Soft Comput (2008) 12:111–120
DOI 10.1007/s00500-007-0193-8

FOCUS

A genetic algorithm-based method for feature subset selection

Feng Tan · Xuezheng Fu · Yanqing Zhang ·
Anu G. Bourgeois

Published online: 31 May 2007

Abstract As a commonly used technique in data preprocessing, feature selection selects a subset of informative attributes or variables to build models describing data. By removing redundant and irrelevant or noise features, feature selection can improve the predictive accuracy and the comprehensibility of the predictors or classifiers. Many feature selection algorithms with different selection criteria has been introduced by researchers. However, it is discovered that no single criterion is best for all applications. In this paper, we propose a framework based on a genetic algorithm (GA) for feature subset selection that combines various existing feature selection methods. The advantages of this approach include the ability to accommodate multiple feature selection criteria and find small subsets of features that perform well for a particular inductive learning algorithm of interest to build the classifier. We conducted experiments using three data sets and three existing feature selection methods. The experimental results demonstrate that our approach is a robust and effective approach to find subsets of features with higher classification accuracy and/or smaller size compared to each individual feature selection algorithm.

Keywords Feature selection · Gene Selection · Genetic algorithm · Microarray gene expression data analysis

1 Introduction

Recent technological developments such as the Internet, hyperspectral imagery, and microarrays have facilitated the emergence of enormous amounts of multivariate data in various applications such as gene expression array analysis, proteomics, information retrieval, and text classification. These applications that involve hundreds to thousands of attributes or variables introduce a challenge to pattern classification or prediction, which makes feature selection critical (Liu et al. 2002; Guyon et al. 2002; Guyon and Elisseeff 2003; Liu et al. 2005a).

Feature selection techniques study how to identify and select informative (discriminative) features for building models which can interpret data better. Feature selection can reduce the computational cost by reducing dimensionality of data, improve the prediction performance and the comprehensibility of the models by eliminating redundant and irrelevant (probable noise) features. Feature selection is different from feature transformation (or feature extraction) which creates new features by combining the original features. On the other hand, feature selection maintains the original meanings of the selected features, which is desirable in some domains.

Many feature selection algorithms have been proposed in the literature. All these methods search for optimal or near optimal subsets of features that optimize a given criterion. Feature selection algorithms can be divided into two categories based on whether the selection criterion depends on the learning algorithm used to construct the classifier or predictor. Filter methods utilize the intrinsic properties of

F. Tan (✉) · X. Fu · Y. Zhang · A. G. Bourgeois
Department of Computer Science,
Georgia State University, Atlanta, GA 30302, USA
e-mail: ftan@student.gsu.edu

X. Fu
e-mail: xfu1@gsu.edu

Y. Zhang
e-mail: yzhang@cs.gsu.edu

A. G. Bourgeois
e-mail: anu@cs.gsu.edu

the data to select subsets of features as a preprocessing step, independently of the chosen classifier. Features are assessed by their relevance or discriminant powers with regard to targeted classes. On the other hand, wrapper methods utilize the learning machine of interest to assess subsets of features according to their predictive or classification performance. Wrappers can often find small feature subset with high accuracy because the features match well with the learning methods. However, wrappers typically require extensive computation. It is argued that filters have better generalization properties since it is independent of any specific learning method. Among the proposed feature selection algorithms, feature ranking approaches that score or rank features by certain ranking criterion and use rankings of features as the base of selection mechanism are particularly attractive because of their simplicity, scalability, and good empirical success. Computationally, feature ranking is efficient since it requires only the computation of n scores and sorting the scores. Statistically, it is robust against overfitting because it introduces bias but it may have considerably less variance (Guyon and Elisseeff 2003; Hastie et al. 2001). Based on the ranks of features, subsets of significant features can be selected to build a predictor or classifier.

Different researchers have introduced varying feature selection criteria. Some feature selection methods use criteria based on statistics, such as χ^2-statistics (Liu and Setiono 1995), T-Statistics (Liu et al. 2002), F-Statistics (Peng et al. 2005), MIT correlation (also known as signal-to-noise statistic) Golub (1999), Fisher criterion (Furey et al. 2000), some use information-theoretic criteria including information gain (Liu 2004), mutual information (Guyon and Elisseeff 2003; Peng et al. 2005), and entropy-based measure (Dash and Liu 1999; Liu et al. 2005b). Some other approaches utilize machine learning approaches, such as support vector machines (SVMs) (Guyon et al. 2002; Weston et al. 2000; Mao et al. 2005) decision trees (Breiman and Forest), and evolutionary algorithm (Yang and Honavar 1998; Jirapech-Umpai and Aitken 2005) for feature ranking or selection.

Since these criteria are very diverse and motivated by various theoretic arguments, they may produce substantially different outcomes when applied to same data set. It has been pointed out that various selection criteria are biased with respect to dimensionality and no single criterion is best for all applications (Dy and Brodley 2004; Chuang et al. 2004). Such phenomena are also supported in our experiments. As a consequence, the performance of the classifiers built upon these feature selection methods varies as well. This discordance caused by various selection criteria makes the interpretation of the data difficult. Moreover, it makes us hard to decide which method is best fit for new unknown data sets. Hence, exploring ways to combine multiple criteria or develop multi-objective criteria seems a reasonable approach to study.

