A Novel Clustering-Based Feature Subset selection algorithm for High dimensional data

A . Chandra Obula Reddy[1.a], C . Ravi[2.b], A . Ananda Kumar Yadav[3.c]
[1]Asst.Professor, Annamacharya Institute of Technology and Sciences, Kadapa.
[2]PG Student, Annamacharya Institute of Technology and Sciences, Kadapa.
[3]PG Student, Sri Venkateswara Institute of Science & Technology, Kadapa.

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 selection algorithm, FAST, is proposed and experimentally evaluated in this Paper. 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. Extensive experiments are carried out to compare FAST and several representative feature selection algorithms, namely, FCBF, Relieff, CFS, Consist, and FOCUS-SF, with respect to four types of well-known classifiers, namely, the probability-based Naive Bayes, The tree-based C4.5, the instance-based IB1, and the rule-based RIPPER before and after feature selection. The results, on 35 publicly Available real-world high dimensional image, microarray, and text data, demonstrate that FAST not only produces smaller subsets of Features but also improves the performances of the four types of classifiers.

Index Terms—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, we will focus on the filter method in this paper. 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, employed the distributional

clustering of words to reduce the dimensionality of text data. 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 (according to some criterion) than its neighbors. The result is a forest and each tree in the forest represents a cluster. In our study, we apply graph theoretic clustering methods to features. In particular, we adopt the minimum spanning tree (MST) based clustering algorithms, because they do not assume that data points are grouped around centers or separated by a regular geometric curve and have been widely used in practice.

2. RELATED WORK

Feature subset selection can be viewed as the process of identifying and removing as many irrelevant and redundant features as possible. This is because: (i) irrelevant features do not redound to getting a better predictor for that they provide mostly information which is already present in other feature. Of the many feature subset selection algorithms, some can effectively eliminate irrelevant features but fail to handle redundant features. Traditionally, feature subset selection research has focused on searching for relevant features. A well known example is Relief.

Relief-F extends Relief, enabling this method to work with noisy and incomplete data sets and to deal with multi-class problems, but still cannot identify redundant features. However, along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms, and thus should be eliminated as well, FCBF and CMIM are examples that take into consideration the redundant features. CFS is achieved by the hypothesis that a good feature subset is one that contains features highly correlated with the target, yet uncorrelated with each other. FCBF is a fast filter method which can identify relevant features as well as redundancy among

Relevant features without pair wise correlation analysis. CMIM iteratively picks features which maximize their mutual information with the class to predict, conditionally to the response of any feature already picked. Different from these algorithms, our proposed FAST algorithm employs clustering based method to choose features, hierarchical clustering to remove redundant features. Quite different from these hierarchical clustering based algorithms, our proposed FAST algorithm uses minimum spanning tree based method to cluster features. Meanwhile, it does not assume that data points are grouped around centers or separated by a regular geometric curve. Moreover, our proposed FAST does not limit to some specific types of data.

3. FEATURE SUBSET SELECTION ALGORITHM

3.1 Framework and definitions

The feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, "good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.

