International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 9 Issue 1 –JULY 2014.

An Approach of Fast Clustering On High Dimensional Data for Efficiency and Effectiveness

P. Naveen Kumar
Mtech-CSE, Teegala Krishna Reddy Engineering College, Medbowli, RN.Reddy nager, Meerpet, Andhra Pradesh, India.
Email:naveenkumar143@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 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, Relief-F, 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 searchspace 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. Pereira et al., Baker et al., and Dhillon et al. 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 neighbourhood graph of instances, then delete any edge in the graph that is much longer/shorter (according to some criterion) than its neighbours. The result is a forest and each tree in the forest represents a cluster. In our study, we apply graphtheoreticclustering 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 centres or separated by a regular geometric curve and have been widely used in practice.

2 FEATURE SUBSET SELECTION ALGORITHMS

2.1 Framework and definitions

Irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, "*good featuresubsets 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 Keepingthese 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. In order to more precisely introduce the algorithm, and because our proposed feature subset selection framework involves irrelevant feature removal and redundant feature elimination, we firstly present the traditional definitions of relevant and redundant features, then provide our definitions based on variable correlation as follows. John et al. [33] presented a definition of relevant features. Suppose $F$to be the full set of features, $Fi \in F$ be a feature, $Si = F - \{Fi\}$ and $S'i \subseteq Si$. Let $s'i$be a value assignment of all features in S',fia value assignment of feature Fi, and ca 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 Cif and only if there exists some $s'i$, $fi$and c, such that, for probability $(S'i = s'i, Fi = fi) > 0$,relevant to the target concept; (ii) when $S'i \subseteq Si$, from the definition we may obtain that $(C|Si, Fi) = p(C|Si)$. 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 $Si - S'$. 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. As a result, redundant features do not contribute to getting better interpreting ability to the target concept. It is formally defined by Yu and Liu based on Markov blanket [36]. The definitions of Markov blanket and redundant feature are introduced as follows, respectively.

Definition *2: (Markov blanket)* Given a feature $Fi \in F$, let $Mi \subset F (Fi \in Mi)$, $Mi$is said to be a Markov blanket for Fiif and only if $(F - Mi - \{Fi\}, C|Fi, Mi) = p(F - Mi - \{Fi\}, C|Mi)$.

Definition *3: (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. Mutual information measures how much the distribution of the feature values and target classes differ from statistical independence. The *symmetric uncertainty* is defined as follows

$(X, Y) = 2 \times Gain(X|Y)$

$H(X) + H(Y)$. (1)

Where,

1) (X) is the entropy of a discrete random variable X. Suppose (x) is the prior probabilities for all

Values of X, (X) is defined by

$(X) = -\Sigma\ x \in X$

$(x) \log 2\ (x)$. (2)

2) *Gain* $(X|)$ is the amount by which the entropy of Ydecreases. It reflects the additional information about Yprovided by $X$and is called the information gain which is given by

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

$= H(Y) - H(Y|X)$. (3)

Where $(X|)$ is the conditional entropy which

$p(C = c|S' i = s' i, Fi = fi) \neq p(C = c|S' i = s' i)$.

Otherwise, feature $Fi$is an *irrelevant feature*. Definition 1 indicates that there are two kinds of relevant features due to different S': (i) when $S'i = Si$, from the definition we can know that $Fi$is directly quantifies the remaining entropy (i.e. uncertainty) of a random variable Xgiven that the value of another random variable $Y$is known. Suppose (x) is the prior probabilities for all values of $X$and $(x|y)$ is the posterior probabilities of Xgiven the values of Y, $H(X|Y)$ is defined by

$(X|Y) = -\Sigma\ y \in Y(y) \Sigma\ x \in X p(x|y) \log 2\ p(x|y)$. (4)

Information gain is a symmetrical measure. That is the amount of information gained about Xafter observing $Y$is equal to the amount of information gained about Yafter observing X. This ensures that the order of two variables (e.g.,(X, Y) or (Y,X)) will not affect the value of the measure. Symmetric uncertainty treats a pair of variables symmetrically, it compensates for information gain's bias toward variables with more values and normalizes its value to the range [0,1]. A value 1 of (X, Y) indicates that knowledge of the value of either one completely predicts the value of the other and the value 0 reveals that $X$and $Y$are independent. Although the entropy based measure handles nominal or discrete variables, they can deal with continuous features as well, if the values are discretized properly in advance. 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 *FCorrelation* between a pair of features, the feature redundance *F-Redundancy* and the representative feature *R-Feature* of a feature cluster can be defined as follows.