Hsu et al. (2002) studied the behavior and relationship between rank combination and score combination by introducing a concept called rank/score graph. They showed that under certain condition rank combination outperforms score combination. Chuang et al. (2004) applied rank combination to combine different feature selection methods. The ranks of features are combined by using a weighted sum (or average) from each of the component rankings obtained from individual feature selection method. It is showed that the combination approach performs better than each individual feature selection method in many cases.

In this paper, we propose a framework based on a genetic algorithm (GA) for feature subset selection that combines various existing feature selection methods. The goal is to effectively utilize useful information from different feature selection methods to select better feature subsets with smaller size and/or higher classification performance in comparison with the existing methods. Multiple selection criteria are combined by a genetic algorithm to improve feature subset selection. The advantage of accommodating multiple selection criteria gives us the ability of finding such better feature subsets. Our approach is also independent of the inductive learning algorithm used to build the classifier. To evaluate our method, we conducted experiments using three data sets and three existing feature selection algorithms, that is, entropy-based (Dash and Liu 1999), T-statistics, and SVM-recursive feature elimination (RFE) (Guyon et al. 2002). These existing algorithms are used to provide candidate features for GA to select feature subsets. Our approach is applied on two microarray data sets [colon cancer data (Alon et al. 1999) and prostate cancer data (Singh 2002) and ionosphere database [28]. Experimental results show that our approach is robust and effective in finding small subsets of informative features with higher classification accuracy and/or smaller size compared to each individual feature selection algorithm.

The rest of the paper is organized as follows. Section 2 describes our framework for hybrid feature selection. Support Vector Machine which is used as the classifier in our experiments is briefly introduced in Sect. 3. Section 4 describes three existing feature selection algorithms used in experiments. Section 5 compares our method with the three feature selection algorithms with experimental results. Conclusions and discussions are presented in Sect. 6.

2 Our approach

The idea of our hybrid approach (shown in Fig. 1.) is to absorb valuable outcomes from multiple feature selection algorithms to find subsets of informative features that have smaller size and/or better classification performance than the individual algorithms. A genetic algorithm (GA) in the

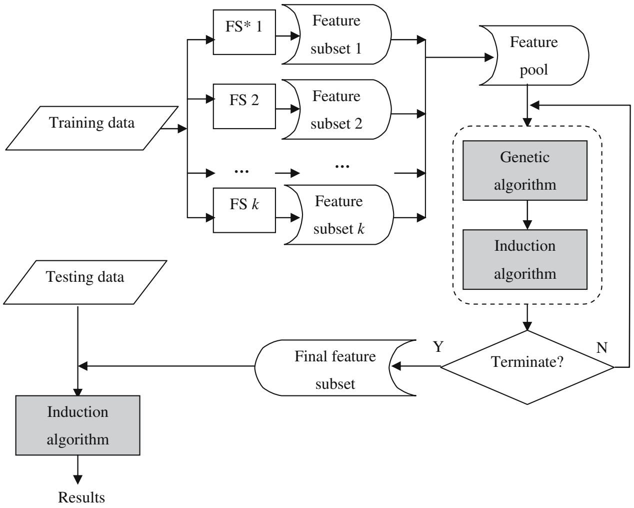

Fig. 1 Framework of hybrid feature selection
* Feature Selection Algorithm

framework accomplishes the fusion of multiple feature selection criteria. We limit ourselves to supervised feature selection in this paper.

In the first stage, several existing feature selection methods are applied on a data set. Then the feature subsets produced by these methods are fed into the feature pool that is used by the GA in the second stage. Then the GA will try to search an optimal or near optimal feature subset from this feature pool. Genetic algorithms can search a pool of hypotheses (called population) containing complex interacting parts. Each individual (hypothesis) of the current population is evaluated according to a specified fitness function (associated with an induction algorithm in here). A new population is generated by applying genetic operations (selection, crossover, and mutation). Our genetic algorithm is designed to maximize classification accuracy and minimize the size of feature subsets.

2.1 Feature pool

The feature pool is a collection of candidate features to be selected by the genetic algorithm to find an optimal or near optimal feature subset. Instead of using all features from the original data, we take sets of features selected by multiple feature selection algorithms to form the pool. Thus the feature pool contains valuable outcomes from different selection criteria. Some feature selection algorithms can automatically generate a subset of important features, while others produce a mere ranking. In the latter case, we need to determine a cut-off point for a ranked list of features to obtain a feature subset. Given a ranking of features, it is unclear how to threshold the ranking to select only important variables and to exclude those that is pure noise. One common practice is to simply select the top-ranked features—say, top 20. A deficiency of this simple approach is that it leads to the selection of a redundant subset. Several recent studies have addressed such redundancy (Peng et al. 2005; Yu and Liu 2003). Any combination or number of feature selection algorithms can be used to generate the feature pool for input to the GA.

