# Redundancy Based Feature Selection for Microarray Data

Lei Yu
Department of Computer Science & Engineering
Arizona State University
Tempe, AZ 85287-8809
leiyu@asu.edu

Huan Liu
Department of Computer Science & Engineering
Arizona State University
Tempe, AZ 85287-8809
hliu@asu.edu

## ABSTRACT

In gene expression microarray data analysis, selecting a small number of discriminative genes from thousands of genes is an important problem for accurate classification of diseases or phenotypes. The problem becomes particularly challenging due to the large number of features (genes) and small sample size. Traditional gene selection methods often select the top-ranked genes according to their individual discriminative power without handling the high degree of redundancy among the genes. Latest research shows that removing redundant genes among selected ones can achieve a better representation of the characteristics of the targeted phenotypes and lead to improved classification accuracy. Hence, we study in this paper the relationship between feature relevance and redundancy and propose an efficient method that can effectively remove redundant genes. The efficiency and effectiveness of our method in comparison with representative methods has been demonstrated through an empirical study using public microarray data sets.

## 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 redundancy, gene selection, microarray data

## 1. INTRODUCTION

The rapid advances in gene expression microarray technology enable simultaneously measuring the expression levels for thousands or tens of thousands of genes in a single experiment [19]. Analysis of microarray data presents unprecedented opportunities and challenges for data mining in areas such as gene clustering [2, 10], sample clustering and class discovery [2, 7, 20], sample classification [1, 7], and gene selection [20, 22]. In sample classification, a microarray data set is provided as a training set of labeled samples. The task is to build a classifier that accurately predicts the classes (diseases or phenotypes) of novel unlabeled samples. A typical data set may contain thousands of genes but only a small number of samples (often less than a hundred). The number of samples is likely to remain small at least for the near future due to the expense of collecting microarray samples [6]. The nature of relatively high dimensionality but small sample size in microarray data can cause the known problem of "curse of dimensionality" and overfitting of the training data [6]. Therefore, selecting a small number of discriminative genes from thousands of genes is essential for successful sample classification [5, 7, 22]. Research on feature selection is receiving increasing attention in gene selection for sample classification.

Feature selection, a process of choosing a subset of features from the original ones, is frequently used as a preprocessing technique in data mining [4, 15]. It has proven effective in reducing dimensionality, improving mining efficiency, increasing mining accuracy, and enhancing result comprehensibility [3, 12]. Feature selection methods can broadly fall into the *wrapper* model and the *filter* model [12]. The wrapper model uses the predictive accuracy of a predetermined mining algorithm to determine the goodness of a selected subset. It is computationally expensive for data with a large number of features [3, 12]. The filter model separates feature selection from classifier learning and relies on general characteristics of the training data to select feature subsets that are independent of any mining algorithms. In gene selection, the filter model is often adopted due to its computational efficiency [7, 22].

Among existing gene selection methods, earlier methods often evaluate genes in isolation without considering gene-to-gene correlation. They rank genes according to their individual relevance or discriminative power to the targeted classes and select top-ranked genes. Some methods based on statistical tests or information gain have been shown in [7, 16]. These methods are computationally efficient due to linear time complexity $O(N)$ in terms of dimensionality $N$. However, they share two shortcomings: (1) certain domain knowledge or trial-and-error is required to determine the threshold for the number of genes selected (e.g., a total of 50 genes were selected in Golub et al' work [7]); and (2) they cannot remove redundant genes. The issue of "redundancy" among genes is recently raised in gene selection literature. It is pointed out in a number of studies [5, 23] that simply combining a highly ranked gene with another


*SIGKDD* '04, August 22-25, 2004, Seattle, WA, USA

highly ranked gene often does not form a better gene set because these two genes could be highly correlated. The effects of redundancy among selected genes are two-fold. On one hand, the selected gene set can have a less comprehensive representation of the targeted classes than one of the same size but without redundant genes; on the other hand, in order to include all representative genes in the selected gene set, redundant genes will unnecessarily increase the dimensionality of the selected gene set, which will in turn affect the mining performance on the small sample [6]. Some latest methods [5, 22] take into account the gene-to-gene correlation and remove redundant genes through pair-wise correlation analysis among genes. They can handle redundancy to certain extent but require time complexity of $O(N^2)$. In addition, they still need to decide the threshold for the number of selected genes.