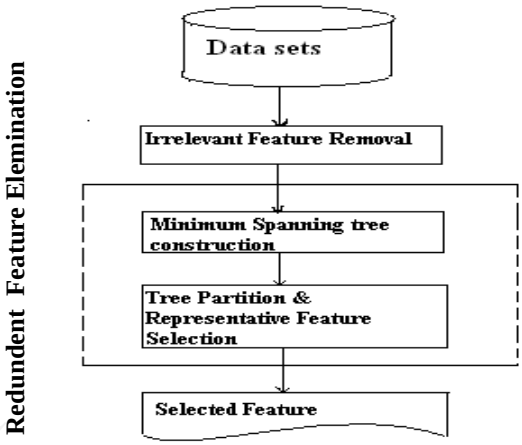

Fig 1: Framework of the proposed feature subset selection algorithm

Keeping these in mind, we develop a novel algorithm which can efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset. We achieve this through a new feature selection framework (shown in Fig.1) which composed of the two connected components of irrelevant feature removal and redundant feature elimination. The former obtains features relevant to the target concept by eliminating irrelevant ones, and the latter removes redundant features from relevant ones via choosing representatives from different feature clusters, and thus produces the final subset. The irrelevant feature removal is straightforward once the right relevance measure is defined or selected, while the redundant feature elimination is a bit of sophisticated. In our proposed FAST algorithm, it involves (i) the construction of the minimum spanning tree (MST) from a weighted complete graph; (ii) the partitioning of the MST into a forest with each tree representing a cluster; and (iii) the selection of representative features from the clusters. the traditional definitions of relevant and redundant features, a definition of relevant features. Suppose F to be the full set of features, $Fi\varepsilon\ F$ be a feature, $Si = F - \{Fi\}$ and $S'_i \subseteq Si$. Let s'_i be a value assignment of all features in S'_i, fi a value-assignment of feature Fi, and c a value-assignment of the target concept C. The definition can be formalized as follows.

Definition 1: (Relevant feature) Fi is relevant to the target concept C if and only if there exists some $s'i$, fi and c, such that, for probability $p(S'i = s'i, Fi = fi) > 0$, $p(C = c \mid S'i = s'i, Fi = fi) \neq p(C = c \mid S'i = s'i)$. Otherwise, feature Fi is an irrelevant feature. Definition 1 indicates that there are two kinds of relevant features due to different $S'i$: (i) when $S'i = Si$, from the definition we can know that Fi is directly relevant to the target concept; (ii) when $S'i \subsetneq Si$, from the definition we may obtain that $p(C|Si, Fi) = p(C|Si)$. It seems that Fi is irrelevant to the target concept. However, the definition shows that feature Fi is relevant when using $S'i \cup \{Fi\}$ to describe the target concept. The reason behind is that either Fi is interactive with $S'i$ or Fi is redundant with $Si - S'i$. In this case, we say Fi is indirectly relevant to the target concept. Most of the information contained in redundant features is already present in other features. The definitions of Markov blanket and redundant feature are introduced as follows, respectively.

Definition 2: (Redundant feature) Let S be a set of features, a feature in S is redundant if and only if it has a Markov Blanket within S. 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, the measure of correlation between either two features or a feature or a feature and the target concept.

The symmetric uncertainty is defined as follows

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

Where,

1) H(x) is the entropy of a discrete random variable X. Suppose p(x) is the prior probabilities for all values of X, H(X) is defined by

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

2) Gain (X|Y) is the amount by which the entropy of Y decreases. It reflects the additional information about Y provided by X and is called the information gain which is given by

Gain (X|Y) = H(X) - H (X|Y)
= H(Y) – H (Y|X)

Where H (X|Y) is the conditional entropy which quantifies the remaining entropy (i.e. uncertainly) of a random variable X given that the value of another random variable Y is known. Suppose p(x) is the prior probabilities for all values of X and p(x|y) is the posterior probabilities of X given the values of Y, H(X|Y) is defined by

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

Information gain is a symmetrical measure. That is the amount of information gained about X after observing Y is equal to the amount of information gained about Y after observing X. This ensures that the order of two variables. 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 feature cluster can be defined as follows.

Definition 3: (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.

Definition 4: (F-Correlation) The correlation between any pair of features Fi and Fj ($Fi, Fj \in F \wedge i \neq j$) is called the F-Correlation of Fi and Fj , and denoted by $SU(Fi, Fj)$.

Definition 5: (F-Redundancy) Let $S = \{F1, F2, ..., Fi, ..., Fk<|F|\}$ be a cluster of features. if $\exists Fj \in S$, $SU(Fj, C) \geq SU(Fi, C) \wedge SU(Fi, Fj) > SU(Fi, C)$ 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).

Definition 6: (R-Feature) A feature $Fi \in S = \{F1, F2, ..., Fk\}$ ($k < |F|$) is a representative feature of the cluster S (i.e. Fi is a R-Feature) if and only if, $Fi = \text{argmax}_{Fj \in S} SU(Fj, C)$. This means the feature, which has the strongest TRelevance, can act as a R-Feature for all the features in the cluster. According to the above definitions, feature subset selection can be the process that identifies and retains the strong T-Relevance features and selects R-Features from feature clusters. The behind heuristics are that 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.

