# Gene selection by sequential search wrapper approaches in microarray cancer class prediction

Iñaki Inza, Basilio Sierra, Rosa Blanco and Pedro Larrañaga

Department of Computer Science and Artificial Intelligence
University of the Basque Country
P.O. Box 649
E-20080 Donostia-San Sebastián
Basque Country, Spain
Tel: (+34) 943015026
Fax: (+34) 943219306
e-mail: inza@si.ehu.es

**Abstract:** In the last years, there has been a large growth in gene expression profiling technologies, which are expected to provide insight into cancer related cellular processes. Machine Learning algorithms, which are extensively applied in many areas of the real world, are not still popular in the Bioinformatics community. We report on the successful application of four well known supervised Machine Learning methods (IB1, Naive-Bayes, C4.5 and CN2) to cancer class prediction problems in three DNA microarray datasets of huge dimensionality (*Colon, Leukemia* and *NCI-60*). The essential gene selection process in microarray domains is performed by a sequential search engine, evaluating the goodness of each gene subset by a wrapper approach which executes, by a leave-one-out process, the supervised algorithm to obtain its accuracy estimation. By the use of the gene selection procedure, the accuracy of supervised algorithms is significantly improved and the number of genes of the classification models is notably reduced for all datasets.

## 1. Introduction

A right and accurate cancer classification allows to the medical staff the application of specific therapies and treatments, related with the specific cancer type [14]. Although cancer classification has been improved over the past three decades, it has been traditionaly based on the morphological appearance of the tumor. However, this non-automatic and systematic classification has obvious and serious limitations, so related with human errors and interpretations. It is well known that similar appearances can follow significantly different clinical courses and can belong to different cancer types, showing non-equal responses to the same therapy or treatment. In this way, a systematic and unbiased approach for an accurate recognition and classification of different tumor types is highly desired in medical environments.

Development of high throughput data acquisition technologies in biological sciences, and specificaly in genome sequencing, together with advances in digital storage and computing, have begun to transform biology, from a data poor science to a data rich science. In order to manage and treat with all this new biological data, the Bioinformatics discipline powerfully emerges.

These advances in the last decade in genome sequencing have lead to the spectacular development of a new technology, named *DNA microarray*, which can be included into the Bioinformatics discipline. DNA microarray allows the monitoring and measurement of the expression levels of thousands of genes simultaneously in an organism. A systematic and computational analysis of these microarray datasets is an interesting way to study and understand many aspects of the underlying biological processes.

There has been a significant recent interest in the development of new methods for functional interpretation of these microarray gene expression datasets. The analysis [6] frequently involves class prediction (supervised classification), regression, feature selection (in this case, gene selection), outlier detection, principal component analysis, discovering of gene relationships and cluster analysis (unsupervised classification). In this way, DNA microarray datasets are an appropiate starting point to carry out the explained systematic and automatic cancer classification [14].

Cancer classification is divided in two major issues: the discovery of previously unknown types of cancer (class discovery) and the assignment of tumor samples to already known cancer types (class prediction). While the class discovery is related with the cluster analysis (or unsupervised classification), class prediction is related with the application of supervised classification techniques.

Our work is focused on class prediction for cancer classification. Starting from a set of samples for which the classification (the class type) is known, we tackle the construction of the predictive model which discriminates among the different cancer types of the problem. Thus, our goal is the maximization of the predictive accuracy of built models.

In the last decade there has been a big growth in the accumulation of information in Economic, Marketing, Medicine, Finance, etc. databases. The larger size of these databases and the improvement of computer related technologies inspired the development of a set of techniques that are grouped under the *Machine Learning* (ML) [13, 30] term and that discover and extract knowledge in an automated way. By approaching an analysis as a search for knowledge, the discovery of previously unknown relationships in the data is the core of the procces for the induction of the classifier. Although ML techniques have successfully solved classification problems in many different areas of the real world, its application is nowadays emerging as a powerful tool to solve DNA microarray problems.

For our purpose of cancer class prediction, we propose the use of four well known ML supervised classification algorithms with completely different approaches to learning and a long tradition in different classification tasks: IB1, Naive-Bayes, C4.5 and CN2. We have experimented them in three well known DNA cancer classification microarray datasets.

