ISSN:0975-9646

Filter versus Wrapper Feature Subset Selection in Large Dimensionality Micro array: A Review

Binita Kumari[#1], Tripti Swarnkar[*2]

*[#1]Department of Computer Science - [*2]Department of Computer Applications, ITER, SOA University*
Orissa, INDIA

Abstract— **The microarray can track the expression levels of thousands of genes simultaneously. The high dimensional feature vectors of microarray impose a high dimensional cost as well as the risk of overfitting during classification. Thus it is necessary to reduce the dimension through ways like feature selection. Two basic approaches of feature selection appear: filter and wrapper techniques.**

In this paper, we make the people aware of the various techniques of feature selection.

Keywords— **Microarray, Feature Selection, Filter, Wrapper, Classification.**

I. Introduction

Accurate disease diagnosis is vital for the successful application of specific treatments. The DNA microarray technology is providing great opportunities in reshaping the biomedical science. An orderly and computational analysis of microarray datasets is a motivating way to study and understand many aspects of underlying biological process. Parallel to these technological advances has been the development of machine learning methods to analyse and understand the data generated by this new kind of experiments. The analysis involves class prediction (supervised classification), regression, feature selection, principal component analysis, outlier detection, discovering of gene relationships and cluster analysis (unsupervised classification) [1,3].

For most biological problems, information about type (class) of each cell line exists indicating whether the tissue is diseased or healthy. By means of the interesting class information, the DNA microarray analysis can be formulated as a classic supervised classification task.

Feature selection can be applied to both supervised and unsupervised learning; we focus here on the problem of supervised learning (classification), where the class labels are known beforehand.

A DNA microarray is a multiplex technology which is being used in molecular biology which consists of an arrayed series of thousands of spots of DNA which are called features. Microarray technology is used to study the expression of many genes at a time. The high dimensional [2,5] feature vectors of microarray data often impose a high computational cost as well as the risk of "overfitting" at the time of classification. Thus it is necessary to reduce the dimensionality through ways like feature selection.

A microarray chip or data can be analyzed as shown in figure 1.First the microarray dataset is normalized so that there are no missing values and the data is scaled between a specific range. Then feature selection is done as a result of which we get the key genes. Then the classification or clustering is done and the output is interpreted to get the required biological information

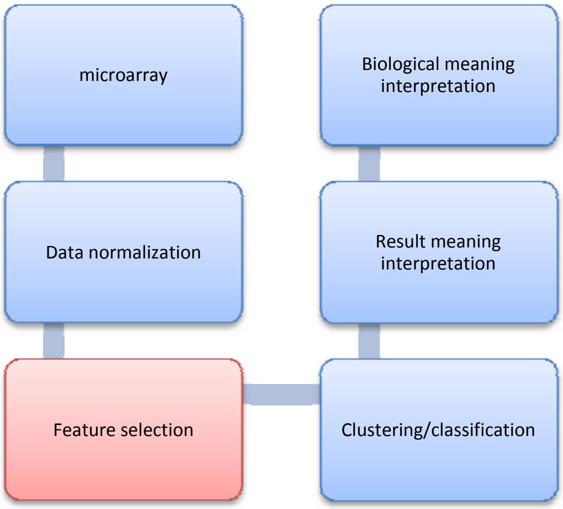

microarray
Data normalization
Feature selection
Clustering/classification
Result meaning interpretation
Biological meaning interpretation

Fig.1 Microarray chip analysis

The selection of relevant features and elimination of irrelevant ones is a great problem. Before an induction algorithm can be applied to a training dataset to make decisions about test cases, it must decide about which attributes to be selected and which to be ignored.

Irrelevant features increase the measurement cost, decrease the classification accuracy and add to making the computation complex. Obviously, one would like to use only those attributes that are relevant to the target concept.

The rest of the paper is organized as follows: a brief review of the existing techniques of filter feature selection and wrapper feature selection, classifiers used in section II, comparative results in section III followed by conclusion.

II. REVIEW OF EXISTING TECHNIQUES

Feature selection (also known as subset selection) entails choosing the feature subset that maximizes the prediction or classification accuracy. The best subset contains the least number of features that most contribute towards accuracy.

