

# A New Proposed Feature Subset Selection Algorithm Based on Maximization of Gain Ratio

Arpita Nagpal<sup>(✉)</sup> and Deepti Gaur

Computer Science Department, Northcap University, Gurgaon, India  
{arpitanagpal, deeptigaur}@ncuindia.edu

**Abstract.** Feature subset selection is one of the techniques to extract the highly relevant subset of original features from a dataset. In this paper, we have proposed a new algorithm to filter the features from the dataset using a greedy stepwise forward selection technique. The Proposed algorithm uses gain ratio as the greedy evaluation measure. It utilizes multiple feature correlation technique to remove the redundant features from the data set. Experiments that are carried out to evaluate the Proposed algorithm are based on number of features, runtime and classification accuracy of three classifiers namely Naïve Bayes, the Tree based C4.5 and Instant Based IB1. The results have been compared with other two feature selection algorithms, i.e. Fast Correlation-Based Filter Solution (FCBS) and Fast clustering based feature selection algorithm (FAST) over the datasets of different dimensions and domain. A unified metric, which combines all three parameters (number of features, runtime, classification accuracy) together, has also been taken to compare the algorithms. The result shows that our Proposed algorithm has a significant improvement than other feature selection algorithms for large dimensional data while working on a data set of image domain.

**Keywords:** Classification · Feature selection · Filter method · Mutual information

## 1 Introduction

Nowadays, the data is increasing at tremendous speed in terms of volume and number of features. The major difficulty is called as ‘the curse of dimensionality’ [20]. The learning algorithms are greatly affected by the dimensionality of the data. The task of dimension reduction is the process of reducing the number of features or variables of an object under consideration.

Dimension reduction and feature selections are two techniques which can reduce the attributes of data for classification tasks. But still there lies some difference in both the techniques. Dimensionality reduction is creating new attributes as a combination of the old attributes like in PCA (Principal Component Analysis) whereas in feature selection some attributes whose information overlap with other attributes called redundant attributes are removed from the dataset. PCA involves feature transformation and obtains a set of transformed features rather than a subset of the original features [17]. Dimension

reduction of the Big Data (i.e. Voluminous data) can help in the easy processing of data for some decision-making. Generally, there are four types of feature selection techniques filter, wrapper, embedded and hybrid technique. The filter approach to feature selection is to remove the features based on the general characteristics of the training data without involving any learning algorithm [4, 6]. The wrapper method involves a classification algorithm to determine the efficiency of the subset generated. The embedded methods are specific to the learning algorithm and during its training phase feature selection algorithms are applied. Hybrid approach which has been recently been proposed for high dimensional data is a combination of both filter and wrapper method [1, 2], [4], [5].

Janecek et al. [6] have proven by experimental results that among the feature selection methods, wrappers tend to produce the smallest feature subsets with very comprehensive classification accuracy, in many cases the best overall dimension reduction methods. But the wrapper method is computationally expensive as it needs the learning algorithm to evaluate the selected features performance and find the final selected set. When there is a small set of features wrapper can be applied, but when the number of features becomes very large, filter model is usually chosen due to its computational accuracy and efficiency [4]. In high dimensional dataset, feature selection becomes more efficient than a dimension reduction technique of PCA.

This work is based on feature selection using a filter approach. A new algorithm which can efficiently remove both irrelevant and redundant features for Big Data has been developed. Here, we have experimentally shown its performance by taking some of the high dimensional data sets. Our method calculates Gain Ratio between each feature and the class attributes. The feature having highest gain ratio value becomes the first node of the list. Second highest gain ratio value feature is now considered. It's gain ratio is calculated with the first feature and if it satisfies the condition then it is added to the list otherwise ignored. This method is based on the multiple feature correlation technique in which next feature's correlation is checked with the correlation of all the features present in the list. If the feature not present in the list satisfies the condition, then it is added to the list otherwise it is denied and not involved in further computations. Final list depict the set of selected features. Multiple feature correlation technique reduces the number of computations and produces a subset of independent feature free from overlap of information contained in them. The Proposed algorithm is tested on 9 datasets from different domains (image, microarray and text data). Our method has been compared with two filter feature selection algorithms: Fast-Correlation based Filter (FCBS) [4] and FAST [7] and it shows that the Proposed algorithm outperforms them in terms of classification accuracy, runtime and number of features selected in most data sets.

The rest of the paper has been organized as follows: in Sect. 2, related work will be explained. In Sect. 3, we describe the theoretical background. Section 4 discusses the complete algorithm with an example. Section 5 gives the result and comparisons with other algorithm based on dimension of data sets and domain. Finally, in Sect. 6 we draw conclusions based on the experimental results.

## 2 Related Work

Feature selection aims to find the subset of features from the original set of features by removing irrelevant and redundant features. Within the filter model of feature selection there have been many algorithms proposed such as Relief [11], Relief-F [12], FOCUS, FOCUS-2[13], Correlation based Feature Selection (CFS) [8], FCBS[4], FAST[7]. Relief [11] and Correlation based Feature Selection (CFS) [8] methods remove only the irrelevant features. The algorithms proposed later remove both irrelevant and redundant features.

Liu and Yu [10], has given that the general process of feature selection is divided into four processes: subset generation, subset evaluation, stopping criterion and subset validation.

