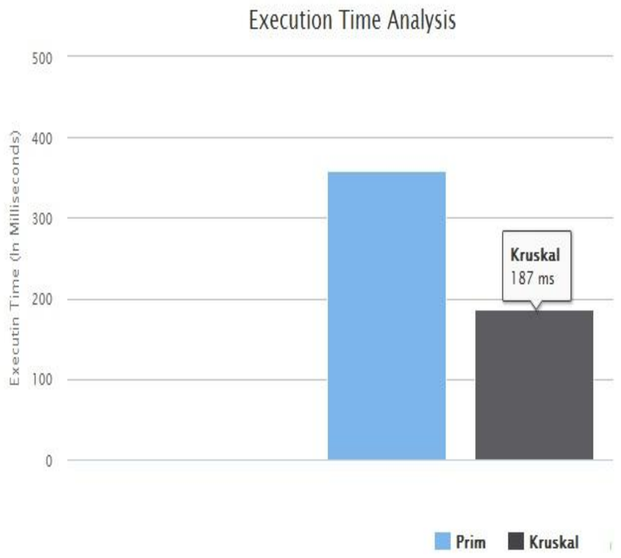
Clusters Created

Cluster No.	Features	T-Relevance
1	[lvx4, om1, lvf, dummy, thalrest, rldv5, cday, rldv5e, age]	[0.18065372510763003, 0.14273315749592783, 0.09504551808814005, 0.11474615071159641, 0.14226010269907755, 0.11906272127411502, 0.09768578647365952, 0.11021037890983608, 0.11467123996208545]
2	[cxmain]	[0.275700573460055]
3	[relrest]	[0.13248729522896688]
4	[thaldur]	[0.11821416110177616]
5	[id]	[0.3429926963481016]
6	[exang]	[0.21921885052628132]
7	[lmt]	[0.30859851315704645]
8	[painexer]	[0.18410458787218822]
9	[tpeakbps]	[0.11643551580215977]
10	[trestbps]	[0.11474615071159641]
11	[rcadist]	[0.14693577298945426]
12	[rcaprox]	[0.2965537006013325]
13	[cp]	[0.17516707858670133]
14	[slope]	[0.2283922385278686]
15	[laddist]	[0.2069735559744347]

VIII. ANALYSIS

The major amount of work for Algorithm involves the computation of SU values for T-Relevance and F-Correlation, which has linear complexity in terms of the number of instances in a given data set. The first part of the algorithm has a linear time complexity (m) in terms of the number of features m. Assuming $(1 \leq k \leq m)$ features are selected as relevant ones in the first part, when $k = 1$, only one feature is selected. Thus, there is no need to continue the rest parts of the algorithm, and the complexity. In proposed system the analysis of Prim's and Kruskal's algorithm is done and the best method is being selected based on the time complexity to generate and select the efficient features by eliminating redundant and irrelevant data.

Based on the analysis, Kruskal's algorithm is consider to be the efficient algorithm as compared to Prim's as the time complexity in Kruskal's is less than of Prim's.



IX. CONCLUSION

In this paper, we have proposed a clustering algorithm, FAST for high dimensional data. The algorithm includes (i) irrelevant features removal (ii) construction of a minimum spanning tree (MST) from, and (iii) partitioning the MST and selecting the representative features. Feature subset selection should be able to recognize and remove as much of the unrelated and redundant information. In the proposed algorithm, a cluster will be used to develop a MST for faster searching of relevant data from high dimensional data. Each cluster will be treated as a single feature and thus volume of data to be processed is drastically reduced. FAST algorithm will obtain the best proportion of selected features, the best runtime, and the best classification accuracy after eliminating redundant and irrelevant data.

Overall the system will be effective in generating more relevant and accurate features which can provide faster results.

REFERENCES

[1] 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 YEAR 2013.

[2] Karthikeyan.P, High Dimensional Data Clustering Using Fast Cluster Based Feature Selection , Int. Journal of Engineering Research and Applications, March 2014, pp.65-71.

[3] B.Swarna Kumari, M.Doorvasulu Naidu, Feature Subset Selection Algorithm for Elevated Dimensional Data By using Fast Cluster, In International Journal Of Engineering And Computer Science Volume 3 Issue Page No. 7102-7105, 7 July, 2014.

[4] Sumeet Pate, E.V. Ramana, A Search Engine Based On Fast Clustering Algorithm for High Dimensional Data, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), Volume 3, Issue 10, October 2014, ISSN: 2278 – 909X.

[5] Comparative study of various clustering techniques with FAST, International Journal of Computer Science and Mobile Computing, Volume 3, Issue 10, October 2014, ISSN: 2320-088X.

[6] Press W.H., Flannery B.P., Teukolsky S.A. and Vetterling W.T., Numerical recipes in C. Cambridge University Press, Cambridge, 1988.

[7] Das S., Filters, wrappers and a boosting-based hybrid for feature Selection, In Proceedings of the Eighteenth International Conference on Machine Learning, pp 74-81, 2001.

[8] Fayyad U. and Irani K., Multi-interval discretization of continuous-valued attributes for classification learning, In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp 1022-1027,

[9] 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.

[10] https://archive.ics.uci.edu/ml/datasets/Heart+Disease