Efficiently Handling Feature Redundancy in High-Dimensional Data

Lei Yu
Department of Computer Science & Engineering
Arizona State University
Tempe, AZ 85287-5406
leiyu@asu.edu

Huan Liu
Department of Computer Science & Engineering
Arizona State University
Tempe, AZ 85287-5406
hliu@asu.edu

ABSTRACT

High-dimensional data poses a severe challenge for data mining. Feature selection is a frequently used technique in preprocessing high-dimensional data for successful data mining. Traditionally, feature selection is focused on removing irrelevant features. However, for high-dimensional data, removing redundant features is equally critical. In this paper, we provide a study of feature redundancy in high-dimensional data and propose a novel correlation-based approach to feature selection within the filter model. The extensive empirical study using real-world data shows that the proposed approach is efficient and effective in removing redundant and irrelevant features.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; I.2.6 [**Artificial Intelligence**]: Learning; I.5.2 [**Pattern Recognition**]: Design Methodology —*feature evaluation and selection*

Keywords

Feature selection, redundancy, high-dimensional data

1. INTRODUCTION

Data mining is a process that consists of major steps such as preprocessing, mining, and post-processing. Feature selection is frequently used as a preprocessing step to data mining. It is a process of choosing a subset of original features by removing irrelevant and/or redundant ones. Feature selection has been effective in removing irrelevant and redundant features, increasing efficiency in mining tasks, improving mining performance like predictive accuracy, and enhancing result comprehensibility [3, 5, 9]. Feature selection algorithms can broadly fall into the filter model or the wrapper model [4, 9]. The filter model relies on general characteristics of the training data to select some features without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm in feature selection and uses its performance to evaluate and determine which features are selected. It tends to give superior performance as it finds features better suited to the predetermined mining algorithm, but it also tends to be more computationally expensive than the filter model [3]. When the number of features becomes very large, the filter model is usually chosen due to its computational efficiency.

In recent years, data has become increasingly larger in both rows (i.e., number of instances) and columns (i.e., number of features) in many applications such as text categorization [22], genome projects [21], and customer relationship management [17]. This enormity may cause serious problems to many data mining algorithms with respect to scalability and mining performance. For example, high dimensional data (i.e., data sets with hundreds or thousands of features) can contain high degree of irrelevant and redundant information which may greatly degrade the performance of mining algorithms. Therefore, feature selection becomes very necessary for data mining tasks when facing high dimensional data nowadays. However, this trend of increase on both size and dimensionality also poses severe challenges to feature selection algorithms in terms of efficiency and effectiveness. Some of the recent research efforts in feature selection have been focused on these challenges from handling a huge number of instances [13, 15] to dealing with high dimensional data [4, 16, 21]. The efforts in the latter introduce algorithms in a hybrid model which combines the advantages of both filter and wrapper algorithms to achieve best possible performance with a particular mining algorithm on high dimensional data with similar time complexity of filter algorithms. However, these new algorithms do not reduce the time complexity of previous filter algorithms. In this work, we aim to develop an efficient filter solution for feature selection in high-dimensional data which can effectively remove both irrelevant and redundant features and is less costly in computation than the currently available methods.

In section 2, we review previous work within the filter model and point out their problems in the context of high dimensionality. In section 3, we describe a correlation-based measure used in our approach, introduce our definition of feature redundancy based on a novel concept, **predominant correlation**, and propose a new algorithm that can effectively select good features based on correlation analysis with less than quadratic time complexity. In section 4, we

SIGKDD '03, August 24-27, 2003, Washington, DC, USA

evaluate the efficiency and effectiveness of this algorithm via extensive experiments on real-world data comparing with other representative feature selection algorithms. In section 5, we conclude our work with some possible extensions.

2. PREVIOUS WORK AND PROBLEMS

Within the filter model, different feature selection algorithms can be further categorized into two groups, namely, feature weighting algorithms and subset search algorithms, based on whether they evaluate the goodness of features individually or through feature subsets. Below, we discuss the advantages and shortcomings of algorithms in each group and show the need of a new algorithm.