A. Feature Selection

Feature selection (also known as subset selection) is a process commonly used in machine learning, where a subset of features is selected from the available data for application of a learning algorithm [5]. So we prefer the model with the smallest possible number of parameters that adequately represent the data. Selecting the best feature subset is a NP complete problem. The task is challenging because first, the features which do not appear relevant singly may be highly relevant when taken with other features. Second, relevant features may be redundant so that omission of some of them will remove unnecessary complexity. An exhaustive search of all possible subsets of features will guarantee the best feature subset. The best subset contains the least number of features that most contribute towards accuracy.

There are two approaches of feature selection [10]:
Forward selection:
(i)Start with no variables.(ii)Add the variables one by one, at each step adding the feature that has the minimum error.(iii)Repeat the above step until any further addition does not signify any decrease in error.

Backward selection:
(i)Start with all variables.(ii)Remove the variables one by one, at each step removing the feature that has the highest error.(iii)Repeat the above step until any further removal increases the error significantly

The two broad categories of feature subset selection have been proposed: filter and wrapper [4,5]. Filter techniques assess the relevance of features by looking at the intrinsic properties of the data. In filter criteria, all the features are scored and ranked based on certain statistical criteria. The features with the highest ranking values are selected and the low scoring features are removed.. Filter methods (fig 2) are fast and independent of the classifier but ignore the feature dependencies and also ignores the interaction with the classifier. They also easily scale to very high-dimensional dataset. As a result feature selection need to be done only once and then different classifiers can be evaluated. The common disadvantage of filter methods is that they ignore the interaction with the classifier and each feature is considered independently thus ignoring feature dependencies In addition, it is not clear how to determine the threshold point for rankings to select only the required features and exclude noise.

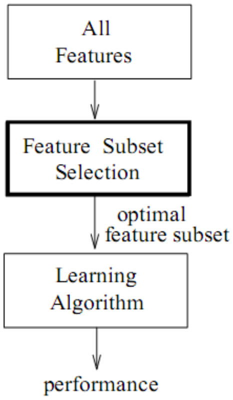


Fig.2 The feature filter approach

Wrapper methods embed the model hypothesis search within within the feature subset search. In this setup, a search procedure in the space of possible feature subsets is defined, and various subsets of features are generated and evaluated. The evaluation of a specific subset of features is obtained by training and testing a specific classification model, rendering this approach tailored to a specific classification algorithm. To search the space of all feature subsets, a search algorithm is then 'wrapped' around the classification model. However, as the space of feature subsets grows exponentially with the number of features, heuristic search methods are used to guide the search for an optimal subset.

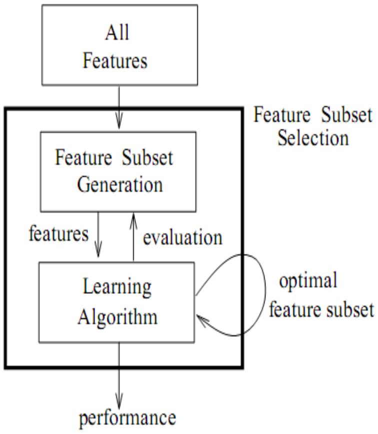


Fig.3 The feature Wrapper approach

Thus feature selection [4,6] is of considerable importance in classification as it (i)Reduces the effects of curse of dimensionality(ii)Helps in learning the model(iii)Minimizes cost of computation(iv)Helps in achieving good accuracy

Model	Advantages	Disadvantages	Examples
Filter	Fast, Scalable, Independent of classifier, Better computational complexity	Ignores interaction with classifier	Chi-square, Euclidean distance, i-test, Information gain, Correlation based feature selection, Markov blanket filter, Fast correlation based feature selection
Wrapper	Simple, Interacts with classifier, Models feature dependencies, Good classification accuracy, Minimizes computational cost	Computationally intensive	Sequential forward selection, Sequential backward selection, Randomized hill climbing, Genetic algorithms

Table 1. The advantages and disadvantages of filter and wrapper approach

B Algorithms

Feature filter algorithms

There are many filter algorithms. Some are described as follows:

(1) $\chi 2$-**Statistic**: This criterion measures the worth of a feature by computing the value of the $\chi 2$ statistic[7,9] with respect to the class.

