ISSN:0975-9646

Khedkar S.A. et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3274 - 3280

A Survey on Clustered Feature Selection Algorithms for High Dimensional Data

Khedkar S.A., Bainwad A. M., Chitnis P. O.

CSE Department, SRTM University, Nanded
SGGSIE & T, Vishnupuri(MS) India

***Abstract*— In machine learning, feature selection is preprocessing step and can be effectively reduce high dimensional data, remove irrelevant data, increase learning accuracy, and improve result comprehensibility. High dimensionality of data take over efficiency and effectiveness points of view in feature selection algorithm. Efficiency stands required time to find a subset of features, and the effectiveness belongs to good quality of the subset of features. In feature selection technique high dimensional data contains many irrelevant and redundant features. Irrelevant features make available no useful information in any context, and redundant features provide no more information than the selected features. Good feature subsets contain features highly predictive of (correlated with) the class, yet not predictive of (uncorrelated with) each other. A subset of useful features to produce compatible results as the original set of features is identified from feature selection.**

Keywords-Feature subset selection; graph-theoretic clustering; filter method.

I. INTRODUCTION

In machine learning, feature selection, also known as variable subset selection, is the process of selecting a subset of relevant features for use in model construction. Feature selection techniques have benefits when constructing correlated models: improved model to interpret the hidden meaning, shorter (small) training times, and enhanced generalization by reducing over fitting. Feature selection is helpful as part of the data analysis process, as it identifies important features for prediction. Choosing a subset of good features according to target concepts, feature subset selection has been effective to reduce dimensionality, removing irrelevant data, increasing learning accuracy, and improving comprehensibility. Feature subset selection algorithms for machine learning applications can be divided into four main categories: Wrapper, Filter, Hybrid, and Embedded methods.

Wrapper methods use a predetermined learning model to score a feature subsets. A wrapper methods train a fresh model for new subset, they have high accuracy but are expensive to compute and also limited in generality of selected features. Filter methods are faster than wrapper methods but produces a features set which is independent from learning algorithms with better generality. Filter methods measures include the correlation coefficient, Mutual Information, distance and consistency measurements to sort a good subset. Filtering approach to feature selection involves a greater degree of search through the feature space but the accuracy of the algorithms is not guaranteed. Embedded algorithms integrates feature subset selection as a training process and they are fixed to learning methods, hence more efficient than Wrapper and Filter methods. Decision tree algorithms are best example of embedded methods. A combination of filter methods and wrapper methods form a hybrid methods which achieves best possible performance with a specific learning algorithm with similar time complexity like the filter methods. The wrapper methods tend to over fit on small training sets. The main benefits of filter methods are they are faster and they have ability to scale to large datasets. With respect to the filter feature selection methods, the application of cluster analysis clearly give practical demonstration and explanation to be more effective than traditional feature selection algorithms. The distributional clustering of words is agglomerative in nature and reduce the high dimensionality of text data since each word cluster can be treated as single feature but are expensive compute.

In cluster analysis, most of the applications use a graph-theoretic methods because they produce good results. The graph-theoretic clustering is simple since it compute a neighborhood graph of instances, then delete any edge in graph that is much short or long than its neighbors. The graph theoretic clustering results in forest and trees in forest represents a cluster. In this survey graph-theoretic clustering algorithms are used to features, particularly minimum spanning tree based clustering algorithms.

II. FEATURE SUBSET SELECTION

Feature as a group for suitability is evaluated by a subset selection a subset of features. Feature subset selection methods are divided into Wrappers, Filters, Embedded and Hybrid methods. Embedded techniques are embedded in and specific to a model. Wrappers use a search algorithm to search through the space of possible features and evaluate every subset by running a model on the subsets. Wrappers are computationally expensive and they have a risk of over fitting to the model. Filters are like Wrappers in the search approach, but instead of evaluating a filter against a model, a simpler filter is evaluated. Two popular filter metrics for classification problems correlation and mutual information, although both are not true metrics. There are, however, true metrics that are functions of the mutual information. Other available filter metrics are: Correlation-based feature selection, Consistency-based feature selection, and Class separability, which include Error probability, Inter class distance, probabilistic distance, and Entropy.