3.2 Algorithm and 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 ($1 \leq i \leq m$) in the first step. The features whose $SU(Fi, C)$ values are greater than a predefined threshold θ comprise the target-relevant feature subset $F' = \{F'1, F'2, ..., F'k\}$ ($k \leq m$). 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. Thus for

graph G, we build a MST, which connects all vertices such that the sum of the weights of the edges is the minimum, using the well-known Prim algorithm. The weight of edge (F' i, F' j) is F-Correlation $SU(F'\ i, F'\ j)$. After building the MST, in the third step, we first remove the edges $E = \{(F'\ i, F'\ j) \mid (F'\ i, F'\ j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$, whose weights are smaller than both of the T-Relevance $SU(F'\ i, C)$ and $SU(F'\ j, C)$, from the MST. Each deletion results in two disconnected trees $T1$ and $T2$. Assuming the set of vertices in any one of the final trees to be V (T), we have the property that for each pair of vertices ($F'\ i, F'\ j \in V$ (T)), $SU(F'\ i, F'\ j) \geq SU(F'\ i, C) \vee SU(F'\ i, F'\ j) \geq SU(F'\ j, C)$ always holds. From Definition 6 we know that this property guarantees the features in V (T) are redundant. This can be illustrated by an example. Suppose the MST shown in Fig.2 is generated from a complete graph G. In order to cluster the features, we first traverse all the six edges, and then decide to remove the edge ($F0$, $F4$) because its weight $SU(F0, F4) = 0.3$ is smaller than both $SU(F0, C) = 0.5$ and $SU(F4, C) = 0.7$. This makes the MST is clustered into two clusters denoted as V ($T1$) and V ($T2$). Each cluster is a MST as well. Take V ($T1$) as an example. From Fig.2 we know that $SU(F0, F1) > SU(F1, C)$, $SU(F1, F2) > SU(F1, C) \wedge SU(F1, F2) > SU(F2, C)$, $SU(F1, F3) > SU(F1, C) \wedge SU(F1, F3) > SU(F3, C)$. We also observed that there is no edge exists between $F0$ and $F2$, $F0$ and $F3$, and $F2$ and $F3$. Considering that $T1$ is a MST, so the $SU(F0, F2)$ is greater than $SU(F0, F1)$ and $SU(F1, F2)$, $SU(F0, F3)$ is greater than $SU(F0, F1)$ and $SU(F1, F3)$, and $SU(F2, F3)$ is greater than $SU(F1, F2)$ and $SU(F2, F3)$. Thus, $SU(F0, F2) > SU(F0, C) \wedge SU(F0, F2) > SU(F2, C)$, $SU(F0, F3) > SU(F0, C) \wedge SU(F0, F3) > SU(F3, C)$, and $SU(F2, F3) > SU(F2, C) \wedge SU(F2, F3) > SU(F3, C)$ also hold. As the mutual information between any pair (Fi, Fj)(i, j = 0, 1, 2, 3 $\wedge$ $i \neq j$) of $F0$, $F1$, $F2$, and $F3$ is greater than the mutual information between class C and Fi or Fj , Features $F0$, $F1$, $F2$, and $F3$ are redundant.

