Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 13(2): 113-122, Jun. 2015 Regular paper

ModifiedFAST: A New Optimal Feature Subset Selection Algorithm

Arpita Nagpal and Deepti Gaur[*], ***Member, KIICE***

Department of Computer Science and Information Technology, ITM University, Gurgaon 122017, India

Abstract

Feature subset selection is as a pre-processing step in learning algorithms. In this paper, we propose an efficient algorithm, ModifiedFAST, for feature subset selection. This algorithm is suitable for text datasets, and uses the concept of information gain to remove irrelevant and redundant features. A new optimal value of the threshold for symmetric uncertainty, used to identify relevant features, is found. The thresholds used by previous feature selection algorithms such as FAST, Relief, and CFS were not optimal. It has been proven that the threshold value greatly affects the percentage of selected features and the classification accuracy. A new performance unified metric that combines accuracy and the number of features selected has been proposed and applied in the proposed algorithm. It was experimentally shown that the percentage of selected features obtained by the proposed algorithm was lower than that obtained using existing algorithms in most of the datasets. The effectiveness of our algorithm on the optimal threshold was statistically validated with other algorithms.

Index Terms: Entropy, Feature selection, Filter model, Graph-based clustering, Mutual information

I. INTRODUCTION

Most real-world data cannot be applied directly to any data mining algorithm due to the dimensional nature of the dataset. In knowledge mining, similar patterns in a large dataset are clustered. The major difficulty faced due to an increasing number of features and, correspondingly, dimensionality, has been called ‘the curse of dimensionality’ [1]. Therefore, feature selection is an important pre-processing step before any model is applied to the data. Moreover, with an increase in the number of features, the complexity grows exponentially.

In order to reduce the dimensionality of the data, feature selection is of paramount importance in most real-world tasks. Feature selection involves removing redundant and irrelevant features, so that the remaining features can still represent the complete information. Sometimes, additional features make no contribution to the performance of the classification task. Reducing the dimensionality of a dataset provides insight into the data even before the application of a data mining process.

A subset of the features is derived using the values of the features, as calculated by mathematical criteria that provide information about the contribution of each feature to the dataset. Various evaluation criteria exist, according to which feature selection can be broadly classified into four categories: the filter approach, the wrapper approach, the embedded approach, and the hybrid approach [2-5]

In filter feature selection, the feature subset is selected as a pre-processing step before the application of any learning and classification processes. Thus, feature selection is independent of the learning algorithm that will be applied.

Received 23 February 2015, **Revised** 27 May 2015, **Accepted** 10 June 2015

***Corresponding Author** Deepti Gaur (E-mail: deeptigaur@itmindia.edu, Tel: +91-124-2365811)
Department of Computer Science and Information Technology, ITM University, Gurgaon 122017, India.

Open Access http://dx.doi.org/10.6109/jicce.2015.13.2.113 print ISSN: 2234-8255 online ISSN: 2234-8883

This is known as the filter approach because the features are filtered out of the dataset before any algorithms are applied. This method is usually faster and computationally more efficient than the wrapper approach [2].

In the wrapper method [6], as in simulated annealing or genetic algorithms, features are selected in accordance with the learning algorithm which will then be applied. This approach yields a better feature subset than the filter approach, because it is tuned according to the algorithm, but it is much slower than the filter approach and has to be rerun if a new algorithm is applied.

The wrapper model requires the use of a predetermined learning algorithm to determine which feature subset is the best. It results in superior learning performance, but tends to be more computationally expensive than the filter model. When the number of features becomes very large, it is preferable to use the filter model due to its computational efficiency [4].

A recently proposed feature selection technique is the ensemble learning-based feature selection method. The purpose of the ensemble classifier is to produce many diverse feature selectors and combine their outputs [7].

This paper presents an algorithm that is simple and fast to execute. The algorithm works in steps. First, irrelevant features are removed using the concept of symmetric uncertainty (SU). A threshold value of SU is established, enabling irrelevant features to be removed. The threshold value employed in this algorithm was tested against various other threshold values, and we found that other threshold values led to the identification of a different number of relevant features. After obtaining a set of relevant features, a minimum spanning tree (MST) is constructed. The redundant features are removed from the MST based on the assumption that the correlation among two features should always be less than their individual correlation with the class. Features that are found to have a greater correlation with the class than their correlation with other features are added to the cluster, and other features are ignored. The next section describes the existing algorithms in this field. Section III explains the definitions and provides an outline of the work. Section IV discusses how the algorithm is implemented and analysed. Section V discusses datasets and performance metrics. Results are discussed in section VI, and the last section contains the conclusion.

II. RELATED WORK

The goal of feature selection is to determine a subset of the features of the original set by removing irrelevant and redundant features.

Liu and Yu [8] have established that the general process of feature selection can be divided into four processes: subset generation, subset evaluation, stopping criterion, and subset validation.