In feature selection technique high dimensional data contains many irrelevant and redundant features. Irrelevant features make available no useful information in any context, and redundant features provide no more information than the selected features. Irrelevant features do

not contribute to the predetermined accuracy and redundant features do not redound to getting a good predictor. Therefore feature selection is the process of identifying as many irrelevant and redundant features and removing them. The feature subset selection algorithms can eliminate irrelevant features but do not handle redundant features [20], [24], [27], [29], [32], and [37]. Some other algorithms can eliminate irrelevant features as well as handles redundant features [22], [31], [42].

A. Feature Selection Definitions

Let X be the original set of features, with cardinality $|X| = n$. The *continuous* feature selection problem refers to the assignment of weights w_i to each feature $x_i \in X$ in such a way that the order corresponding to its theoretical relevance is preserved. The feature selection problem can be seen as a search in a hypothesis space (set of possible solutions). In the case of the binary problem, the number of potential subsets to evaluate is 2^n.

Definition (Feature Selection) *Let* J(X') *be an evaluation measure to be optimized defined as* $J : X' \subset X \rightarrow R$. *The feature subset selection is viewed as:*

- Set $|X| = m<n$. Find $X' \subset X$, such that $J(X')$ is maximum.
- Set a value J_0, this is, the maximum J that is going to be tolerated. Find the $X' \subset X$ with smaller $|X'|$, such that $J(X) > J_0$.
- Find the compromise among minimizing $|X'|$ and maximizing $J(X')$.

Note that, optimal subset of feature is not unique always.

B. Characteristics of Feature Selection Algorithms

Figures The feature selection algorithms have following important characteristics:

1) Search Organization: A search algorithm is useful for driving the feature selection process using a specific strategy. In general, a search procedure examines only a part of the search space. When a specific state has to be visited, the algorithm uses the information of the previously visited states and eventually heuristic knowledge about non-visited ones [33].

2) Generation of Successor: Mechanism by which possible variants (successor candidates) of the current hypothesis are proposed. Up to five different operators can be considered to generate a successor for each state: Forward, Backward, Compound, Weighting, and Random [33].

3) *Evaluation measure:* Function by which successor candidates are evaluated, allowing to compare different hypothesis to guide the search process [33].

III. Feature Subset Selection Algorithms

Irrelevant features as well as redundant features largely affect the learning machines accuracy. Thus, to identify and remove as much of the irrelevant and redundant information as possible, feature subset selection should be useful. "Good feature subsets contain features highly correlated with the class i.e. predictive of class, yet uncorrelated with each other i.e. not predictive of each class." A feature subset selection algorithms can efficiently and effectively handle both irrelevant and redundant features, and obtains a good feature subsets. The two connected components of *irrelevant feature removal* and *redundant feature elimination* are composed to design feature selection framework (shown in Figure 1). The first i.e. *irrelevant feature removal* obtains features which are relevant to the target concept by eliminating irrelevant ones, and the second eliminates redundant features from relevant ones via selecting representatives from various feature clusters, and hence results the final subset.

