International Journal of Research Studies in Science, Engineering and Technology
Volume 1, Issue 7, October 2014, PP 1-6
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

# A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data

**Yogesh R. Shepal**

Second Year (IV$^{TH}$ SEM),
M.Tech (CSE),
PRRMEC, Shabad, R.R. Dist. ,A.P.
*yogeshshepal@gmail.com*

**Ashraf Shaikh**

Assistant Professor, M.Tech (CSE),
PRRMEC, Shabad, R.R. Dist. ,A.P.
*ashrafanjum09@gmail.com*

**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. While the efficiency concerns the time required to find a subset of features, the effectiveness is related to the quality of the subset of features. The FAST algorithm works in two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature that is strongly related to target classes is selected from each cluster to form a subset of features. 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 clustering method. The efficiency and effectiveness of the FAST algorithm are evaluated through an empirical study.*

**Keywords:** *Feature subset selection, filter method, feature clustering, graph-based clustering*

## 1. INTRODUCTION

With the aim of choosing a subset of good features with respect to the target concepts, feature subset selection is an effective way for reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. Many feature subset selection methods have been proposed and studied for machine learning applications. They can be divided into four broad categories: the Embedded, Wrapper, Filter, and Hybrid approaches. The embedded methods incorporate feature selection as a part of the training process and are usually specific to given learning algorithms, and therefore may be more efficient than the other three categories. Traditional machine learning algorithms like decision trees or artificial neural networks are examples of embedded approaches. The wrapper methods use the predictive accuracy of a predetermined learning algorithm to determine the goodness of the selected subsets, the accuracy of the learning algorithms is usually high. However, the generality of the selected features is limited and the computational complexity is large. The filter methods are independent of learning algorithms, with good generality. Their computational complexity is low, but the accuracy of the learning algorithms is not guaranteed. The hybrid methods are a combination of filter and wrapper methods by using a filter method to reduce search space that will be considered by the subsequent wrapper. They mainly focus on combining filter and wrapper methods to achieve the best possible performance with a particular learning algorithm with similar time complexity of the filter methods. The wrapper methods are computationally expensive and tend to over fit on small training sets. The filter methods, in addition to their generality, are usually a good choice when the number of features is very large. Thus, focus will be on the filter method in this report. With respect to the filter feature selection methods, the application of cluster analysis has been demonstrated to be more effective than traditional feature

selection algorithms. In cluster analysis, graph-theoretic methods have been well studied and used in many applications. Their results have, sometimes, the best agreement with human performance. The general graph-theoretic clustering is simple: Compute a neighborhood graph of instances, then delete any edge in the graph that is much longer/shorter than its neighbors. The result is a forest and each tree in the forest represents a cluster. In the proposed study, a graph theoretic clustering method has been applied to features. In particular, the minimum spanning tree (MST) based clustering algorithms is adopted because it is assumed that data points are grouped around centers or separated by a regular geometric curve and have been widely used in practice.

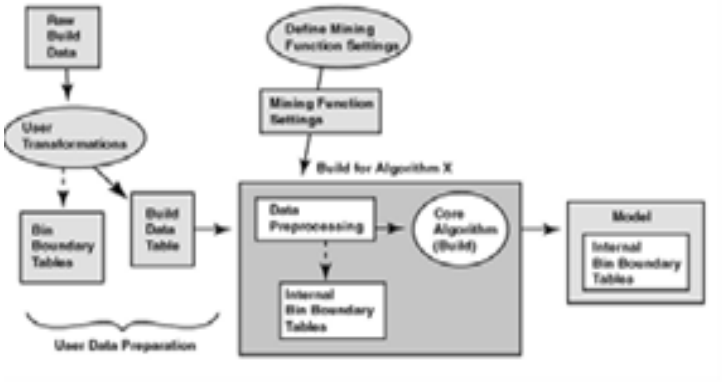


**Fig.** MST.

## 2. RELATED WORK

### 2.1 Text Classification Algorithm

The Text classification contains many problems, which has been widely studied in the data mining, machine learning, database, and information retrieval communities with applications in a number of diverse domains, such as target marketing, medical diagnosis, news group filtering, and document organization. The text classification technique assumes categorical values for the labels, though it is also possible to use continuous values as labels. The latter is referred to as the regression modeling problem. The problem of text classification is closely related to that of classification of records with set-valued features; however, this model assumes that only information about the presence or absence of words is used in a document. In reality, the frequency of words also plays a helpful role in the classification process, and the typical domain-size of text data (the entire size) is much greater than a typical set-valued classification problem.