2.2 Representation of hypotheses

Each individual represents a feature subset. The individuals are encoded by n-bit binary vectors. The bit with value 1 in a vector represents the corresponding feature being selected, while the bit with value 0 means the opposite.

2.3 Fitness function

The genetic algorithm is designed to optimize two objectives: maximize classification accuracy of the feature subset and minimize the number of features selected. To do so, we define the following fitness function:

$$F = w * c(x) + (1 - w) * (1/s(x))$$

where x is a feature vector representing a feature subset selected and w is a parameter between 0 and 1. The function is composed of two parts. The first part is a weighted classification accuracy $c(x)$ from the classifier and the second part is weighted size $s(x)$ of the feature subset represented by x. For a given w, the fitness of an individual x is increased as the classification accuracy of the x increases and decreased as the size of x increases. Increasing the value of w means that we give more priority on the classification accuracy over the size. On the other hand, reducing the value of w will give more penalties on the size of x. By adjusting w, we can achieve a tradeoff between the accuracy and the size of the feature subset obtained.

2.4 Induction algorithm

The genetic algorithm is independent of the inductive learning algorithm used by the classifier. Different induction algorithms, such as Naïve Bayes, artificial neural network, and decision trees can be flexibly incorporated into our method. In this paper, we use SVM classifier (Burges 1998) in the experiments.

2.5 Genetic operators

(1) Selection: Roulette wheel selection is used to probabilistically select individuals from a population for later breeding. The probability of selecting individual h_i is

determined by:

$$P(h_i) = \frac{F(h_i)}{\sum_{i=1}^{p} F(h_i)}$$

where $F(h_i)$ is the fitness value of h_i. The probability that an individual will be selected is proportional to its own fitness and is inversely proportional to the fitness of the other competing hypothesis in the current population.

(2) Crossover: we use single-point crossover operator. The cross-over point i is chosen at random so that the first i bits are contributed by one parent and the remaining bits by the second parent.

(3) Mutation: each individual has a probability p_m to mutate. We randomly choose a number of n bits to be flipped in every mutation stage.

3 Support vector machines

Support vector machines (SVMs) is a new generation learning system based on recent advances in statistical learning theory (Burges 1998; Vapnik 1998). SVMs create a decision boundary (the maximal-margin separating hyperplane) between the positive group and negative group and select the most relevant examples involved in the decision process (called support vectors). If the data is linearly separable, the construction of the hyperplane is always possible. Otherwise, SVMs can use kernels which nonlinearly map into a higher dimensional feature space so that a separating hyperplane can be found. We adopt linear SVM in this work:

$$K(x_i, x_j) = < x_i, x_j > \tag{1}$$

where x_i and x_j are two data instances in a d-dimensional Euclidian space.

For a linear kernel SVM, the margin width can be calculated as the following:

$$w = \sum_{i=1}^{N_s} \alpha_i y_i x_i \tag{2}$$

$$\text{margin width} = 2/\|w\| \tag{3}$$

where N_s is the number of support vectors, which are defined to be the training samples with $0 < \alpha_i \leq C$. C is the penalty parameter of the error term.

4 Feature selection methods

To evaluate our method, we adopt three feature selection algorithms in the experiments, two filters (entropy-based, T-statistics) and one wrapper (SVM–RFE) to form the feature pool. All of the three methods generate a mere ranking of features. We then pick a number of top-ranked features from each ranking and input them into the feature pool. The performances of the three algorithms are described in Sect. 4.

4.1 Entropy-based feature ranking

The entropy-based method (Dash and Liu 1999) is based on the fact that entropy is lower for orderly configurations and higher for disorderly configurations. From this point of view, it is assumed that removing an irrelevant feature would reduce the entropy more than that for a relevant feature. The algorithm ranks the features in descending order of relevance by finding the descending order of the entropies after removing each feature one at a time. The entropy measure of a data set of N instances is calculated as the following:

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N}(s_{ij} \times \log s_{ij} + (1 - s_{ij}) \times \log(1 - s_{ij})) \tag{4}$$

$$S_{ij} = e^{-\alpha \times D_{ij}} \tag{5}$$

$$\alpha = \frac{-\ln 0.5}{\overline{D}} \tag{6}$$

where S_{ij} is the similarity measure based on distance between two instances x_i and x_j with all numeric features (similarity between two instances with nomial features is measured using Hamming distance) and α is a parameter. D_{ij} is the Euclidean distance between the two instances. $\overline{D}$ is the average distance among the instances. This method can be used for unsupervised data since no class information is needed.