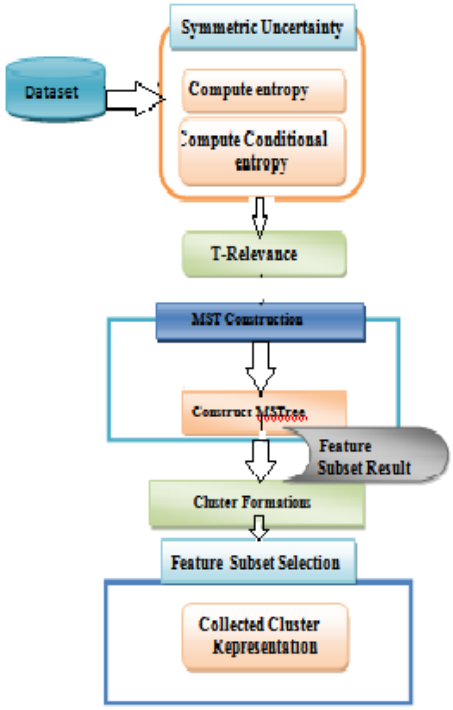

Figure 1: Framework of feature subset selection algorithms

A definition of relevant features is presented as 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_i', fi a value-assignment of feature Fi, and c a value-assignment of the target concept C. The relevant features can be formally defined as.

Definition 1: (Relevant feature) Fi is relevant to the target concept C if and only if there exists some si, fi and c, such that, for probability $p(S_i' = s'i, Fi = fi) > 0$,
$p(C = c \mid S_i' = s'i, Fi = fi) \neq p(C = c \mid Si = si)$.
Otherwise, feature Fi is an *irrelevant feature*.

However, the definition gives that feature Fi is relevant when using $Si \cup \{Fi\}$ to describe the target concept. The reason behind is that either Fi is interactive with Si or Fi is redundant with $Si - S_i$. , we say Fi is indirectly relevant to the target concept.

Most of the information is already present in other features is also contained in redundant features. As a result, the redundant features do not contribute to getting better

interpreting ability to the target concept. This is defined by Liu and Yu based on Markov blanket.

Definition 2: (Markov blanket) Given a feature $Fi \in F$, let $Mi \subset F$ ($Fi \notin Mi$), Mi is said to be a Markov blanket for Fi if and only if

$p(F - Mi - \{Fi\}, C \mid Fi, Mi) = p(F - Mi - \{Fi\}, C \mid Mi)$.

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.

Redundant features are not necessary for a best subset because their values are completely uncorrelated with target concepts while relevant features have strong correlation with target concept so relevant features are invariably necessary for a best subset. Thus, conception of feature relevance and feature redundancy are normally in terms of feature correlation and feature-target concept correlation. How much distribution of the feature values and target classes differ from statistical independence is measured through mutual information. This is a straight forward estimation of correlation between feature values and target classes or between feature values. To derive *symmetric uncertainty* (SU) from the mutual information by normalizing it to the entropies feature values and target classes or between feature values. Thus, we select symmetric uncertainty as the measure of correlation between either two features or a feature and the target concept.

The *symmetric uncertainty* can be defined as

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

Where,

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

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

2) *Gain* ($X|Y$) is the amount by which the entropy of Y reduces. It returns the additional information about Y provided by X and is called the information gain which is defined by

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

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

$$H(X|Y) = -\Sigma_{y \in Y} p(y) \Sigma_{x \in X} p(x|y) \log 2\ p(x|y).$$

Information gain is a symmetrical measure. Symmetric uncertainty deals a couple 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 $SU(X, Y)$ indicates that knowledge of the value of either one completely guess the value of the other and the value 0 reveals that X and Y are independent. If the values are discretized properly in advance, the entropy based measure handles nominal or discrete variables, and they can deal with continuous features also. Given $SU(X, Y)$ the symmetric uncertainty of variables X and Y, the correlation *F-Correlation* between a pair of features, the feature redundancies *F-Redundancy*, the relevance *T-Relevance* between a feature and the target concept C, 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 $SU(Fi, C)$.

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

Definition 5: (F-Correlation) The correlation between any pair of features Fi and Fj ($Fi, Fj \in F \wedge i \neq j$) is called the *F-Correlation* of Fi and Fj, and denoted by $SU(Fi, Fj)$.

Definition 6: (F-Redundancy) Let $S = \{F_1, F_2, \ldots, F_i, \ldots, F_k \mid F\}$ be a cluster of features if $\exists Fj \in S$, $SU(Fj, C) \geq SU(Fi, C) \wedge SU(Fi, Fj) > SU(Fi, C)$ is always corrected for each $Fi \in S$ ($i \neq j$), then Fi are redundant features with respect to the given Fj (i.e. each Fi is a *F-Redundancy*).