However, it is well known that the accuracy of supervised ML methods is not monotonic regarding the inclusion of features [21]: irrelevant or redundant features, depending on the specific characteristics of the classifier, may degrade the accuracy of the classification model. In this sense, given the entire set of features, we aim to find the feature subset with the best predictive accuracy for a certain classifier. This problem is known in the ML community as the Feature Subset Selection problem and it has been tackled with success in so different types of problems [27].

As microarray datasets have a very large number of predictive genes (usually more than $1,000$) and a small number of samples (usually less than 100), a reduction in the number of genes to build a classifier is an essential part of any microarray study. This makes it always possible to find a perfect discriminator if we are not careful in restricting the complexity of the classification model [6]. To avoid this problem, it is mandatory to look for very simple classifiers with few predictive genes, compromising between simplicity and accuracy. Moreover, for diagnostic purposes it is important to find small subsets of genes that are sufficiently informative to distinguish between cells of different types [4]. All the studies also show that the major part of the genes measured in a DNA microarray are not relevant for an accurate distinction between cancer classes [14]. To this end we suggest a simple combinatorial, sequential, classic search mechanism, named Sequential Forward Selection [18], which performs the major part of its search near the empty subset of genes. Each found gene subset is evaluated by a wrapper [21] scheme: once the classification algorithms is fixed, it is trained with the gene subset encountered by the sequential search algorithm, estimating the error percentage, and assigning it as the value of the evaluation function of the gene subset.

The wrapper approach, which is very popular in ML applications, is not extensively used in DNA microarray tasks and few works of the field make use of it [5, 25, 35]: the work [5] with the Naive-Bayes classifier and [25] with K-NN. The work [35] does not perform a search procedure by a wrapper scheme, and it only uses the wrapper approach to decide among subsets of genes of different cardinalities that were previously found for Gaussian classifier, Logistic regression and K-NN. As far as we know, C4.5 and CN2 have never been used in a wrapper framework in microarray domains. This is the first work where a systematic search-wrapper experimentation for a set of four ML classifiers in DNA microarray domains is carried out. In this way, our aim is to make contributions in the expansion of the wrapper scheme in this kind of domains.

The rest of the paper is organized as follows. The supervised classifiers included in the study an(usually more than $1,000$) d the method for the selection of genes are described in the next section. Section 3 surveys other class prediction approaches in DNA microarray datasets. Experimental results are shown in Section 4. The last section briefly summarizes the work and presents ways for future research in the field.

## 2. Learning supervised classifiers by Feature Subset Selection

### 2.1 Machine Learning classifiers

In our study, four well known ML supervised classifiers, with completely different approaches to learning, are applied to perform the class prediction in microarray cancer datasets. All the algorithms are selected due to their simplicity and their long standing tradition in classification studies. To understand the following explanations about the characteristics of the algorithms, it must be taken into account that for microarray datasets, the predictive genes are the features (or variables) of the problem and the cell samples are represented by the instances or cases. Note also that microarray datasets do not have missing values and all the gene numeric values are continuous.

The IB1 [1] is a case-based, Nearest-Neighbor classifier. To classify a new test sample, all training instances are stored and the nearest training instance regarding the test instance is found: its class is retrieved to predict this as the class of the test instance. As the variables of microarray datasets have continuous values, the measure of the distance between two samples is performed using the euclidean metric.

The Naive-Bayes (NB) rule [7] uses the Bayes theorem to predict the class for each case, assuming that the predictive genes are independent given the category. To classify a new sample characterized by $d$ genes $\mathbf{X} = (X_1, X_2, ..., X_d)$, the NB classifier applies the following rule:

$$c_{N-B} = \arg\max_{c_j \in C} p(c_j) \prod_{i=1}^{d} p(x_i|c_j)$$

where $c_{N-B}$ denotes the class label predicted by the Naive-Bayes classifier and the possible classes of the problem are grouped in $C = \{c_1, \ldots, c_l\}$. A normal distribution is assumed to estimate the class conditional densities for predictive genes. Despite its simplicity, the NB rule has obtained better results than more complex algorithms in many domains.

