ISSN 2278-3083
Volume 4, No.2, March - April 2015
International Journal of Science and Applied Information Technology
Available Online at http://www.warse.org/ijsait/static/pdf/file/ijsait03422015.pdf

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

G. Hemanth Kumar[1], Ch. Sreenubabu[2]
[1]PG Scholar, Dept of CSE, GMRIT, Rajam, India, hemanthkumar.gullipalli@gmail.com
[2]Associate Professor, Dept of CSE, GMRIT, Rajam, India, sreenubabu.ch@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. Based on these criteria, a fast clustering-based feature subset selection algorithm (FAST) is proposed, it involves (i) removing irrelevant features (ii) constructing a minimum spanning tree from relative ones and (iii) partitioning the MST and selecting representative features.

Index Terms: Feature subset selection, Minimum-spanning tree, Redundant features, Relevant features.

1. INTRODUCTION

It is widely recognized that a large number of features can adversely affect the performance of inductive learning algorithms, and clustering is not an exception. However, while there exists a large body of literature devoted to this problem for supervised learning task, feature selection for clustering has been rarely addressed. The problem appears to be a difficult one given that it inherits all the uncertainties that surround this type of inductive learning. Particularly, that there is not a single performance measure widely accepted for this task and the lack of supervision available.

1.1 Feature Selection

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 data points). The archetypal case is the use of feature selection in analyzing DNA microarrays, where there are many thousands of features, and a few tens to hundreds of samples. 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.

Many feature subset selection methods have been proposed in Figure 1.

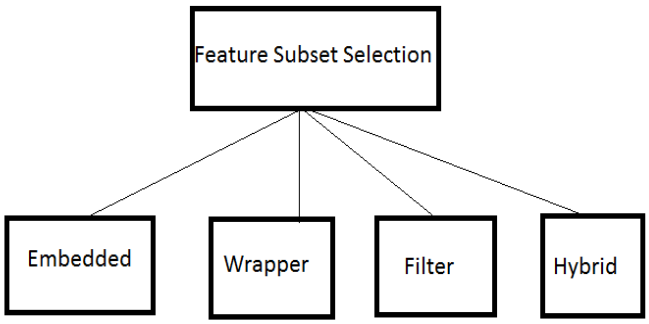

Figure 1.Categories of Feature Subset Selection

Figure.1 shows the categorization of selection methods studied for machine learning applications the Embedded, Wrapper, Filter, and Hybrid approaches.

1.2 Embedded

The embedded methods incorporate feature selection as a part of the training process. Specific to learning algorithms [4]. Computationally less than wrapper methods.

1.3 Wrapper

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. Computational complexity is large.

1.4 Filter

The filter methods are independent of learning algorithms with good generality. Computational cost is low[6], [7].The accuracy of learning algorithm is not guaranteed

1.5 Hybrid

The hybrid methods are a combination of filter and wrapper methods [8], [9]. 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 fit on small training sets.

2. FEATURE SUBSET SELECTION ALGORITHM ANALYSIS

Definitions

Relevant Feature: F_i is relevant to the target concept C if and only if there exists some s_i', f_i and c, such that for probability $p(S_i'= s_i', F_i = f_i)>0$, $p(C=c| S_i'= s_i, F_i = f_i)\neq (C=c| S_i'= s_i')$.Otherwise F_i is an irrelevant feature.

Markov blanket: Given a feature $F_i \in F$, let $M_i \subset F$ ($F_i \notin M$),M_i is said to be Markov blanket for F_i if and only if

$$p(F-M_i-\{F_i\}, C|F_i,M_i)=p(F-M_i-\{F_i\}, C|M_i).$$

Redundant Feature: Let S be a set of features, a feature in S is redundant if and only if it has a Markov blanket in S.

Relative features have strong correlation with target concept so are always necessary for best subset, while redundant features are not because they are completely correlated with each other. Moreover good features subsets contain features highly correlated with (predictive of) the class yet uncorrelated with (predictive of) each other