Definition 7: (R-Feature) A feature $F_i \in S = \{F_1, F_2, \ldots, F_k\}$ $(i < \mid F \mid)$ is a representative feature of the cluster S (i.e. Fi is a *R-Feature*) if and only if,

$$F_i = argmax_{F_j \in S} SU(F_j, C)$$

This means the feature, which has the strongest *T-Relevance*, can act as an *R-Feature* for all the features in the cluster.

According to the above definitions, feature subset selection can be the process that identifies and retains the strong *T-Relevance* features and selects *R-Feature*s from feature clusters. The behind heuristics are that

1) irrelevant features have no/weak correlation with target concept;

2) redundant features are assembled in a cluster and a representative features can be taken out of the cluster.

Definition 8: (Predominant feature) A relevant feature is predominant iff it does not have any inexact Markov blanket in the current set.

A. Relief-F

The ReliefF (Relief-F) algorithm [36] (Kononenko, 1994) is not limited to two class problems, is more robust and can deal with incomplete and noisy data. Similarly to Relief, ReliefF randomly selects an instance Ri (line 3), other than searches for k of its nearest neighbors from the similar class called nearest hits H_j (line 4), and also k nearest neighbors from each of the dissimilar classes, called nearest misses $M_j(C)$ (lines 5 and 6). It upgrades the quality estimation $W[A]$ for all attributes A depending on their values for R_i, hits H_j and misses $M_j(C)$ (lines 7, 8 and 9). The upgrade formula is same as to that of Relief (lines 5 and 6 on Figure 1), excluding that we average the contribution of all the hits and all the misses. The contribution for each class of the misses is weighted with the initial probability of that class $P(C)$ (estimated from the training set). Since the contributions of hits and misses in each step to be in [0, 1] and also symmetric to ensure that misses' probability weights sum to 1. As the class of hits is missing in the sum we have to divide each probability weight with factor $1 - P$ (*class* (R_i)) (which represents the sum of probabilities for the misses' classes). The process is repeated for m times.

Input: *for each training instance a vector of attribute values and the class value*
Output: *the vector W of estimations of the qualities of attributes*

1 set all weights $W[A] = 0.0$;
2 for i = 1 to m do begin
3 randomly select an instance R_i;
4 find k nearest hits H_j;
5 for each class $C \neq class(R_i)$ do
6 from class C find k nearest misses $M_j(C)$;
7 for $A = 1$ to a do
8

$$W[A] = W[A] - \sum_{j=1}^{k} \frac{diff(A, R_i, H_j)}{m \cdot k} -$$

9

$$\sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^{k} diff(A, R_i, M_j(C)) \right] / (m \cdot k)$$

10 end;

Selection of k hits and misses is the basic difference to Relief and ensures greater robustness of the algorithm concerning noise. User-defined parameter k controls the locality of the estimates. For most purposes it can be safely set to 10 (see (Kononenko, 1994).To deal with incomplete data we change the diff function. Missing values of attributes are treated probabilistically. We calculate the probability that two given instances have different values for given attribute conditioned over class value:

- if one instance (e.g., $I1$) has unknown value:
  $diff(A, I_1, I_2) = 1 - P(value(A, I_2) | class(I_1))$
- if both instances have unknown value:

$$diff(A, I_1, I_2) = 1 - \sum_{V}^{nvalues(A)} [P(V | class(I_1)) \times P(V | class(I_2))]$$

Conditional probabilities are inexactd with relative frequencies from the training set.

