International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 10, October 2014

Search Engine Based On A Fast Clustering Algorithm for High Dimensional Data

SUMEET SUBHASH PATE[1]
P.G Scholar, Dept of CSE, VVIT,
Chevella, Hyderabad, India .

E.V.RAMANA[2]
Assistant Professor, Dept of CSE, VVIT,
Chevella, Hyderabad, India.

Abstract *— We come across various search engines, but giving different Keywords to the Search engine and getting the best results is still a tedious process because we get redundant and irrelevant data. This is mainly due to the selected features that are used to find the results. The paper aims at proposing a search engine that uses the fast clustering algorithm for removing this 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 framework is introduced that decouples relevance analysis and redundancy analysis. We develop a clustering -based method for relevance and redundancy analysis for feature selection and perform searching based on the selected features. While the efficiency concerns the time required to find a subset of features, the effectiveness determines the quality of the subset of features. Based on these criteria, a fast clustering-based feature selection algorithm, FAST, has been selected to be used for the search engine in our proposed paper. The clustering-based strategy of FAST 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 complicity. The features selected by fast are input dataset for the proposed search engine, thus as a result the time complexity is reduced for the data search which is proved by the obtained results.*

Index Terms *— Feature subset selection, FAST algorithm, Minimum spanning tree, Search engine.*

I. INTRODUCTION

Data mining refers uses a variety of techniques to identify nuggets of information or decision-making knowledge in bodies of data, and extracting them in such a manner that they can be directly put into use in the areas such as decision support, estimation prediction and forecasting. The data is often huge, but as it 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 [1]. Feature subset selection is an effective way for reducing dimensionality, removing irrelevant data, increasing learning accuracy and improving result comprehensibility with the aim of choosing a subset of good features with respect to the target concepts,. Many feature subset selection methods are already proposed and studied for machine learning applications which can be divided into four broad categories namely the Wrapper, Embedded, Filter, and Hybrid approaches. The embedded methods incorporate feature selections as a part of the training process and are usually specific to given learning algorithms, and therefore are more efficient than the other three categories [2]. Based on the MST method, we propose search engine based on a fast searching clustering-based feature Selection algorithm (FAST). The FAST algorithm works in the following two steps. Initially clusters are formed by using graph-theoretic clustering methods. In the second step, the features that are strongly related to target classes are selected from each cluster to generate the final subset of features. Features selected from different clusters are relatively independent; the clustering-based strategy of FAST has higher probability of producing a subset of useful and independent features [5].

II. RELATED WORK

Traditionally, feature subset selection generally focused on searching for relevant features only 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. But, Relief is ineffective in removing redundant features as the two predictive but highly correlated features are likely to be highly weighted. Relief-F

ISSN: 2278 – 909X

[7] 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. Relief algorithm assigns a weight to each feature that reflects its ability to distinguish among the classes, and then selects those features with weights that exceed a user specified threshold. 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.

Also as discussed earlier different approaches are 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. The accuracy of the learning algorithms [2] 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

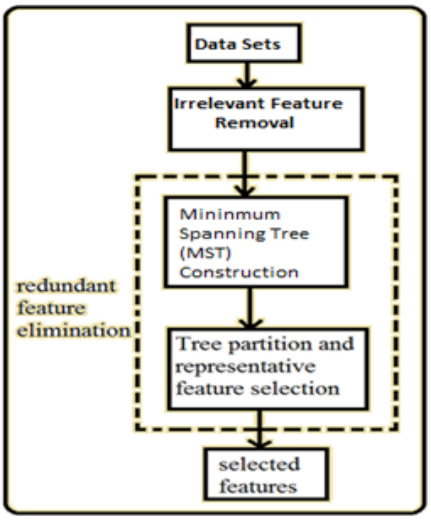

Fig 1: *Feature subset selection process.*

The *symmetric uncertainty* (*SU*) [8] is derived from the mutual information by normalizing it to the entropies of feature values or feature values and target classes Therefore; I 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

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

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

$$H(X) = -\sum_{x \in X} p(x) log2p\,(x)$$

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

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

Where H($X|\ Y$) is the conditional entropy.

Given that ($X,\ Y$)be the symmetric uncertainty of variables *X* and *Y*, the relevance *T-Relevance*[9] 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 [2].

T-Relevance - The relevance between the feature $Fi \in F$ 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.

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

F-Redundancy **-** Let *S*= {*F*1, *F*2, *Fi, Fk*<|*F*|} be a cluster of features.

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

R-Feature **-** A feature *Fi*$\in$ *S*= {*F*1, *F*2... *Fk*} (*k* < |*F* |) is a representative feature of the cluster *S* (i.e. *Fi* is an *R-Feature)* if and only if, *Fi* = argmax*Fj*$\in$*S SU* (*Fj, 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.

IV. MST CONSTRUCTION

With the F-Correlation value computed above, 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

ISSN: 2278 – 909X

minimized. If the graph is not connected, then it finds a minimum spanning forest (a minimum spanning tree for each connected component).

Description:

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. The sample tree is as follows [4],

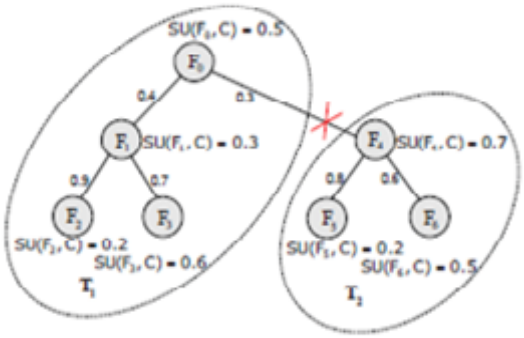

Fig 2: *Clustering with MST construction*

V. FAST ALGORITHM

Inputs: D (F1, F2 … Fm, C) (High Dimensional Dataset).
Output: S-Selected feature subset for searching.