Each of the algorithms described in literature uses one of the subset generation techniques either complete search, sequential search or random search, all of either in forward, backward or bidirectional directions [3, 10]. After generating a subset, this subset needs to be evaluated. Form data set with 'n' features  $2^n$  subsets can be generated. A search algorithm is often employed to find the optimal feature subsets. To find which the optimal subset is, each algorithm has an evaluation technique based on different heuristics. Generally, the statistics used as evaluation measures are distance based measures, Information gain [4], correlation coefficient [9], consistency measures [3]. The process terminates when it reaches a stopping criterion.

CFS (Correlation-based Feature Selection), one of the feature selection method uses forward greedy feature selection method. It is based on the hypothesis that a good feature subset is the one that contains features highly correlated to the class, yet uncorrelated to each other [8].

Fast-Correlation based Filter (FCBS) algorithm given by Yu and Liu [4] removes both irrelevant and redundant features by the use symmetric uncertainty as an evaluation measure. It uses the concept of correlation based on information theory of mutual information and entropy to calculate the uncertainty of a random variable. It removes features by performing a pairwise correlation.

Recently, an algorithm named FAST has been developed by Song et al. (2013) which uses symmetric uncertainty as an evaluation measure to remove both irrelevant and redundant features. It generates a Minimum Spanning Tree (MST) by calculating the correlation of each feature with every other feature. Then it partitions the MST in forest with each tree representing a cluster, represented features are then selected from each cluster. These representative features form a final subset of features.

Quite different from FAST algorithm, our Proposed algorithm uses Gain Ratio as an evaluation measure to remove irrelevant and redundant features. It does not calculate the correlation of each feature with every other feature instead, it calculates correlation with only those present in the feature subset formed. The problem becomes NP-hard if data are high dimensional and we keep on finding correlation of each feature with every other feature in the dataset. Unlike, FCBF it does not perform pairwise correlation, instead it performs a correlation with all present in the feature subset. It is a greedy forward selection method and adds features in forward direction.

### 3 Theoretical Background

Entropy is a measure of uncertainty of a random variable. If  $X$  is a discrete random variable with alphabet  $x$  and probability mass function  $p(x) = \text{PR} \{X = x\}$ . The probability mass function is denoted by  $p(x)$ . The entropy  $H(X)$  of a discrete random variable  $X$  is defined by:

$$H(X) = - \sum_x p(x) \log_2 p(x) \quad (1)$$

Information gain is a measure of the amount of information that one random variable contains about another random variable. The information gain  $I(X; Y)$  give the relation between two variables. Information gain or Mutual Information [16] is given by

$$\begin{aligned} I(X; Y) &= H(Y) - H(Y|X) = H(X) - H(X|Y) \\ &= H(Y) + H(X) - H(X, Y) \end{aligned} \quad (2)$$

Where  $H(X)$ ,  $H(Y)$  is the individual entropies of two random variables  $X$  and  $Y$ . Entropy of  $X$  is based on the individual probabilities of variables in  $X$ . It gives the diversification in values.

Our algorithm is based on the concept of information theory. It is based on the assumption that if there are two different random variables, then larger the value of information gain between them, stronger the relationship they share. Theorem 1 as given in [14].

**Theorem 1:** For any discrete random variable  $Y$  and  $Z$   $I(Y; Z) \geq 0$ . Moreover,  $I(Y; Z) = 0$  if and only if  $Y$  and  $Z$  are independent.

Let  $D$  be a full feature set,  $f_i \in D$  be one of the feature.  $C$  is the target class attribute. The definitions can be defined as follows:

**Definition 1 (Relevance):** A feature  $f_i$  is relevant to the class  $C$  if and only if  $GR(f_i, C) \geq \theta$  i.e. the gain ratio value between a feature and class attribute should be greater than a predefined threshold  $\theta$ .

**Definition 2 (Redundancy):** Let  $D$  be a full set of features in the dataset, Two features  $f_i$  and  $f_j$  are redundant peers if and only if  $GR(f_i, f_j) > GR(f_i, C) \cap GR(f_i, f_j) > GR(f_j, C)$ . Otherwise they are not redundant.

Redundant features are those that contain much of the common information. Definition 2 indicates that a feature is redundant to other feature in the dataset, if the correlation between them is greater than the correlation of feature with the Class  $C$ . Here, Information gain ( $I$ ) is used to measure the correlation between the two features. The drawback of information gain is that it is biased towards the feature with all different values. It prefers to select the attribute having large number of different values in its instances. For example, consider an attribute `student_ID` in the students' record data that has all different values in it. Therefore, the information gained on this attribute will be maximal. To overcome this drawback, we use an extension of information gain

known as gain ratio, which overcomes this bias. It normalizes information gain using a “split info” value defined as:

$$\text{SplitInfoB}(D) = - \sum_{i=1}^u \frac{|D_i|}{|D|} \times \log_2\left(\frac{|D_i|}{|D|}\right) \tag{3}$$

The data set  $D$  is split into  $u$  portions of attribute  $B$ .  $|D_i|$  is total number of tuples in the  $i$ th portion and  $|D|$  is the number of attributes in complete dataset. Splitinfo gives the information generated after partitioning whereas information gain, measures the information with respect to classification that is obtained based on the same partitioning. The gain ratio is defined as

$$\text{Gain Ratio}(B) = \text{Gain}(B)/\text{SplitInfo}(B) \tag{4}$$

**Definition 3 (maxGR):** It is the largest value of Gain ratio(GR) found after calculating the gain ratio of the new feature  $f_i$  ( $f_i \in D$ ) with all the features already present in the tree.