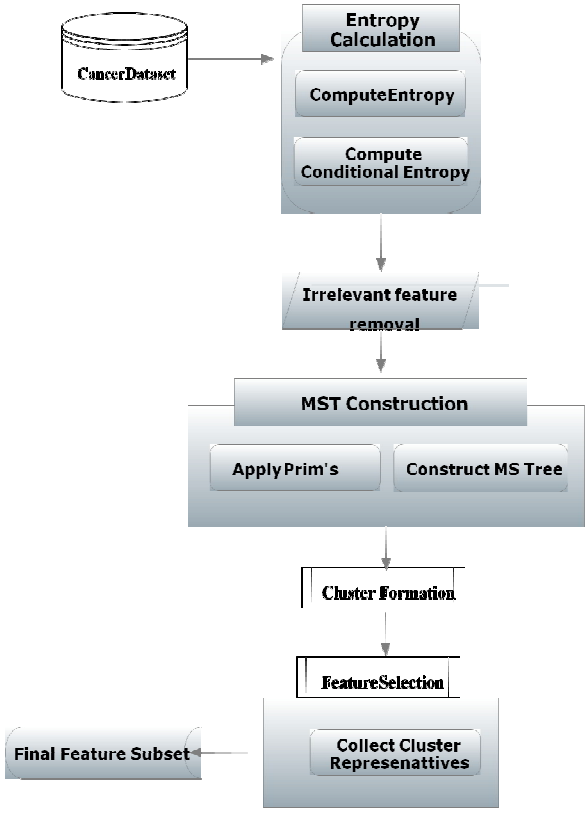

Figure 2: Framework of the FAST Algorithm

2.1 Load Data and Classify

Load the data into the process. Cancer dataset (Figure.2) has taken to preprocess for removing missing values, noise and outliers. Then the given dataset must be converted into the arff format which is the standard format for WEKA toolkit. From the arff format, only the attributes and the values are extracted and stored into the database. By considering the last column of the dataset as the class attribute and select the distinct class labels from that and classify the entire dataset with respect to class labels.

2.2 Information Gain Computation

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.

To find the relevance of each attribute with the class label, Information gain is computed in this module. This is also said to be Mutual Information measure. 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) 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

The symmetric uncertainty is defined as follows:

$$\text{Gain}(X|Y) = H(X) - H(X|Y)$$

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

To calculate gain, we need to find the entropy and conditional entropy values. The equations for that are given below:

$$H(X) = -\sum_{x \in X} p(x)\log_2 p(x)$$

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

Where p(x) is the probability density function and p(x|y) is the conditional probability density function.

2.3 T-Relevance Calculation

The relevance between the feature $F_i \in F$ and the target concept C 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, we say that F_i is a strong T-Relevance feature.

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

After finding the relevance value, the irrelevant attributes will be removed with respect to the threshold value.

2.4 F-Correlation Calculation

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

The equation symmetric uncertainty, which is used for finding the relevance between the two attributes with respect to each label. SU(F'_i ,F'_j)($i \neq j$) as the weight of the edge between vertices F'_i and F'_j.

2.5 MST Construction

With the F-Correlation value computed above, the Minimum Spanning tree is constructed. For that, we use Prim's algorithm which form MST effectively.

Prim'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).

2.5.1 Description:

- Create a forest F (a set of trees), where each vertex in the graph is a separate tree.
- Create a set S containing all the edges in the graph
- 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.

The sample tree is as follows,

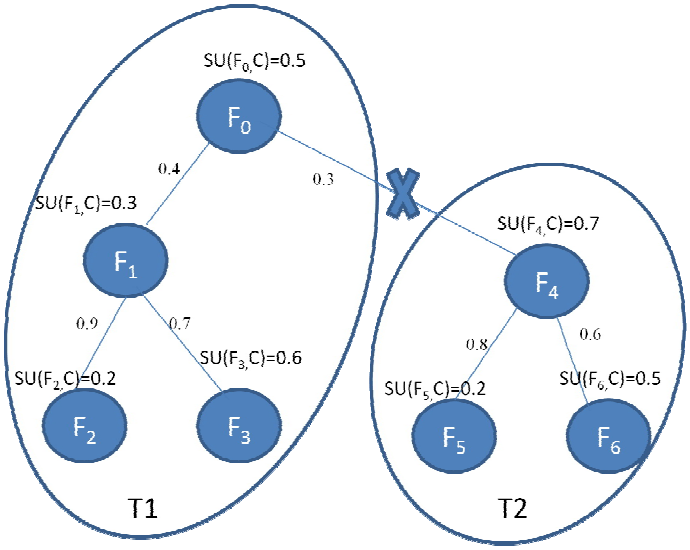

Figure 3:Example of Clustering features

In this tree, the vertices represent the relevance value and the edges represent the F-Correlation value.

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 interwoven. Moreover, the decomposition of complete graph is NP-hard. Thus for graph G, we build an MST, which connects all vertices such that the sum of the weights of the edges is the minimum, using the well known Prim's algorithm. The weight of edge (F'_i , F'_j)is F-Correlation SU(F'_i , F'_j).