B. FCBF

The relevance and redundancy analysis can be realized by an algorithm, called FCBF (Fast Correlation-Based Filter) [42], [45]. FCBF algorithm involves two connected steps: (1) a relevant features subset selection, and (2) selection of main i.e. primary features from relevant ones. For a data set S with N features and class C, the algorithm finds a set of predominant features S_{best} .

input: $S(F_1, F_2, ..., F_{N,C})$ // a training data set
δ □// a predefined threshold
output: S_{best} // a selected subset

1 begin
2 for $i = 1$ to N do begin
3 calculate $SU_{i,c}$ for Fi;
4 if ($SU_{i,c} >$ □)
5 append Fi to S_{list} ;
6 end;
7 order S_{list} in descending $SU_{i,c}$ value;
8 $F_j = getFirstElement(S_{list})$;
9 do begin
10 $F_i = getNextElement(S_{list}, F_j)$;
11 if (F_i <> NULL)
12 do begin
13 if ($SU_{i,j} \geq SU_{i,c}$)
14 remove Fi from S_{list} ;
15 $Fi = getNextElement(S_{list}, F_i)$;
16 end until (Fi == NULL);
17 $F_j = getNextElement(S_{list}, Fj)$;
18 end until (F_j == NULL);
19 $S_{best} = S_{list}$;
20 end;

In the prior step (lines 2 - line 7), it calculates the SU value for each features selects relevant features into S'_{list} based on a predefined threshold δ, and orders them in a descending order according to their SU values. In the posterior step (lines 8 –line 18), it next processes the ordered list S_{list} to select primary features. A feature F_j that has already been determined to be a predominant (primary) feature can always be used to filter out other features for which F_j forms an inexact Markov blanket. Since the feature with the highest C-correlation does not have any inexact Markov blanket, it must be one of the predominant features. So the iteration starts from the very first element in S_{list} (line 8) and continues. For all the remaining features (from the one right next to F_j to the last one in S_{list}), if F_j happens to form an inexact Markov blanket for Fi (line 13), Fi will be removed from S_{list} . After first round of filtering features based on F_j, the algorithm will take the remaining feature right next to F_j as the new reference (line 17) to replicate the filtering process. The algorithm stops until no more predominant (primary) features can be selected. Figure 2 illustrates how predominant (primary) features are selected with the rest features removed as redundant ones.

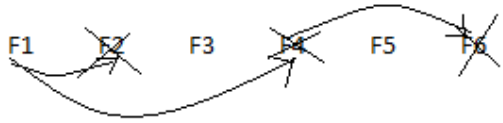

Figure 2: Selection of predominant features

In Figure 2, six features are selected as relevant ones and ranked according to their C-correlation values, with $F1$ being the most relevant one. In the first round, $F1$ is selected as a predominant feature, and $F2$ and $F4$ are removed based on $F1$. In the second round, $F3$ is selected, and $F6$ is removed based on $F3$. In the last round, $F5$ is selected.

C. CFS

The CFS (Correlation-based Feature Selector) [22] is a simple filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and

uncorrelated with each other. Irrelevant features should be ignored because they will have low correlation with the class. Redundant features should be screened out as they will be highly correlated with one or more of the remaining features. The acceptance of a feature will depend on the extent to which it predicts classes in areas of the instance space not already predicted by other features. CFS's feature subset evaluation function is for ease of reference:

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

Where, M_s is the heuristic "merit" of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation (f ∈ S), and $\overline{r_{ff}}$ is the average feature-feature inter-correlation.

The numerator of above equation can be thought of as providing an indication of how predictive of the class a set of features are; the denominator of how much redundancy there is among the features.

CFS calculates feature-class and feature-feature correlations using one of the measures and then searches the feature subset space. The subset with the highest merit (as measured) found during the search is used to reduce the dimensionality of both the original training data and the testing data. Both reduced datasets may then be passed to a machine learning scheme for training and testing. It is important to note that the general concept of correlation-based feature selection does not depend on any one module (Such as discretization). A more sophisticated method of measuring correlation may make discretization unnecessary. Similarly, any conceivable search strategy may be used with CFS.