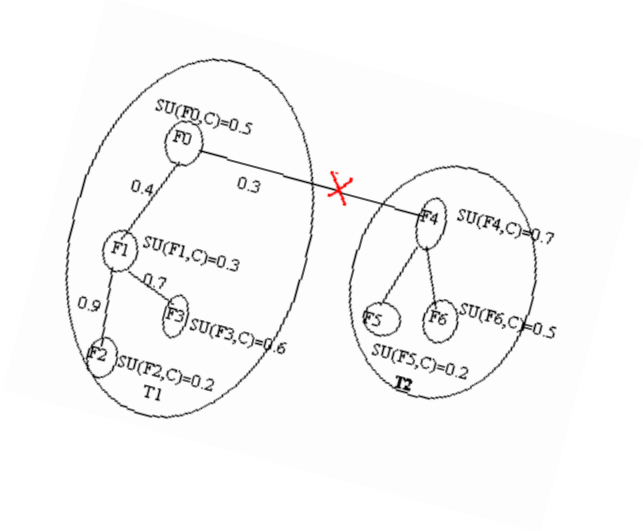

Fig. 2: Example of the clustering step

After removing all the unnecessary edges, a forest Forest is obtained. Each tree $Tj \in$ Forest represents a cluster that is denoted as V (Tj), which is the vertex set of Tj as well. As illustrated above, the features in each cluster are redundant, so for each cluster V (Tj) we choose a representative feature $Fj\ R$ whose T-Relevance $SU(Fj\ R, C)$ is the greatest. All $Fj\ R$ (j = 1...|Forest|) comprise the final feature subset $\cup Fj\ R$.

The details of the FAST algorithm is shown in Algorithm 1.

Algorithm 1: FAST

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 ====
1 for i = 1 to m do
2 T-Relevance = SU (Fi, C)
3 if T-Relevance > θ then
4 S = S $\cup$ {Fi };
//==== Part 2 : Minimum Spanning Tree Construction ====
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 Algorithm to generate the minimum spanning tree
//==== Part 3 : Tree Partition and Representative Feature Selection ====
10 Forest = minSpanTree
11 for each edge □□□ ∈ Forest do
12 if SU(□ ′ □, □ ′ □) < SU(□ ′ □, □) ∧ SU(□ ′ □, □ ′ □) < SU(□ ′ □,□) then
13 Forest = Forest − □□□
14 S = □
15 For each tree □□ ∈ Forest do
16 □□ □ = argmax□ ′ □ ∈ □□ SU(□ ′ □,□)
17 S = S ∪ {□□ □ };
18 return S

Time complexity analysis: The major amount of work for Algorithm 1 involves the computation of □□ 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 □(□) in terms of the number of features □. Assuming □(1 ≤ □ ≤ □) features are selected as relevant ones in the first part, when □ = 1, only one feature is selected. Thus, there is no need to continue the rest parts of the algorithm, and the complexity is □(□). When 1 < □ ≤ □, the second part of the algorithm firstly constructs a complete graph from relevant features and the complexity is □(□2), and then generates a MST from the graph using Prim algorithm whose time complexity is □(□2). The third part partitions the MST and chooses the representative features with the complexity of □(□). Thus when 1 < □ ≤ □, the complexity of the algorithm is □(□+□2). This means when □ ≤ √ □, FAST has linear

complexity □(□), while obtains the worst complexity □(□2) when □ = □. However, □ is heuristically set to be ⌊ √ □ ∗ lg□⌋ in the implementation of FAST. So the complexity is □(□ ∗ lg2□), which is typically less than □(□2) since □□2□ < □. This can be explained as follows. Let □(□) = □ − lg2□, so the derivative □′(*m*) = 1 − 2 lg □/□, which is greater than zero when □ > 1.

Feature selection algorithms in a fair and reasonable way, we set up our experimental study as follows. 1) The proposed algorithm is compared with five different types of representative feature selection algorithms. They are (i) FCBF, (ii) ReliefF, (iii) CFS , (iv) Consist, and (v) FOCUSSF, respectively. FCBF and ReliefF evaluate features individually. For FCBF, in the experiments, we set the relevance threshold to be the □□ value of the ⌊ □/log□⌋ □*h* ranked feature for each data set (□ is the number of features in a given data set) . ReliefF searches for nearest neighbors of instances of different classes and weights features according to how well they differentiate instances of different classes. The other three feature selection algorithms are based on subset evaluation. CFS exploits best-first search based on the evaluation of a subset that contains features highly correlated with the target concept, yet uncorrelated with each other. The Consist method searches for the minimal subset that separates classes as consistently as the full set can under best-first search strategy. FOCUS-SF is a variation of FOCUS . FOCUS has the same evaluation strategy as Consist, but it examines all subsets of features. 2) Four different types of classification algorithms are employed to classify data sets before and after feature selection. They are (i) the probability-based Naive Bayes (NB), (ii) the tree-based C4.5, (iii) the instance-based lazy learning algorithm IB1, and (iv) the rule-based RIPPER, respectively. Naive Bayes utilizes a probabilistic method for classification by multiplying the individual probabilities of every feature-value 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, and so on. The tree comprises of nodes (features) that are selected by information entropy. Instance-based learner IB1 is a single-nearest neighbor algorithm. 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.