In the subset generation step, the starting point of the search needs to be decided first. A search strategy is then implemented: exhaustive search, heuristic search, or random search. All of the search strategies operate in three directions to generate a feature subset: forward (adding a feature to a selected subset that begins with the empty set), backward (eliminating features from a selected subset that begins with the full original set) and bidirectional (both adding and removing features). The generated subset is then evaluated based on mathematical criteria. These mathematical criteria are divided into the filter, wrapper, hybrid, and embedded approaches. Within the filter model, many algorithms have been proposed, including Relief [9], Relief-F, Focus, FOCUS-2 [10], correlation-based feature selection (CFS) [11], fast-correlation-based filter (FCBS) [12], and FAST [13]. Some of these algorithms, including Relief, FCBS, and FAST, employ a threshold to decide whether a feature is relevant.

The Relief algorithm for feature selection uses distance measures as a feature subset evaluation criterion. It weighs (ranks) each feature under different classes. If the score exceeds a specified threshold, then the feature is retained; otherwise it is removed [9].

CFS algorithms are based on the concept of information theory. The FCBS algorithm developed by Yu and Liu [12] removes both irrelevant and redundant features, using SU as a goodness measure. It uses the concept of correlation based on the information theoretic concept of mutual information and entropy to calculate the uncertainty of a random variable.

Hall and Smith [3] have developed a CFS algorithm for feature selection that uses correlation to evaluate the value of features. This approach is based on the idea that good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other [11]. In this algorithm, it is necessary to find the best subset from the 2^n possible subsets of the n features. The subset generation strategy used is the best-first search. In this strategy, the feature class and the feature-feature correlation matrix is given as input. Initially, the feature set is empty; as each feature is added, its value is evaluated. The feature set with the highest evaluation is chosen. Subsequently, the next subsets are added to the best one and evaluated. Finally, the best subset found is returned [11]. This approach uses Pearson correlation as a heuristic for calculating the value of each feature subset that is generated. The search terminates when the best subset is found. It stops when five consecutive fully expanded non-improving subsets are found.

The subset generation and evaluation process terminates when it reaches the stopping criterion. Different stopping criteria are used in different algorithms. Some involve a specified limit on the number of features or a threshold value. Finally, validation data are used to validate the selected subset. In real-world applications, we do not have any prior knowledge of the relevance of a feature set. Therefore, observations are made of how changes in the feature set affect the performance of the algorithm. For example, the classification error rate or the proportion of selected features can be used as a performance indicator for a selected feature subset [8].

The adjusted Rand index, which is a clustering validation measure, has been used as a measure of correlation between the target class and a feature. Such metrics rank the features by making a comparison between the partition given by the target class and the partition of each feature [14].

General graph-theoretic clustering uses the concept of relative neighbourhood graphs [15, 16]. Using the concept of neighbourliness, we define RNG(V) as the representative of the family of the graph consisting of a finite set of points V. RNG(V) is called the relative neighbourhood graph.

Some clustering-based methods construct a graph using the k-nearest-neighbour approach [17]. Chameleon, which is a hierarchical clustering algorithm, uses a k-nearest-neighbour graph approach to construct a sparse graph. It then uses an algorithm to find a cluster by repeatedly combining these subclusters. Chameleon is applicable only when each cluster contains a sufficiently large number of vertices (data items).

According to Xu et al. [18], the MST clustering algorithm has been widely used. They have used this approach to represent multidimensional gene expression data. It is not pre-assumed that the data points are separated by a regular geometric curve or are grouped around centres in any MST-based clustering algorithm.

Many graph-based clustering methods take advantage of the MST approach to represent a dataset. Zhong et al. [17] employed a two-round MST-based graph representation of a dataset and developed a separated clustering algorithm and a touching clustering algorithm, encapsulating both algorithms in the same method. Zahn [19] divided a dataset into different groups according to structural features (distance, density, etc.) and captured these distinctions with different techniques. Grygorash et al. [20] proposed two algorithms using MST-based clustering algorithms. The first algorithm assumes that the number of clusters is predefined. It constructs an MST of a point set and uses an inconsistency measure to remove the edges. This process is repeated for k clusters and forms a hierarchy of clusters until k clusters are obtained. The second algorithm proposed partitions the point set into a group of clusters by maximizing the overall standard deviation reduction.

Shi and Malik [21] published a normalized cut criterion. Ding et al. [22] proposed a min-max clustering algorithm, in which the similarity or association between two subgraphs is minimized, while the similarity within each subgraph is maximized. It has been shown that the min-max cut leads to more balanced cuts than the ratio cut and normalized cut [21].

Our algorithm uses Kruskal's algorithm, which is an MST-based technique of for clustering features. The clusters formed are not based on any distance or density measures. A feature is added to the cluster if it is not redundant to any of the features already present in it.