D. FAST

Based on the Minimum Spanning Tree method a feature subset selection FAST algorithm [1]. The FAST algorithm works as, initially features are separated into clusters by using graph-theoretic clustering methods and then the most representative features which is strongly related to target classes is selected from each cluster to form the final subset of features. Features in various clusters are comparatively independent, the clustered strategy of feature subset selection algorithm has a high chances of producing a subset of independent and useful features. This algorithm requires the construction of the minimum spanning tree; from a weighted complete graph; the partitioning of the minimum spanning tree into a forest with each tree representing a cluster; and the selection of representative features from the clusters.

For a data set D with m features $F = \{F1, F2, ..., Fm\}$ and class C, we compute the *T-Relevance* $SU(Fi, C)$ value for each feature Fi $(1 \le i \le m)$ in the first step. The features whose $SU(Fi, C)$ values are greater than a predefined threshold θ comprise the target-relevant feature subset $F' = \{F'1, F'2, ..., F'k\}$ $(k \le m)$.

In the second step, we first calculate the *F-Correlation* $SU(F'i, F'j)$ value for each pair of features $F'i$ and $F'j(F'i, 'j \in F' \wedge i \neq j)$. Then, viewing features $F'i$ and $F'j$ as vertices and $SU(F'i, F'j)$ $(i \neq j)$ as the weight of the edge between vertices $F'i$ and $F'j$, a weighted complete graph $G = (V,E)$ is constructed where $V = \{F'i \mid F'i \in F' \wedge i \in [1, k]\}$ and $E = \{(F'i, F'j) \mid (F'i, F'j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$. As symmetric uncertainty is symmetric further the *F-Correlation* $SU(F'i, F'j)$ is symmetric as well, thus G is an undirected graph.

The complete graph G reflects the correlations among all the target-relevant features. Unfortunately, graph G has k vertices and $k(k-1)/2$ edges. For high dimensional data, it is heavily dense and the edges with different weights are strongly interweaved. Moreover, the decomposition of complete graph is NP-hard [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* $SU(F'i, F'j)$. After building the MST, in the third step, initially remove the edges $E = \{(F'i, F'j) \mid (F'i, F'j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$, whose weights are smaller than both of the *T-Relevance* $SU(F'i, C)$ and $SU(F'j, C)$, from the MST. Each deletion results in two disconnected trees $T1$ and $T2$. Assuming the set of vertices in any one of the final trees to be $V(T)$, we have the property that for each pair of vertices $(F'i, F'j \in V(T))$, $SU(F'i, F'j) \ge SU(F'i, C) \vee SU(F'i, F'j) \ge SU(F'j, C)$ always holds.

Algorithm: FAST

inputs: $D(F_1, F_2, ..., F_m, C)$ - *the given data set* θ - *the T-Relevance threshold.*

output: S - *selected feature subset.*

```
// Part 1: Irrelevant Feature Removal
1       for i = 1 to m do
2       T-Relevance = SU (Fi, C)
3       if T-Relevance > θ then
4               S = S ∪ {Fi};
5       end if
6       end for
//Part 2: Minimum Spanning Tree Construction
7 G = NULL; //G is a complete graph
8 for each pair of features {F'i, F'j} ⊂ S do
9       F-Correlation = SU (F'i , F'j )
10      Add Fi and / or F'j to G with F-Correlation as
        the weight of the corresponding edge,
11 end for
12  minSpanTree = Prim (G); //Using Prim Algorithm to
        generate the minimum spanning tree
        //Part 3: Tree Partition and Representative Feature
        Selection
13 Forest = minSpanTree
14 for each edge Eij ∈ Forest do
15      if SU(F'i, F'j ) < SU(F'i, C) ∧ SU(F'i, F'j ) <
        SU(F'j , C) then
13 Forest = Forest − Eij
14      end if
15 end for
16 S = ϕ
17 for each tree Ti ∈ Forest do
18      FjR = argmax F'k ∈ Ti SU(F'k,C)
19      S = S ∪ {F'jR};
20 end for
21 return S.
```

IV. EXPERIMENTAL STUDY

A. Experimental Setup

To evaluate the performance of feature subset selection algorithms and compare with other feature selection algorithms the experimental set up as follows.