The C4.5 [32] represents a classification model by a decision tree. It is run with the default values of its parameters. The tree is constructed in a top-down way, dividing the training set and beginning with the selection of the best variable in the root of the tree. The selection of the best feature is performed by the maximization of a splitting criterio which is based on an informatic theoretic approach. For each continuous attribute, a threshold that maximizes the splitting criterion is found by sorting the cases of the dataset on their values of the attribute: every pair of adjacent values suggest a threshold in their midpoint, and the threshold that yields the best value of the splitting criterion is selected. A descendant of the root node is then created for each possible value of the selected feature, and the training cases are sorted to the appropiate descendant node. The entire process is then recursively repeated using the training cases associated with each descendant node to select the best feature to test at that point in the tree. The process stops at each node of the tree when all cases in that point of the tree belong to the same category or the best split of the node does not surpass a fixed chi-square significancy threshold. Then, the tree is simplified by a pruning mechanism to avoid overspecialization.

The CN2 [8] algorithm represents a classification model by a set of IF-THEN rules, where the THEN part represents the class predicted for the samples that match the conditions of the IF part. It is run with the default values of its parameters. CN2 is based on an information theoretic approach with a significance metric to improve rule quality and to avoid overspecialization of the results. When a significant rule is found, CN2 removes those examples it covers from the training set and adds the rule to the end of the rule list. To use induced rules to classify test examples, CN2 tries each rule in order until one is found whose conditions are satisfied by the example being classified. If no induced rules are satisfied, the final default rule assigns the most common class in the training set to the test case. In a similar way to C4.5, CN2 sorts the cases in the dataset on their values of a continuous attribute, and every threshold in the midpoint of each pair of adjacent values is considered to construct the IF-THEN rules. The difference with respect to C4.5 is that CN2 only considers the split points with different associated class labels.

Even though a decision tree can be converted into a set of IF-THEN rules, while CN2 rules are independent to each other, C4.5 rules are dependent on each other. It must be noted that C4.5 and CN2, by their own, can discard some of the presented features to build their classification models. On the other hand, IB1 and NB include all the presented variables in the model.

Apart from the prediction accuracy, the explanation ability of the classifier is also very important. To support the diagnostic process in everyday practice, physicians and biologists need a classifier that is able to explain its decisions, being such transparent decisions much more acceptable by them. For this reason, other promising techniques, such as neural nets, are not included among our classification models, due to their low human-transparency [17, 29].

Due to the low number of samples of microarray datasets, the *leave-one-out cross-validation* (LOOCV) procedure [19], a special case of $k$-fold cross-validation, is used to perform the accuracy estimation. In the leave-one-out technique, the supervised algorithm is run $k$ times, where $k$ is the number of examples of the dataset. Each time $k-1$ examples are used for training and the remaining example is used for testing, where each example is used only once for testing. The LOOCV estimate of accuracy is the overall number of correct classifications, divided by $k$, the number of examples in the dataset. LOOCV estimation is almost unbiased from the real accuracy [19] and it is considered as the most reliable estimator. However, its computational cost can only be assumed, as in our case, when few examples are presented.

### 2.2 Gene selection process: Feature Subset Selection

The basic problem of ML is concerned with the induction of a model that classifies a given object into one of several known classes. In order to induce the classification model, each object is described by a pattern of $d$ features. Here, the ML community has formulated the following question: *are all of these d descriptive features useful for learning the 'classification rule'?* On trying to respond to this question, we come up with the Feature Subset Selection (FSS) [27] approach which can be reformulated as follows: *given a set of candidate features, select the 'best' subset in a classification problem.* In our case, the 'best' subset will be the one with the best predictive accuracy.

Most of the supervised learning algorithms perform rather poorly when faced with many irrelevant or redundant (depending on the specific characteristics of the classifier) features. In this way, the FSS proposes additional methods to reduce the number of features so as to improve the performance of the supervised classification algorithm.