### 2.2 Association Rule Mining

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule{mathrm{onions,potatoes}}Rightarrow{mathrm{burger}} found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production and bioinformatics. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

### 2.3 Feature Selection Algorithm

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more

information than the currently selected features, and irrelevant features provide no useful information in any context. Feature selection techniques are a subset of the more general field of feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or datapoints). Feature selection techniques provide three main benefits when constructing predictive models:

• Improved model interpretability,
• Shorter training times,
• Enhanced generalization by reducing over fitting.

Feature selection is also useful as part of the data analysis process, as shows which features are important for prediction, and how these features are related.

## 3. PROPOSED SYSTEM

Feature subset selection can be viewed as the process of identifying and removing as many irrelevant and redundant features as possible. This is because irrelevant features do not contribute to the predictive accuracy and redundant features do not redound to getting a better predictor for that they provide mostly information which is already present in other feature(s). Of the many feature subset selection algorithms, some can effectively eliminate irrelevant features but fail to handle redundant features yet some of others can eliminate the irrelevant while taking care of the redundant features.

Proposed FAST algorithm falls into the second group. Traditionally, feature subset selection research has focused on searching for relevant features. A well-known example is Relief which weighs each feature according to its ability to discriminate instances under different targets based on distance-based criteria function. However, Relief is ineffective at removing redundant features as two predictive but highly correlated features are likely both to be highly weighted. Relief-F extends Relief, enabling this method to work with noisy and incomplete data sets and to deal with multiclass problems, but still cannot identify redundant features.

### Advantages

- Good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.
- The efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset.
- Generally all the six algorithms achieve significant reduction of dimensionality by selecting only a small portion of the original features.
- The null hypothesis of the Friedman test is that all the feature selection algorithms are equivalent in terms of runtime.

FAST Algorithm is a classic algorithm for frequent item set mining and association rule learning over transactional databases. This FAST algorithm inbuiltly contains an algorithm called Apriori, which proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

**inputs**: $D(F1, F2, ..., Fm, C)$ - the given data set
$\theta$ - the T-Relevance threshold.

**output**: $S$ - selected feature subset .

**1** **for** $i = 1$ *to* $m$ **do**
**2** T-Relevance = **SU** ($Fi$, $C$)
**3** **if** T-Relevance > $\theta$ **then**
***4*** $S = S \cup \{Fi\}$;
**5** $G$ = NULL; //G is a complete graph

**6** **for** *each pair of features* $\{F'_i, F'_j\} \subset S$ **do**
**7** F-Correlation = **SU** ($F'_i$, $F'_j$ )
**8** *Add F'i and/or F' j to G with* F-Correlation as the weight of the corresponding edge;
**9** minSpanTree = *Prim* (*G*); //Using Prim Algorithmtogenerate the min spanning tree
**10** Forest = minSpanTree
**11** **for** *each edge* $E_{i,j} \in$ Forest **do**
**12** **if SU** ($F'_i$, $F'_j$ ) < **SU(**$F'_i$, C**)** $\wedge$ **SU**($F'_i$, $F'_j$) < **SU(**$F'_j$, C**) then**
**13** Forest = Forest − $E_{ij}$
**14** $S = \phi$
**15** **for** *each tree* $T_i \in$ Forest **do**
16 $F^j_R$ = argmax $F_k \in T_i$ **SU(**$F'_k$, C**)**
**17** $S = S \cup \{F^j_R\}$;
**18** return *S*