Suppose at any step ( $i-1$ ), some features are already present in tree  $U$ . To add a new feature  $f_i$  ( $f_i \in D \setminus U_{i-1}$ ) to the tree find  $GR(f_i; f_j)$  for each feature  $f_j \in U$ . The largest value of  $GR(f_i; f_j)$  is the maxGR value.

**Definition 4 (Relevantf):** It gives us the relation between the feature  $f_i$  ( $f_i \in D$ ) and the class  $C$ . The relevance between the feature and the class is found on the basis of the value of the Gain ratio value between them. It is denoted by  $GR(f, c)$ .

**Definition 5 (Redundantf):** A feature is added as the tree node if,  $f_i \in D$  and  $\text{maxGR}(X; Y) < GR(Y, C)$ . This means that both features ( $X, Y$ ) are not correlated with each other.

Our algorithm is based on the assumption that stronger the relation between two variables, larger the value of gain ratio they will have. A feature becomes redundant if its relation is stronger with any of the feature already present in the feature subset than its relation with its class attribute.

## 4 Algorithm and Analysis

The Proposed algorithm depicted in Figs. 1 and 2 is based on the methodology of gain ratio as described before. Input to the algorithm is all the features given in the data set  $D = \{f_1, f_2, \dots, f_m\}$  and Class  $C$ . The first task is to remove the irrelevant features. Relevantf gives the value of  $GR(f_i, C)$ . If this value is greater than the predefined threshold, then those features are kept as relevant ones, others are ignored.

The next step is to remove redundant features. To do this, first arrange the list of all relevant features ( $D'$ ) found along with their Gain ratio value ( $G'$ ) in descending order. The first feature in the  $D'$  list will be the feature having the highest value of gain ratio with the class. According to our assumption made the first feature in  $D'$  is most relevant and has to be kept.

```

Input: D(f1,f2,.....fm, C), the given dataset
      θ // Relevant Threshold
Output: F //final feature subset
Begin:
// Irrelevant feature removal
1. For each feature, i=1 to m do
2. Relevantf= GR(fi,C)
3. If Relevantf > θ
4. Append fi to Snewlist
5. End;
6. End

//Redundant Feature removal from the elements in Snewlist
which has 'k' relevant features

7. D'=Arrange Snewlist in Descending order.
8. G'=Arrange GR(f,C) in descending order.
9. ΔT.firstnode ← get first element from G'
10. Y ← second element from G'
11. L ← 1
12. F=sub (ΔT,Y,L,G')
13. End;

```

**Fig. 1.** Algorithm

To find correlation in polynomial time, we arrange it in descending order and find correlation of new feature with every feature in tree U ignoring redundant features found.

This step starts with an empty list named T. First add the first feature from D' and keeps on adding non redundant features into it. Check the first feature with every other feature in terms of the redundancy, according to Definition 5. The features found redundant with the first feature is ignored and their redundancy is never again checked with any other feature.

Suppose at any step (i-1), some features are already present in Tree U. To add a new feature f<sub>i</sub> (f<sub>i</sub> ∈ D \ U<sub>i-1</sub>) to the list first it finds GR (U<sub>s-1</sub>; f<sub>i</sub>). The following criterion is used:

$$f_{(i)} = \operatorname{argmax} \operatorname{GR}(U_{s-1}; f_i)$$

This means that from the feature in the list which has the largest value of GR with f<sub>i</sub> is picked.

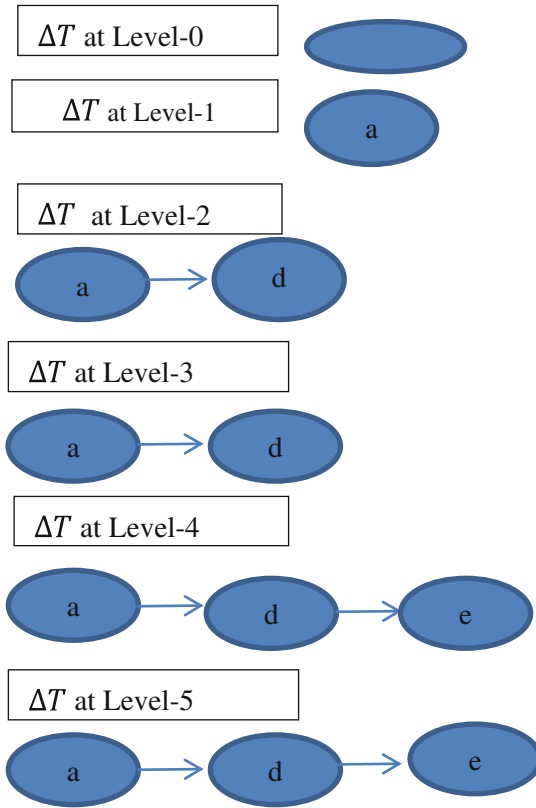
This approach maximizes the gain ratio between feature subset found and the class attribute. If maxGR with all features in U at step i-1 is less than GR of new feature and class C then a new feature gets added to the list. If this condition is not satisfied, then the feature has lost the opportunity to be added in the list and hence ignored. Figure 1 gives a method to remove the irrelevant features. The remaining features are then arranged in descending order. Pass the first node to list T, the first element from D'.