2.6 Cluster Formation

After building the MST, in the third step, we first remove the edges(Figure.3) whose weights are smaller than both of the T-Relevance SU(F'_i, C) and SU(F'_j, C), from the MST.

After removing all the unnecessary edges, a Forest is obtained. Each tree T_j € Forest represents a cluster that is denoted as V (T_j), which is the vertex set of T_j as well. The features in each cluster are redundant, so for each cluster V (T_j) we choose a representative feature F^j_R whose T-Relevance SU(F^j_R,C) is the greatest.

FAST Algorithm:

Inputs: D($F1$, $F2$, ..., Fm, C) - the given data set
θ- the T-Relevance threshold.

Output: S - selected feature subset.

//==== Part 1 : Irrelevant Feature Removal ====
for i = 1 to m **do**
T-Relevance = SU (F_i , C)
if T-Relevance >θ
then
S = S ∪ {F_i } ;

//==== Part 2 : Minimum Spanning Tree Construction ====
G = NULL; //G is a complete graph
foreach pair of features {F'_i , F'_j } ⊂ S **do**
F-Correlation = SU (F'_i, $F'j$)
//Add F_i' and/or F_j' to G with F-Correlation as the weight of the corresponding edge.
minSpanTree = Prim (G); //Using Prim Algorithm to generate the minimum spanning tree Selection ====

//==== Part 3: Tree Partition and Representative Feature==
Forest = minSpanTree
foreach edge E_{ij}∈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 = ϕ
for each tree Ti ∈Forest **do**
F^j_R= argmax$_{F'k∈Ti}$SU(F',C)
S = S ∪ {F^j_R };
Return S;//Feature Subset

3. RESULTS

Some of the implementation results shown here.