III. FEATURE SUBSET SELECTION METHOD

Entropy feature selection is a pre-processing step in many applications. Subsets are generated and then evaluated based on a heuristic. The main aim is to find the subset which has relevant and non-redundant features. In order to evaluate and classify features, we have used the information theoretic concept of information gain and entropy. Entropy is a measure of the unpredictability of a random variable. Entropy is related to the probabilities of the random variable rather than their actual values. Entropy is defined as:

$$\mathrm{H}(X) = -\sum_{\mathrm{x}\in\mathrm{Z}} \mathrm{p}(x)\log_2 \mathrm{p}(x). \qquad (1)$$

Here, X is a discrete random variable with the alphabet Z and the probability mass function $\mathrm{p}(x) = \Pr\{X = x\}, x \in Z$.

If X and Y are discrete random variables, Eqs. (2) and (3) give the entropy of Y before and after observing X.

$$\mathrm{H}(Y) = -\sum_{y\in Y} p(y)\log_2 p(y), \qquad (2)$$

$$\mathrm{H}(Y|X) = -\sum_{x\in X} p(x)\sum_{y\in Y} p(y|x)\log_2 p(y|x). \qquad (3)$$

The amount by which the entropy of Y decreases reflects the additional information about Y provided by X and is called the information gain [23]. Information gain is a measure of the amount of information that one random variable contains about another random variable. The information gain I(X, Y) is the relative entropy between the product distribution and the joint distribution [24].

$$\mathrm{I}(Y, X) = \sum_{x\in X}\sum_{y\in Y} p(x,y) log\frac{p(x,y)}{p(x)p(y)}\ , \qquad (4)$$

$$\begin{aligned}\mathrm{Gain}(Y|X) &= \mathrm{H}(Y) - \mathrm{H}(Y|X)\\ &= \mathrm{H}(X) - \mathrm{H}(X|Y)\\ &= \mathrm{H}(Y) + \mathrm{H}(X) - \mathrm{H}(X, Y). \qquad (5)\end{aligned}$$

Information gain is the amount of information obtained about X after observing Y, and is equal to the amount of information obtained about Y after observing X. Unfortunately, information gain is biased in favour of features with many outcomes. That is, attributes with many diverse values will appear to gain more information than those with less diverse values even if they are actually no more informative. Symmetrical uncertainty [25] compensates for the bias of information gain toward attributes with more values and normalizes its value to the range [0,1]. SU is defined as:

$$\mathrm{SU} = 2 * \left[\frac{\text{gain}}{\mathrm{H}(Y)+\mathrm{H}(X)}\right]. \tag{6}$$

Sotoca and Pla [26] have used conditional mutual information as an information theoretic measure. Conditional mutual information I(X; Y/Z) can be defined with regard to two random variables X and Z, over the same dataset, and the relevant variable Y, representing the class labels. I(X; Y/Z) can be interpreted as how much information the feature space X can predict about the relevant variable Y that the feature space Z cannot [24].

A. Outline

The procedure of removing irrelevant and redundant features is shown in Fig. 1.

The goal of obtaining relevant features is accomplished by defining an appropriate threshold for the SU value. If the attributes from the table meet this condition, then they considered relevant and kept. Subsequently, pair-wise correlation is performed to remove redundant features and a complete weighted graph is created. An MST is then constructed. If features meet the criterion defined by the threshold value, they are added to the cluster of the feature subset, and otherwise are ignored.

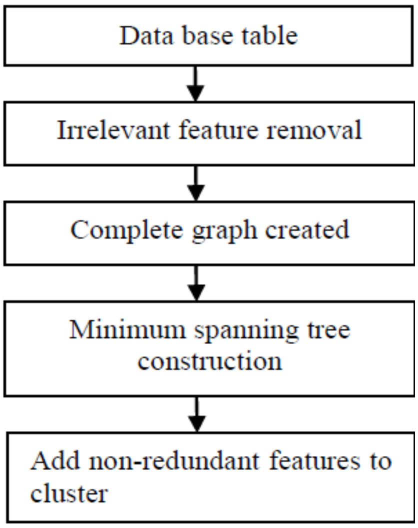
Data base table
Irrelevant feature removal
Complete graph created
Minimum spanning tree construction
Add non-redundant features to cluster

Fig. 1. Flowchart of feature selection [27].

IV. METHODOLOGY AND ANALYSIS

Our algorithm uses the concept of SU described above to solve the problem of feature selection. It is a two-step process; first, irrelevant features are removed, and then redundant features are removed from the dataset and the final feature subset is formed.