FSS can be viewed as a search problem [23], with each state in the search space specifying a subset of the possible features of the task. Exhaustive evaluation of possible feature subsets is usually unfeasible in practice due to the large amount of computational effort required. In this way, any feature selection method must determine four basic issues that define the nature of the search process:

*1. The starting point in the space.* It determines the direction of the search. One might start with no features and successively add them, or one might start with all features and successively remove them. One might also select an initial state somewhere in the middle of the search space.

*2. The organization of the search.* It determines the strategy of the search. Roughly speaking, the search strategies can be *complete* or *heuristic* (see [27] for a review of FSS algorithms). When we have more than 10-15 features the search space becomes huge and a *complete* search strategy is unfeasible. As FSS is a classic NP-hard optimization problem, the use of search *heuristics* is justified. Among *heuristic* algorithms, there are *deterministic heuristic* and *non-deterministic heuristic* algorithms. Classic deterministic heuristic FSS algorithms are sequential forward and backward selection (SFS and SBS [18]), floating selection methods (SFFS and SFBS [31]) and best-first search [21]. They are deterministic in the sense that all the runs over the same data always obtain the same solution. *Non-deterministic heuristic* search appears in a motivation to avoid getting stuck in local maximum. Randomness is used to escape from local maximum and this implies that one should not expect the same solution from different runs. Two classic implementations of non-deterministic search engines are Genetic Algorithms [34], Estimation of Distribution Algorithms [16] and Simulated Annealing [11].

*3. Evaluation strategy of feature subsets.* The evaluation function identifies the promising areas of the search space. The objective of FSS algorithm is its maximization. The search algorithm uses the value returned by the evaluation function for helping to guide the search. Some evaluation functions carry out this objective looking only at the characteristics of the data, capturing the relevance of each feature or set of features to define the target concept: these type of evaluation functions are grouped below the *filter* strategy. However, Kohavi and John [21] report that when the goal of FSS is the maximization of the accuracy, the features selected should depend not only on the features and the target concept to be learned, but also on the learning algorithm. Thus, they proposed the *wrapper* concept: this implies that the FSS algorithm conducts a search for a good subset using the induction algorithm itself as a part of the evaluation function, the same algorithm that will be used to induce the final classification model. Once the classification algorithm is fixed, the idea is to train it with the feature subset found by the search algorithm, estimating the accuracy and assigning it as the value of the evaluation function of the feature subset. In this way, representational biases of the induction algorithm which are used to construct the final classifier are included in the FSS process. It is claimed by many authors [21, 27] that the wrapper approach obtains better predictive accuracy estimates than the filter approach. However, its computational cost must be taken into account.

*4. Criterion for halting the search.* An intuitive approach for stopping the search is the non-improvement of the evaluation function value of alternative subsets. Another classic criterion is to fix an amount of possible solutions to be visited along the search.

In our microarray problems, we propose to use Sequential Forward Selection (SFS) [18], a classic and well known hill-climbing, deterministic search algorithm which starts from an empty subset of genes. It sequentially selects genes until no improvement is achieved in the evaluation function value. As the totality of previous microarray studies note that very few genes are needed to discriminate between different cell classes, we consider that SFS could be an appropiate search engine because it performs the major part of its search near the empty gene subset.

To assess the goodness of each proposed gene subset for a specific classifier, a wrapper approach is applied. In the same way as supervised classifiers when no gene selection is applied, this wrapper approach estimates, by the LOOCV procedure, the goodness of the classifier using only the gene subset found by the search algorithm. Thus, the microarray dataset is projected maintaining the values of the selected genes and the class variable for all cell samples: the goodness of the proposed gene subset, using the specific classifier, is estimated by the explained LOOCV technique over this projected dataset.

## 3. Related work in microarray datasets for class prediction

Classic statistical classification techniques (Discriminant Analysis, Gaussian and Logistic Classifiers,...) [26, 35], Support Vector Machines [4, 10], Neural Networks [15, 28] and K-NN [2, 14, 25, 35] are the most broadly used class prediction procedures in microarray domains. As far as we know, only two works use decision trees [3, 12] and only our previous study [5] employs the NB classifier. In our knowledge, the CN2 algorithm or classifiers of the same IF-THEN rules family are not still used in microarray domains.