In this paper, we tackle gene selection by developing a novel redundancy based approach that overcomes the above two shortcomings in an efficient way. The remainder of this paper is organized as follows. In Section 2, we introduce gene and feature selection, review notions of feature relevance, and identify the need for feature redundancy analysis. In Section 3, we provide formal definitions on feature redundancy, and reveal the relationship between feature relevance and feature redundancy. In Section 4, we describe correlation measures and present our redundancy based filter method. Section 5 contains experimental evaluation and discussions. Section 6 concludes this work.

## 2. GENE, FEATURE, AND FEATURE RELEVANCE

In sample classification and gene selection, the microarray data for analysis is in the form of a gene expression matrix (shown in Figure 1), in which each column represents a gene and each row represents a sample with label $c_i$. For each sample the expression levels of all the genes in study are measured, so $f_{ij}$ is the measurement of the expression level of the $j$th gene for the $i$th sample where $j = 1, ..., N$ and $i = 1, ..., M$. The format of a microarray data set conforms to the normal data format of machine learning and data mining, where a gene can be regarded as a feature or attribute and a sample as an instance or a data point.

| Gene 1 | Gene 2 | . | . | . | Gene $N$ | |
|---|---|---|---|---|---|---|
| $f_{11}$ | $f_{12}$ | . | . | . | $f_{1N}$ | $c_1$ |
| $f_{21}$ | $f_{22}$ | . | . | . | $f_{2N}$ | $c_2$ |
| . | . | . | | | . | . |
| . | . | | . | | . | . |
| . | . | | | . | . | . |
| $f_{M1}$ | $f_{M2}$ | . | . | . | $f_{MN}$ | $c_M$ |

**Figure 1: An example of gene expression matrix.**

Let $F$ be a full set of features and $G \subseteq F$. In general, the goal of feature selection can be formalized as selecting a minimum subset $G$ such that $\mathbf{P}(C \mid G)$ is equal or as close as possible to $\mathbf{P}(C \mid F)$, where $\mathbf{P}(C \mid G)$ is the probability distribution of different classes given the feature values in $G$ and $\mathbf{P}(C \mid F)$ is the original distribution given the feature values in $F$ [13]. We call such a minimum subset an *optimal* subset, illustrated by the example below.

EXAMPLE 1. **(Optimal subset)** *Let features $F_1, ..., F_5$ be Boolean. The target concept is $C = g(F_1,\ F_2)$ where $g$ is a Boolean function. With $F_2 = \overline{F_3}$ and $F_4 = \overline{F_5}$, there are only eight possible instances. In order to determine the target concept, $F_1$ is indispensable; one of $F_2$ and $F_3$ can be disposed of (note that $C$ can also be determined by $g(F_1,\ \overline{F_3})$), but we must have one of them; both $F_4$ and $F_5$ can be discarded. Either $\{F_1,\ F_2\}$ or $\{F_1,\ F_3\}$ is an optimal subset. The goal of feature selection is to find either of them.*

In the presence of hundreds or thousands of features, researchers notice that it is common that a large number of features are not informative because they are either irrelevant or redundant with respect to the class concept [22]. Traditionally, feature selection research has focused on searching for relevant features. Although empirical evidence shows that along with irrelevant features, redundant features also affect the speed and accuracy of mining algorithms [8, 12, 13], there is little work on explicit treatment of feature redundancy. We next present a classic notion of feature relevance and employ the same example above to illustrate why it alone cannot handle feature redundancy.

Based on a review of previous definitions of feature relevance, John, Kohavi, and Pfleger classified features into three disjoint categories, namely, strongly relevant, weakly relevant, and irrelevant features [11]. Let $F_i \in F$ and $S_i = F - \{F_i\}$. These categories of relevance can be formalized as follows.