The irrelevant feature removal process evaluates whether a feature is relevant to the class or not. For each data set in Table 1, the algorithm given in Fig. 2 is applied to determine the relevant features corresponding to different threshold values. The input to this algorithm is the dataset with m features, data = {F_0,F_1,F_2,………F_m }, the class C, and a threshold value. For each feature, information gain and SU for the class SU(F_i, C) is determined according to Eqs. (5) and (6). In order to determine the relevance of a feature, a user-defined threshold is applied on the SU value [4]. A subset FS of relevant features can be determined by a threshold on the SU value (θ), such that F_i ϵ S', 1<= i <= h, $Su_{i,c}$ > = θ. A relevant feature subset FS={F_0',F_1',F_2', ……..F_h'} is formed, where $h < m$.

Here, for the input dataset D, the algorithm starts with the empty set as the starting point and then adds features to it. It uses a heuristic or sequential search strategy for subset generation. Each subset generated is evaluated by SU. The search iterates and keeps all relevant features found as subsets if they meet a predefined threshold. The algorithm outputs a feature subset.

Time Complexity: The time complexity of the method applied above is quite low. It has linear complexity and is dependent on the size of the data, as defined by the number of features and number of instances in the dataset. The time complexity is O(m), if the number of features is m. Each feature is referred once and its relevance is checked according to the threshold value.

```
Input: Data (F_0, F_1, F_2,……,F_m) // data set of m features
       Θ // predefined threshold
       C // class label corresponding to each instance
Output: R  // relevant feature
        FS // relevant feature subset
    Begin
    FS=0;
    For i=1 to m do
    Z=MI(F_i,C);
    R=SU(F_i,C);
    If R > θ then,
    FS=FS U {F_i}
    Return FS;
    End;
```

Fig. 2. ModifiedFAST algorithm for removing irrelevant features.

```
Input: FS // relevant feature subset
        Data ($F_0, F_1, F_2,……,F_m$) // data set of m features
Output: cluster // final feature subset
    Begin;
    For each pair of features $\{f_i,f_j\} \in$ FS do
    Correlation(i,j)= SU($f_i,f_j$);
    Construct a 2D correlation matrix with value of correlation
    between each feature.
    SpanTree [edge, cost]=kruskal(M);
    For each edge E(i,j) $\in$ SpanTree
    If cost E(i,j)< SU (fi, C) && cost E(i,j)< SU(fj,C)
    If cluster $\nexists$ i
    cluster=cluster $\cup$ i
    If cluster $\nexists$ j
    cluster=cluster $\cup$ j
    Return cluster;
    End;
```

Fig. 3. ModifiedFAST algorithm for redundant feature removal.

In order to remove redundant features from the subset obtained above, a graph-based clustering method is applied. The vertices of the graph are the relevant features and the edges represent the correlation between two features. In order to compute the correlation among two features ($f_i,f_j \in FS$), the correlation SU(f_i,f_j) is calculated. This SU(f_i,f_j) is calculated for all possible feature pair f_i and f_j where $i \neq j$. Therefore, the value of each edge is the correlation value, and the value of each node in the graph is the value of SU calculated above for each feature with its class label.

This results in a weighted complete graph with $h(h-1)/2$ edges (where h is the number of nodes), and the correlation value is expressed as the weights on the edges. In the case of high-dimensional data where the numbers of vertices are very large, the complete graph is very dense with a large number of vertices and edges. For further work on this dataset to be feasible, the number of edges needs to be reduced. An MST is constructed to make the graph somewhat less dense. The edges are removed using Kruskal's algorithm. In order to remove the redundant features from the spanning tree, only features with an edge weight less than the correlation of both features with the class are added to the cluster. The algorithm is described in Fig. 3.

V. EMPIRICAL STUDY

In this section, we evaluate the performance of the algorithm described above by testing it on eight datasets of varying dimensionality, ranging from low-dimension datasets to large-dimension datasets. The ModifiedFAST algorithm was evaluated in terms of the percentage of selected features, runtime and classification accuracy.

Table 1. Summary of data sets

Dataset	No. of instances	No. of features	No. of classes	Domain
WarpPIE10p	210	750	10	Image, face
WarpAR10P	130	510	10	Image, face
Chess	3196	36	2	Text
Coil2000	134	86	2	Text
Email word subject	64	242	2	Text
tox-171	100	256	4	Microarray
Pix10P	100	256	10	Image, face
orlaws10p	100	256	10	Image, face

A. Data Set

In order to validate the proposed algorithm, eight samples of real datasets with different dimensionality were chosen from the UCI machine learning repository and from featureselection.asu.edu/datasets. The datasets contain text, microarray, or image data. The details of these datasets are described in Table 1.

B. Performance Parameters

Many feature selection algorithms are available in the literature. Some algorithms perform better than others with regard to individual metrics, but may perform less well from the viewpoint of other metrics.

