Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution

Lei Yu LEIYU@ASU.EDU
Huan Liu HLIU@ASU.EDU
Department of Computer Science & Engineering, Arizona State University, Tempe, AZ 85287-5406, USA

Abstract

Feature selection, as a preprocessing step to machine learning, has been effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving comprehensibility. However, the recent increase of dimensionality of data poses a severe challenge to many existing feature selection methods with respect to efficiency and effectiveness. In this work, we introduce a novel concept, predominant correlation, and propose a fast filter method which can identify relevant features as well as redundancy among relevant features without pairwise correlation analysis. The efficiency and effectiveness of our method is demonstrated through extensive comparisons with other methods using real-world data of high dimensionality.

1. Introduction

Feature selection is frequently used as a preprocessing step to machine learning. It is a process of choosing a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion. Feature selection has been a fertile field of research and development since 1970's and shown very effective in removing irrelevant and redundant features, increasing efficiency in learning tasks, improving learning performance like predictive accuracy, and enhancing comprehensibility of learned results (Blum & Langley, 1997; Dash & Liu, 1997; Kohavi & John, 1997). 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 genome projects (Xing et al., 2001), text categorization (Yang & Pederson, 1997), image retrieval (Rui et al., 1999), and customer relationship management (Ng & Liu, 2000). This enormity may cause serious problems to many machine learning algorithms with respect to scalability and learning 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 learning algorithms. Therefore, feature selection becomes very necessary for machine learning tasks when facing high dimensional data nowadays. However, this trend of enormity on both size and dimensionality also poses severe challenges to feature selection algorithms. Some of the recent research efforts in feature selection have been focused on these challenges from handling a huge number of instances (Liu et al., 2002b) to dealing with high dimensional data (Das, 2001; Xing et al., 2001). This work is concerned about feature selection for high dimensional data. In the following, we first review models of feature selection and explain why a filter solution is suitable for high dimensional data, and then review some recent efforts in feature selection for high dimensional data.

Feature selection algorithms can broadly fall into the filter model or the wrapper model (Das, 2001; Kohavi & John, 1997). The filter model relies on general characteristics of the training data to select some features without involving any learning algorithm, therefore it does not inherit any bias of a learning algorithm. The wrapper model requires one predetermined learning algorithm in feature selection and uses its performance to evaluate and determine which features are selected. As for each new subset of features, the wrapper model needs to learn a hypothesis (or a classifier). It tends to give superior performance as it finds features better suited to the predetermined learning algorithm, but it also tends to be more computationally expensive (Langley, 1994). When the number of features becomes very large, the filter model is usually a choice due to its computational efficiency.

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

To combine the advantages of both models, algorithms in a hybrid model have recently been proposed to deal with high dimensional data (Das, 2001; Ng, 1998; Xing et al., 2001). In these algorithms, first, a goodness measure of feature subsets based on data characteristics is used to choose best subsets for a given cardinality, and then, cross validation is exploited to decide a final best subset across different cardinalities. These algorithms mainly focus on combining filter and wrapper algorithms to achieve best possible performance with a particular learning algorithm at the same time complexity of filter algorithms. In this work, we focus on the filter model and aim to develop a new feature selection algorithm which can effectively remove both irrelevant and redundant features and is less costly in computation than the current available algorithms.

In section 2, we review current algorithms within the filter model and point out their problems in the context of high dimensionality. In section 3, we describe correlation measures which form the base of our method in evaluating feature relevance and redundancy. In section 4, we first propose our method which selects good features for classification based on a novel concept, **predominant correlation**, and then present a fast algorithm with less than quadratic time complexity. In section 5, we evaluate the efficiency and effectiveness of this algorithm via extensive experiments on various real-world data sets comparing with other representative feature selection algorithms, and discuss the implications of the findings. In section 6, we conclude our work with some possible extensions.

2. Related Work

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 representative algorithms in each group.

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 (Blum & Langley, 1997; Kohavi & John, 1997). 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 (Kira & Rendell, 1992). The key idea of Relief is to estimate 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. Relief 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 (Kira & Rendell, 1992). Many other algorithms in this group have similar problems as Relief does. They can only capture the relevance of features to the target concept, but cannot 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 learning algorithms and thus should be eliminated as well (Hall, 2000; Kohavi & John, 1997). 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.