(2) **Information gain**: This criterion measures the worth of a feature by measuring the information gain with respect to the class. Information gain is given by

InfoGain = $H(Y) - H(Y|X)$,

where X and Y are features

Both, the information gain and the $\chi 2$ statistic, are biased in favor of features with higher dispersion.

(3) **Symmetrical uncertainty**: This criterion measures the worth of a feature by measuring the symmetrical uncertainty with respect to the class, and compensates for information gain's bias.

SU = 2.0 ×InfoGain/($H(Y) + H(X)$)

.

(4) **ReliefF**: This is a feature weighting algorithm that is sensitive to feature interactions. The key idea of ReliefF is to rate features according to how well their values distinguish among instances of different classes and to how well they cluster instances of the same class. To this end, ReliefF repeatedly chooses a single instance at random from the data, and then locates the nearest instances of the same class and the nearest instances pertaining to different classes. The feature values of these instances are used to update the scores for each feature

Classification algorithms

In this study we use three well-known classifiers, namely the decision tree learner C4.5, the simple Bayesian classifier naïve Bayes, and a support vector machine (SVM) (Vapnik, 1998) to demonstrate the advantages and disadvantages of feature selection algorithms. For a more thorough discussion of the first two algorithms and the corresponding feature selection methods, we refer to (Witten and Frank, 1999; Hall, 1999). Decision trees have been popular in practice due to their simplicity, fast evaluation speed, and interpretability. The training of decision trees directly on high dimensional microarray cancer data can sometimes overfit the data, generating an overly large tree. Removing irrelevant and redundant information results in smaller, more predictive trees. naıve Bayes assumes that features are independent given the class. Its performance on data sets with redundant features can be improved by removing such features. A forward search strategy is normally used with naıve Bayes[3,4] as it should immediately detect dependencies when harmful redundant features are added.

SVMs use a kernel function to implicitly map data to a high dimensional space. Then, they construct the maximum margin hyperplane by solving an optimization problem on the training data. Sequential minimal optimization (SMO) (Platt, 1998) is used in this paper to train an SVM[9]. SVMs have been shown to work well for high dimensional microarray data sets (Furey et al., 2000). However, due to the high computational cost it is not very practical to use the wrapper method to select genes for SVMs, as will be shown in our experimental results section.

Correlation-based feature selection

CFS evaluates a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them .

$\mathrm{CFS}_S = kr_{\mathrm{cf}}/\surd k + k(k - 1)r_{\mathrm{ff}}$

where CFS_S is the score of a feature subset S containing k features, r_{cf} is the average feature to class correlation ($\mathrm{f} \in S$), and r_{ff} is the average feature to feature correlation. The distinction between normal filter algorithms and CFS is that while normal filters[3,5,9] provide scores for each feature independently, CFS presents a heuristic "merit" of a feature subset and reports the best subset it finds.

C. Classifiers Used

The classifiers used can broadly be classified as: Support vector based classification methods and Non support vector based classification methods [2,3].

Support Vector Machine based classification methods:
Support vector machines (SVMs) (Vapnik, 1998) are perhaps the single most important development in supervised classification of recent years. SVMs often achieve superior classification performance compared to other learning algorithms across most domains and tasks; they are fairly insensitive to the curse of dimensionality and are efficient enough to handle very large-scale classification in both sample and variables. In clinical bioinformatics, they have allowed the construction of influential experimental cancer diagnostic models based on gene expression data with thousands of variables and as little as few dozen samples. Moreover, several efficient and high quality implementations of SVM algorithms (e.g. Joachims, 1999; Chang and Lin, 2003, http://www.csie.ntu.edu.tw/~cjlin/libsvm) facilitate application of these techniques in practice. The first generation of SVMs were limited to binary classification tasks. But, most real-life diagnostic tasks are not binary. Moreover, all other things being equal, multicategory classification is significantly harder than binary classification. Fortunately, several algorithms have emerged during the last few years that allow multicategory classification with SVMs.