Some classical methods used as performance metrics are:

1) **The number of selected features**: This is the main task of any feature selection algorithm. An algorithm that removes all irrelevant and redundant features may result in a lower number of selected features in cases of high-dimensional data. Two algorithms can be compared on this basis.
2) **Runtime:** A main task of any computer application today is to reduce the required runtime. An algorithm that gives the best feature subset in the least time is preferred. If classification accuracy of two algorithms does not significantly vary, then this parameter can be used to compare the algorithms. The value of this parameter depends upon the total number of features available in the dataset. It also depends upon the machine on which processing is taking place.
3) **Classification accuracy**: This metric assesses the performance of a feature selection algorithm on the given classification algorithm. Classifier accuracy refers to the ability of a classifier to correctly predict the class label of unseen data. If the accuracy of any classifier increases after applying feature selection, then that feature selection algorithm is acceptable.

For each dataset, we obtained the number of features after running the algorithm. Different feature selection algorithms were compared on the basis of the percentage of selected features. We then applied the naïve Bayes, the tree-based C4.5, and the rule based IB1 classification algorithms on each newly generated feature subset to calculate the classification accuracy by 10-fold cross validation.

The main target of developing a feature subset for the classification task is to improve the performance of classification algorithms. Accuracy is also the main metric for measuring classification performance. However, the number of selected features is also important in some cases.

We have developed a new performance evaluation measure resulting in a formula that integrates the number of features selected with the classification accuracy of an algorithm.

Δ denotes the classification accuracy/number of feature selected. The number of features selected is a negative metric and the classification accuracy is a positive metric, such that values of *Δ* > 1 indicate a better algorithm.

C. Statistical Significance Validation

Performance metrics are necessary to make observations about different classifiers. The purpose of statistical significance testing is to collect evidence representing the general behaviour of the classifiers from the results obtained by an evaluation metric. One of the tests used is the Friedman test.

The Friedman test [28] is a non-parametric approach. It can be used as a measure to compare the rank of *k* algorithms over *d* datasets. It provides a test of significance for data with ranks <6. If the value of *k* is >5, then the level of significance or the rank of the algorithm can be seen in the chi-square distribution table. The data are treated as the matrix $\{x_{ij}\}_{dxk}$, where *d* is the number of datasets (called blocks) and *k* is the number of columns that have different algorithms.

$$M = \frac{\mathbf{12}}{\boldsymbol{dk(k+1)}}\boldsymbol{X}\sum \boldsymbol{R_j^2} - \mathbf{3}\boldsymbol{d}(\boldsymbol{k}+\mathbf{1}). \tag{7}$$

The null hypothesis of the Friedman test considered here is that there is no difference between the feature selection algorithms based on the percentage of selected features. The decision rule then states that the null hypothesis should be rejected if *M* is greater than the critical value. If this hypothesis gets rejected, then a posthoc test is needed to compare the performance of the algorithms. The Nemenyi test or the Bonferroni-Dunn test can be used as the posthoc test.

VI. RESULTS AND ANALYSIS

We used several threshold values in our experiment. Different feature selection algorithms, such as FCBF [12], Relief, CFS [11], and FAST [13] have likewise used different threshold values to evaluate their results. Yu and Liu [12] have suggested the relevant threshold value on SU to be the ⌊m/log m⌋-th ranked feature for each dataset. In FAST [13] feature selection, features are ranked by using the ($\sqrt{m}$ * lg *m*)-th ranked feature as the threshold of the SU value, where *m* is the number of features. Table 2 shows the relevant features found after implementing the filter feature selection algorithm described in Fig. 2 with a range of threshold values.

The algorithm given in Fig. 3 was applied on all the features found in Table 2 for each dataset. The final feature subset was recorded. Table 3 shows the different percentages corresponding to different thresholds. In order to choose the best thresholds for this algorithm can work, the percentage of selected features, runtime, and classification accuracy values for each threshold value were observed. Tables 4–6 record the classification accuracy along with the calculations of the proposed performance parameter *Δ* for each dataset, using the naïve Bayes classifier, the tree-based C4.5 classifier, and the instance-based IB1 classifier, respectively.

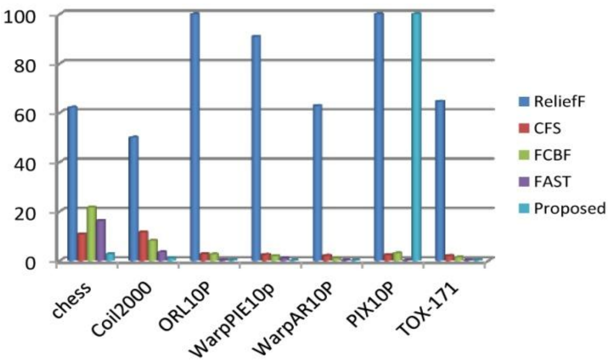

Fig. 4. Comparison of five feature selection algorithms based on the percentage of features selected.