<p><b>Algorithm:</b> sub(<math>\Delta T, Y, L, G'</math>)</p> <p><b>Input:</b> (<math>\Delta T</math>, Value of Gain ratio's <math>G'</math>, Next element <math>Y</math>, Current depth <math>L</math>, number of relevant features <math>k</math>)</p> <p><b>Output:</b> Final tree, <math>F</math></p> <p><b>Begin:</b></p> <ol style="list-style-type: none"> <li>1. <math>F = \Delta T.L^{\text{th}}</math> Node;</li> <li>2. For every element <math>X \in F</math> {</li> <li>3.     Calculate <math>GR(X; Y)</math></li> <li>4.     Extract <math>\max GR(X; Y)</math> as defined in Definition 3</li> <li>5. }</li> <li>6. If(<math>\max GR(X; Y) &lt; GR(Y, C)</math>) {</li> <li>7.     Extract <math>Y</math> from <math>G'</math></li> <li>8.     Add to <math>\Delta T.L^{\text{th}}</math> Node</li> <li>9. }</li> <li>10. If(<math>L \leq k</math>) {</li> <li>11.     <math>Y = \text{get next element from } G'</math>;</li> <li>12.     <math>L \leftarrow L + 1</math>;</li> <li>13.     sub(<math>\Delta T, Y, L, G'</math>)</li> </ol> <p><b>End;</b></p>
---

**Fig. 2.** The subroutine of the algorithm

Then  $T$  is passed to the subroutine sub given in Fig. 2. Figure 2 keeps on adding the new features to  $T$  according to  $\max GR$  explained in Definition 3. If  $\max GR(X; Y) > GR(Y, C)$  then the algorithm will ignore the feature and move to the next feature. Suppose the number of features found after removing irrelevant features is  $k$ . The searching process terminates when all the ' $k$ ' features in the dataset have been examined.

To clarify the search procedure, we explain the algorithm through an example. Let the set of features in the dataset is  $D = \{a, b, c, d, e, f\}$  and the class attribute is  $C$ . ' $m$ ' is the number of features in the dataset. Here,  $m = 6$ . Suppose after going through steps 1 to 5 in Fig. 1 all feature's Relevantf value except feature  $f$  satisfies the predetermined threshold and gets added in the Snewlist. Snewlist becomes  $\{a, b, c, d, e\}$ .  $k$  i.e. the number of features in Snewlist becomes 5. To further process Snewlist, it is arranged in descending order in  $G' \{a, d, b, e, c\}$  according to their gain ratio value with the class. Figure 3 depicts the search procedure to add the new feature to the empty data structure  $T$ . First the list  $T$  is empty  $\Delta T = \{\}$ . At level 1, the first node ' $a$ ', from  $G'$  is taken and added to the list. Second value from  $G'$  i.e.  $d$  is taken. Calculate  $GR(a, d)$  and compare it with  $GR(d, \text{class } C)$  according to step 6 of Fig. 2. ' $d$ ' satisfies the condition so list adds ' $d$ ' to itself. Pick the third element from  $G'$  which is ' $b$ '. Calculate  $GR(a, b)$  and  $GR(d, b)$ . Pick the maximum  $GR$  value from them and compare it with  $GR(b, \text{class})$ . According to steps 6 to 8 of Fig. 2, it does not satisfy the condition, so it is not added to the list and it never used again in further comparisons. The next element from the list is ' $e$ '. Repeat the same steps with  $e$  as done with last element ' $b$ '. Here, we found that  $e$  satisfies the condition so it is added to the  $T$ . The last element is ' $c$ ' Calculate  $GR(a, c)$ ,  $GR(d, c)$  and  $GR(e, c)$ . Take out the maximum value of  $GR$  out of these three values and Compare it with  $GR(c, \text{Class})$ . It is observed that  $c$  does not satisfy the condition



**Fig. 3.** Depicting the search procedure

so it is filtered out. Finally  $\Delta T$  at level-5 contains the feature subset after removing both irrelevant and redundant features.

## 5 Empirical Study

### 5.1 Datasets

In our experiment we have employed 11 datasets which are from different domains, text, image, Microarray. The number of features varies from 36 to 10,000 features. The data sets used in the experiments are taken from the UCI machine learning repository [18], [tunedit.org/repo/Data/Text](http://tunedit.org/repo/Data/Text) and [featureselection.asu.edu](http://featureselection.asu.edu) [21]. Microarray datasets of Colon\_1 [22] actually contained 22883 features, but a preprocessing strategy [23] has reduced the features to 8826. Leukemia [24] is one of the cancer classification data which comes under microarray domain has also been pre-processed. The features with a value less than 20 and more than 16000 were removed. The summary of a data set is given in Table 1.