***Definition** 4: (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 (Fi, C). If (Fi, C) is greater than a predetermined threshold θ, we say that $Fi$is a strong *T-Relevance* feature.

2.2 Algorithm and analysis

The proposed FAST algorithm logically consists of three 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* $S(Fi, C)$ value for each feature $Fi (1 \leq i \leq m)$ in the first step. The features whose (Fi, C) values are greater than a predefined threshold θcomprise 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*$(F' i, F' j)$ 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 $V = \{F' i | F' i \in F' \wedge i \in [1, k]\}$ and $E = \{(F' i, F' j) | (F' i, F' j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$. As symmetric uncertainty is symmetric further the *F Correlation* $(F' i, F' j)$ is symmetric as well, thus $G$is an undirected graph. The complete graph Greflects the correlations among all the target-relevant features. Unfortunately, graph Ghas $k$vertices and $(k-1)/2$ 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 [26]. 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 [54]. The weight of edge $(F' i, F' j)$ is *F Correlation* $(F' i, F' j)$. After building the MST, in the third step, we first remove the edges $E = \{(F' i, F' j) | (F' i, F' j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$, whose

weights are smaller than both of the *T Relevance* ($F'i$, 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 finaltrees to be $V(T)$, we have the property that for each pair of vertices ($F'i$, $F'j \in V(T)$), $(F'i, F'j) \geq SU(F'i, C) \vee SU(F'i, F'j) \geq SU(F'j, C)$ 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 ($F0$, $F4$) = 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 ($F0$, $F2$) is greater than ($F0$, $F1$) 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 Fior Fj, features $F0$, $F1$, $F2$, and $F3$ are redundant. 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 FjR whose *T-Relevance* (FjR, C) is the greatest. All FjR ($j = 1...|Forest|$) comprise the final feature subset $\cup FjR$.

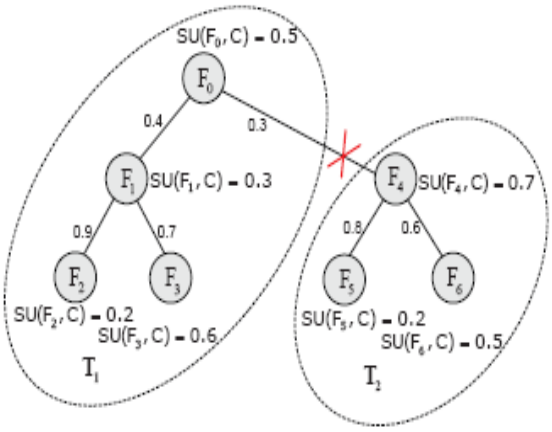

Fig. 1: Example of the clustering step

Time complexity analysis. The major amount of work for Algorithm 1 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 is (m). When $1 < k \leq m$, 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 (k). Thus when $1 < k \leq m$, the complexity of the algorithm is ($m+k2$). This means when $k \leq \sqrt{m}$, FAST has linear complexity (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 ($m * \lg 2m$), which is typically less than $O(m2)$ since $lg2m < m$. This can be explained as follows. Let (m) = $m - \lg 2m$, so the derivative $f'(m) = 1 - 2 \lg e/m$, which is greater than zero when $m > 1$. So (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 ($m2$). Thus, on high dimensional data, the time complexity of FAST is far more less than ($m2$). This makes FAST has a better runtime performance with high dimensional data.

3 EXPERIMENTALSTUDY

3.1 Data source

For the purposes of evaluating the performance and effectiveness of our proposed FAST algorithm, verifying whether or not the method is potentially useful in practice, and allowing other researchers to confirm our results, 35 publicly available data sets1 were used. The numbers of features of the 35 data sets vary from 37 to 49152 with a mean of 7874. The dimensionality of the 54.3% data sets exceeds 5000, of which 28.6% data sets have more than 10000 features. The 35 data sets cover a range of application domains such as text, image and bio microarray data classification. Table 12 shows the corresponding statistical information. Note that for the data sets with continuous-valued features, the well-known off-the-shelf MDL method was used to discretize the continuous values.