Binary SVMs: The main idea of binary SVMs [6] is to implicitly map data to a higher dimensional space via a kernel function and then solve an optimization problem to identify the maximum-margin hyperplane that separates training instances. The hyperplane is based on a set of boundary training instances, called *support vectors* [7]. New instances are classified according to the side of the hyperplane[2] they fall into. The optimization problem is most often formulated in a way that allows for non-separable data by penalizing misclassifications.

Multiclass SVMs: one-versus-rest (*OVR*) This is conceptually the simplest multiclass SVM method. Here, *k* binary SVM classifiers are constructed: class 1 (positive) versus all other classes (negative), class 2 versus all other classes, . . ., class *k* versus all other classes. The combined OVR [2] decision function chooses the class of a sample that corresponds to the maximum value of *k* binary decision functions specified by the furthest 'positive' hyperplane. By doing so, the decision hyperplanes calculated by *k* SVMs 'shift', which questions the optimality of the multicategory classification. This approach is computationally costly, since we need to solve *k* quadratic programming (QP) optimization problems of size *n*. Moreover, this technique does not currently have theoretical justification such as the analysis of generalization, which is a relevant property of a robust learning algorithm

Multiclass SVMs: one-versus-one (*OVO*) This method involves the construction of binary SVM [2] classifiers for all pairs of classes; in total there are _*k*2 _ = [*k(k* − 1*)*]/2 pairs . other words, for every pair of classes, a binary SVM problem is solved (with the underlying optimization problem to maximize the margin between two classes). The decision function assigns an instance to a class that has the largest number of votes, so-called *Max Wins strategy*. If ties still occur, each sample will be assigned a label based on the classification provided by the furthest hyperplane.

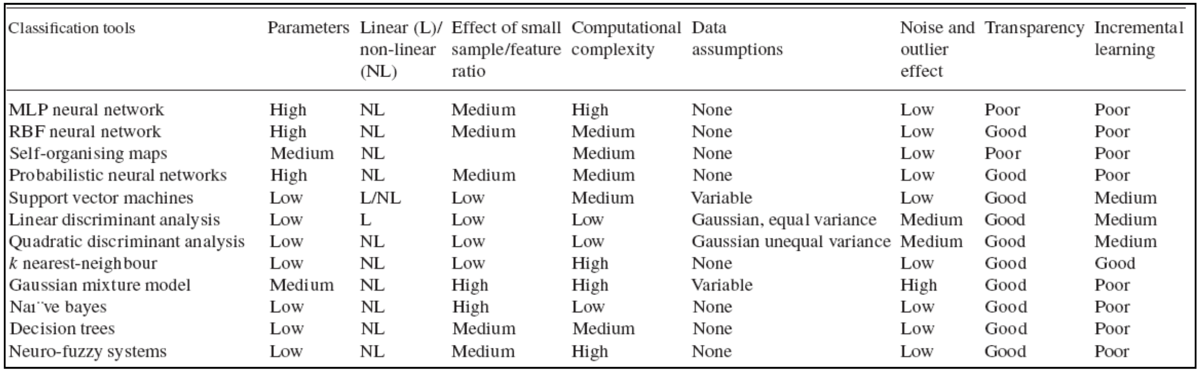
Multiclass SVMs: DAGSVM The training phase of this algorithm is similar to the OVO approach using multiple binary SVM classifiers; however, the testing phase of DAGSVM requires the construction of a rooted binary decision directed acyclic graph (DDAG) using _*k*2_ classifiers. Each node of this graph is a binary SVM for a pair of classes, say (*p*, *q*). On the topologically lowest level there are *k* leaves corresponding to *k* classification[3,6] decisions. Every non-leaf node (*p*, *q*) has two edges—the left edge corresponds to decision 'not *p*' and the right one corresponds to 'not *q*'. The choice of the class order in the DDAG list can be arbitrary as shown empirically in Platt *et al*. (2000). In addition to inherited advantages from the OVO method, DAGSVM is characterized by a bound on the generalization error.

Non-Support vector machine based classification methods:
In addition to five MC-SVM [2] methods, three popular classifiers, *K*-nearest neighbors (KNNs), backpropagation neural networks (NNs) and probabilistic neural networks (PNNs), are also used. These learning methods have been extensively and successfully applied to gene expression-based cancer diagnosis [2].