3) When evaluating the performance of the feature subset selection algorithms, four metrics, (i) the proportion of selected features (ii) the time to obtain the feature subset, (iii) the classification accuracy, and (iv) the Win/Draw/Loss record, 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. The Win/Draw/Loss record presents three values on a given measure, i.e. the numbers of data sets for which our proposed

4 EMPIRICAL STUDY

algorithm FAST obtains better, equal, and worse performance than other five feature selection algorithms, respectively. The measure can be the proportion of selected features, the runtime to obtain a feature subset, and the classification accuracy, respectively.

Experimental procedure

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.

Results and analysis

In this section we present the experimental results in terms of the proportion of selected features, the time to obtain the feature subset, the classification accuracy, and the Win/Draw/Loss record.

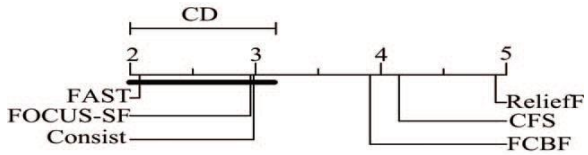

Fig. 3: Proportion of selected features comparison of all feature selection algorithms against each other with the Nemenyi test.

Runtime

1) Generally the individual evaluation based feature selection algorithms of FAST, FCBF and ReliefF are much faster than the subset evaluation based algorithms of CFS, Consist and FOCUS-SF. FAST is consistently faster than all other algorithms. The runtime of FAST is only 0.1% of that of CFS, 2.4% of that of Consist, 2.8% of that of FOCUSSF, 7.8% of that of ReliefF, and 76.5% of that of FCBF, respectively. The Win/Draw/Loss records show that FAST outperforms other algorithms as well.

2) For image data, FAST obtains the rank of 1. Its runtime is only 0.02% of that of CFS, 18.50% of that of ReliefF, 25.27% of that of Consist, 37.16% of that of FCBF, and 54.42% of that of FOCUS-SF, respectively. This reveals that FAST is more efficient than others when choosing features for image data.

3) For microarray data, FAST ranks 2. Its runtime is only 0.12% of that of CFS, 15.30% of that of Consist, 18.21% of

that of ReliefF, 27.25% of that of FOCUSSF, and 125.59% of that of FCBF, respectively.

4) For text data, FAST ranks 1. Its runtime is 1.83% of that of Consist, 2.13% of that of FOCUS-SF, 5.09% of that of CFS, 6.50% of that of ReliefF, and 79.34% of that of FCBF, respectively. This indicates that FAST is more efficient than others when choosing features for text data as well.

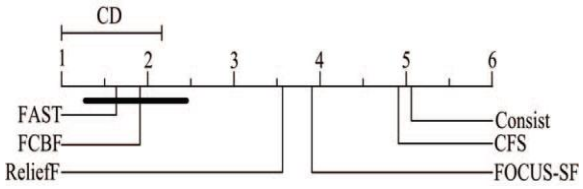

Fig. 4: Runtime comparison of all feature selection algorithms against each other with the Nemenyi test

The above results shows with □ = 0.1 on the 35 data sets. The results indicate that the runtime of FAST is statistically better than those of ReliefF, FOCUS-SF, CFS, and Consist, and there is no consistent evidence to indicate statistical runtime differences between FAST and FCBF.