3.2 Experiment setup

To evaluate the performance of our proposed FAST algorithm and compare it with other feature selection

1. The data sets can be downloaded at: http://archive.ics.uci.edu/ml/, http://tunedit.org/repo/Data/ Text-wc, http://featureselection.asu.edu/datasets.php, http://www.lsi.us.es/aguilar/datasets.html
   2. F, I, and T denote the number of features, the number of instances, and the number of classes, respectively. 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 [68], [71], (ii) ReliefF [57], (iii) CFS [29], (iv) Consist [14], and (v) FOCUSSF [2], respectively. FCBF and ReliefF evaluate features individually. For FCBF, in the experiments, we set the relevance threshold to be the SU value of the $\lfloor m/\log m \rfloor th$ ranked feature for each data set (m is the number of features in a given data set) as suggested by Yu and Liu [68], [71]. 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 [2]. FOCUS has the same evaluation strategy as Consist, but it examines all subsets of features. Considering the time efficiency, FOUCS-SF replaces exhaustive search in FOCUS with sequential forward selection. For our proposed FAST algorithm, we heuristically set θ to be the SU value of the $\lfloor \sqrt{m} * \lg m \rfloor th$ ranked feature for each data set.
   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, and it classifies entities taking the class of the closest associated vectors in the training set via distance metrics. It is the simplest among the algorithms used in our study. Inductive rule learner RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [12] 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 [65], 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 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.

TABLE 1: Summary of the 35 benchmark data sets

3.3 Experimental procedure

In order to make the best use of the data and obtainstable 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 [21]. Randomizing the order of the inputs can help diminish the order effectsIn the experiment, for each feature subset selection algorithm, we obtain M×N feature subsets *Subset* and the corresponding runtime *Time* with each data set. Average *Subset* and *Time*, we obtain the number of selected features further the proportion of selected features and the corresponding runtime for each feature selection algorithm on each data set. For each classification algorithm, we obtain M×N classification *Accuracy*for each feature selection algorithm and each data set. Average these *Accuracy*, we obtain mean accuracy of each classification algorithm under each feature selection algorithm and each data set. The procedure *ExperimentalProcess* shows the details.

3.4 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. For the purpose of exploring the statistical significance of the results, we performed a nonparametric Friedman test followed by Nemenyi post-hoc test, as advised by Demsar and Garcia and Herrerato to statistically compare algorithms on multiple data sets. Thus the Friedman and the Nemenyi test results are reported as well.3.*4.1 Proportion of selected features*

Table 2 records the proportion of selected features of the six feature selection algorithms for each data set. From it we observe that 1) generally all the six algorithms achieve significant reduction of dimensionality by selecting only a small portion of the original features. FAST on average obtains the best proportion of selected features of 1.82%. The Win/Draw/Loss records show FAST wins other algorithms as well. 2) For image data, the proportion of selected features of each algorithm has an increment compared with the corresponding average proportion of selected features on the given data sets except Consist has an improvement. This reveals that the five algorithms are not very suitable to choose features for image data compared with for microarray and text data. FAST ranks 3 with the proportion of selected features of 3.59% that has a tiny margin of 0.11% to the first and second best proportion of selected features 3.48% of Consist and FOCUS-SF, and a margin of 76.59% to the worst proportion of selected features 79.85% of ReliefF. 3) For microarray data, the proportion of selected features has been improved by each of the six algorithms compared with that on the given data sets. This indicates that the six algorithms work well with microarray data. FAST ranks 1 again with the proportion of selected features of 0.71%. Of the six algorithms, only CFS cannot choose features for two data sets whose dimensionalities are 19994 and 49152, respectively. 4) For text data, FAST ranks 1

again with a margin of 0.48% to the second best algorithm FOCUS-SF.