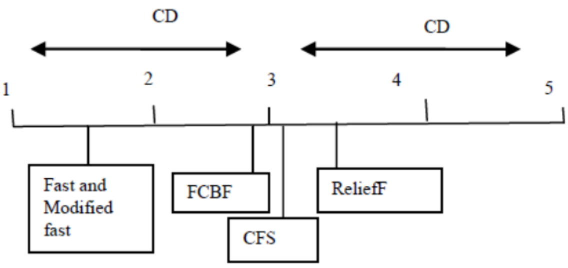

Fig. 5. Ranking of feature selection algorithms based on the Nemenyi test.

Table 2. The number of relevant features found using different threshold values

Threshold dataset	m/log$m$	√m*log$m$	0	0.01	0.2	0.4	0.6	0.8	1	$m^{1/3}$*log$m$	√m/ log$m$
Chess	13	27	36	14	1	0	0	0	0	31	33
Coil2000	18	45	73	51	2	0	0	0	0	56	65
WarpAR10P	322	449	510	510	510	499	0	0	0	489	502
WarpPIE10p	464	652	750	750	750	714	0	0	0	714	737
Email word subject	0	0	242	231	1	0	0	0	0	14	124
Orlaws10P	149	218	256	256	256	256	146	0	0	241	250
Pixraw10P	150	218	256	256	256	256	256	126	0	241	250
Tox-171	0	0	256	256	256	0	0	0	0	0	21

Table 3. Percentage of selected features using different threshold values

Threshold dataset	m/log$m$	√m*log$m$	0	0.01	0.2	0.4	0.6	0.8	1	$m^{1/3}$*log$m$	√m/ log$m$
Chess	19.44	47.22	94.44	33.33	2.77	0	0	0	0	77.77	69.44
WarpPIE10p	0.041	0.041	0.041	0.041	0.041	0.041	0	0	0	0.041	0.041
Coil2000	17.44	46.51	53.48	52.32	1.162	0	0	0	0	17.44	46.51
WarpAR10P	0.041	0.041	0.041	0.041	0.041	0.041	0	0	0	0.041	0.041
Email word subject	0	0	8.26	4.13	0.413	0	0	0	0	4.545	24.38
Orlaws10P	0.390	0.39	0.39	0.390	0.390	0.390	0.39	0	0	0.390	0.390
Pixraw10P	26.56	52.343	100	64.45	64.45	64.45	64.45	26.95	0	52.34	94.53
Tox-171	0	0	0.390	0.390	0.390	0	0	0	0	0	0.390

Table 4. Accuracy calculated using the naïve Bayes classifier

Threshold dataset	**Domain**	m/log$m$	√m*log$m$	0.01	0.2	Δ (performance metric)
Chess	**Text**	82.41	84.44	86.60	86.60	2.59
Coil2000	**Text**	89.55	89.55	89.55	89.55	1.711
Email word subject	**Text**	0	0	85.93	85.93	20.8
Average (text)		85.98	85.98	87.36	87.36	7.6
WarpPIE1p	**Image**	22.85	22.85	22.85	22.85	557.31
Orlaws10P	**Image**	62	62	62	62	158.97
Pixraw10P	**Image**	93	93	92	92	1.427
WarpAR1P	**Image**	20.76	21.53	21.53	21.53	525.1
Average (image)		49.65	49.845	49.595	49.595	310.7
Tox-171	**Microarray**	0	0	37	42	94.87

Tables 2–6 prove that different threshold values result in different numbers of features selected for the same dataset. In Tables 4–6, all values for each threshold have not been displayed due to space considerations. We observed experimentally that a threshold value of 0.2 yields optimal results. We also found that after feature selection, all three classifiers result in good classification accuracy for text datasets.

The datasets were categorized as low-dimension (D < 200) and large-dimension (D > 200). Tables 7 and 8 compare the results obtained with a threshold value of 0.2 using this algorithm with the results obtained using other existing feature selection algorithms.

When different algorithms are evaluated in terms of the percentage of features selected, we conclude that the ModifiedFAST feature selection algorithm obtains the least percentage of features in most of the datasets. This is shown in Fig. 4. ModifiedFAST yielded better results than other algorit hms in many of the datasets. The FAST algorithm took second place, followed by FCBF, CFS, and ReliefF. ModifiedFAST had the highest performance in the Chess, coil2000, WarpPIE10p, and WarpAR10P datasets. The