DEFINITION 1. **(Strong relevance)** *A feature $F_i$ is strongly relevant iff*

$$\mathbf{P}(C \mid F_i,\ S_i) \neq \mathbf{P}(C \mid S_i)\ .$$

DEFINITION 2. **(Weak relevance)** *A feature $F_i$ is weakly relevant iff*

$$\mathbf{P}(C \mid F_i,\ S_i) = \mathbf{P}(C \mid S_i),\ \texttt{and}$$

$$\exists\ S_i' \subset S_i,\ \textit{such that}\ \mathbf{P}(C \mid F_i,\ S_i') \neq \mathbf{P}(C \mid S_i')\ .$$

COROLLARY 1. **(Irrelevance)** *A feature $F_i$ is irrelevant iff*

$$\forall\ S_i' \subseteq S_i,\ \mathbf{P}(C \mid F_i,\ S_i') = \mathbf{P}(C \mid S_i')\ .$$

Strong relevance of a feature indicates that the feature is always necessary for an optimal subset; it cannot be removed without affecting the original conditional class distribution. Weak relevance suggests that the feature is not always necessary but may become necessary for an optimal subset at certain conditions. Irrelevance (following Definitions 1 and 2) indicates that the feature is not necessary at all. According to these definitions, it is clear that in previous Example 1, feature $F_1$ is strongly relevant, $F_2$, $F_3$ weakly relevant, and $F_4$, $F_5$ irrelevant. An optimal subset should include all strongly relevant features, none of irrelevant features, and a subset of weakly relevant features. However, it is not given in the definitions which of weakly relevant features should be selected and which of them removed. Therefore, we can conclude that feature relevance alone cannot effectively help remove redundant features and there is also a need for feature redundancy analysis.

## 3. DEFINING FEATURE REDUNDANCY

Notions of feature redundancy are normally in terms of feature correlation. It is widely accepted that two features are redundant to each other if their values are completely correlated (for example, features $F_2$ and $F_3$ in Example 1). In reality, it may not be so straightforward to determine feature redundancy when a feature is correlated (perhaps partially) with a set of features. We now formally define feature redundancy in order to devise an approach to explicitly identify and eliminate redundant features. Before we proceed, we first introduce the definition of a feature's Markov blanket given by Koller and Sahami (1996).

DEFINITION 3. **(Markov blanket)** *Given a feature $F_i$, let $M_i \subset F$ ($F_i \notin M_i$), $M_i$ is said to be a Markov blanket for $F_i$ iff*

$$\mathbf{P}(F - M_i - \{F_i\},\ C \mid F_i,\ M_i) = \mathbf{P}(F - M_i - \{F_i\},\ C \mid M_i)\ .$$

The Markov blanket condition requires that $M_i$ subsume not only the information that $F_i$ has about $C$, but also about all of the other features. It is pointed out in Koller and Sahami (1996) that an optimal subset can be obtained by a backward elimination procedure, known as *Markov blanket filtering*: let $G$ be the current set of features ($G = F$ in the beginning), at any phase, if there exists a Markov blanket for $F_i$ within the current $G$, $F_i$ is removed from $G$. It is proved that this process guarantees a feature removed in an earlier phase will still find a Markov blanket in any later phase, that is, removing a feature in a later phase will not render the previously removed features necessary to be included in the optimal subset. According to previous definitions of feature relevance, we can also prove that strongly relevant features cannot find any Markov blanket. Since irrelevant features should be removed anyway, we exclude them from our definition of redundant features. Hence, our definition of redundant feature is given as follows.

DEFINITION 4. **(Redundant cover)** *A Markov blanket $M_i$ of a weakly relevant feature $F_i$ is called a redundant cover of $F_i$ .*

DEFINITION 5. **(Redundant feature)** *Let $G$ be the current set of features, a feature is redundant and hence should be removed from $G$ iff it has a redundant cover within $G$ .*

From the property of Markov blanket, it is easy to see that a redundant feature removed earlier remains redundant when more features are removed. Figure 1 depicts the relationships between definitions of feature relevance and redundancy introduced so far. It shows that an entire feature set can be conceptually divided into four basic disjoint parts: irrelevant features (I), redundant features (II, part of weakly relevant features), weakly relevant but non-redundant features (III), and strongly relevant features (IV). An optimal subset essentially contains all the features in parts III and IV. It is worthy to point out that although parts II and III are disjoint, different partitions of them can result from the process of Markov blanket filtering. In previous Example 1, either of $F_2$ or $F_3$, but not both, should be removed as a redundant feature.