**Table 1.** Summary of data sets

Data set	Number of instances	Number of features	Number of classes	Domain
WarpAR10P	130	2400	10	Image, face
Chess	3196	36	2	Text
WarpPIE10p	210	2420	10	Image, Face
Arcene	100	10000	2	Image
Coil2000	134	86	2	Text
Email word subject	64	242	2	Text
tox-171	100	5749	3	Microarray
Pix10P	100	10000	10	Image,face
orlawn10p	100	10304	10	Image, face
Colon_I	37	8826	2	Microarray
Leukemia	72	7128	3	Microarray

### 5.2 Experimental Results and Comparisons

To verify the experimental results three parameters percentage of selected features, runtime and Classification accuracy has been calculated. Two feature selection algorithms FCBF and FAST have been compared with the Proposed algorithm on these three parameters. In the performed experiments, the relevant threshold,  $\theta$  is the GR((ain Ratio) value of the  $\lfloor m/\log m \rfloor_{th}$  ranked feature for all the datasets. This threshold is as a suggestion given by Yu and Liu [4]. By applying, 10-fold cross validation classification accuracies of three different classifiers (Naïve Bayes, C4.5, and IB1) on all the datasets accurate results could be achieved.

The Proposed algorithm has also been compared with FCBF and FAST on the basis of multi-criteria Metric EARR (Extended Adjusted Ratio of Ratios) proposed by Wang [15]. Under one unified metric, the classification accuracy with a runtime and number of features selected are integrated. This unified metric EARR evaluates the performance by taking the ratio of the metric values. Let  $D = \{D_1, D_2, \dots, D_n\}$  be a set of n data sets, and  $A = \{A_1, A_2, \dots, A_n\}$  be a set of M FSS algorithms. Then, the EARR of  $A_i$  to  $A_j$  over  $D_k$  be defined as:

$$EARR_{A_i, A_j}^{D_k} = \frac{acc_i^k / acc_j^k}{1 + \alpha \cdot \log(t_i^k / t_j^k) + \beta \cdot \log(n_i^k / n_j^k)} \quad (1 \leq i \neq j \leq M, 1 \leq k < N) \quad (5)$$

$\alpha$  and  $\beta$  are user defined parameters, which tells us how much importance should be given to the runtime and number of features selected respectively.  $acc_i^k$  is the accuracy of ith algorithm of ‘k’ th dataset.  $t_i^k$  and  $n_i^k$  are the runtime and number of selected features of dataset ‘k’ on ith algorithm respectively.

This will allow the user to tell how much the runtime and feature selected should dominate accuracy. When comparing multiple feature selection algorithms, it takes the arithmetic mean of the  $EARR_{A_i, A_j}^{D_k}$  of  $A_i$  to another algorithm  $A_j$  on  $D$ . It is defined as:

$$EARR_{A_i}^{D_k} = \frac{1}{M-1} \sum_{j=1 \& j \neq i}^M EARR_{A_i, A_j}^{D_k} \quad (6)$$

The larger the value of EARR, better is the corresponding algorithm on a given data set  $D$  [15].

### 5.3 Dimension Wise Comparison of Feature Selection Algorithms

Tables 2 and 3 depict the comparison of low dimensional data set and large dimensional data set respectively, of three feature selection algorithm on the basis of three parameters (percentage of feature selected, runtime and accuracy of three different classifiers). According to our assumption low dimension datasets consisting of less than 200 features and high dimensional data sets are datasets with features greater than 200 or more. Accuracy on all three classification algorithm on all the data sets is conducted in Weka [19].

We observe from Tables 2 and 3 that the Proposed algorithm is performing better in terms of percentage of the selected features and runtime for large dimensional data rather than for low dimensional data. The Proposed algorithm is securing first position for all the evaluation metrics for Arcene data, PIX10P, ORL10P set which have around 10, 000 features. So, we can say that this technique works well on large dimensional data sets.

### 5.4 Domain Wise Comparison of Feature Selection Algorithms

Data sets used in the experiments belongs to three types of Domain Text, Image and Microarray. Tables 4, 5, 6, 7, 8 gives the average values obtained domain wise for three performance metrics (Percentage of selected features, Runtime and Classification accuracy). If we observe the percentage of the selected features and runtime in Tables 4 and 5, data sets in image domain are showing a comparable performance. In case of text dataset, there is an improvement in some cases. Table 4 elaborates averages of the three feature selection algorithm on each data set domain wise.

Table 4 shows that the Proposed algorithm selects fewer features than FCBF and FAST in all three domains text, Image and Microarray. Further Comparisons depicts that:

1. For text datasets, Proposed algorithm selects 5.89 percent of the features. FCBF and FAST have a margin of 1.28 percent between them.
2. For image datasets, Proposed algorithm ranks 1 by selecting 0.334 percent of features with a margin of 0.15 from FCBF and 5.25 from FAST.
3. For Microarray datasets, Proposed algorithm selects 1.25 and 6.7 lesser percentages of features than FCBF and FAST respectively.