K-nearest neighbours: The main idea of KNN is that it treats all the samples as points in the *m*-dimensional space (where *m* is the number of variables) and given an unseen sample *x*, the algorithm classifies it by a vote of *K*-nearest training instances as determined by some distance metric, typically Euclidean distance.

Backpropagation neural networks: NNs are feed-forward neural networks with signals propagated only forward through the layers of units. These networks are comprised of (1) an input layer of units, which we feed with gene expression data; (2) hidden layer(s) of units; and (3) an output layer of units, one for each diagnostic category, so-called *1-of-n encoding*. The connections among units have weights and are adjusted during the training phase (epochs of a neural network) by backpropagation learning algorithm. This algorithm adjusts weights by propagating the error between network outputs and true diagnoses backward through the network and employs gradient descent optimization to minimize the error function. This process is repeated until a vector is found of weights that best fits the training data. When training of a neural network is complete, unseen data instances are fed to the input units, propagated forward through the network and the network outputs classifications.

Probabilistic neural networks: PNNs [2] belong to the family of Radial Basis Function neural networks which are feed-forward neural networks with only one hidden layer. The primary difference between an NN with one hidden layer and an RBF network is that for the latter one, the inputs are passed directly to the hidden layer *without weights*. The Gaussian density function is used in a hidden layer as an activation function.A key advantage of RBF networks is that they are trained much more efficiently than NNs.



Classification tools	Parameters	Linear (L)/ non-linear (NL)	Effect of small sample/feature ratio	Computational complexity	Data assumptions	Noise and outlier effect	Transparency	Incremental learning
MLP neural network	High	NL	Medium	High	None	Low	Poor	Poor
RBF neural network	High	NL	Medium	Medium	None	Low	Good	Poor
Self-organising maps	Medium	NL		Medium	None	Low	Poor	Poor
Probabilistic neural networks	High	NL	Medium	Medium	None	Low	Good	Poor
Support vector machines	Low	L/NL	Low	Medium	Variable	Low	Good	Medium
Linear discriminant analysis	Low	L	Low	Low	Gaussian, equal variance	Medium	Good	Medium
Quadratic discriminant analysis	Low	NL	Low	Low	Gaussian unequal variance	Medium	Good	Medium
k nearest-neighbour	Low	NL	Low	High	None	Low	Good	Good
Gaussian mixture model	Medium	NL	High	High	Variable	High	Good	Poor
Naı¨ve bayes	Low	NL	High	Low	None	Low	Good	Poor
Decision trees	Low	NL	Medium	Medium	None	Low	Good	Poor
Neuro-fuzzy systems	Low	NL	Medium	High	None	Low	Good	Poor

Table 2. The properties of classification tools

D Experimental procedure

Selecting genes using feature-ranking filters
(1) Use a filter to rank all the genes in the data.

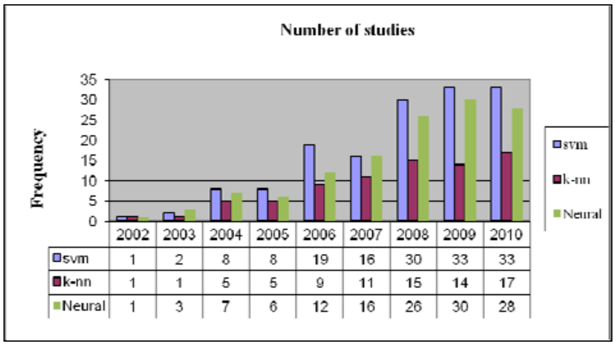
(2) Choose the first $n - 1$ genes as the best feature subset.

Note that the data has to be discretized before $\chi 2$, information gain and symmetrical uncertainty filters can be applied.

Selecting genes using a wrapper method
(1) Choose a machine learning algorithm[2,4] to evaluate the score of a feature subset.
(2) Choose a search algorithm.
(3) Perform the search, keeping track of the best subset encountered.
(4) Output the best subset encountered.