In determining relevant features and redundant features, it is advisable to use efficient approximation methods for two reasons. First, searching for an optimal subset based on the definitions of feature relevance and redundancy is combinatorial in nature. It is obvious that exhaustive or complete search is prohibitive with a large number of features. Second, an optimal subset is defined based on the full population where the true data distribution is known. It is generally assumed that a training data set is only a small portion of the full population, especially in a high-dimensional space. Therefore, it is not proper to search for an optimal subset from the training data as over-searching the training data can cause over-fitting [9]. We next present our method.

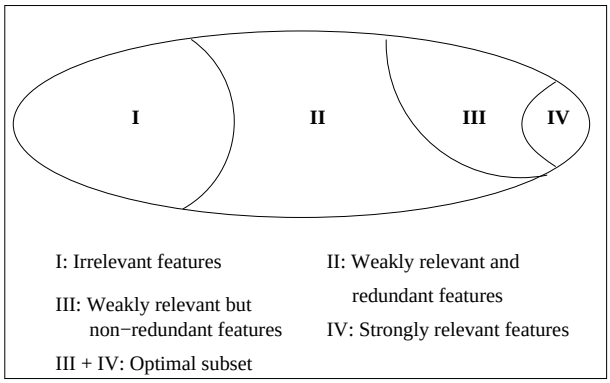


**Figure 2: A view of feature relevance and redundancy.**

## 4. AN APPROXIMATION METHOD

In this section, we first introduce our choice of correlation measure for gene selection, and then describe our approximation method based on the previous definitions.

### 4.1 Correlation Measures

In gene selection, there exist broadly two types of measures for correlation between genes or between a gene and the target class: linear and non-linear. Since linear correlation measures may not be able to capture correlations that are not linear in nature, in our approach we adopt non-linear correlation measures 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

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

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

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

where $P(x_i)$ is the prior probabilities for all values of $X$, and $P(x_i \mid y_i)$ 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*, given by

$$IG(X \mid Y) = H(X) - H(X \mid Y)\ .$$

Information gain tends to favor variables with more values and can be normalized by their corresponding entropy. In

this work, we use *symmetrical uncertainty* ($SU$), defined as

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

which compensates for information gain's bias toward features with more values and restricts its values to the range $[0, 1]$. A value of 1 indicates that knowing the values of either feature completely predicts the values of the other; a value of 0 indicates that $X$ and $Y$ are independent. Entropy-based measures handle nominal or discrete values, and therefore continuous values in gene expression data need to be discretized [14] in order to use entropy-based measures.

## 4.2 Methodology and Analysis

In developing an approximation method for both relevance and redundancy analysis, our goal is to efficiently find a feature subset that approximates the optimal subset (parts III and IV in Figure 2). We first differentiate two types of correlation between features and the class.

DEFINITION 6. **(Individual $C$-correlation)** *The correlation between any feature $F_i$ and the class $C$ is called individual $C$-correlation, denoted by $ISU_i$ .*

DEFINITION 7. **(Combined $C$-correlation)** *The correlation between any pair of features $F_i$ and $F_j$ $(i \neq j)$ and the class $C$ is called combined $C$-correlation, denoted by $CSU_{i,j}$ .*

For combined $C$-correlation, we treat features $F_i$ and $F_j$ as one single feature $F_{i,j}$ and the cartesian product of the domains of $F_i$ and $F_j$ as the domain of $F_{i,j}$.