**Table 2.** Comparison of feature selection algorithm for low- dimensional data sets

Data set	Feature selection algorithm	Percentage of selected features	Runtime (s)	Accuracy by Naïve Bayes	Accuracy by C4.5	Accuracy byIB1
Chess	FCBF	16.6	25.36	92.5	94.08	92.05
	FAST	5.5	164.42	75.4	75.4	70.3
	Proposed	11.11	56.24	67.08	69.8	63.7
Coil2000	FCBF	3.48	0.190	92.5	95.5	94.77
	FAST	2.3	1.093	94.04	94.02	93.04
	Proposed	2.32	0.32	94.06	95.52	94.77

**Table 3.** Comparison of feature selection algorithm for large – dimensional data sets

Data set	Feature selection algorithm	Percentage of selected features	Runtime (s)	Accuracy by Naïve Bayes	Accuracy by C4.5	Accuracy by IB1
ORL10P	FCBF	0.39	0.53	48	44	45
	FAST	0.39	69.21	49	50	59
	Proposed	0.39	0.36	62	63	58
WarpAR10P	FCBF	0.041	7.15	16.15	16.92	17.69
	FAST	0.19	1476	15.03	18.69	15.33
	Proposed	0.041	8.35	33.84	33.07	26.15
Warp PIE10p	FCBF	0.041	11.19	32.95	34.28	35.23
	FAST	0.133	7411	33.75	36.60	34.92
	Proposed	0.041	0.78	32.8	34.28	35.2
Arcene	FCBF	0.01	23.143	58	54	61
	FAST	0.01	5164	57	55	59
	Proposed	0.03	94.8	64	64	58
PIX10P	FCBF	1.953	0.482	90	98	97
	FAST	27.34	60.3	89.07	98.07	97
	Proposed	1.17	0.78	90	98.08	97
TOX-171	FCBF	0.39	0.0040	40	37	46
	FAST	0.39	0.0025	40	37	44
	Proposed	0.39	0.44	40	40	47
Email-word subject	FCBF	8.26	0.157	67.18	67.18	57.8
	FAST	24.38	8.23	67	66.08	59.9
	Proposed	4.54	0.046	67.18	67.18	60.9
Colon_1	FCBF	0.05	4.72	32.43	89.18	81.08
	FAST	0.30	8.66	72.5	88.5	80.8
	Proposed	0.02	2.79	83.78	75.65	81.08
Leukemia	FCBF	0.02	42.64	52.77	52.77	56.94
	FAST	0.07	69.21	77.16	52.77	55
	Proposed	0.01	28.46	52.77	52.77	56.94

**Table 4.** Comparison based on percentage of selected features

Dataset	Domain	Proposed	FCBF	FAST
Chess	Text	11.11	16.6	5.5
Coil2000	Text	2.32	3.48	2.3
Email word	Text	4.25	8.26	24.38
Average (Text)		5.89	9.44	10.72
WarpAR10P	Image	0.041	0.041	0.19
Warp PIE10p	Image	0.041	0.041	0.133
Arcene	Image	0.03	0.01	0.01
PIX10P	Image	1.17	1.953	27.34
ORL10P	Image	0.39	0.39	0.39
Average (Image)		0.3344	0.487	5.61
Tox-171	Microarray	4.54	8.26	24.38
Leukemia	Microarray	0.01	0.02	0.07
Colon_I	Microarray	0.02	0.05	0.30
Average (Microarray)		1.52	2.77	8.25

**Table 5.** Comparison based on runtime

Dataset	Domain	Proposed	FCBF	FAST
Chess	Text	56.24	25.6	164.42
Coil2000	Text	0.32	0.190	1.093
Email word	Text	0.046	0.157	8.23
Average (Text)		18.868	8.649	57.91
WarpAR10P	Image	8.35	7.15	1476
Warp PIE10p	Image	0.78	11.19	7411
Arcene	Image	94.8	11.19	7411
PIX10P	Image	0.78	0.482	60.3
ORL10P	Image	0.36	0.53	69.21
Average (Image)		21.008	6.1084	3285.5
Tox-171	Microarray	0.44	0.0040	0.0025
Leukemia	Microarray	28.46	42.64	69.21
Colon_I	Microarray	2.79	4.724	8.66
Average (Microarray)		10.56	15.788	25.95

If we observe the runtime, the Proposed algorithm is showing a significant improvement compared to FAST and FCBF algorithm in all the domains. Table 5 depicts the average runtime of all three algorithms domain wise.

For text data set, the Proposed algorithm is 39.05 times faster at runtime than FAST. For Image data set, it is 3,264.49 times faster than the FAST algorithm. In case of microarray data, the runtime of Proposed algorithm has decreased by 5.22 and 15.39 as compared to FCBF and FAST respectively.

Tables 6, 7, 8 depicts the 10-fold cross validation classification accuracies of three classifiers on 11 data sets domain wise.

**Table 6.** Average accuracy domain wise for Naïve Bayes classifier

Dataset	Domain	Proposed	FCBF	FAST
Chess	Text	67.08	92.5	75.4
Coil2000	Text	94.02	92.5	94.04
Email word	Text	67.18	67.18	67
Average (Text)		76.09	84.06	78.81
WarpAR10P	Image	33.84	16.5	15.03
Warp PIE10p	Image	32.8	32.95	33.75
Arcene	Image	64	58	57
PIX10P	Image	90	90	89.07
ORL10P	Image	62	48	49
Average (Image)		56.52	49.09	48.77
Tox-171	Microarray	40	40	40
Leukemia	Microarray	52.77	52.77	77.16
Colon_l	Microarray	83.78	32.43	72.5
Average (Microarray)		58.85	41.73	63.22