The major amount of work for Algorithm 1 involves the computation of $SU$ values for *T- elevance* 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 $O(m)$ in terms of the number of features $m$. Assuming $k(1 \le k \le m)$ 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 is $O(m)$. When $1 < k \le m$, the second part of the algorithm firstly constructs a complete graph from relevant features and the complexity is $O(k2)$, and then generates a MST from the graph using Prim algorithm whose time complexity is $O(k2)$. The third part partitions the MST and chooses the representative features with the complexity of $O(k)$. Thus when $1 < k \le m$, the complexity of the algorithm is $O(m+k2)$. This means when $k \le \sqrt{m}$, FAST has linear complexity $O(m)$, while obtains the worst complexity $O(m2)$ when $k = m$. However, $k$ is heuristically set to be $\lfloor \sqrt{m} - \lg m \rfloor$ in the implementation of FAST. So the complexity is $O(m - \lg 2m)$, which is typically less than $O(m2)$ since $\lg 2m < m$. This can be explained as follows. Let $f(m) = m - \lg 2m$, so the derivative $f'(m) = 1 - 2 \lg e/m$, which is greater than zero when $m > 1$. So $f(m)$ is a increasing function and it is greater than $f(1)$ which is equal to 1, i.e., $m > \lg 2m$, when $m > 1$. This means the bigger the $m$ is, the farther the time complexity of FAST deviates from $O(m2)$. Thus, on high dimensional data, the time complexity of FAST is far more less than $O(m2)$. This makes FAST has a better runtime performance with high dimensional data.

**Analysis**

The proposed FAST algorithm logically consists of tree steps: (i) removing irrelevant features, (ii) constructing a MST from relative ones, and (iii) partitioning the MST and selecting Representative features.

For a data set $D$ with $m$ features $F = \{F1, F2, ..., Fm\}$ and class $C$, we compute the *T-Relevance* $SU(Fi, C)$ value for each feature $Fi$ $(1 \le i \le m)$ in the first step. The features whose $SU(Fi, C)$ values are greater than a predefined threshold $\theta$ comprise the target-relevant feature subset

$F' = \{F'1, F'2, ..., F'k\}$ $(k \le m)$.

In the second step, we first calculate the *F-Correlation* $SU(F'i, F'j)$ value for each pair of features $F'i$ and $F'j$ $(F'i, F'j \in F' \wedge i \neq j)$. ). Then, viewing features $F'i$ and $F'j$ as vertices and $SU(F'i, F'j)$ $(i \neq j)$ as the weight of the edge between vertices $F'i$ and $F'j$, a weighted complete graph $G = (V,E)$ is constructed where

$V = \{F'i \mid F'i \in F' \wedge i \in [1, k]\}$ and $E = \{(F'i, F'j) \mid (F'i, F'j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$. As symmetric uncertainty is symmetric further the *F-Correlation* $SU(F'i, F'j)$ is symmetric as well, thus $G$ is an undirected graph. The complete graph $G$ reflects the correlations among all the target-relevant features. Unfortunately, graph $G$ has $k$ vertices and $k(k-1)/2$ edges. For high dimensional data, it is

heavily dense and the edges with different weights are strongly interweaved. Moreover, the decomposition of complete graph is NP-hard.