The algorithms are compared with different feature selection algorithms, like (i) FCBF [42], [45], (ii) Relief-F [36], (iii) CFS [22], (iv) FAST [1], respectively. FCBF and Relief-F evaluate features separately. For FCBF, in the experiments, the relevance threshold to be the SU value of the $\lfloor m/\log m \rfloor th$ ranked feature for every data set (m is the number of features in a given data set). Relief-F searches for nearest neighbors of instances of different classes and weights features according to how well they differentiate instances of different classes. CFS uses best-first search based on the evaluation of a subset that contains features highly predictive of the target concept, yet not predictive of each other. For FAST algorithm, set θ to be the SU value of the $\lfloor \sqrt{m} * \lg m \rfloor th$ ranked feature for each data set.

Different types of classification algorithms are used to classify data sets prior and after feature selection. Such as (i) the tree-based C4.5, (ii) the probability-based Naive Bayes (NB), (iii) the rule-based RIPPER, (iv) the instance-based lazy learning algorithm IB1, respectively. Naive Bayes employs 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 etc. 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) 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.

When evaluating the performance of the feature subset selection algorithms, different metrics, such as (i) the proportion of selected features (ii) the time to obtain the feature subset, (iii) the classification accuracy, 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.

B. CFS

In order to make the best use of the data and obtain stable results, a (M = 5) × (N = 10)-cross-validation strategy is used. That is, for each data set, each feature subset selection algorithm and each classification algorithm, the 10-fold cross-validation is repeated M = 5 times, with each time the order of the instances of the data set being randomized. This is because many of the algorithms exhibit order effects, in that certain orderings dramatically improve or degrade performance. Randomizing the order of the inputs can help diminish the order effects. In 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 *Experimental Process* shows the details.

Procedure: *Experimental Process*
1 M = 5, N = 10
2 DATA = *{*$D_1, D_2, \ldots, D_{35}$*}*
3 Learners = *{*NB, C4.5, IB1, RIPPER*}*
4 FeatureSelectors = *{*FAST, FCBF, ReliefF, CFS*}*
5 for each *data* ∈ DATA do
6 for each *times* ∈ [1, M] do
7 randomize instance-order for *data*
8 generate N bins from the randomized *data*
9 for each *fold* ∈ [1, N] do
10 *TestData* = bin[*fold*]
11 *TrainingData* = *data* - *TestData*
12 for each *selector* ∈ FeatureSelectors do
13 (*Subset*, *Time*) = *selector*(*TrainingData*)
14 *TrainingData′* = select *Subset* from *TrainingData*
15 *TestData′* = select *Subset* from *TestData*
16 for each *learner* ∈ Learners do
17 *classifier* = *learner*(*TrainingData′*)
18 *Accuracy* = apply *classifier* to *TestData′*
19 end for
20 end for
21 end for
22 end for
23 end for

REFERENCES

[1] Qinbao Song, Jingjie Ni and Guangtao Wang, A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data, IEEE Transactions on Knowledge and Data Engineering vol:25 no:1 year 2013.

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

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

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

[16] Dhillon I.S., Mallela S. and Kumar R., A divisive information theoretic feature clustering algorithm for text classification, J. Mach. Learn. Res., 3, pp 1265-1287, 2003.

[17] Dougherty, E. R., Small sample issues for microarray-based classification. Comparative and Functional Genomics, 2(1), pp 28-34, 2001.

[18] Fayyad U. and Irani K., Multi-interval discretization of continuous-valued attributes for classification learning, In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp 1022-1027, 1993.

[19] Fleuret F., Fast binary feature selection with conditional mutual Information, Journal of Machine Learning Research, 5, pp 1531-1555, 2004.