**Table 7.** Average accuracy domain wise for C4.5 classifier

Dataset	Domain	Proposed	FCBF	FAST
Chess	Text	69.8	95.5	94.02
Coil2000	Text	95.52	95.5	94.02
Email word	Text	67.18	67.18	66.08
Average (Text)		77.5	86.06	84.70
WarpAR10P	Image	33.07	16.92	18.69
Warp PIE10p	Image	32.8	32.95	33.75
Arcene	Image	64	54	55
PIX10P	Image	98.08	98	98.07
ORL10P	Image	<b>63</b>	<b>44</b>	<b>50</b>
Average (Image)		58.19	49.17	51.102
Tox-171	Microarray	40	37	37
Leukemia	Microarray	52.77	52.77	52.77
Colon_I	Microarray	75.65	89.18	88.5
Average (Microarray)		56.14	59.65	59.42

We observe that Naïve Bayes classification accuracy after applying feature selection algorithm has a positive improvement in all the domains. It is noted that:

1. Under Image dataset domain FCBF, FAST is 86.8 percent and 86.2 percent better in accuracy than Proposed algorithm.
2. For Arcene dataset, i.e. a large dimensional dataset the Naïve Bayes classification accuracy of Proposed algorithm has been increased by 6, 7 times by FCBF and FAST respectively.

**Table 8.** Average accuracy domain wise for IB1 classifier

Dataset	Domain	Proposed	FCBF	FAST
Chess	Text	63.7	92.05	70.3
Coil2000	Text	94.77	94.77	93.04
Email word	Text	60.9	57.8	59.9
Average (Text)		75.35	81.54	74.41
WarpAR10P	Image	26.15	17.69	15.33
Warp PIE10p	Image	35.2	35.2	34.92
Arcene	Image	58	61	59
PIX10P	Image	97	97	97
ORL10P	Image	58	45	59
Average (Image)		54.87	51.178	53.05
Tox-171	Microarray	47	46	44
Leukemia	Microarray	56.94	56.94	55
Colon_I	Microarray	81.08	81.08	80.8
Average (Microarray)		61.67	61.34	59.93

3. For WarpAR10p, datasets i.e. a large dimensional dataset the Naïve Bayes classification accuracy of Proposed algorithm has been increased by 17.34 and 18.81 times for FCBF and FAST respectively.
4. The Proposed algorithm has increased the classification accuracy of the Naïve Bayes classifier by 0.9 times than that of FAST and is performing almost the same in case of FCBF for WarpPie10p dataset.
5. For microarray dataset Colon\_I Naïve Bayes accuracy of Proposed algorithm has increased by 38.7 percent and 86.5 percent from FCBF and FAST respectively.

For C4.5 classifier, we observe from Table 7 that:

1. For Image dataset, FCBF and FAST have 9.02, 7.08 lesser accuracy value than the Proposed algorithm.
2. For Arcene dataset, one of a large dimensional data set in image domain the classification accuracy of Proposed algorithm has been increased by 10, 9 times by FCBF and FAST respectively.
3. For WarpAR10p datasets i.e. a large dimensional image datasets the classification accuracy of Proposed algorithm has been increased by 16.15 and 14.38 times for FCBF and FAST respectively.
4. For WarpPIE10p dataset Proposed algorithm is performing almost in a similar way as FCBF and FAST.
5. In case of microarray datasets Tox-171 the accuracy of Proposed algorithm has increased 3 times from FAST and FCBF. In case of Colon\_I it has increased 13.53 times and 12.85 times for FCBF and FAST respectively.

We observed the classification accuracies for IB1 classifier in Table 8 and noted that:

1. For Image dataset FCBF and FAST have a difference of 3.69 and 1.82 respectively, than Proposed algorithm.

2. For Arcene dataset, i.e. a large dimensional dataset the classification accuracy of Proposed algorithm has been decreased by 3, 1 margin by FCBF and FAST respectively.
3. For WarpAR10p datasets one of a large dimensional datasets the classification accuracy of Proposed algorithm has been increased by 8.46 and 10.82 value for FCBF and FAST respectively.
4. For WarpPie10p dataset Proposed algorithm is performing almost in a similar way as FCBF and FAST.
5. For microarray data, Proposed algorithm has increased the classification accuracy for Tox-171 and Colon\_1 dataset by 5.4 percent and 1.4 percent by FAST algorithm.

In order to further explore, which feature selection algorithm is performing better, a unified, multi-criteria metric EARR (Wang et al. 2013) has been used. We have taken number of features selected, runtime and Naïve Bayes classification accuracies values from Tables 4, 5, 6 respectively. Using Eq. 5, we calculate the EARR values of Proposed algorithm with FCBF and FAST on all datasets. Table 9 depicts the different EARR values found between two different algorithms.

According to Wang et al., the value of  $EARR A_i, A_j$  is greater than (or equal to, or smaller than) that of  $EARR A_i, A_j$  indicates that  $A_i$  is better than (or equal to, or worse than)  $A_j$ .

The Proposed algorithm when compared with FCBF on Image dataset, the EARR (Proposed, FCBF) value is 1.15 which is greater than EARR (FCBF, Proposed) in value of 0.868 at  $\alpha = \beta = 0.001$ . When comparing Proposed algorithm with FAST on image data set gives a value of EARR (Proposed, FAST) as 1.16 and EARR (FAST, Proposed) as 0.85. Here also we can say that Proposed is better than FAST.

The EARR value of text dataset at  $\alpha = \beta = 0.001$ , when compared with FCBF is EARR (Proposed, FCBF) is 0.905 and EARR (FCBF, Proposed) is 1.115. Comparing Proposed algorithm with FAST algorithm gives a value of EARR (Proposed, FAST) as 0.965 and EARR (FAST, Proposed) as 1.035. Here we found that FCBF and FAST are better than Proposed.