As it was explained in the first section, all microarray works are concerned about the necessity of a gene selection procedure to improve the predictive accuracy of their class prediction model. A filter technique is used in the major part of the works. The wrapper approach is only used in [5, 25]. The work [35] is a hybrid of filter and wrapper approaches. Only the works [5, 25] state the gene selection task as a search problem in the space of gene subsets.

Before the selection process starts, all the works except [5] fix the number of genes in the classifiers they want to compare. In this way, we think that the application of the SFS search tool in the space of gene subsets, which does not previously fix a specific number of genes, implies an improvement.

From the point of view of the employed accuracy estimation technique, the major part of the works use a hold-out procedure, using a portion of the samples to train a predictive model and using the rest of the instances as a test set to estimate the goodness of the classifier. However, other works use more sophisticated validation techniques, such as $k$-fold cross-validation [12]. LOOCV also appears in the literature [5, 25]. Although we consider that LOOCV, due to its unbiased nature [19], is the most suited estimation procedure for microarray datasets, the employment of the hold-out procedure can be justified in many works because a portion of the samples arise from a specific medical-center or study and other samples arise from a different source. In this way, it is common to use the samples of the first medical-center to train the classifier and the samples of the second source to measure its accuracy.

## 4. Experimental results

We test the classification accuracy of our four ML inducers in the following three well known microarray class prediction datasets:

- the *Colon* dataset is presented by Ben-Dor et al. (2000) [4]. This dataset is composed by 62 samples of colon ephitelial cells. Each sample is characterized by 2,000 genes. The samples were collected from colon-cancer patients. The 'tumor' (22 samples) biopsies were collected from tumors, and the 'normal' (40 samples) biopsies were collected from healthy parts of the colons of the same patient. The task is to predict the status of biopsy samples;

- *Leukemia* dataset is presented by Golub et al. (1999) [14]. It contains 72 instances of leukemia patients involving 7,129 genes. The class to be predicted is the specific type of acute leukemia of the patient: acute myeloid leukemia-AML (25 patients) or acute lymphoblastic leukemia-ALL (47 patients);

- *NCI-60* dataset is presented by Ross et al. (2000) [33] and it asses the gene expression profiles in 60 human cancer cell lines that were characterized pharmacologically by treatment with more than 70,000 different drug agents, one at time and independently. The dataset is based on a study of the efficacy of anticancer drugs on tumor tissues. The dataset includes cell lines derived from cancers of colorectal (7 cell lines), renal (8 cell lines), breast (8 cell lines), ovarian (6 cell lines), prostate (2 cell lines), lung (9 cell lines) and central nervous system (6 cell lines) origin, as well as leukemias (6 cell lines) and melanomas (8 cell lines).

Experiments were run in a SGI-Origin200 computer using *MLC++* [22] ML library of programs for the presented classifiers.

Table 1 shows the LOOCV accuracy estimates (coupled with their associated standard deviations) of IB1, NB, C4.5 and CN2 in the microarray datasets, when no gene selection procedure is applied, facing the algorithms with the whole set of genes.

Table 1: LOOCV accuracy percentage and associated standard deviation when all genes are used for each classifier and dataset.

| *Dataset* | IB1 | NB | C4.5 | CN2 |
|---|---|---|---|---|
| *Colon* | $74.19 \pm 5.60$ | $53.23 \pm 6.39$ | $87.10 \pm 4.29$ | $77.42 \pm 5.35$ |
| *Leukemia* | $86.11 \pm 4.10$ | $84.72 \pm 4.27$ | $84.72 \pm 4.27$ | $75.00 \pm 5.14$ |
| *NCI-60* | $70.00 \pm 5.97$ | $48.33 \pm 6.51$ | $46.66 \pm 6.49$ | $28.33 \pm 5.87$ |

It must be noted that the classification trees built by C4.5 have $4, 2$ and 8 genes for *Colon, Leukemia* and *NCI-60* datasets, respectively. The induced trees have, including branch and decision nodes, $9, 5$ and 20 nodes for the same datasets.