In relevance analysis, the individual $C$-correlation for each feature is measured and ranked. Without choosing any threshold, we heuristically treat all features as relevant features which are subject to redundancy analysis. In Section 2, we apply the concept of redundant cover to *exactly* determine feature redundancy. When it comes to *approximately* determine feature redundancy, the key is to define approximate redundant cover. In our method, we approximately determine the redundancy between two features based on both their individual $C$-correlations and combined $C$-correlation. We assume that a feature with a larger individual $C$-correlation value contains by itself more information about the class than a feature with a smaller individual $C$-correlation value. For two features $F_i$ and $F_j$ with $ISU_i \geq ISU_j$, we choose to evaluate whether feature $F_i$ can form an approximate redundant cover for feature $F_j$ (instead of $F_j$ for $F_i$) in order to maintain more information about the class. In addition, if combining $F_j$ with $F_i$ does not provide more predictive power in determining the class than $F_i$ alone, we heuristically decide that $F_i$ forms an approximate redundant cover for $F_j$. Therefore, an approximate redundant cover is defined as follows.

DEFINITION 8. **(Approximate redundant cover)** *For two features $F_i$ and $F_j$, $F_i$ forms an approximate redundant cover for $F_j$ iff $ISU_i \geq ISU_j$ and $ISU_i \geq CSU_{i,j}$ .*

Recall that Markov blanket filtering, a backward elimination procedure based on a feature's Markov blanket in the current set, guarantees that a redundant feature removed in an earlier phase will still find a Markov blanket (redundant cover) in any later phase when another redundant feature is removed. It is easy to verify that this is not the case for backward elimination based on a feature's approximate redundant cover in the current set. For instance, if $F_j$ is the only feature that forms an approximate redundant cover for $F_k$, and $F_i$ forms an approximate redundant cover for $F_j$, after removing $F_k$ based on $F_j$, further removing $F_j$ based on $F_i$ will result in no approximate redundant cover for $F_k$ in the current set. However, we can avoid this situation by removing a feature only when it can find an approximate redundant cover formed by a predominant feature, defined as follows.

DEFINITION 9. **(Predominant feature)** *A feature is predominant iff it does not have any approximate redundant cover in the current set.*

Predominant features will not be removed at any stage. If a feature $F_j$ is removed based on a predominant feature $F_i$ in an earlier phase, it is guaranteed that it will still find an approximate redundant cover (the same $F_i$) in any later phase when another feature is removed. Since the feature with the highest $ISU$ value does not have any approximate redundant cover, it must be one of the predominant features and can be used as the starting point to determine the redundancy between the rest features. In summary, our approximation method of relevance and redundancy analysis is to find all predominant features and eliminate the rest. It can be summarized by an algorithm shown in Table 1.

---

**Algorithm RBF:**

**Relevance analysis**

1 Order features based on decreasing $ISU$ values

**Redundancy analysis**

2 Initialize $F_i$ with the first feature in the list

3 Find and remove all features for which $F_i$ forms an approximate redundant cover

4 Set $F_i$ as the next remaining feature in the list and repeat step 3 until the end of the list

---

**Table 1: A two-step Redundancy Based Filter (RBF) algorithm.**

We now analyze the efficiency of RBF before conducting an empirical study. Major computation of the algorithm involves $ISU$ and $CSU$ values, which has linear time complexity in terms of the number of instances in a data set. Given dimensionality $N$, the algorithm has linear time complexity $O(N)$ in relevance analysis. To determine predominant features in redundancy analysis, it has a best-case time complexity $O(N)$ when only one feature is selected and all of the rest features are removed in the first round, and a worse-case time complexity $O(N^2)$ when all features are selected. In general cases when $k$ $(1 < k < N)$ features are selected, based on the predominant feature identified in the previous round, RBF typically removes a large number of features in the current round. This makes RBF substantially faster than algorithms of subset evaluation based on traditional greedy sequential search, as will be demonstrated by the running time comparisons reported in Section 5. As to space complexity, the algorithm only requires space linear to dimensionality $N$ to compute and store $ISU$ values in relevance analysis, as $CSU$ values can be dynamically computed in redundancy analysis.