Subset search algorithms search through candidate feature subsets guided by a certain evaluation measure (Liu & Motoda, 1998) which captures the goodness of each subset. An optimal (or near optimal) subset is selected when the search stops. Some existing evaluation measures that have been shown effective in removing both irrelevant and redundant features include the consistency measure (Dash et al., 2000) and the correlation measure (Hall, 1999; Hall, 2000). Consistency measure attempts to find a minimum number of features that separate classes as consistently as the full set of features can. An inconsistency is defined as two instances having the same feature values but different class labels. In Dash et al. (2000), different search strategies, namely, exhaustive, heuristic, and random search, are combined with this evaluation measure to form different algorithms. The time complexity is exponential in terms of data dimensionality for exhaustive search and quadratic for heuristic search. The complexity can be linear to the number of iterations in a random search, but experiments show that in order to find best feature subset, the number of iterations required is mostly at least quadratic to the number of features (Dash et al., 2000). In Hall (2000), a correlation measure is applied to evaluate the good-

ness of feature subsets based on the hypothesis that a good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other. The underlying algorithm, named CFS, also exploits heuristic search. Therefore, with quadratic or higher time complexity in terms of dimensionality, existing 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 algorithm which can effectively identify both irrelevant and redundant features with less time complexity than subset search algorithms.

3. Correlation-Based Measures

In this section, we 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 with the class but not highly correlated with 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. In this sense, the problem of feature selection boils down to find a suitable measure of correlations between features and a sound procedure to select features based on this measure.

There exist broadly two approaches to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information theory. Under the first approach, the most well known measure is *linear correlation coefficient*. For a pair of variables (X, Y), the linear correlation coefficient r is given by the formula

$$r = \frac{\sum_i (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_i (x_i - \overline{x_i})^2}\sqrt{\sum_i (y_i - \overline{y_i})^2}} \quad (1)$$

where $\overline{x_i}$ is the mean of X, and $\overline{y_i}$ is the mean of Y. The value of r lies between -1 and 1, inclusive. If X and Y are completely correlated, r takes the value of 1 or -1; if X and Y are totally independent, r is zero. It is a symmetrical measure for two variables. Other measures in this category are basically variations of the above formula, such as *least square regression error* and *maximal information compression index* (Mitra et al., 2002). There are several benefits of choosing linear correlation as a feature goodness measure for classification. First, it helps remove features with near zero linear correlation to the class. Second, it helps to reduce redundancy among selected features. It is known that if data is linearly separable in the original representation, it is still linearly separable if all but one of a group of linearly dependent features are removed (Das, 1971). However, it is not safe to always assume linear correlation between features in the real world. Linear correlation measures may not be able to capture correlations that are not linear in nature. Another limitation is that the calculation requires all features contain numerical values.

To overcome these shortcomings, in our solution we adopt the other approach and choose a 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

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)), \quad (2)$$

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

$$H(X|Y) = -\sum_j P(y_j)\sum_i P(x_i|y_j)\log_2(P(x_i|y_j)), \quad (3)$$

where $P(x_i)$ is the prior probabilities for all values of 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* (Quinlan, 1993), given by

$$IG(X|Y) = H(X) - H(X|Y). \quad (4)$$

According to this measure, a feature Y is regarded more correlated to feature X than to feature Z, if $IG(X|Y) > IG(Z|Y)$. About information gain measure, we have the following theorem.

Theorem Information gain is symmetrical for two random variables X and Y.

Proof Sketch: To prove $IG(X|Y) = IG(Y|X)$, we need to prove $H(X) - H(X|Y) = H(Y) - H(Y|X)$. This can be easily derived from $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$. □

Symmetry is a desired property for a measure of 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* (Press et al., 1988), defined as follows.

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \quad (5)$$

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. These 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 (Fayyad & Irani, 1993; Liu et al., 2002a). Therefore, we use symmetrical uncertainty in this work.

4. A Correlation-Based Filter Approach

4.1. 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). More specifically, suppose a data set S contains 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), then a subset S' of relevant features can be decided by a threshold SU value δ, such that $\forall f_i \in S', 1 \le i \le N$, $SU_{i,c} \ge \delta$.