Table 5. Accuracy calculated using the C4.5 classifier

Threshold Dataset	Domain	m/log$m$	√m*log$m$	0.01	0.2	Δ (performance metric)
Chess	**Text**	94.08	94.55	94.08	94.08	2.82
Coil2000	**Text**	95.52	95.52	95.52	95.52	1.82
Email word subject	**Text**	0	0	85.93	85.93	20.8
Average (text)		94.8	94.8	91.84	91.84	8.48
WarpPIE1p	**Image**	26.66	26.66	26.66	26.66	650.2
Orlaws10P	**Image**	59	59	59	59	151.2
Pixraw10P	**Image**	92	93	84	84	1.303
WarpAR1P	**Image**	19.23	21.53	21.53	21.5	525.1
Average (image)		49.22	50.04	47.79	47.79	331.98
Tox-171	**Microarray**	0	0	35	41	89.74

Table 6. Accuracy calculated using the IB1 classifier

Threshold Dataset	Domain	m/log$m$	√m*log$m$	0.01	0.2	Δ (performance metric)
Chess	**Text**	93.17	92.45	91.99	91.99	2.75
Coil2000	**Text**	94.02	93.28	94.02	94.02	1.79
Email word subject	**Text**	0	0	71.87	71.87	17.40
Average (text)		93.595	93.595	85.96	85.96	7.31
WarpPIE1p	**Image**	20	20	20	20	487.8
Orlaws10P	**Image**	51	51	51	51	130.76
Pixraw10P	**Image**	98	98	98	98	1.52
WarpAR1P	**Image**	21.53	25.38	25.38	25.38	619.02
Average (image)		47.63	48.59	48.59	48.59	309.7
Tox-171	**Microarray**	0	0	47	43	120.51

proportion of selected features was improved in all these datasets. This shows that the optimal threshold value of 0.2 greatly improves the performance of the classification algorithms. In order to further rank the algorithms based on statistical significance, the Friedman test is used. It is used to compare k algorithms over d datasets by ranking the algorithms. The value of M given in Eq. (7) is calculated and compared with the critical value at α= 1%.

The test results falsified the null hypothesis, showing that all feature selection algorithms performed differently in terms of the percentage of selected features.

We then applied the Nemenyi test as a posthoc test [29] to explore the actual range by which pairs of algorithms differed from each other. As stated by the Nemenyi test, two classifiers are found to perform differently if the corresponding average ranks (R_x-R_y, where R_x and R_y are the average ranks of algorithms x and y, respectively) differ by at least the critical difference (CD).

$$CD = \boldsymbol{q}_{\propto}\sqrt{\frac{k(k+1)}{6N}}\ . \qquad (8)$$

In Eq. (8), k is the number of algorithms, N is number of datasets, and $\boldsymbol{q}_{\propto}$ is based on the Studentized range statistic divided by √2. Fig. 5 shows the results for these five algorithms with $\propto$ = 0.1 on seven datasets. The CD value is compared with the mean rank of each algorithm using ModifiedFAST. Overall, we found that the rank of ModifiedFAST was slightly higher than FAST, and much higher than other algorithms.

Table 7. Comparison of feature selection algorithms for low-dimensional datasets

Dataset	Feature selection algorithm	Percentage of selected features	Runtime (ms)
Chess	ReliefF	62.16	12660
	CFS	10.81	352
	FCBF	21.62	60
	FAST	16.22	105
	ModifiedFAST	2.7	5.4
Coil2000	ReliefF	50.00	304162
	CFS	11.63	1483
	FCBF	8.14	875
	FAST	3.49	866
	ModifiedFAST	1.16	1.25

Table 8. Comparison of feature selection algorithms for large-dimensional datasets

Dataset	Feature selection algorithm	Proportion of selected features
ORL10P	ReliefF	99.97
	CFS	2.76
	FCBF	2.61
	FAST	0.30
	ModifiedFAST	0.39
WarpPIE10p	ReliefF	91.00
	CFS	2.52
	FCBF	1.98
	FAST	1.07
	ModifiedFAST	0.133
WarpAR10P	ReliefF	62.89
	CFS	2.12
	FCBF	1.04
	FAST	0.21
	ModifiedFAST	0.19
PIX10P	ReliefF	100.00
	CFS	2.35
	FCBF	3.04
	FAST	0.15
	ModifiedFAST	64.45
TOX-171	ReliefF	64.60
	CFS	2.09
	FCBF	1.41
	FAST	0.28
	ModifiedFAST	0.39

VII. CONCLUSION

This paper finds an optimal threshold value that can be used for most feature selection algorithms, resulting in good subsets. The thresholds given by earlier feature selection algorithms are not optimal and need to be changed. We observed that different threshold values can result in different numbers of features selected and a different level of accuracy due to changes in the number of selected features. The best threshold found for the proposed algorithm was 0.2.

We have compared the performance of the Modified FAST algorithm, based on the percentage of features selected and classification accuracy, with the values given by other feature selection algorithms, such as FAST, ReliefF, CFS, and FCBF. For text datasets, we found that the proposed algorithm resulted in improved classification accuracy than it was before the algorithm was applied for all the classifiers. On the basis of the proposed performance parameter Δ, we observed that all values were positive for all three classifiers, demonstrating that ModifiedFAST is a better algorithm as assessed by a combination of classification accuracy and the number of selected features.