4.2 T-statistics

T-statistics is a classical feature selection approach (Liu et al. 2002) which has proven effective. It assesses whether the means of two groups are statistically different from each other. Each sample is labeled with $1, -1$. For each feature f_j, the mean μ_j^1 (resp. μ_j^{-1}) and standard deviation δ_j^1 (resp. δ_j^{-1}) are calculated using only the samples labeled 1 (resp. -1). Then a score $T(f_j)$ can be obtained by Eq. (7).

$$T(f_j) = \frac{|\mu_j^1 - \mu_j^{-1}|}{\sqrt{\frac{(\delta_j^1)^2}{n_1} + \frac{(\delta_j^{-1})^2}{n_{-1}}}} \tag{7}$$

Where n_1 (resp. $n{-}1$) is the number of samples labeled as 1 (resp. -1). When making a selection, those features with the highest scores are considered as the most discriminatory features.

4.3 SVM–RFE

Guyon et al. (2002) proposed a backward feature elimination algorithm by removing one "worst" gene (i.e., the one that changed the objective or cost function J least after being removed) at one time.

$$J = |w|^2/2 \tag{8}$$

in which w is calculated by Eq. (2), because only linear SVM is adopted. The change of J caused by removing the ith feature is approximated by optimal brain damage (OBD) algorithm (LeCun et al. 1990):

$$\Delta J(i) = \frac{\partial J}{\partial w_i}\Delta w_i + \frac{\partial^2 J}{\partial w_i^2}(\Delta w_i)^2 \tag{9}$$

At the optimum of J, the first order is neglected and the second order becomes

$$\Delta J(i) = (\Delta w_i)^2 \tag{10}$$

Because removing the ith feature means $\Delta w_i = w_i$, w_i^2 is taken as the ranking criterion. The feature with the smallest w_i^2 is removed due to its smallest effect on classification. The iterative procedure of RFE is as follows:

(1) train SVM with the training data
(2) compute the ranking criterion for all features
(3) remove the feature with smallest ranking criterion

5 Experiments

Microarray technology is having a significant impact on molecular biology. By allowing the monitoring of expression levels of thousands of genes (features) in cells simultaneously, it leads to a more complete understanding of the molecular variations among tumors and hence to a finer and more reliable classification. Many fields including drug discovery and toxicological research will certainly benefit from the use of DNA microarray technology.

Two practical realities constrain the analysis of microarray data (Somorjai et al. 2003). One is the "curse of dimensionality": the number of features characterizing these data is in the thousands or tens of thousands. The other is the "curse of dataset sparsity": the number of samples is comparatively limited. These two curses are believed to significantly deteriorate the performance of a classifier. Therefore, it is important to be able to remove redundant and irrelevant genes and find a subset of discriminative genes for accurate diagnosis of disease. In this work, we did the first two experiments using two microarray data sets. The third experiment using a radar database is in a different application domain.

Cross-validation procedure is commonly used to evaluate the performance of a classifier. In k-fold cross-validation, the data is divided into k subsets of (approximately) equal size. We train the classifier k times, each time leaving out one of the subsets from training, but using only the omitted subset to compute the classification accuracy. Leave-one-out (LOO) cross-validation (CV) is a special case of k-fold cross-validation where k equals the sample size. Leave-one-out cross-validation (LOOCV) is used in our first experiment. With a test data set available in Experiment 2, we use fivefold cross-validation to obtain the training accuracy.

Our focus is on using a GA to improve classification accuracy and minimize the size of feature subsets in comparison with each individual feature selection algorithm, not on comparing the effects of different induction algorithms (classifiers) on feature selection. Thus we only use SVM with linear kernel as the classifier in the experiments. However, it is flexible to incorporate different induction algorithms into our hybrid approach. Moreover, since we are not focusing on optimizing the performance of SVMs, no efforts has been made to find the optimal parameters for SVM. In each experiment, every feature subset is classified using the same linear SVM with same parameters. To save time, the population size and number of generations used in the experiments by our genetic algorithm are relatively small. It is possible to achieve better results if more iterations or larger population size are allowed. All experiments are implemented in a PC with Pentium 4 (2.4 GHz) and 512M RAM. All algorithms are coded in C++ and Matlab R14.

5.1 Experiment 1

Colon cancer data set (Alon et al. 1999) contains 62 tissues (samples) among which there are 40 tumor tissues and 22 normal tissues collected from colon-cancer patients. Gene expression information of colon cancer on more than 6,500 genes were measured using oligonucleotide microarray and 2,000 of them with highest minimum intensity were extracted to form a matrix of 62 tissues × 2,000 gene expression values. For the sake of simplicity, we identify the genes (features) with their column indexes in the matrix.

First, the three feature selection methods (Entropy-based, T-statistics, and SVM-RFE) are applied to the data set and three rankings of features are obtained. Next, we pick a number of top-ranked features (e.g., top-2 features, top-4 features, etc.) to get a few feature subsets. Then, SVM classifies the data set using these feature subsets. The classification accuracy of the feature subsets selected from the three rankings is presented in Table 1.