Our method is suboptimal due to the way individual and combined $C$-correlations are used for relevance and redundancy analysis. It is fairly straightforward to improve the optimality of the results by considering more complex combinations of features in evaluating feature relevance and redundancy, which in turn increases time complexity. To improve result optimality without increasing time complexity, further effort is needed to design and compare different heuristics in determining a feature's approximate redundant cover. In our previous work [24], the redundancy between a pair of features is approximately determined based on their individual $C$-correlations to the class and the correlation between themselves which is measured by their symmetrical uncertainty value. The method has similar complexity to RBF, but it does not directly consider the overall predictive power of two features when determining feature redundancy, while RBF does based on their combined $C$-correlation.

# 5. EMPIRICAL STUDY

In this section, we empirically evaluate the efficiency and effectiveness of our method on gene expression microarray data. The efficiency of a feature selection algorithm can be directly measured by its running time over various data sets. As to effectiveness, since we often do not have prior knowledge about which genes are irrelevant or redundant in microarray data, we adopt two indirect criteria: (a) number of selected genes and (b) predictive accuracy on selected genes. For gene selection in sample classification, it is desirable to select the smallest number of genes which can achieve the highest predictive accuracy on new samples.

## 5.1 Experimental Setup

In our experiments, we select four microarray data sets which are frequently used in previous studies: colon cancer, leukemia, breast cancer, and lung cancer [1]. Note that except for colon data, each original data set comes with training and test samples that were drawn from different conditions. Here we combine them together for the purpose of cross validation. The details of these data sets are summarized in Table 2.

**Table 2: Summary of microarray data sets.**

| Title | # Genes | # Samples | # Samples per class | |
|---|---|---|---|---|
| Colon cancer | 2000 | 62 | tumor 40 | normal 22 |
| Leukemia | 7129 | 72 | ALL 47 | AML 25 |
| Lung cancer | 12533 | 181 | MPM 31 | ADCA 150 |
| Breast cancer | 24481 | 97 | relapse 46 | non-relapse 51 |

Two representative filter algorithms are chosen in comparison with RBF in terms of the evaluation criteria identified before. One algorithm representing feature ranking methods is ReliefF, which searches for nearest neighbors of instances of different classes and ranks features according to their importance in differentiating instances of different classes. A subset of features is selected from top of the ranking list [18]. The other algorithm is a variation of the CFS algorithm, denoted by CFS-SF (Sequential Forward). CFS uses symmetrical uncertainty to measure the correlation between each feature and the class and between two features, and exploits best-first search in searching for a feature subset of maximum overall correlation to the class and minimum correlation among selected features [8]. Sequential forward selection is used in CFS-SF as previous experiments show CFS-SF runs much faster to produce similar results than CFS. A widely used classification algorithm C4.5 [17] is adopted to evaluate the predictive accuracy of the selected features. We use programs for ReliefF, CFS-SF, and C4.5 from Weka's collection [21]. RBF is also implemented in the Weka environment.

For each data set, we first apply all the feature selection algorithms in comparison, and obtain the running time and the selected genes for each algorithm. Note that in applying ReliefF, the number of nearest neighbors is set to 5 and all instances are used in weighting genes. We then apply C4.5 on the original data set and each of the three newly obtained data sets (with only the selected genes), and obtain overall classification accuracy by leave-one-out cross-validation, a performance validation procedure adopted by many researchers due to the small sample size of microarray data [5, 22]. The experiments were conducted on a Pentium III PC with 256 KB RAM.

## 5.2 Results and Discussions

Table 3 reports the running time, number of selected genes, and the leave-one-out accuracy for each feature selection algorithm. As shown in the table, RBF produced selected genes in seconds for each data set. ReliefF took from 4 seconds (on colon data) to 6 minutes (on lunge cancer data) to produce the results. CFS-SF produced its result on colon data in 5 seconds, but it failed on the other three data sets as the program ran out of memory after a period of time ($>$ 10 minutes) due to its $O(N^2)$ space complexity in terms of the number of genes $N$. These observations clearly show the superior efficiency of RBF for gene selection in high-dimensional microarray data.