For Microarray dataset, EARR value (Proposed, FCBF) is 1.411 and EARR value (FCBF, Proposed) is 0.708. This implies that Proposed algorithm is better than FCBF. In case of FAST algorithm, EARR (Proposed, FAST) is 0.931 and EARR (FAST, Proposed) is 1.072 indicates FAST is better than Proposed.

So, from Table 9 it can be observed that the value of EARR Proposed algorithm is better in case of an image dataset when compared with both FCBF and FAST. The Proposed algorithm is also better than FCBF in case of microarray data. However, for text datasets FCBF and FAST are better than the Proposed algorithm.

**Table 9.** EARR values calculated to compare two algorithms

	Image			Text			Microarray		
	Proposed	FCBF	FAST	Proposed	FCBF	FAST	Proposed	FCBF	FAST
Proposed		0.8688	0.85		1.115	1.035		0.708	1.072
FCBF	1.15		0.990	0.9051		0.936	1.4116		1.512
FAST	1.16	1.0133		0.965	1.129		0.9319	0.660	

## 6 Conclusion

In this paper, a new feature selection technique which can work well on high dimensional data has been introduced. To reach this goal a correlation based filter approach using gain ratio is implemented to find the optimal and complete feature subset. The Proposed algorithm can eliminate both irrelevant and redundant features. By extensive experiments and calculations, we have shown that it works well for large dimensions (thousands of features) data set for classification.

The performance of the Proposed algorithm has been compared with two of the existing feature selection algorithms FCBF and FAST on the different data sets. Datasets belong to three domain, text, image and microarray data. We found that the proposed method ranks 1 for image data set and microarray dataset in case of three classifiers naïve Bayes, C4.5 and IB1. Comparing the Proposed algorithm on a unified metric, also led us to decide that the Proposed algorithm is better for the image dataset than FCBF and FAST. FCBF and FAST are good alternatives for text dataset and micro array data sets.

Our further work will be to extend this method for much higher dimensionality (more than ten thousand features). We will try to explore the different methodology which can improve text data.

## References

1. Kohavi, R., John, G.H.: Wrapper for feature subset selection. *Artif. Intell.* **97**, 273–324 (1997)
2. Das, S.: Filter, wrapper and a boosting-based hybrid for feature selection. In: *Proceedings of Eighteenth International Conference on Machine Learning*, pp. 74–81 (2001)
3. Dash, M., Liu, H., Motoda, H.: Consistency based feature selection. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) *PAKDD 2000*. LNCS, vol. 1805, pp. 98–109. Springer, Heidelberg (2000)
4. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC (2003)
5. Huang, J., Cai, Y., Xu, X.: A filter approach to feature selection based on mutual information. In: *5th IEEE international Conference* (2006)
6. Andreas, G.K., Janecek, A., Gansterer, W.N., Demel, M.A., Ecker, G.F.: On the relationship between feature selection and classification accuracy. In: *JMLR: Workshop and Conference Proceedings*, vol. 4, pp. 90–105 (2008)
7. Song, Q., Ni, J., Wang, G.: A fast clustering based feature subset selection algorithm for high dimensional data. *IEEE Trans. Knowl. Data Eng.* **25**(1), 1–14 (2013)
8. Hall M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: *Proceedings of 17th International Conference on Machine Learning*, pp. 359–366 (2000)
9. Hall, M.A.: Correlation based feature selection for machine learning. Thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand (1999)
10. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **17**(4), 491–502 (2005)



11. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: Proceedings of 10th National Conference Artificial Intelligence, pp. 129–134 (1992)
12. Kononenko, I.: Estimating attributes: analysis and extensions of RELIEF. In: Proceedings of European Conference Machine Learning, pp. 171–182 (1994)
13. Almuallim H., Dietterich T.G.: Algorithms for Identifying Relevant Features, Proc. Ninth Canadian Conf. Artificial Intelligence, pp. 38–45 (1992)
14. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991)
15. Wang, G., Song, Q., Sun, H., Zhang, X., Xu, B., Zhou, Y.: A feature subset selection algorithm automatic recommendation method. *J. Artif. Intell. Res.* **47**, 1–34 (2013)
16. Gray, R.M.: Entropy and Information Theory. Springer, New York (1991)
17. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 301–302 (2002)
18. Blake, C., Merz: UCI repository of machine learning databases. <http://www.ics.uci.edu>
19. Witten, I.H., Frank, E., Hall, M.A., Mining, D.: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2011)
20. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality, 1st edn. Wiley-Interscience, New York (2007)
21. Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H.: Advancing feature selection research. Technical report, Arizona State University (2011)
22. Laiho, P., Kokko, A., Vanharanta, S., Salovaara, R., Sammalkorpi, H., Jarvinen, H., Mecklin, J.P., Karttunen, T.J., Tuppurainen, K., Davalos, V., Schwartz, S., Arango, D., Makinen, M.J., Aaltonen, L.A.: Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis. *Oncogene* **26**(2), 312–320 (2007)
23. Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Gerald, W., Loda, M., Lander, E.S., Golub, T.R.: Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Nat. Acad. Sci. USA* **98**(26), 15149–15154 (2001)
24. Golub, T.R., Slonim, D.R., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–537 (1999)