We can see that SVM–RFE provides the highest accuracy of the three except in the first case with a subset of top-2 features. With a subset of top-16 features, SVM–RFE achieves highest accuracy of 98.3% while the accuracies of the other two are 64.5 and 88.7%, respectively. In general, T-statistics gives acceptable performance. Entropy-based method purely

Table 1 LOO Accuracy of entropy-based, T-statistic and SVM–RFE on colon cancer data

Top features	Entropy-based (%)	T-statistic(%)	SVM–RFE (%)
2	64.5	79.0	75.9
4	64.5	88.7	89.7
8	64.5	88.7	96.6
16	64.5	88.7	98.3
32	64.5	88.7	96.6
64	66.1	88.7	94.8
128	74.2	90.3	93.1
256	80.7	88.7	91.4
512	85.5	83.9	86.2
1024	83.9	80.7	84.5
2000	83.9	83.9	79.3

scores features based on the entropy value of the system without considering the class information, which may explain its worst classification performance of the three. However, it can be used for unsupervised data and may be less prone to overfitting. SVM–RFE assesses features by tightly binding with the classifier (SVM). It ranks features with the magnitude of the weights of a linear discriminant classifier. We think that this may account for the good performance of SVM–RFE in the experiments. In our implementation, SVM–RFE takes much more time than the others to rank the features because it needs to train the SVM to do the ranking.

Due to the consideration of the cost of performing the necessary clinical test and analysis, a small size of informative gene subset (e.g., no more than 20 genes) is usually preferred for data analysis for a given accuracy. The top-20 genes ranked by the three algorithms on colon cancer data set are presented in Table 2. It shows only two genes with column index 14 and 1,423 are shared by T-statistics and SVM–RFE and entropy-based has no common feature with others in top-20 features. Besides the top-20 features, we notice that the ranks of other features are also very different in the three algorithms.

After the three rankings of features are obtained, we choose a number of top-ranked genes from the rankings and input them to the feature pool used by the GA. The classification performances of the feature selection algorithms and the domain knowledge can affect how the feature pool is formed. The genetic algorithm uses the following parameter settings in this experiment:

- Population size: 20–30
- Number of generations: 10
- Probability of crossover: 1
- Probability of mutation: 0.001

Table 3 shows the feature subsets selected by the GA and the classification accuracy of the subsets on colon cancer data. We test several feature pools (no more than 20 features in total) with different values of parameter w in the fitness function. Each feature pool contains a different number of top-ranked genes from the three methods. The results demonstrate that the feature subsets selected by our GA can accomplish the two goals: either achieve higher accuracy with smaller size or equal accuracy with smaller size compared to the feature subsets of the same size level selected by the other three.

As we can see from the table, reducing w does affect the size of feature subsets selected. Smaller values of w impose more penalties on the size of the subsets being selected. Therefore using smaller w tends to select smaller subsets. In general, reducing w reduces the accuracy as well. However, there are a few exceptions in Table 3. For example, in the (4, 8, 8) feature pool, the GA chooses a subset of 9 features reaching 100% accuracy with $w = 0.75$. This subset is smaller than the one of 12 features with $w = 0.85$, but their accuracies are the same (100%). In addition, the subset obtains higher accuracy than the one of 10 features with $w = 0.8$. These indicate that there may exist redundancy, interaction and correlations between these features so that the feature subset with smaller size can achieve higher accuracy.

5.2 Experiment 2

We further test the three feature selection methods and our GA on prostate cancer data (Singh 2002). The training set contains 52 prostate tumor samples and 50 non-tumor (normal) prostate samples with 12,600 genes. An independent set of testing samples is also available, which is from a

Table 2 Top-20 Features from entropy-based, T-statistics, and SVM–RFE on colon cancer data

Feature selection algorithms	Top-20 Features
Entropy-based	169, 1451, 1430, 1538, 375, 445, 1277, 1660, 603, 761, 1055, 1150, 1697, 609, 1170, 825, 1590, 1910, 803, 1264
T-Statistics	493, **1423**, 249, 377, 765, 245, 267, 66, **14**, 822, 1772, 625, 897, 137, 1674, 111, 1635, 513, 1892, 286
SVM–RFE	175, 70, **14**, 15, **1423**, 1378, 115, 164, 1791, 110, 1024, 35, 206, 38, 3, 1976, 415, 65, 16, 1325

The numbers in bold are the common gene(s)/feature(s) selected by two methods

Table 3 LOO Accuracy of GA on colon cancer data

w	Feature pool*	GA	LOO Accuracy (%)
0.85	2, 4, 4	6 (14, 15, 70, 175, 249, 493)	96.6
	4, 4, 4	7 (14, 15, 70, 175, 249, 377, 493)	96.6
	4, 8, 8	12 (14, 15, 70, 164, 175, 245, 267, 377, 493, 1378, 1423, 1451)	100
0.8	2, 4, 4	3 (70, 175, 493)	91.9
	4, 4, 4	4 (14, 70, 493, 1430)	93.5
	4, 8, 8	10 (14, 15, 66, 70, 175, 245, 493, 1378, 1423, 1430)	98.4
0.75	2, 4, 4	2 (377, 1423)	88.7
	4, 4, 4	3 (14, 377, 493)	91.9
	4, 8, 8	9 (14, 15, 70, 175, 267, 493, 1430, 1451, 1538)	100
0.7	2, 4, 4	1 (377)	83.9
	4, 4, 4	2 (249, 377)	91.9
	4, 8, 8	3 (70, 267, 1451)	90.3