REFERENCES

[1] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: Wiley-Interscience, 2007.

[2] J. Huang, Y. Cai, and X. Xu, “A filter approach to feature selection based on mutual information,” in *Proceedings of the 5th IEEE International Conference on Cognitive Informatics (ICCI)*, Beijing, China, pp. 84-89, 2006.

[3] M. A. Hall and L. A. Smith, “Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper,” in *Proceedings of the 12th International Florida AI Research Society Conference*, Orlando, FL, pp. 235-239, 1999.

[4] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97, no. 1, pp. 273-324, 1997.

[5] L. Yu and H. Liu, “Feature selection for high-dimensional data: a fast correlation-based filter solution,” in *Proceedings of the 20th International Conference on Machine Learning (ICML2003)*, Washington DC, pp. 856-863, 2003.

[6] S. Das, “Filters, wrappers and a boosting-based hybrid for feature selection,” in *Proceedings of the 18th International Conference on Machine Learning (ICML2001)*, Williamstown, MA, pp. 74-81, 2001.

[7] D. Guan, W. Yuan, Y. K. Lee, K. Najeebullah, and M. K. Rasel, “A review of ensemble learning based feature selection,” *IETE Technical Review*, vol. 31, no. 3, pp. 190-198, 2014.

[8] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, 2005.

[9] K. Kira and L. A. Rendell, “The feature selection problem: traditional methods and a new algorithm,” in *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI)*, San Jose, CA, pp. 129-134, 1992.

[10] H. Almuallim and T. G. Dietterich, “Efficient algorithms for identifying relevant features,” in *Proceedings of the 9th Canadian Conference on Artificial Intelligence*, pp. 1-8, 1992.

[11] M. A. Hall, “Correlation-based feature selection for machine learning,” Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, 1999.

[12] L. Yu and H. Liu, “Feature selection for high-dimensional data: a fast correlation-based filter solution,” in *Proceedings of the 20th International Conference on Machine Learning (ICML2003)*, Washington, DC, pp. 856-863, 2003.

[13] Q. Song, J. Ni, and G. Wang, “A fast clustering-based feature subset selection algorithm for high-dimensional data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1-14, 2013.

[14] J. M. Santos and S. Ramos, “Using a clustering similarity measure for feature selection in high dimensional data sets,” in *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA)*, Cairo, Egypt, pp. 900-905, 2010.

[15] J. W. Jaromczyk and G. T. Toussaint, “Relative neighborhood graphs and their relatives,” *Proceedings of the IEEE*, vol. 80, no. 9,

pp. 1502-1517, 1992.

[16] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261-268, 1980.

[17] C. Zhong, D. Miao, and R. Wang, "A graph-theoretical clustering method based on two rounds of minimum spanning trees," *Pattern Recognition*, vol. 43, no. 3, pp. 752-766, 2010.

[18] Y. Xu, V. Olman, and D. Xu, "Minimum spanning trees for gene expression data clustering," *Genome Informatics*, vol. 12, pp. 24-33, 2001.

[19] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 68-86, 1971.

[20] O. Grygorash, Y. Zhou, and Z. Jorgensen, "Minimum spanning tree based clustering algorithms," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, Arlington, VA, pp. 73-81, 2006.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[22] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proceedings IEEE International Conference on Data Mining (ICDM 2001)*, San Jose, CA, pp. 107-114, 2001.

[23] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: Wiley, pp. 12-49, 1991.

[25] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1988.

[26] J. M. Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognition*, vol. 43, no. 6, pp. 2068-2081, 2010.

[27] A. Nagpal, D. Gaur, and S. Gaur, "Feature selection using mutual information for high-dimensional data sets," in *Proceedings of 2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, India, pp. 45-49, 2014.

[28] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86-92, 1940.

[29] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton University, NJ, 1963.

Arpita Nagpal

received a bachelor's degree from Punjab Technical University, Punjab, India and a Master's in Technology from Jaypee Institute of information and Technology, Noida in 2011. She is currently a Ph.D. student in the Department of Computer Science at ITM University, Gurgaon, India. Her research interests include data mining and machine learning.

Deepti Gaur

received an M.Tech in CSE degree from BIT Mesra Ranch, India and a Ph.D. from Banashali University, Banasthali India. Dr. Deepti Gaur is currently an associate professor at ITM University Gurgaon, Haryana, India. She has 17 years of teaching and research experience in the field of computer science and information technology. She had successfully completed a project of the AICTE govt. of India, based on data mining and machine learning. She has published more than 25 research papers in international journals and reputable international conferences. She was an organizing chair of an IEEE International Conference (IACC 2014) in India.

http://dx.doi.org/10.6109/jicce.2015.13.2.113