The answer to the second question is more complicated because it may involve analysis of pairwise correlations between all features (named f-correlation), which results in a time complexity of $O(N^2)$ associated with the number of features N for most existing algorithms. To solve this problem, we propose our method below.

Since f-correlations are also captured by SU values, in order to decide whether a relevant feature is redundant or not, 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 features in S' is high enough to cause redundancy so that one of them may be removed from S'. For a feature f_i in S', 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 $SU_{j,i}$ for $\forall f_j \in S'$ $(j \ne i)$, we will also obtain quantified estimations about the extent to which f_i is correlated to (or predicted by) the rest relevant features in S'. Therefore, it is possible to identify highly correlated features to f_i in the same straightforward manner as we decide S', using a threshold SU value equal or similar to δ. We can do this for all features in S'. 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 the highly correlated features for a given feature f_i within S', it is more reasonable to use the c-correlation level between f_i and the class concept, $SU_{i,c}$, as a reference. The reason lies on the common phenomenon - a feature that is correlated with one concept (e.g., the class) at a certain level may also be correlated with some other concepts (features) at the same or an even higher level. Therefore, even the correlation between this feature and the class concept is larger than some threshold δ and thereof making this feature relevant to the class concept, this correlation is by no means predominant. To be more precise, we define the concept of **predominant correlation** as follows.

Definition 1 (Predominant correlation). The correlation between a feature f_i ($f_i \in S$) and the class C is predominant *iff* $SU_{i,c} \ge \delta$, and $\forall f_j \in S'$ $(j \ne i)$, there exists no f_j such that $SU_{j,i} \ge SU_{i,c}$.

If there exists such f_j to a feature f_i, we call it a redundant peer to f_i and use S_{P_i} to denote the set of all redundant peers for f_i. Given $f_i \in S'$ and S_{P_i} $(S_{P_i} \ne \emptyset)$, we divide S_{P_i} into two parts, ${S_{P_i}}^+$ and ${S_{P_i}}^-$, where ${S_{P_i}}^+ = \{f_j | f_j \in S_{P_i}, SU_{j,c} > SU_{i,c}\}$ and ${S_{P_i}}^- = \{f_j | f_j \in S_{P_i}, SU_{j,c} \le SU_{i,c}\}$.

Definition 2 (Predominant feature). A feature is predominant to the class, *iff* its correlation to the class is predominant or can become predominant after removing its redundant peers.

According to the above definitions, 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. We now propose three heuristics that together can effectively identify predominant features and remove redundant ones among all relevant features, without having to identify all the redundant peers for every feature in S', and thus avoids pairwise analysis of f-correlations between all relevant features.

Our assumption in developing these heuristics 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 predict the class while reducing redundancy in the data.

Heuristic 1 ($if\ S_{P_i}{}^{+} = \emptyset$). Treat f_i as a predominant feature, remove all features in $S_{P_i}{}^{-}$, and skip identifying redundant peers for them.

Heuristic 2 ($if\ S_{P_i}{}^{+} \neq \emptyset$). Process all features in $S_{P_i}{}^{+}$ before deciding whether or not to remove f_i. If none of them becomes predominant, follow Heuristic 1; otherwise only remove f_i and decide whether or not to remove features in $S_{P_i}{}^{-}$ based on other features in S'.

Heuristic 3 (*starting point*). The feature with the largest $SU_{i,c}$ value is always a predominant feature and can be a starting point to remove other features.