* The three numbers in a feature pool represent the number of top features selected from entropy-based, T-Statistics and SVM–RFE, respectively

Table 4 Training and testing accuracy of entropy-based, T-Statistic and SVM–RFE on prostate cancer data

Top features	Training accuracy (%)			Testing accuracy (%)		
	Entropy-based	T-Statistics	SVM–RFE	Entropy-based	T-Statistics	SVM–RFE
2	59.8	76.5	84.3	73.5	97.1	73.5
4	59.8	78.4	86.3	73.5	97.1	70.6
8	61.8	86.3	96.1	73.5	88.2	73.5
16	62.8	83.3	100	73.5	88.2	85.3
32	63.7	89.2	100	73.5	88.2	94.1
64	64.7	90.2	100	73.5	76.5	91.2
128	63.7	91.2	99.0	73.5	91.2	91.2
256	63.7	93.1	95.1	73.5	82.4	91.2
512	67.7	93.1	95.1	76.5	82.4	91.2
1024	68.6	91.2	94.1	73.5	85.3	94.1
2048	71.6	91.2	93.1	73.5	88.2	94.1
4096	76.5	89.2	92.2	82.4	94.1	94.1
8192	87.3	90.2	91.2	97.1	97.1	94.1
12600	89.2	89.2	91.2	97.1	97.1	94.1

different experiment and has a nearly tenfold difference in overall microarray intensity from the training data. The testing set contains 34 samples (25 tumor and 9 normal samples).

Table 4 demonstrates the training accuracy and testing accuracy from the three algorithms. SVM–RFE performs better in terms of higher training accuracy. The highest testing accuracy achieved by SVM–RFE is 94.1%, which is lower that the highest accuracy (97.1%) obtained by the other two. Again, we compare the top-20 features ranked from the three methods in Table 5 and find out that no genes are shared by the three. There are only two common genes (205 and 12,153) shared by T-Statistics and SVM–RFE.

Since this data set is relatively large with 12,600 features, we run the GA with smaller population size and fewer generations to reduce time consumption:

- Population size: 10
- Number of generations: 5
- Probability of crossover: 1
- Probability of mutation: 0.001

Table 6 presents the results of applying the GA on the prostate cancer data. From all the cases in the table, the GA obtains 94.1% testing accuracy, which is the highest one that can be

Table 5 Top-20 features from entropy-based, T-statistics, and SVM–RFE on prostate cancer data

Feature selection algorithms	Top-20 features
Entropy-based	4234, 1058, 2789, 2474, 575, 4502, 6472, 12354, 5041, 3474, 727, 9994, 1585, 6365, 7249, 5823, 8052, 11401, 11926, 9926
T-Statistics	6185, 10138, 3879, 7520, 4365, 9050, **205**, 5654, 3649, **12153**, 3794, 9172, 9850, 8136, 7768, 5462, 12148, 9034, 4833, 8965
SVM–RFE	10234, **12153**, 8594, 9728, 11730, **205**, 11091, 10484, 12495, 49, 12505, 10694, 1674, 7079, 2515, 11942, 8058, 8658, 8603, 7826

The numbers in bold are the common gene(s)/feature(s) selected by two methods

Table 6 Training and testing accuracy of GA on prostate cancer data

w	Feature pool*	GA	Training accuracy (%)	Testing accuracy (%)
0.85	2, 4, 4	5 (4234, 6185, 7520, 8594, 10138)	93.1	94.1
	4, 4, 4	7 (2474, 2789, 4234, 6185, 7520, 8594, 10234)	98.0	94.1
	4, 8, 8	10 (205, 3879, 4234, 5654, 6185, 7520, 10138, 10234, 11091, 11730)	99.0	94.1
0.8	2, 4, 4	4 (3879, 6185, 9728, 10234)	92.2	94.1
	4, 4, 4	6 (2474, 2789, 4234, 6185, 7520, 10234)	98.0	94.1
	4, 8, 8	8 (205, 8594, 9728, 10234, 10484, 11091, 11730, 12153)	96.1	94.1
0.75	2, 4, 4	3 (6185, 10234, 12153)	89.2	94.1
	4, 4, 4	3 (3879, 10234, 12153)	91.2	94.1
	4, 8, 8	4 (205, 3879, 9728, 10234)	91.2	94.1
0.7	2, 4, 4	1 (6185)	85.3	94.1
	4, 4, 4	2 (3879, 10234)	89.2	94.1
	4, 8, 8	3 (205, 8594, 10138)	91.2	94.1

* The three numbers in a feature pool represent the number of top features selected from Entropy-based, T-statistics and SVM–RFE, respectively

reached by SVM–RFE. This testing accuracy is lower than the one obtained by the two feature subsets (with top-2 and top-4 features) selected by T-statistics. However the training accuracies of these two feature subsets from T-statistics are very low. As to entropy-based method, although it can also get 97.1% testing accuracy, it requires too many features. By reducing the value of parameter w associated with a feature pool, we can get a feature subset with smaller size. From Table 4, we can see that for a given feature pool, the accuracy is reduced as well in most cases when a smaller w is used. All the feature subsets selected by the GA from the feature pools achieve higher training accuracy than those subsets with equal or the next larger size from all the three methods.