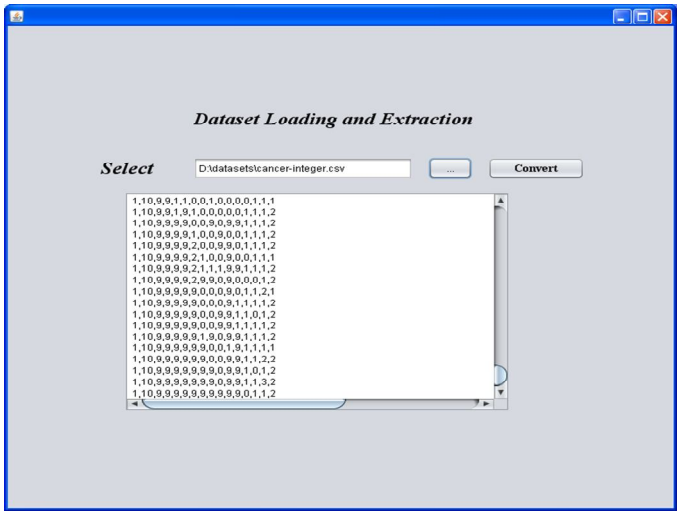

Figure 4.Dataset Loading

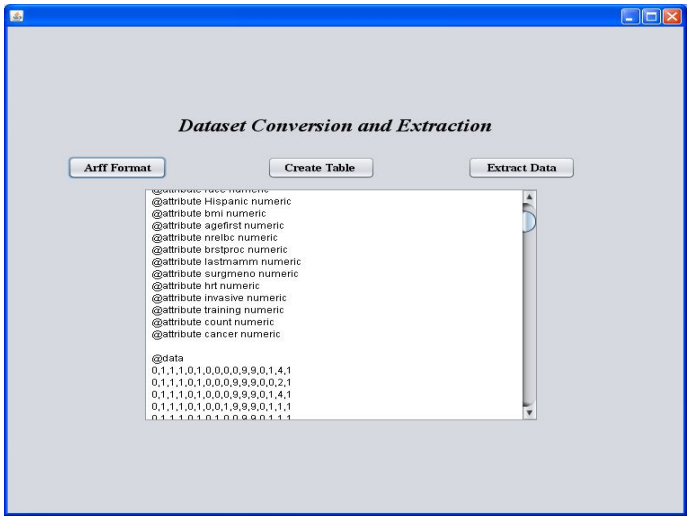

Figure 5. Dataset Conversation

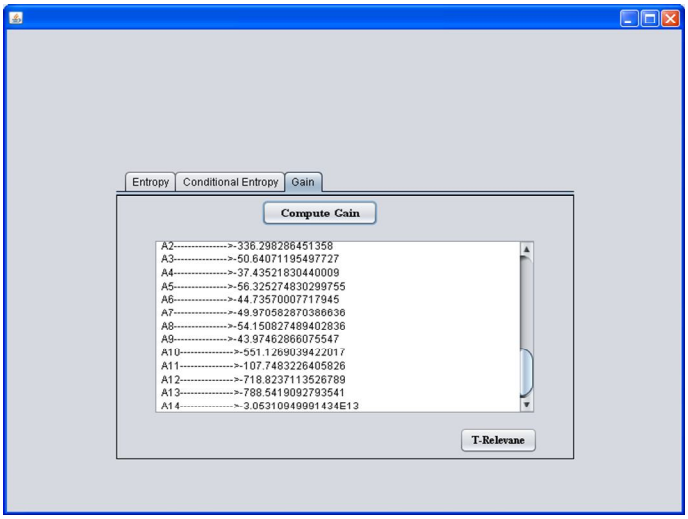

Figure 6. Gain Calculation

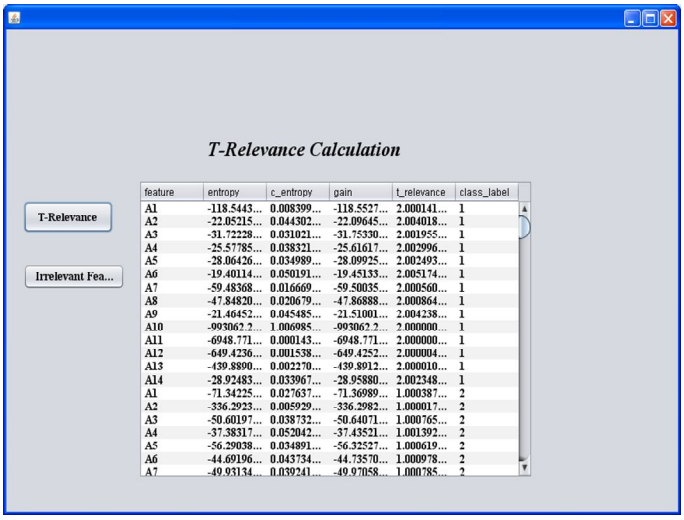

Figure 7. T-Relevance

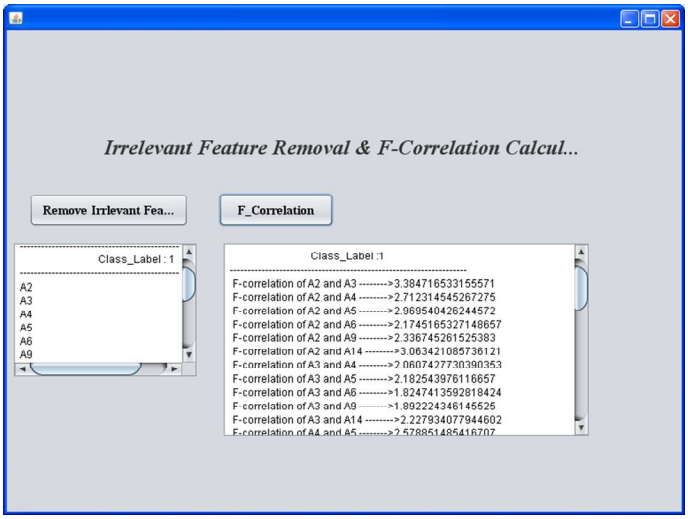

Figure 8.Irrelevancy & F-Correlation

4. CONCLUSION

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. We have applied FAST algorithm on different datasets. For the future work, we plan to explore different types of correlation measures.

5. REFERRENCES

[1]. 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.

[2] 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.

[3] Dash M., Liu H. and Motoda H., **Consistency based feature Selection,** In Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, pp 98-109, 2000.

[4] Yu L. and Liu H., **Redundancy based feature selection for microarray data,** In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 737-742, 2004.

[5] Yu L. and Liu H., Feature selection for high-dimensional data: **a fast correlation-based filter solution**, in Proceedings of 20th International Conference on Machine Leaning, 20(2), pp 856-863, 2003.

[6] Dash M. and Liu H., **Feature Selection for Classification**, Intelligent Data Analysis, 1(3), pp 131-156, 1997.

[7] Yu L. and Liu H., **Efficiently handling feature redundancy in high dimensional data,** in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03). ACM, New York, NY, USA, pp 685-690, 2003.

[8] Xing E., Jordan M. and Karp R., Feature selection for high-dimensional genomic microarray data, In Proceedings of the Eighteenth International Conference on Machine Learning, pp 601-608, 2001.

[9]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.