4.2. 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 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 Heuristic 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 peers. The iteration starts from the first element (Heuristic 3) 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 a redundant peer to a feature f_q, f_q will be removed from S'_{list} (Heuristic 2). 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}.

```
input:  S(f_1, f_2, ..., f_N, C)  // a training data set
        δ                         // 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} ≥ δ)
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} ≥ 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

The first part of the above algorithm has a linear time complexity in terms of the number of features 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 peers 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 $O(MN \log N)$.

5. Empirical Study

The objective of this section is to evaluate our proposed algorithm in terms of speed, number of selected features, and learning accuracy on selected features.

5.1. Experiment Setup

In our experiments, we choose three representative feature selection algorithms in comparison with FCBF. One is a feature weighting algorithm, ReliefF (an extension to Relief) which searches for several nearest neighbors to be robust to noise and handles multiple classes (Kononenko, 1994); the other two are sub-

set search algorithms which exploit sequential forward search and utilizes correlation measure or consistency measure to guide the search, denoted as CorrSF and ConsSF respectively. CorrSF is a variation of the CFS algorithm mentioned in section 2. The reason why we prefer CorrSF to CFS is because both experiments in Hall (1999) and our initial experiments show that CFS only produces slightly better results than CorrSF, but CorrSF based on sequential forward search 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 two different learning algorithms, C4.5 (Quinlan, 1993) and NBC (Witten & Frank, 2000), 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 algorithms and FCBF is also implemented in Weka environment (Witten & Frank, 2000). All together 10 data sets are selected from the UCI Machine Learning Repository (Blake & Merz, 1998) and UCI KDD Archive (Bay, 1999). A summary of data sets is presented in Table 1.

For each data set, we run all four feature selection algorithms, FCBF, ReliefF, CorrSF, ConsSF, respectively, and record the running time and the number of selected features for each algorithm. We then apply C4.5 and NBC on the original data set as well as each newly obtained data set containing only the selected features from each algorithm and record overall accuracy by 10-fold cross-validation.

5.2. 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 the other three algorithms, 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 value.

From Table 2, it is also clear that FCBF achieves the highest level of dimensionality 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 and 4 show the learning accuracy of C4.5 and NBC respectively on different feature sets. From the averaged accuracy over all data sets, we observe that, in general, (1) FCBF improves the accuracy of both C4.5 and NBC; and (2) of the other three algorithms, only CorrSF 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 is practical for feature selection for classification of high dimensional data. It can efficiently achieve high degree of dimensionality reduction and enhance classification accuracy with predominant features.

6. Conclusions

In this paper, we propose a novel concept of predominant correlation, introduce an efficient way of analyzing feature redundancy, and design a fast correlation-based filter approach. A new feature selection algorithm FCBF is implemented and evaluated through extensive experiments comparing with related feature selection algorithms. The feature selection results are further verified by applying two different classification algorithms to data with and without feature selection. Our approach demonstrates its efficiency and effectiveness in dealing with high dimensional data for classification. Our further work will extend FCBF to work on data with higher dimensionality (thousands of features). We will study in more detail redundant features and their role in classification, and combine FCBF with feature discretization algorithms to smoothly handle data of different feature types.

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

Title	Running Time				# Selected Features			
	FCBF	CorrSF	ReliefF	ConsSF	FCBF	CorrSF	ReliefF	ConsSF
Lung-cancer	20	50	50	110	5	8	5	4
Promoters	20	50	100	190	4	4	4	4