TABLE 2: Proportion of selected features of the six feature selection algorithms

Data set	Proportion of selected features (%) of					
	FAST	FCBF	CFS	ReliefF	Consist	FOCUS-SF
chess	16.22	21.62	10.81	62.16	81.08	18.92
mfeat-fourier	19.48	49.35	24.68	98.70	15.58	15.58
coil2000	3.49	8.14	11.63	50.00	37.21	1.16
elephant	0.86	3.88	5.60	6.03	0.86	0.86
arrhythmia	2.50	4.64	9.29	50.00	8.93	8.93
fqs-nowe	0.31	2.19	5.63	26.56	4.69	4.69
colon	0.30	0.75	1.35	39.13	0.30	0.30
fbis.wc	0.80	1.45	2.30	0.95	1.75	1.75
AR10P	0.21	1.04	2.12	62.89	0.29	0.29
PIE10P	1.07	1.98	2.52	91.00	0.25	0.25
oh0.wc	0.38	0.88	1.10	0.38	1.82	1.82
oh10.wc	0.34	0.80	0.56	0.40	1.61	1.61
B-cell1	0.52	1.61	1.07	30.49	0.10	0.10
B-cell2	1.66	6.13	3.85	96.87	0.15	0.15
B-cell3	2.06	7.95	4.20	98.24	0.12	0.12
base-hock	0.58	1.27	0.82	0.12	1.19	1.19
TOX-171	0.28	1.41	2.09	64.60	0.19	0.19
tr12.wc	0.16	0.28	0.26	0.59	0.28	0.28
tr23.wc	0.15	0.27	0.19	1.46	0.21	0.21
tr11.wc	0.16	0.25	0.40	0.37	0.31	0.31
embryonal-tumours	0.14	0.03	0.03	13.96	0.03	0.03
leukemia1	0.07	0.03	0.03	41.35	0.03	0.03
leukemia2	0.01	0.41	0.52	60.63	0.08	0.08
tr21.wc	0.10	0.22	0.37	2.04	0.20	0.20
wap.wc	0.20	0.53	0.65	1.10	0.41	0.41
PIX10P	0.15	3.04	2.35	100.00	0.03	0.03
ORL10P	0.30	2.61	2.76	99.97	0.04	0.04
CLL-SUB-111	0.04	0.78	1.23	54.35	0.08	0.08
ohscal.wc	0.34	0.44	0.18	0.03	NA	NA
la2s.wc	0.15	0.33	0.54	0.09	0.37	NA
la1s.wc	0.17	0.35	0.51	0.06	0.34	NA
GCM	0.13	0.42	0.68	79.41	0.06	0.06
SMK-CAN-187	0.13	0.25	NA	14.23	0.06	0.06
new3s.wc	0.10	0.15	NA	0.03	NA	NA
GLA-BRA-180	0.03	0.35	NA	53.06	0.02	0.02
Average(Image)	3.59	10.04	6.68	79.85	3.48	3.48
Average(Microarry)	0.71	2.34	2.50	52.92	0.91	0.91
Average(Text)	2.05	3.25	2.64	10.87	11.46	2.53
Average	1.82	4.27	3.42	42.54	5.44	2.06
Win/Draw/Loss	-	33/0/2	31/0/4	29/1/5	20/2/13	19/2/14

3.4.1 Sensitivity analysis

Like many other feature selection algorithms, our proposed FAST also requires a parameter θ that is the threshold of feature relevance. Different θ values might end with different classification results. In order to explore which parameter value results in the best classification accuracy for a specific classification problem with a given classifier, a 10 fold cross-validation strategy was employed to reveal how the classification accuracy is changing with value of the parameter θ.

Fig. 2 shows the results where the 35 subfigures represent the 35 data sets, respectively. In each subfigure, the four curves denotes the classification accuracies of the four classifiers with the different θ values. The cross points of the vertical line with the horizontal axis represent the default values of the parameter θ recommended by FAST, and the cross points of the vertical line with the four curves are the classification accuracies of the corresponding classifiers with the θ values. From it we observe that:

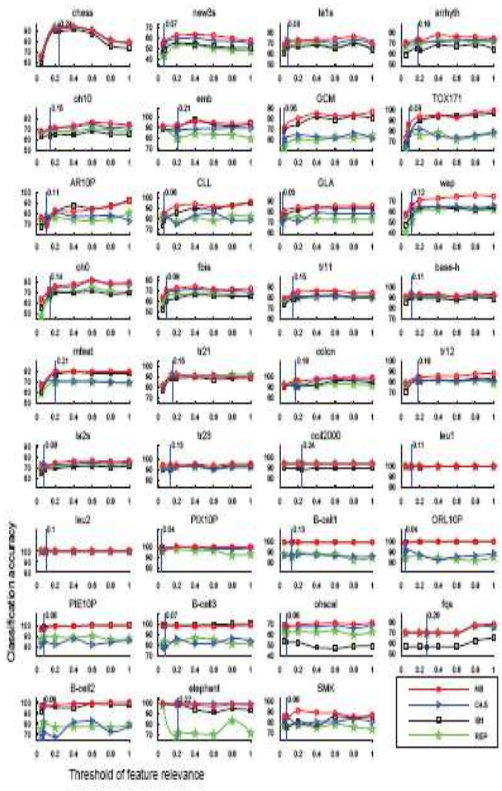

Fig. 2: Accuracies of the four classification algorithms with different θ values.

1) Generally, for each of the four classification algorithms, (i) different θ values result in different classification accuracies; (ii) there is a θ value where the corresponding classification accuracy is the best; and (iii) the θ values, in which the best classification accuracies are obtained, are different for both the different data sets and the different classification algorithms. Therefore, an appropriate θ value is desired for a specific classification problem and a given classification algorithm.

2) In most cases, the default θ values recommended by FAST are not the optimal. Especially, in a few cases (e. g., data sets GCM, CLL-SUB-11, and TOX- 171), the corresponding classification accuracies are very small.

3) For each of the four classification algorithms, although the θ values where the best classification accuracies are obtained are

different for different data sets, The value of 0.2 is commonly accepted because the corresponding classification accuraciesare among the best or nearly the best ones. When determining the value of θ, besides classification accuracy, the proportion of the selected features should be taken into account as well. This is because improper proportion of the selected features results in a large number of features are retained, and further affects the classification efficiency.

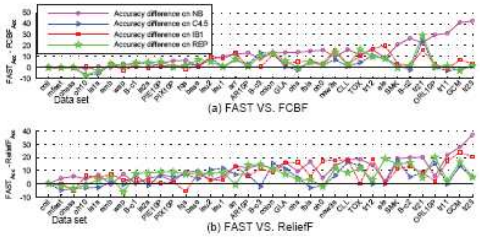

Fig. 3: Accuracy differences between FAST and the comparing algorithms

Just like the default θvalues used for FAST in the experiments are often not the optimal in terms of classification accuracy, the default threshold values used for FCBF and ReliefF (CFS, Consist, and FOCUS-SF do not require any input parameter) could be so. In order to explore whether or not FAST still outperforms when optimal threshold values are used for the comparing algorithms, 10-fold cross validation methods were firstly used to determine the optimal threshold values and then were employed to conduct classification for each of the four classification methods with the different feature subset selection algorithms upon the 35 data sets. The results reveal that FAST still outperforms both FCBF and ReliefF for all the four classification methods, Fig. 3 shows the full details. At the same time, Wilcox on signed ranks tests [75] with α= 0.05 were performed to confirm the results as advised by Demsar. All the p values are smaller than 0.05, this indicates that the FAST is significantly better than both FCBF and ReliefF (please refer to Table 3 for details).

TABLE 3: *p* values of the Wilcoxon tests

Alternative hypothesis	NB	C4.5	IB1	REPPER
FAST > FCBF	8.94E-07	0.0013	5.08E-05	8.78E-05
FAST > ReliefF	4.37E-07	3.41E-04	4.88E-06	7.20E-06

Note that the optimal θvalue can be obtained via the cross-validation method. Unfortunately, it is very time consuming.

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 FOCUSS Fare 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 Proceedings of IEEE international Conference on Data Mining Workshops, pp 350-355, 2009.

[11] Chikhi S. and Benhammada S., ReliefMSS: a variation on a feature ranking ReliefF algorithm. Int. J. Bus. Intell. Data Min. 4(3/4), pp 375-390, 2009.

[12] Cohen W., Fast Effective Rule Induction, In Proc. 12th international Conf. Machine Learning (ICML'95), pp 115-123, 1995.

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

[14] 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.

[15] 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.

[16] Dash M. and Liu H., Consistency-based search in feature selection. Artificial Intelligence, 151(1-2), pp 155-176, 2003.