5.3 Experiment 3

Ionosphere database [28] contains radar data collected by a system in Goose Bay, Labrador. The targets were free electrons in the ionosphere. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those that do not. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal. There are 351 instances with 34 attributes in the database. This is a binary classification task to identify “good” radar and “bad” radar.

Since the number of instances is relatively large, LOO cross-validation will take a lot of time. Therefore, we use fivefold cross-validation on this data set. The classification accuracies are shown in Table 7. As we have mentioned before, we only use linear SVM to classify the data and did not make effort to optimize the parameters of SVM. So the classification performances on this data set seem not very good in general. A SVM with nonlinear kernels may achieve better accuracies in this case. However, by using the same linear SVM, we can still get a fair comparison of the three algorithms. SVM–RFE provides the best performance in most cases. It can achieve the highest accuracy among the three by 90.31% with a feature subset of top-8 features. T-Statistics can reach its highest accuracy (87.18%) with only top-4 features.

Table 7 Fivefold CV accuracy of entropy-based, T-Statistic and SVM–RFE on ionosphere data

Top features	Entropy-based (%)	T-statistics (%)	SVM–RFE (%)
2	74.93	82.05	74.93
4	76.07	87.18	87.18
8	83.48	85.47	90.31
16	83.48	87.18	90.31
32	86.61	86.61	86.89
34	86.04	86.04	86.04

Table 8 Top-20 features from entropy-based, T-statistics, and SVM–RFE on ionosphere data

Feature selection algorithms	Top-20 features
Entropy-based	**1, 15**, 13, 19, 21, 17, **11**, 10, 14, **23**, 4, 12, 2, **6, 9, 7**, 28, **25**, 20, **5**
T-statistics	3, **5, 1, 7, 9**, 31, 33, 29, 21, 8, **15, 23**, 14, **25**, 13, **11**, 12, **6**, 16, 4
SVM-RFE	**1**, 8, **9, 23**, 34, **6**, 27, 31, 30, 19, 16, 22, **15**, 3, **11, 5**, 18, **25**, 20, **7**

The numbers in bold are the common gene(s)/feature(s) selected by three methods

Although the number of features in this data set is very small compared to the microarray data sets above, we find out that the rankings of features are still various as shown in Table 8. Here are the implementation details of the GA:

- Population size: 10
- Number of generations: 5
- Probability of crossover: 1
- Probability of mutation: 0.001

Table 9 shows the results of our approach on the data set. It suggests the same trend, that is, a smaller w reduces the size of feature subsets selected, which is demonstrated in the previous two experiments as well. In the (4, 8, 8) feature pool, the GA selects a subset of 8 features with 90.31%, which is the highest accuracy that the three methods can achieve. With the equal size of feature subset, SVM–RFE and T-Statistics can only achieve 90.03 and 85.47%, respectively. Besides, our approach finds several feature subsets of size 2, 3, and 4 with higher accuracies compared to the ones with equal or larger size selected by the three methods.

Table 9 Fivefold CV accuracy of GA on ionosphere data

w	Feature pool*	GA	Accuracy (%)
0.9	2, 4, 4	6 (1, 5, 8, 9, 15, 23)	88.60
	4, 4, 4	5 (1, 7, 8, 9, 23)	88.32
	4, 8, 8	8 (1, 3, 8, 9, 23, 27, 31, 34)	90.31
0.85	2, 4, 4	4 (1, 5, 8, 9)	87.75
	4, 4, 4	4 (1, 7, 8, 9)	88.32
	4, 8, 8	8 (1, 6, 8, 9, 23, 27, 31, 34)	90.03
0.8	2, 4, 4	3 (1, 7, 8)	88.32
	4, 4, 4	3 (1, 7, 8)	88.32
	4, 8, 8	5 (1, 5, 7, 29, 31)	88.60
0.75	2, 4, 4	2 (1, 5)	88.03
	4, 4, 4	2 (1, 5)	88.03
	4, 8, 8	4 (1, 5, 15, 34)	87.75

* The three numbers in a feature pool represent the number of top features selected from entropy-based, T-Statistics and SVM-RFE, respectively

6 Discussions and conclusions

Based on different selection criteria and theoretic arguments, various feature selection methods often provide substantially different outcomes when they are applied to same data set, which is supported in our experiments. In the experiments above, the rankings of features produced by the three feature selection methods are very different for the same data sets. This inconsistency makes it difficult to interpret the data. Although SVM–RFE shows better classification performance in the experiments, it may be prone to overfitting because it assesses features by tightly binding with the classifier (SVM). On the other hand, T-Statistics, a filter method independent of any classifier, selects features on the basis of their relevance or discriminant powers with regard to the targeted class. It may be less prone to overfitting and gives better generalization performance. Various selection criteria may be biased in different aspects, which cause difficulty in determining which method is best fit for new unknown data sets.