III. RESULTS

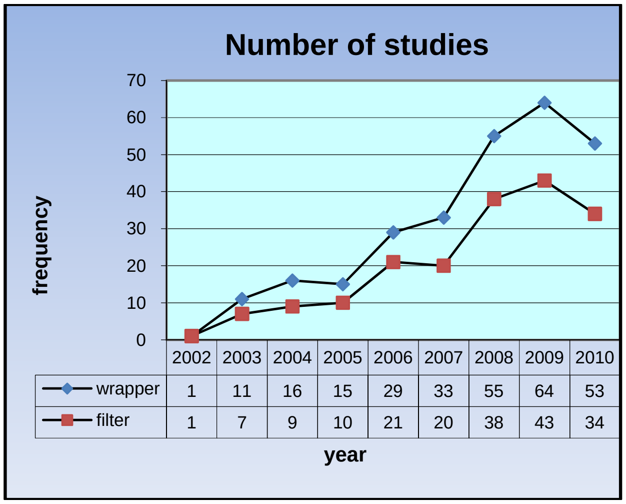
In the last few years the use of wrapper methods has increased a lot in the field of classification. In most of the wrapper methods support vector machine has been used as compared to other classifiers because of its classification accuracy as shown in fig 4.



	2002	2003	2004	2005	2006	2007	2008	2009	2010
svm	1	2	8	8	19	16	30	33	33
k-nn	1	1	5	5	9	11	15	14	17
Neural	1	3	7	6	12	16	26	30	28

Fig.4 frequency of use of SVM, K-NN, Neural Network

The frequency of use of wrapper and filter approach for last few years has been shown in fig.5.



	2002	2003	2004	2005	2006	2007	2008	2009	2010
wrapper	1	11	16	15	29	33	55	64	53
filter	1	7	9	10	21	20	38	43	34

Fig.5 frequency of use of feature wrapper and filter approach

IV. CONCLUSIONS

We have shown in this paper that feature selection algorithms, namely wrappers and filters are very useful in extracting useful information in microarray data analysis. Filter approaches could be recommended for fast data analysis. However, in order to better validate the results and to select few genes wrapper approaches could be recommended. Wrapper approaches can choose the best genes for building classifiers. This is the reason for the increased use of wrapper method in last few years.

Amongst the classifiers, we conclude that support vector machines are widely used because it can achieve superior classification performance compared to other learning algorithms across most domains and tasks; they are fairly insensitive to the curse of dimensionality and are efficient enough to handle very large-scale classification in both sample and variables.

REFERENCES

[1] In˜aki Inza, Pedro Larran˜aga, and Rosa Blanco, Antonio J. Cerrolaza, "Filter versus wrapper gene selection approaches in DNA microarray domains," , in *Proc Artificial Intelligence in Medicine* ,31(2004) 91—103

[2] Alexander Statnikov, Constantin F. Aliferis, Ioannis Tsamardinos, Douglas Hardin, Shawn Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis", in *Proc. Bioinformatics,* 21 (2005) 631 -- 643

[3] Yu Wanga, Igor V. Tetkoa, Mark A. Hallb, Eibe Frankb, "Gene Selection from microarray data for cancer classification-a machine learning approach", in *Proc. Computational Biology and Chemistry,* 29 (2005) 37–46

[4] AlanWee-Chung Liew, HongYan, MengsuYang, "Pattern recognition techniques for the emerging field of bioinformatics: A review" , in *Proc Pattern Recognition,* 38 (2005) 2055 – 2073.

[5] Ian B Jeffery, Desmond G Higgins, and Aedín C Culhane, "Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data," , in *Proc BMC Bioinformatics,* 7:359 (2006) 1471 -- 2105.

[6] Yvan Saeys, In˜ aki Inza, and Pedro Larran˜ aga, "A review of feature selection techniques in bioinformatics," , in *Proc BMC Bioinformatics,* 23 (2007) 2507 -- 2517.

[7] Harish Bhaskar, David C. Hoyle, Sameer Singh, "Machine learning in bioinformatics: A brief survey and recommendations for practitioners", in *Proc. Computers in Biology and Medicine,* 36 (2006) 1104 – 1125

[8] Jianping Hua, Waibhav D. Tembe, and Edward R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," , in *Proc Pattern Recognition,* 42 (2009) 409 -- 424.

[9] Iffat A.Gheyas, Leslie S.Smith, "Feature subset selection in large dimensionality domains", in *Proc. Pattern Recognition,* 43 (2010) 5 -- 13