Feature weighting algorithms assign weights to features individually and rank them based on their relevance to the target concept. There are a number of different definitions on feature relevance in machine learning literature [3, 9]. A feature is good and thus will be selected if its weight of relevance is greater than a threshold value. A well known algorithm that relies on relevance evaluation is Relief [8]. It estimates the relevance of features according to how well their values distinguish between the instances of the same and different classes that are near each other. It randomly samples a number (m) of instances from the training set and updates the relevance estimation of each feature based on the difference between the selected instance and the two nearest instances of the same and opposite classes. Time complexity of Relief for a data set with M instances and N features is $O(mMN)$. With m being a constant, the time complexity becomes $O(MN)$, which makes it very scalable to data sets with both a huge number of instances and a very high dimensionality. However, Relief does not help with removing redundant features. As long as features are deemed relevant to the class concept, they will all be selected even though many of them are highly correlated to each other [8]. Many other algorithms in this group have similar problems with handling redundancy as Relief does. They can only identify relevant features to the target concept according to different relevance criteria, but cannot effectively discover redundancy among features. However, empirical evidence from feature selection literature shows that, along with irrelevant features, redundant features also affect the speed and accuracy of mining algorithms and thus should be eliminated as well [7, 9]. Therefore, in the context of feature selection for high dimensional data where there may exist many redundant features, pure relevance-based feature weighting algorithms do not meet the need of feature selection very well, although they have linear time complexity $O(N)$ in terms of dimensionality N.

Subset search algorithms search through candidate feature subsets guided by a certain evaluation measure [12] which captures the goodness of each subset. An optimal (or near optimal) subset is selected when the search stops. Among existing evaluation measures, correlation measure has been shown effective in removing both irrelevant and redundant features [7]. Correlation measure evaluates the goodness of feature subsets based on the hypothesis that good feature subsets contain features highly correlated to (predictive of) the class, yet uncorrelated to (not predictive of) each other. It requires certain heuristic that takes into account the usefulness of individual features for predicting the class label along with the level of inter-correlation between them. In [7], correlation measure is applied in an algorithm called CFS that exploits heuristic search (best first search) to search for candidate feature subsets. As many other algorithms that exploit heuristic search, CFS has time complexity $O(N^2)$ in terms of dimensionality N. It is known that algorithms with random search can have linear time complexity in terms of the number of subsets evaluated [14], but experiments show that in order to obtain near optimal results the required number of subsets for evaluation is mostly at least quadratic to the number of features N [6]. Therefore, with at least quadratic complexity in terms of dimensionality, subset search algorithms do not have strong scalability to deal with high dimensional data.

To overcome the problems of algorithms in both groups and meet the demand for feature selection for high dimensional data, we develop a novel approach which can effectively identify both irrelevant and redundant features with less time complexity than subset search algorithms.

3. A CORRELATION-BASED APPROACH

3.1 Correlation-based Measures

Before we delve into our new approach, we now discuss how to evaluate the goodness of features for classification. In general, a feature is *good* if it is *relevant* to the class concept but is not *redundant* to any of the other relevant features. If we adopt the correlation between two variables as a goodness measure, the above definition becomes that a feature is good if it is highly correlated to the class but not highly correlated to any of the other features. In other words, if the correlation between a feature and the class is high enough to make it relevant to (or predictive of) the class and the correlation between it and any other relevant features does not reach a level so that it can be predicted by any of the other relevant features, it will be regarded as a good feature for the classification task.

Classical linear correlation analysis is a well known approach to measure the correlation between two random variables. It helps remove features with near zero linear correlation to the class and reduce redundancy among selected features. However, it may not be able to capture correlations that are not linear in nature in the real world. Another limitation is that the calculation requires all features contain numerical values.