In this work, we propose a combinatorial approach that accommodates multiple selection criteria by a genetic algorithm. The experimental results show that our method is capable of finding feature subsets with better classification performance and/or smaller size than each single individual feature selection algorithm does. Besides this, the proposed approach can find some important features that are underrated by some individual algorithms. All of these suggest that our approach might be a viable and feasible approach for feature selection. Additional experiments with our approach are currently in progress.

The experimental results in this paper also demonstrate that selecting a number of top-ranked features does not necessarily obtain an optimal feature subset. A common drawback among feature ranking algorithms is that they implicitly assume that features are orthogonal to each other and assess features in isolation. Features are ranked on the basis of their individual predictive capabilities. Some features with highest individual performance are selected. Thus they can only detect relations between a single feature and class labels. The mutual information such as redundancy or complementariness among features is ignored. In fact, top-ranked features might be strongly related so that using two or more of them may provide little added benefit, In addition, a feature is insignificant according to some feature ranking or selection measurement can provide a significant performance improvement when grouped with other features. These are two well-appreciated issues (redundancy and multivariate prediction) that tend to confound the feature selection (Liu et al. 2005a). Some studies have addressed the redundancy and dependency among features.

In the future, we will address the correlations by incorporating the correlation-based feature selection methods into our approach. One way is to include such a method to construct the feature pool for the genetic algorithm. Another way is to design a fitness function that considers the correlations between features.

Acknowledgments This work is supported in part by NIH under Grant P20 GM065762.

References

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

Breiman L, Forest R Technical Report. Stat. Dept, UCB

Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data mining Knowl Dis 2(2):121–167

Chuang H-Y et al (2004) Identifying significant genes from microarray data. Fourth IEEE symposium on bioinformatics and bioengineering (BIBE'04) p. 358

Dash M, Liu H (1999) Handling large unsupervised data via dimensionality reduction. ACM SIGMOD workshop on research issues in data mining and knowledge discovery

Dy JG, Brodley CE (2004) Feature selection for unsupervised learning. J Mach Learn Res 5:845–889

Furey T, Cristianini N, Bednarski DN, Schummer DM (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16:906–914

Golub TR et al (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection (Kernel Machines Section): JMLR 3:1157–1182

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer series in statistics. Springer, New York

Hsu FD, Shapiro J, Taksa I (2002) Methods of data fusion in information retreival: Rank vs. Score combination. DIMACS Technical report 58

Jirapech-Umpai T, Aitken S (2005) Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. BMC Bioinform 6:148

LeCun Y, Denker JS, Solla SA (1990) Optimum brain damage. Touretzky DS (ed) Advances in neural information processing systems II, Morgan Kaufmann, Mateo

Liu Y (2004) A comparative study on feature selection methods for drug discovery. J Chem Inform Comput Sci 44(5):1823–1828

Liu H, Setiono R (1995) χ^2: feature selection and discretization of numeric attributes. In: Proceedings IEEE 7th international conference on tools with artificial intelligence, pp 338–391

Liu H, Li J, Wong L (2002) A comparative study on feature selection and classification methods using gene expression profiles and proteomic pattern. Genom Inform 13:51–60

Liu H et al. (2005) Evolving feature selection. Intelligent systems, IEEE Vol 20(6), pp 64–76

Liu X, Krishnan A, Mondry A (2005) An entropy-based gene selection method for cancer classification using microarray data. BMC Bioinform 6:76

Mao Y, Zhou X, Pi D, Sun Y, STC Wong (2005) Multiclass cancer classification by using fuzzy support vector machine and binary decision tree with gene selection. J Biomed Biotechnol (2):160–171

Noble WS (2004) Support vector machine applications in computational biology.In: Schoelkopf B, KTsuda, Vert J.-P (eds) Kernel methods in computational biology. MIT, New York, pp 71–92

Peng HC, Long FH, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Patt Anal Mach Intell 27(8):1226–1238

Schölkopf B, Guyon I, Weston J (2003) Statistical learning and kernel methods in bioinformatics. In: Frasconi P, Shamir R (eds) Artificial intelligence and heuristic methods in bioinformatics. vol 183. IOS Press, Amsterdam, pp 1–21

Singh D (2002) Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1:203–209

Singh D et al (2002) Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1:203–209

Somorjai RL, Dolenko B, Baumgartner R (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. Bioinformatics. 12; 19(12):1484–91

Space Physics Group; Applied Physics Laboratory; Johns Hopkins University; Johns Hopkins Road; Laurel; MD 20723

Vapnik V (1998) Statistical learning theory. Wiley, New York

Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. IEEE Intell Syst 13:44–49

Yu L, Liu H (2003) Efficiently handling feature redundancy in high-dimensional data. In: Proceedings of ACM SIGKDD international conference knowledge discovery and data mining (KDD 03), ACM, New york, pp. 685–690

Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2000) Feature Selection for SVMs. Adv Neural Inform Process Syst 13