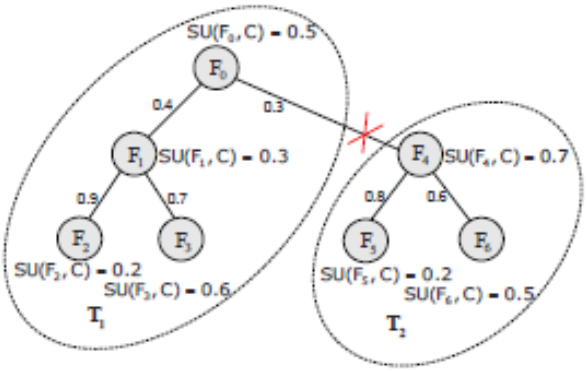


**Fig.** *Example of Clustering*

**System Architecture**

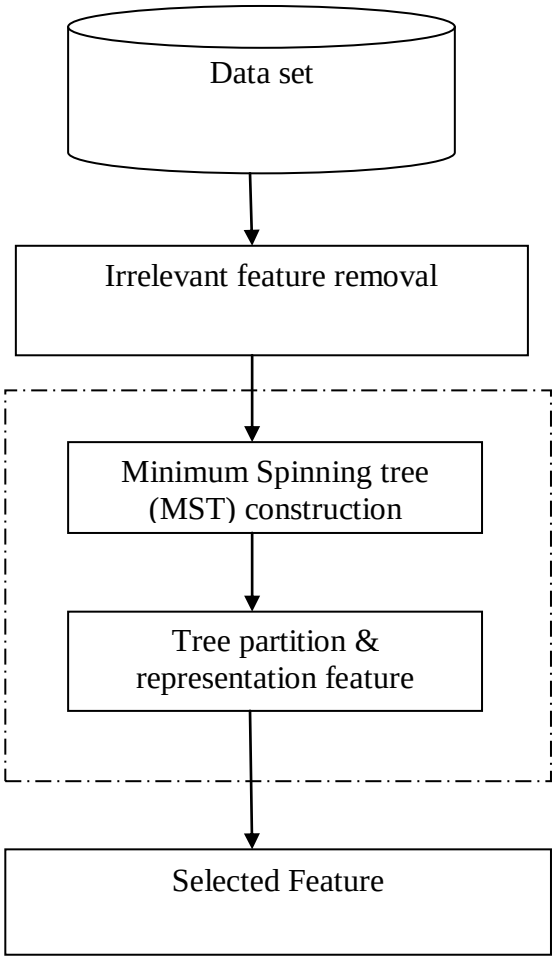


**Fig.** *System Architecture*

## 4. CONCLUSION

The algorithm involves

1) removing irrelevant features.
2) constructing a minimum spanning tree from relative ones.
3) partitioning the MST and selecting representative features.

In the proposed algorithm, a cluster consists of features. Each cluster is treated as a single feature and thus dimensionality is drastically reduced. Generally, the proposed algorithm obtained the best proportion of selected features, the best runtime, and the best classification accuracy confirmed the conclusions.

A novel clustering-based feature subset selection algorithm for high dimensional data is presented. The algorithm involves removing irrelevant features, constructing a minimum spanning tree from relative ones, and partitioning the MST and selecting representative features. In the proposed algorithm, a cluster consists of features. Each cluster is treated as a single feature and thus dimensionality is drastically reduced.

The performance of the proposed algorithm has been compared with those of the five well-known feature selection algorithms FCBF, CFS, Consist, and FOCUS-SF on the publicly available image, microarray, and text data from the four different aspects of the proportion of selected features, runtime, classification accuracy of a given classifier, and the Win/Draw/Loss record.

Generally, the proposed algorithm obtained the best proportion of selected features, the best runtime, and the best classification accuracy for Naive, and RIPPER, and the second best classification accuracy for IB1.

## REFERENCES

[1] Battiti R., Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks, 5(4), pp 537- 550, 1994.

[2] Chanda P., Cho Y., Zhang A. and Ramanathan M., Mining of Attribute Interactions Using Information Theoretic Metrics, In Proceedings of IEEE international Conference on Data Mining Workshops, pp 350-355, 2009.

[3] Chanda P., Cho Y., Zhang A. and Ramanathan M., Mining of Attribute Interactions Using Information Theoretic Metrics, In Proceedings of IEEE international Conference on Data Mining Workshops, pp 350-355, 2009.

[4] 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

[5] Butterworth R., Piatetsky-Shapiro G. and Simovici D.A., On Feature Selection through Clustering, In Proceedings of the Fifth IEEE international Conference on Data Mining, pp 581-584, 2005.

[6] Almuallim H. and Dietterich T.G., Algorithms for Identifying Relevant Features, In Proceedings of the 9th Canadian Conference on AI, pp 38-45, 1992.

[7] Almuallim H. and Dietterich T.G., Learning boolean concepts in the presence of many irrelevant features, Artificial Intelligence, 69(1-2), pp 279-305, 1994.

[8] Arauzo-Azofra A., Benitez J.M. and Castro J.L., A feature set measure based on relief, In Proceedings of the fifth international conference on Recent Advances in Soft Computing, pp 104-109, 2004.

[9] Baker L.D. and McCallum A.K., Distributional clustering of words for text classification, In Proceedings of the 21st Annual international ACM SIGIR Conference on Research and Development in information Retrieval, pp 96- 103, 1998.

[10] Bell D.A. and Wang, H., A formalism for relevance and its application in feature subset selection, Machine Learning, 41(2), pp 175-195, 2000.