Therefore, in our approach we adopt another form of correlation measure based on the information-theoretical concept of *entropy*, a measure of the uncertainty of a random variable. The entropy of a variable X is defined as

$$H(X) = -\sum_i P(x_i) \log_2(P(x_i)), \tag{1}$$

and the entropy of X after observing values of another variable Y is defined as

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_j) \log_2(P(x_i|y_j)), \tag{2}$$

where $P(x_i)$ is the prior probabilities for all values of variable X, and $P(x_i|y_i)$ is the posterior probabilities of X given the values of Y. The amount by which the entropy of X decreases reflects additional information about X provided by Y and is called *information gain* [19], given by

$$IG(X|Y) = H(X) - H(X|Y). \tag{3}$$

According to this measure, a feature Y is regarded more correlated to feature X than to feature Z, if $IG(X|Y) > IG(Z|Y)$.

It is known that information gain is symmetrical for two variables [18], which is desirable for measuring correlations between features. However, information gain is biased in favor of features with more values. Furthermore, the values have to be normalized to ensure they are comparable and have the same affect. Therefore, we choose *symmetrical uncertainty* [18], defined as follows.

$$SU(X,Y) = 2 \ \frac{IG(X|Y)}{H(X) + H(Y)} \tag{4}$$

It compensates for information gain's bias toward features with more values and normalizes its values to the range $[0, 1]$ with the value 1 indicating that knowledge of the value of either one completely predicts the value of the other and the value 0 indicating that X and Y are independent. In addition, it still treats a pair of features symmetrically. Entropy-based measures require nominal features, but they can be applied to measure correlations between continuous features as well, if the values are discretized properly in advance [11]. Therefore, we use symmetrical uncertainty in this work.

3.2 Definitions and Methodology

Using symmetrical uncertainty (SU) as the goodness measure, we are now ready to develop a procedure to select good features for classification based on correlation analysis of features (including the class). This involves two aspects: (1) how to decide whether a feature is relevant to the class or not, and (2) how to decide whether such a relevant feature is redundant or not when considering it with other relevant features.

The answer to the first question can be using a threshold SU value decided by the user, as the method used by many other feature weighting algorithms (e.g., Relief). For a data set S containing N features and a class C, let $SU_{i,c}$ denote the SU value that measures the correlation between a feature F_i and the class C (named C-correlation), We give our definition of relevance as below.

Definition 1. (Relevance) A feature F_i is said to be relevant to the class concept C *iff* $SU_{i,c} \geq \delta$ where δ is the threshold relevance value.

Through this paper, we use S' to denote the set of relevant features, i.e., $S' = \{F_i | SU_{i,c} \geq \delta, F_i \in \text{S}\}$.

The answer to the second question is more complicated because (1) it may involve analysis of pairwise correlations between all features (named F-correlation), which results in time complexity $O(N^2)$ associated with dimensionality N; and (2) it is not clear which is the best way to remove redundant features based on F-correlation information. To solve these two problems, we now propose our method.

Since F-correlations are also captured by SU values, in order to determine feature redundancy, we need to find a reasonable way to decide the threshold level for F-correlations as well. In other words, we need to decide whether the level of correlation between two relevant features in S' is high enough to cause redundancy so that one of them may be removed. For a relevant feature F_i, the value of $SU_{i,c}$ quantifies the extent to which F_i is correlated to (or predictive of) the class C. If we examine the value of $SU_{j,i}$ between F_i and all the rest relevant features (i.e., $\forall F_j \in S'$, $j \neq i$), we can also obtain quantified estimations about the extent to which F_i is correlated to (or predicted by) the rest relevant features. Therefore, it is possible to identify highly correlated features to the concept F_i in the same straightforward manner as we decide relevant features to the class concept, using some arbitrary threshold SU value. We can do this for all relevant features. However, this method only sounds reasonable when we try to determine highly correlated features to one concept while not considering another concept. In the context of a set of relevant features S' already identified for the class concept, when we try to determine highly correlated features for a given feature F_i in S', it is more reasonable to use the C-correlation level $SU_{i,c}$ between F_i and the class concept as a reference. The reason lies on the common phenomenon - a feature that is correlated to one concept (e.g., the class) at a certain level may also be correlated to some other concepts (features) at the same or even higher level. If these features are more correlated to the class concepts, it is natural to think that F_i is redundant to the existence of these features. Therefore, even the correlation between F_i and the target concept is larger than some threshold δ and thereof making this feature relevant to the target concept, this correlation may not be predominant or significant in determining the target concept. Before we give precise definitions about *predominant correlation*, *redundant feature*, and *predominant feature*, we now introduce some additional symbols to facilitate our definitions.