Part1: 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′= {F1′, F2′…F_k′} (k<=M)

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

Part2: 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'j*} ⊂ *S* **do**
7. F-Correlation = **SU** (*F'i* , *F'j*)
8. *Add 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.

Part3: Feature selection.

10. Forest = minSpanTree
11. **for** *each edge* $E_{i,j}$ ∈ Forest **do**
12. **if SU** (F_i', F_j') < **SU**(F_i' , C) ∧ **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_R^j = argmax $F_k \in T_i SU$ (*F'k*, C)
17. $S = S \cup \{ F_R^j \}$;
18. **returns** *S*.
19. Use the selected feature(**S**) for searching relevant information.

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 while the final part of the algorithm is concerned with feature selection based on the value of the Forest.

The detailed classification and structural output of the algorithm is described below.

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 ($1 \le i \le m$).

B. Second step:

Here we first 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 *Fi'* and *Fj'* a weighted complete graph *G= (V,E)* is constructed which is an undirected graph. The complete graph reflects the correlations among the target-relevant feature [3].

C. Third step:

Here unnecessary edges can be removed each tree *Tj* ∈ *Forest* shows a cluster that is denoted as *V (Tj),* which is the vertex set of *Tj*. For each cluster *V (Tj),* select a representative feature who's *T-Relevance SU (FjR, C)* is the highest. All *FjR* (*j* = 1...|*Forest*|) consist of the final feature subset ∪ *FjR* .

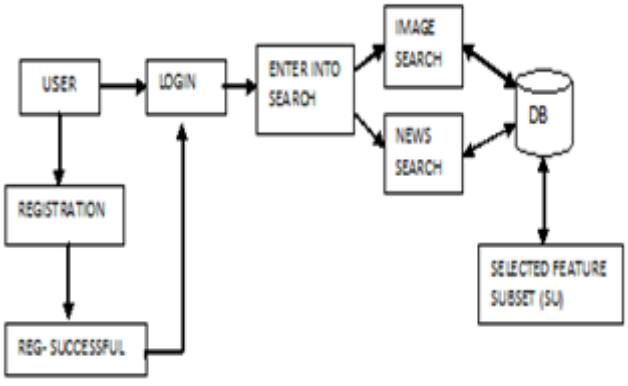

Fig 3: *Data Search.*

VI. EXPERIMENTAL RESULT

The proposed search engine discussed in our paper contains input datasets as images, text, and microarrays. It creates a clustering tree depending on the domain that the admin selects while uploading the file. Our proposed system then stores the file in the cluster by using the minimum

ISSN: 2278 – 909X

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. We have used FAST algorithm in our proposed search engine as FAST 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 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) [2]

Different algorithms such as FAST, FCBF, ReliefF determined following results,

TABLE 1: *Classification Accuracy & Runtime Complexity*

DATASETS \ ALGORITHM		FAST	FCBF	ReliefF
ACCURACY (in %)	IMAGE	81.7	80.93	77.12
	TEXT	81.38	83.15	72.59
	MICRO-ARRAY	83.77	79.48	74.35
RUN-TIME COMPLEXITY (in ms)	IMAGE	1520	4090	8213
	TEXT	6989	8808	107528
	MICRO-ARRAY	1468	1169	8059

The above table shows the classification accuracy of C4.5 classifier along with the run-time complexity.

For image data, the classification accuracy of C4.5 has been improved for FAST, as compared to FCBF and ReliefF.

For microarray data, the classification accuracy of C4.5 has been improved. FAST ranks 1 with a margin of 3.92 to the second best accuracy 79.85% of CFS.

For text data, the classification accuracy of C4.5 has been 81.70% slightly closer to that of the FCBF.

FAST obtains the rank 1 with a margin of 1.26% to the second best accuracy 82.28% of FCBF.

FAST is consistently faster than all other algorithms. The runtime of FAST is, 76.5% of that of ReliefF, and 7.5% of that of FCBF, respectively.

VII. CONCLUSION

In this paper, we have proposed a search engine based on a FAST clustering algorithm 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 use of algorithm for our search engine, a cluster is used to develop a MST for faster searching of relevant data from high dimensional data. Each cluster is treated as a single feature and thus volume of data to be processed is drastically reduced. FAST algorithm obtained the best proportion of selected features, the best runtime, and the best classification accuracy and thus it is being used in our paper to implement the proposed system.

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

REFERENCES

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

[2] Qinbao Song, Jingjie Ni and Guangtao Wang, A Fast Clustering-Based Feature Subset Selection Algorithm For High Dimensional Data, In IEEE Transactions On Knowledge And Data Engineering Vol:25 No:1 Year 2013.

[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] Manjuparkavi A1, Arokiamuthu M2, Cluster Based Speed and Effective Feature Extraction for Efficient Search Engine, In IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014.

[5] Pat Langley, Selection of Relevant Features in Machine Learning, In Proceedings of the AAAI Fall Symposium on Relevance [1994] New Orleans.

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

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

[8] William W. Cohen & Haym Hirsh, Irrelevant Features and the subset selection Problem, In Proceedings of Eleventh International Conference, 121-129, Morgan Kaulimann Publishers, San Francisco, CA.

[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.

ISSN: 2278 – 909X