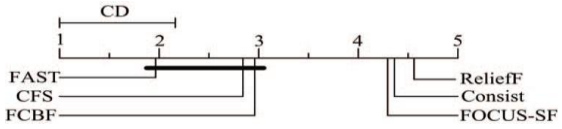

Fig. 5: Accuracy comparison of Naive Bayes with the six feature selection algorithms against each other with the Nemenyi test

we observe that the accuracy of Naïve Bayes with FAST is statistically better than those with
ReliefF, Consist, and FOCUS-SF. But there is no consistent evidence to indicate statistical accuracy differences between Naive Bayes with FAST and with CFS, which also holds for Naive Bayes with FAST and with FCBF.

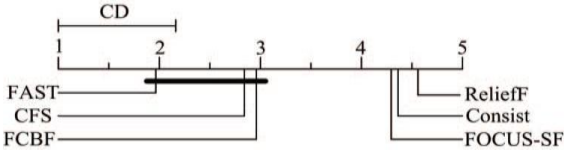

Fig. 6: Accuracy comparison of C4.5 with the six feature selection algorithms against each other with theNamenvi test.

we observe that the accuracy of C4.5 with FAST is statistically better than those with ReliefF, Consist, and FOCUS-SF. But there is no consistent evidence to indicate statistical accuracy differences between C4.5 with FAST and with FCBF, which also holds for C4.5 with FAST and with CFS.

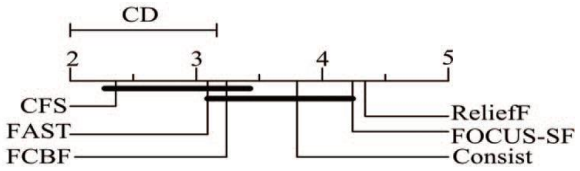

Fig. 7: Accuracy comparison of IB1 with the six feature selection algorithms against each other with theNemenyi test.

we observe that the accuracy of IB1 with FAST is statistically better than those with ReliefF. But there is no consistent evidence to indicate statistical accuracy differences between IB1 with FAST and with FCBF, Consist, and FOCUS-SF, respectively, which also holds for IB1 with FAST and with CFS.

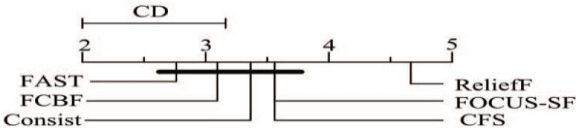

Fig. 8: Accuracy comparison of RIPPER with the six feature selection algorithms against each other with the Nemenyi test

From Fig. 8 we observe that the accuracy of RIPPER with FAST is statistically better than those with ReliefF. But there is no consistent evidence to indicate statistical accuracy differences between RIPPER with FAST and with FCBF, CFS, Consist, and FOCUS-SF, respectively. Microarray data has the nature of the large number of features (genes) but small sample size, which can cause "curse of dimensionality" and over-fitting of the training data [19]. In the presence of hundreds or thousands of features.

5 CONCLUSION

In this paper, we have presented a novel clustering-based feature subset selection algorithm for high dimensional data. The algorithm involves (i) removing irrelevant features, (ii) constructing a minimum spanning tree from relative ones, and (iii) 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. We have compared the performance of the proposed algorithm with those of the five well-known feature selection algorithms FCBF, ReliefF, CFS, Consist, and FOCUS-SF on the 35 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 Bayes, C4.5, and RIPPER, and the second best classification accuracy for IB1. The Win/Draw/Loss records confirmed the conclusions. We also found that FAST obtains the rank of 1 for microarray data, the rank of 2 for text data, and the rank of 3 for image data in terms of classification accuracy of the four different types of classifiers, and CFS is a good alternative. At the same time, FCBF is a good alternative for image and text data. Moreover, Consist and FOCUSSF are alternatives for text data. For the future work, we plan to explore different types of correlation measures, and study some formal properties of feature space.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] Biesiada J. and Duch W., Features election for high-dimensionaldatała Pearson redundancy based filter, AdvancesinSoftComputing, 45, pp 242C249, 2008.

[8] 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.

[9] Cardie, C., Using decision trees to improve case-based learning, In Proceedings of Tenth International Conference on Machine Learning, pp 25-32, 1993.

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