Given a data set S with a set of relevant features S' and a relevant feature F_i, $S_i{}^{+} = \{F_j | F_j \in S', SU_{j,c} > SU_{i,c}\}$, $S_i{}^{-} = \{F_j | F_j \in S', j \neq i, SU_{j,c} <= SU_{i,c}\}$, $S_i{}^{(RS)} = \{F_j | SU_{j,i} \geq SU_{i,c}, F_j \in S_i{}^{+}\ \}$, and $S_i{}^{(RO)} = \{F_j | SU_{i,j} \geq SU_{j,c}, F_j \in S_i{}^{-}\ \}$.

Definition 2. (Predominant correlation). The correlation between a relevant feature F_i and the class C is predominant *iff* $S_i{}^{(RS)} = \varnothing$.

Definition 3. (Redundancy) A relevant feature F_i is regarded as redundant to the existence of any feature F_j (called redundant subject) in $S_i{}^{(RS)}$, and at the same time, any feature F_k (called redundant object) in $S_i{}^{(RO)}$ is regarded as redundant to the existence of F_i.

Our definition of redundancy has two distinct differences to the normal understanding about feature redundancy. First, a feature is normally said to be redundant if one or more of the other features are highly correlated to it. In our definition, the redundancy of a feature is decided not only by its F-correlations to other features but also by its C-correlation and the C-correlations of other features that are correlated to it. Second, redundancy is normally regarded as a symmetrical relationship between features and which one is removed is decided at random. In our definition, we believe that in recognition of redundancy between a pair of features, the feature that is more relevant to the class concept should have priority to be kept to the other feature which is redundant to it. Our assumption is that if two features are found to be redundant to each other and one of them needs to be removed, removing the one that is less relevant to the class concept keeps more information to predicate the class while reducing redundancy in the data. Whether or not a relevant feature F_i is removed after it is decided as a redundant feature to the features in $S_i{}^{(RS)}$ is dependent on the existence of each feature in $S_i{}^{(RS)}$. For instance, for three features F_i, F_j, F_k, if ${S_i}^{(RS)} = \{F_j\}$, ${S_i}^{(RO)} = \{F_k\}$, and if F_j is known to be removed, F_i should be kept, and thus F_k should be removed; otherwise, F_i should be removed, and we need to decide whether or not to remove F_k based on other features in ${S_k}^{(RS)}$.

Definition 4. (Predominant feature). A feature is predominant to the class, *iff* its correlation to the class is predominant or can become predominant after all features in ${S_i}^{(RS)}$ are removed.

According to the above definitions, we have the following theorem (proof is given elsewhere due to the space limit).

THEOREM 1. *The feature that is the most relevant to the class concept is always a predominant feature and can be used as a starting point to remove other features.*

Based on Theorem 1, we can easily obtain a deterministic procedure that can effectively identify predominant features and remove redundant ones among all relevant features in S', without having to identify all the redundant features for every relevant feature, and thus avoids pairwise analysis of F-correlations between all features. This is because we do not need to calculate *SU* values for all pairs of features in advance. Once we get a ranking of the relevance of every feature to the class concept and determine the relevant feature set S', we can analyze the F-correlations on the fly starting from the first feature in the ranking list. After a relevant feature is identified as redundant to one of the already-determined predominant features, it will be removed immediately without further processing for correlation analysis with other features.