We now examine the effectiveness of these three algorithms based on the number of genes selected and the leave-one-out accuracy reported in Table 3. We pick the colon data to explain the difference in gene selection results. As we can see from Table 3, based on the original colon data (2000 genes), 12 out of 62 samples were incorrectly classified, resulting an overall accuracy of 80.65%. RBF selected only 4 genes and helped to reduce the number of misclassified samples to 4 (increasing the overall accuracy to 93.55%). ReliefF also selected 4 genes but only reduced the number of errors to 9 (85.48% on accuracy) since it cannot handle feature redundancy. CFS-SF resulted in 7 errors (88.71% on accuracy), but it selected more genes than RBF. Across the four data sets, we can observe that RBF selected less number of genes than ReliefF and CFS-SF. At the same time, the genes selected by RBF led to the highest accuracy by leave-one-out.

In summary, the above results suggest that RBF is an efficient and effective method for gene selection and is practical for use in sample classification. It is worthy to emphasize that genes selected by RBF are independent of classification algorithms, in other words, RBF does not directly aim to increase the accuracy of C4.5.

[1] http://sdmc.lit.org.sg/GEDatasets/Datasets.html

**Table 3: Comparison of gene selection results: Acc records leave-one-out cross-validation accuracy rate** (%)

| | Full Set | | RBF | | | ReliefF | | | CFS-SF | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | # Genes | Acc | Time (s) | # Genes | Acc | Time (s) | # Genes | Acc | Time (s) | # Genes | Acc |
| Colon cancer | 2000 | 80.65 | 0.47 | 4 | 93.55 | 4.3 | 4 | 85.48 | 4.6 | 26 | 88.71 |
| Leukemia | 7129 | 73.61 | 1.74 | 4 | 87.50 | 22.91 | 60 | 81.94 | N/A | N/A | N/A |
| Lung cancer | 12533 | 96.13 | 4.26 | 6 | 98.34 | 339.39 | 64 | 98.34 | N/A | N/A | N/A |
| Breast cancer | 24481 | 57.73 | 9.65 | 67 | 79.38 | 211.06 | 70 | 59.79 | N/A | N/A | N/A |

# 6. CONCLUSIONS

In this work, we have introduced the importance of removing redundant genes in sample classification and pointed out the necessity of studying feature redundancy. We have provided formal definitions on feature redundancy and proposed a redundancy based filter method with two desirable properties: (1) it does not require the selection of any threshold in determining feature relevance or redundancy; and (2) it combines sequential forward selection with elimination, which substantially reduces the number of feature pairs to be evaluated in redundancy analysis. Experiments on microarray data have demonstrated the efficiency and effectiveness of our method in selecting discriminative genes that improve classification accuracy.

## 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), and CEINT 2004 at ASU.

# 7. REFERENCES

[1] A. Alizadeh and etal. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[2] U. Alon and et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, 96:6745–6750, 1999.

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

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

[5] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the Computational Systems Bioinformatics Conference*, pages 523–529, 2003.

[6] E. R. Dougherty. Small sample issue for microarray-based classification. *Comparative and Functional Genomics*, 2:28–34, 2001.

[7] T. R. Golub and et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[8] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 359–366, 2000.

[9] D. D. Jensen and P. R. Cohen. Multiple comparisions in induction algorithms. *Machine Learning*, 38(3):309–338, 2000.

[10] D. Jiang, J. Pei, and A. Zhang. Interactive exploration of coherent patterns in time-series gene expression data. In *Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 565–570, 2003.

[11] G. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, pages 121–129, 1994.

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

[13] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning*, pages 284–292, 1996.

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

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

[16] F. Model, P. Adorjan, A. Olek, and C. Piepenbrock. Feature selection for DNA methylation based cancer classification. *Bioinformatics*, 17:157–164, 2001.

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

[18] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning*, 53:23–69, 2003.

[19] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.

[20] C. Tang, A. Zhang, and J. Pei. Mining phenotypes and informative genes from gene expression data. In *Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 655–660, 2003.

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

[22] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 601–608, 2001.

[23] M. Xiong, Z. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Research*, 11:1878–1887, 2001.

[24] L. Yu and H. Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. In *Proc. of the 20th International Conference on Machine Learning*, pages 856–863, 2003.