Splice	200	961	2343	34920	6	6	11	10
USCensus90	541	932	7601	161121	2	1	2	13
CoIL2000	470	3756	7751	341231	3	10	12	29
Chemical	121	450	2234	14000	4	7	7	11
Musk2	971	8903	18066	175453	2	10	2	11
Arrhythmia	151	2002	2233	31235	6	25	25	24
Isolet	3174	177986	17025	203973	23	137	23	11
Multi-Features	4286	125190	21711	133932	14	87	14	7
Average	995	32028	7911	109617	7	30	11	12

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

Title	Full Set	FCBF	CorrSF	ReliefF	ConsSF
Lung-cancer	80.83 ±22.92	87.50 ±16.32	84.17 ±16.87	80.83 ±22.92	84.17 ±16.87
Promoters	86.91 ±6.45	87.73 ±6.55	87.73 ±6.55	89.64 ±5.47	84.00 ±6.15
Splice	94.14 ±1.57	93.48 ±2.20	93.48 ±2.20	89.25 ±1.94	93.92 ±1.53
USCensus90	98.27 ±0.19	98.08 ±0.22	97.95 ±0.15	98.08 ±0.22	98.22 ±0.30
CoIL2000	93.97 ±0.21	94.02 ±0.07	94.02 ±0.07	94.02 ±0.07	93.99 ±0.20
Chemical	94.65 ±2.03	95.51 ±2.31	96.47 ±2.15	93.48 ±1.79	95.72 ±2.09
Musk2	96.79 ±0.81	91.33 ±0.51	95.56 ±0.73	94.62 ±0.92	95.38 ±0.75
Arrhythmia	67.25 ±3.68	72.79 ±6.30	68.58 ±7.41	65.90 ±8.23	67.48 ±4.49
Isolet	79.10 ±2.79	75.77 ±4.07	80.70 ±4.94	52.44 ±3.61	69.23 ±4.53
Multi-Features	94.30 ±1.49	95.06 ±0.86	94.95 ±0.96	80.45 ±2.41	90.80 ±1.75
Average	88.62 ±9.99	89.13 ±8.52	89.36 ±9.24	83.87 ±14.56	87.29 ±11.04

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.

References

Bay, S. D. (1999). The UCI KDD Archive. http://kdd.ics.uci.edu.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*, 245–271.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 74–81).

Das, S. K. (1971). Feature selection with a linear dependence measure. *IEEE Transactions on Computers*.

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

Dash, M., Liu, H., & Motoda, H. (2000). Consistency based feature selection. *Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining* (pp. 98–109). Springer-Verlag.

Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the Thirteenth*

Table 4. Accuracy of NBC on selected features for each feature selection algorithm.

Title	Full Set	FCBF	CorrSF	ReliefF	ConsSF
Lung-cancer	80.00 ±23.31	90.00 ±16.10	90.00 ±16.10	80.83 ±22.92	86.67 ±17.21
Promoters	90.45 ±7.94	94.45 ±8.83	94.45 ±8.83	87.82 ±10.99	92.64 ±7.20
Splice	95.33 ±0.88	93.60 ±1.74	93.60 ±1.74	88.40 ±1.97	94.48 ±1.39
USCensus90	93.38 ±0.90	97.93 ±0.16	97.95 ±0.15	97.93 ±0.16	97.87 ±0.26
CoIL2000	79.03 ±2.08	93.94 ±0.21	92.94 ±0.80	93.58 ±0.43	83.18 ±1.94
Chemical	60.79 ±5.98	72.11 ±2.51	70.72 ±4.20	78.20 ±3.58	67.20 ±2.51
Musk2	84.69 ±2.01	84.59 ±0.07	64.85 ±2.09	84.59 ±0.07	83.56 ±1.05
Arrhythmia	60.61 ±3.32	66.61 ±5.89	68.80 ±4.22	66.81 ±3.62	68.60 ±7.64
Isolet	83.72 ±2.38	80.06 ±2.52	86.28 ±2.14	52.37 ±3.17	71.67 ±3.08
Multi-Features	93.95 ±1.50	95.95 ±1.06	96.15 ±0.94	76.05 ±3.26	93.75 ±1.95
Average	82.20 ±12.70	86.92 ±10.79	85.57 ±12.53	80.66 ±13.38	83.96 ±11.31

International Joint Conference on Artificial Intelligence (pp. 1022–1027). Morgan Kaufmann.

Hall, M. (1999). *Correlation based feature selection for machine learning*. Doctoral dissertation, University of Waikato, Dept. of Computer Science.

Hall, M. (2000). Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 359–366).

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

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–324.

Kononenko, I. (1994). Estimating attributes : Analysis and extension of RELIEF. *Proceedings of the European Conference on Machine Learning* (pp. 171–182). Catania, Italy: Berlin: Springer-Verlag.

Langley, P. (1994). Selection of relevant features in machine learning. *Proceedings of the AAAI Fall Symposium on Relevance*. AAAI Press.

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

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers.

Liu, H., Motoda, H., & Yu, L. (2002b). Feature selection with selective sampling. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 395 – 402).

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 301–312.

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

Ng, K., & Liu, H. (2000). Customer retention via data mining. *AI Review*, *14*, 569 – 590.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1988). *Numerical recipes in C*. Cambridge University Press, Cambridge.

Quinlan, J. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Rui, Y., Huang, T. S., & Chang, S. (1999). Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, *10*, 39–62.

Witten, I., & Frank, E. (2000). *Data mining - pracitcal machine learning tools and techniques with JAVA implementations*. Morgan Kaufmann Publishers.

Xing, E., Jordan, M., & Karp, R. (2001). Feature selection for high-dimensional genomic microarray data. *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 601–608).

Yang, Y., & Pederson, J. O. (1997). A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412–420).