3.3 Algorithm and Analysis

Based on the methodology presented before, we develop an algorithm, named **FCBF** (Fast Correlation-Based Filter). As in Figure 1, given a data set S with N features and a class C, the algorithm finds a set of predominant features S_{best} for the class concept. It consists of two major parts. In the first part (line 2-7), it calculates the *SU* value for each feature, selects relevant features into S'_{list} based on the predefined threshold δ, and orders them in descending order according to their *SU* values. In the second part (line 8-20), it further processes the ordered list S'_{list} to remove redundant features and only keeps predominant ones among all the selected relevant features. According to Theorem 1, a feature F_p that has already been determined to be a predominant feature can always be used to filter out other features that are ranked lower than F_p and have F_p as one of its redundant subjects. The iteration starts from the first element in S'_{list} (line 8) and continues as follows. For all the remaining features (from the one right next to F_p to the last one in S'_{list}), if F_p happens to be one of the redundant subjects to a feature F_q (line 14), F_q will be removed from S'_{list}. After one round of filtering features based on F_p, the algorithm will take the currently remaining feature right next to F_p as the new reference (line 19) to repeat the filtering process. The algorithm stops until there is no more feature to be removed from S'_{list}.

The first part of the above algorithm has linear time complexity $O(N)$ in terms of dimensionality N. As to the second part, in each iteration, using the predominant feature F_p identified in the previous round, FCBF can remove a large number of features that are redundant to F_p in the current iteration. The best case could be that all of the remaining features following F_p in the ranked list will be removed; the worst case could be none of them. On average, we can assume that half of the remaining features will be removed in each iteration. Therefore, the time complexity for the second part is $O(N \log N)$ in terms of N. Since the calculation of *SU* for a pair of features is linear in term of the number of instances M in a data set, the overall complexity of FCBF is $O(MN \log N)$.

```
input:   $S(F_1, F_2, ..., F_N, C)$ // a training data set
         $\delta$                    // a predefined threshold
output:  $S_{best}$                  // an optimal subset

1   begin
2     for $i = 1$ to $N$ do begin
3       calculate $SU_{i,c}$ for $F_i$;
4       if ($SU_{i,c} \geq \delta$)
5         append $F_i$ to $S'_{list}$;
6     end;
7     order $S'_{list}$ in descending $SU_{i,c}$ value;
8     $F_p = getFirstElement(S'_{list})$;
9     do begin
10      $F_q = getNextElement(S'_{list}, F_p)$;
11      if ($F_q$ <> NULL)
12        do begin
13          $F'_q = F_q$;
14          if ($SU_{p,q} \geq SU_{q,c}$)
15            remove $F_q$ from $S'_{list}$;
16            $F_q = getNextElement(S'_{list}, F'_q)$;
17          else $F_q = getNextElement(S'_{list}, F_q)$;
18        end until ($F_q$ == NULL);
19      $F_p = getNextElement(S'_{list}, F_p)$;
20    end until ($F_p$ == NULL);
21  $S_{best} = S'_{list}$;
22  end;
```

Figure 1: FCBF Algorithm

4. EMPIRICAL STUDY

The objective of this section is to evaluate our proposed algorithm in terms of speed, degree of dimensionality and redundancy reduction, and classification accuracy on selected features.

4.1 Experiment Setup

In our experiments, we select three feature selection algorithms in comparison with FCBF. One is a partial algorithm of FCBF (line 2-7 in Figure 1, denoted as FCBF-P) which merely ranks each feature in S based on its C-correlation and selects relevant ones according to the threshold δ. Another one is a feature weighting algorithm, ReliefF [10] (an extension to Relief) which searches for several nearest neighbors to be robust to noise and handles multiple classes. The third one is a subset search algorithm (denoted as CFS-SF) which exploits correlation measure and sequential forward search . It is a variation of the CFS algorithm mentioned in section 2. The reason why we prefer CSF-SF to CFS is because both experiments in [7] and our initial experiments show that CFS only produces slightly better results than CSF-SF, but CSF-SF runs faster than CFS based on best first search with 5 nodes expansion and therefore is more suitable for high dimensional data. In addition to feature selection algorithms, we also select a well known classification algorithm, C4.5 [19], to evaluate the accuracy on selected features for each feature selection algorithm.

Table 1: Summary of bench-mark data sets.