[20] Forman G., An extensive empirical study of feature selection metrics for text classification, Journal of Machine Learning Research, 3, pp 1289-1305, 2003.

[21] Guyon I. and Elisseeff A., An introduction to variable and feature selection, Journal of Machine Learning Research, 3, pp 1157-1182, 2003.

[22] Hall M.A., Correlation-Based Feature Subset Selection for Machine Learning, Ph.D. dissertation Waikato, New Zealand: Univ. Waikato, 1999.

[23] Hall M.A. and Smith L.A., Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper, In Proceedings of the Twelfth international Florida Artificial intelligence Research Society Conference, pp 235-239, 1999.

[24] 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

[25] Jaromczyk J.W. and Toussaint G.T., Relative Neighborhood Graphs and Their Relatives, In Proceedings of the IEEE, 80, pp 1502-1517, 1992.

[26] John G.H., Kohavi R. and Pfleger K., Irrelevant Features and the Subset Selection Problem, In the Proceedings of the Eleventh International Conference on Machine Learning, pp 121-129, 1994.

[27] Kira K. and Rendell L.A., The feature selection problem: Traditional methods and a new algorithm, In Proceedings of Nineth National Conference on Artificial Intelligence, pp 129-134, 1992.

[28] Kohavi R. and John G.H., Wrappers for feature subset selection, Artif. Intell., 97(1-2), pp 273-324, 1997.

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

[30] Last M., Kandel A. and Maimon O., Information-theoretic algorithm for feature selection, Pattern Recognition Letters, 22(6-7), pp 799-811, 2001.

[31] Liu H. and Setiono R., A Probabilistic Approach to Feature Selection: A Filter Solution, in Proceedings of the 13th International Conference on Machine Learning, pp 319-327, 1996.

[32] Modrzejewski M., Feature selection using rough sets theory, In Proceedings of the European Conference on Machine Learning, pp 213-226, 1993.

[33] Molina L.C., Belanche L. and Nebot A., Feature selection algorithms: A survey and experimental evaluation, in Proc. IEEE Int. Conf. Data Mining, pp 306-313, 2002.

[34] Park H. and Kwon H., Extended Relief Algorithms in Instance-Based Feature Filtering, In Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), pp 123-128, 2007.

[35] Raman B. and Ioerger T.R., Instance-Based Filter for Feature Selection, Journal of Machine Learning Research, 1, pp 1-23, 2002.

[36] Robnik-Sikonja M. and Kononenko I., Theoretical and empirical analysis of Relief and ReliefF, Machine Learning, 53, pp 23-69, 2003.

[37] Scherf M. and Brauer W., Feature Selection By Means of a Feature Weighting Approach, Technical Report FKI-221-97, Institut fur Informatik, Technische Universitat Munchen, 1997.

[38] Souza J., Feature selection with a general hybrid algorithm, Ph.D, University of Ottawa, Ottawa, Ontario, Canada, 2004.

[39] Van Dijk G. and Van Hulle M.M., Speeding Up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis, International Conference on Artificial Neural Networks, 2006.

[40] Xing E., Jordan M. and Karp R., Feature selection for high-dimensional genomic microarray data, In Proceedings of the Eighteenth International Conference on Machine Learning, pp 601-608, 2001.

[41] Yu J., Abidi S.S.R. and Artes P.H., A hybrid feature selection strategy for image defining features: towards interpretation of optic nerve images, In Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 8, pp 5127-5132, 2005.

[42] Yu L. and Liu H., Feature selection for high-dimensional data: a fast correlation-based filter solution, in Proceedings of 20th International Conference on Machine Leaning, 20(2), pp 856-863, 2003.

[43] Yu L. and Liu H., Efficiently handling feature redundancy in highdimensional data, in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03). ACM, New York, NY, USA, pp 685-690, 2003.

[44] Yu L. and Liu H., Redundancy based feature selection for microarray data, In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 737-742, 2004.

[45] Yu L. and Liu H., Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research, 10(5), pp 1205-1224, 2004.