Title	# Features	# Instances	# Classes
Lung-cancer	57	32	3
Promoters	59	106	2
Splice	62	3190	3
USCensus90	68	9338	3
CoIL2000	86	5822	2
Chemical	151	936	3
Musk2	169	6598	2
Arrhythmia	280	452	16
Isolet	618	1560	26
Multi-features	650	2000	10

The experiments are conducted using Weka's implementation of all these existing algorithms and FCBF is also implemented in Weka environment [20]. All together 10 data sets are selected from the UCI Machine Learning Repository [2] and the UCI KDD Archive [1]. A summary of data sets is presented in Table 1.

For each data set, we run FCBF, CFS-SF, and ReliefF to obtain a set of selected features from each algorithm, and record their running times. As our purpose is to evaluate FCBF, it is not necessary for us to record the running time of FCBF-P (a fraction of that of FCBF), but we still record the set of selected feature for it in order to investigate the level of redundancy reduction by FCBF. We then apply C4.5 on the original data set as well as each newly obtained data set containing only the selected features from each feature selection algorithm and record overall classification accuracy by 10-fold cross-validation.

4.2 Results and Discussions

Table 2 records the running time and the number of selected features for each feature selection algorithm. For ReliefF, the parameter k is set to 5 (neighbors) and m is set to 30 (instances) throughout the experiments. From Table 2, we can observe that for each algorithm the running times over different data sets are consistent with our previous time complexity analysis. From the averaged values in the last row of Table 2, it is clear that FCBF runs significantly faster (in degrees) than CFS-SF and ReliefF, which verifies FCBF's superior computational efficiency. What is interesting is that ReliefF is unexpectedly slow even though its time complexity becomes $O(MN)$ with a fixed sample size m. The reason lies on that searching for nearest neighbors involves distance calculation which is more time consuming than the calculation of symmetrical uncertainty values.

From Table 2, it is also clear that FCBF achieves the highest level of dimensionality and redundancy reduction by selecting the least number of features (with only one exception in USCensus90), which is consistent with our theoretical analysis about FCBF's ability to identify redundant features.

Tables 3 shows the classification accuracy of C4.5 on different feature sets. From the averaged accuracy over all data sets, we observe that, in general, (1) FCBF improves the accuracy of C4.5; and (2) of the other three algorithms, only CSF-SF can enhance the accuracy of C4.5 to the same level as FCBF does. From individual accuracy values, we also observe that for most of the data sets, FCBF can maintain or even increase the accuracy.

The above experimental results suggest that FCBF can achieve what it is designed for and performs best among all feature selection algorithms under comparison in different aspects - it can efficiently achieve high degree of dimensionality and redundancy reduction and enhance classification accuracy with predominant features most of times. Therefore, FCBF is practical for feature selection for classification of high dimensional data.

5. CONCLUSIONS

In summary, a feature is *good* if it is *predominant* in predicting the class concept, and feature selection for classification is a process that identifies all predominant features to the class concept and removes the rest. In this work, we point out the importance of removing redundant features in high-dimensional data. After reviewing existing feature selection algorithms, we propose an efficient correlation-based algorithm based on the concepts of C-correlation and F-correlation. The superior performance of our method is established through extensive experiments on high dimensional data. Our next goal is to apply FCBF to data with higher dimensionality in the range of thousands of features.

Acknowledgements

We gratefully thank anonymous reviewers and Area Chair for their constructive comments. This work is in part supported by grants from NSF (No. 0127815, 0231448), Prop 301 (No. ECR A601), Ford and IMES at ASU for H. Liu.

6. REFERENCES

[1] S. D. Bay. The UCI KDD Archive, 1999. http://kdd.ics.uci.edu.

[2] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[3] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[4] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74–81, 2001.

[5] M. Dash and H. Liu. Feature selection for classifications. *Intelligent Data Analysis: An International Journal*, 1(3):131–156, 1997.

[6] M. Dash, H. Liu, and H. Motoda. Consistency based feature selection. In *Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining*, pages 98–109. Springer-Verlag, 2000.

[7] M. Hall. *Correlation Based Feature Selection for Machine Learning.* PhD thesis, University of Waikato, Dept. of Computer Science, 1999.

Table 2: Running time (in ms) and number of selected features for each feature selection algorithm.

	Running Time			# Selected Features			
	FCBF	CSF-SF	ReliefF	FCBF	FCBF-P	CSF-SF	ReliefF
Lung-cancer	20	50	50	5	19	8	5
Promoters	20	50	100	4	4	4	4
Splice	200	961	2343	6	6	6	11
USCensus90	541	932	7601	2	9	1	2
CoIL2000	470	3756	7751	3	12	10	12
Chemical	121	450	2234	4	39	7	7
Musk2	971	8903	18066	2	25	10	2
Arrhythmia	151	2002	2233	6	43	25	25
Isolet	3174	177986	17025	23	386	137	23
Multi-Features	4286	125190	21711	14	26	87	14
Average	995	32028	7911	7	57	30	11

Table 3: Accuracy of C4.5 on selected features for each feature selection algorithm.

Title	Full Set	FCBF	FCBF-P	CFS-SF	ReliefF
Lung-cancer	80.83 ±22.92	87.50 ±16.32	84.17 ±16.87	84.17 ±16.87	80.83 ±22.92
Promoters	86.91 ±6.45	87.73 ±6.55	87.73 ±6.55	87.73 ±6.55	89.64 ±5.47
Splice	94.14 ±1.57	93.48 ±2.20	93.48 ±2.20	93.48 ±2.20	89.25 ±1.94
USCensus90	98.27 ±0.19	98.08 ±0.22	98.19 ±0.31	97.95 ±0.15	98.08 ±0.22
CoIL2000	93.97 ±0.21	94.02 ±0.07	94.02 ±0.07	94.02 ±0.07	94.02 ±0.07
Chemical	94.65 ±2.03	95.51 ±2.31	95.40 ±2.21	96.47 ±2.15	93.48 ±1.79
Musk2	96.79 ±0.81	91.33 ±0.51	96.35 ±0.51	95.56 ±0.73	94.62 ±0.92
Arrhythmia	67.25 ±3.68	72.79 ±6.30	66.59 ±9.22	68.58 ±7.41	65.90 ±8.23
Isolet	79.10 ±2.79	75.77 ±4.07	79.68 ±2.62	80.70 ±4.94	52.44 ±3.61
Multi-Features	94.30 ±1.49	95.06 ±0.86	94.20 ±1.48	94.95 ±0.96	80.45 ±2.41
Average	88.62 ±9.99	89.13 ±8.52	88.98 ±9.81	89.36 ±9.24	83.87 ±14.56

[8] K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134. Menlo Park: AAAI Press/The MIT Press, 1992.

[9] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[10] I. Kononenko. Estimating attributes : Analysis and extension of RELIEF. In F. Bergadano and L. De Raedt, editors, *Proceedings of the European Conference on Machine Learning*, pages 171–182, Catania, Italy, 1994. Berlin: Springer-Verlag.

[11] H. Liu, F. Hussain, C. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, 2002.

[12] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining.* Boston: Kluwer Academic Publishers, 1998.

[13] H. Liu, H. Motoda, and L. Yu. Feature selection with selective sampling. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 395 – 402, 2002.

[14] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In L. Saitta, editor, *Proceedings of International Conference on Machine Learning (ICML-96), July 3-6, 1996*, pages 319–327, Bari, Italy, 1996. San Francisco: Morgan Kaufmann Publishers, CA.

[15] H. Liu, L. Yu, M. Dash, and H. Motoda. Active feature selection using classes. In *Proceedings of the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-03)*, pages 474–485, 2003.

[16] A. Y. Ng. On feature selection: learning with exponentially many irrelevant features as training examples. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 404–412, 1998.

[17] K. Ng and H. Liu. Customer retention via data mining. *AI Review*, 14(6):569 – 590, 2000.

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

[19] J. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[20] I. Witten and E. Frank. *Data Mining - Pracitcal Machine Learning Tools and Techniques with JAVA Implementations.* Morgan Kaufmann Publishers, 2000.

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

[22] Y. Yang and J. O. Pederson. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, 1997.