The set of IF-THEN rules induced by CN2 shows $10, 9$ and 23 genes for *Colon, Leukemia* and *NCI-60* datasets, respectively. The induced sets of rules have, including the default 'class majority' rule, $7, 6$ and 15 individual IF-THEN rules for the same datasets.

Table 2 reflects the LOOCV estimates and the number of selected genes when SFS is applied over ML algorithms.

Table 2: LOOCV accuracy percentage with the associated standard deviation (first row) and the cardinalities (second row) of finally selected gene subsets for each classifier and dataset when SFS is applied.

| *Dataset* | IB1 | NB | C4.5 | CN2 |
|---|---|---|---|---|
| *Colon* | $91.94 \pm 3.49$ | $87.10 \pm 4.29$ | $95.16 \pm 2.75$ | $91.94 \pm 3.49$ |
| | 4 | 2 | 3 | 3 |
| *Leukemia* | $100.00 \pm 0.0$ | $95.83 \pm 2.37$ | $95.83 \pm 2.37$ | $97.22 \pm 1.95$ |
| | 3 | 4 | 2 | 3 |
| *NCI-60* | $85.00 \pm 4.15$ | $70.00 \pm 5.97$ | $76.66 \pm 5.51$ | $67.33 \pm 6.06$ |
| | 6 | 6 | 7 | 6 |

With the aid of the SFS gene selection technique, all supervised classifiers improve their accuracy results in our three datasets with respect to the no gene selection approach. In all cases except for C4.5 in *Colon* dataset, when a cross-validated paired $t$-test is applied [9], these accuracy differences between no gene selection and SFS approaches are statistically significant at the 0.05 significance level.

In the case of NB and IB1 algorithms, this accuracy improvement is coupled with a dramatic reduction in the number of genes needed to build the learned models. Although this gene reduction is slight in the case of C4.5 and a bit more pronounced for CN2, the SFS approach causes the desired accuracy improvement. In this way, we confirm the results of Kohavi (1995) in ML problems about the capability of the FSS approach to improve the accuracy of C4.5.

The case of C4.5 in *Leukemia* dataset must be mentioned. When no gene selection approach is applied, the decision tree is built with two genes, that is, the same number of genes proposed by

SFS over C4.5 in the same dataset. However, the SFS approach helps C4.5 in the detection of genes that build a more accurate model. This occurs because the learning process of C4.5 prefers genes that are closely correlated with the class label and does not directly take into account the accuracy level. Thus, by means of the application of the wrapper approach, which focuses its attention on the accuracy level, genes that build a more accurate tree can be found. Figure 1 shows the decision trees induced when no FSS and SFS are applied.

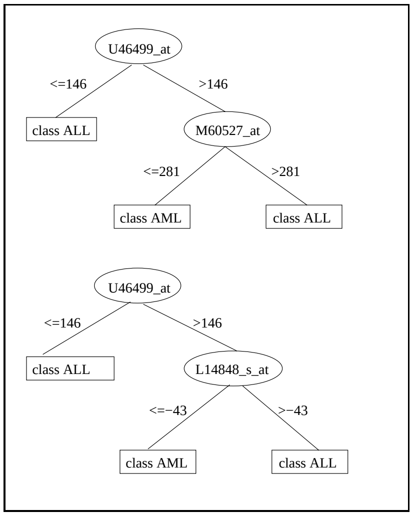

Figure 1: Decision trees induced by C4.5 when no FSS is applied (top) and SFS is applied (bottom). Genes are denoted with their GenBank accession numbers: U46499-at (Glutathione s-transferase, microsomal), M60527-at (DCK Deoxycytidine kinase) and L14848-s-at (MHC class I-related protein mRNA).

Among three datasets, *NCI-60* needs the largest number of genes. This is due to the large number of different classes in *NCI-60*, needing the classifiers a larger number of predictive genes to discriminate among the nine classes of the problem.

Although the number of genes for C4.5 and CN2 is slightly reduced by the SFS approach, the number of tree nodes (C4.5) and IF-THEN rules (CN2) is not reduced in the classification models when the wrapper approach is applied. The C4.5 trees finally selected by the wrapper procedure have, including branch and decision nodes, 9, 5 and 21 nodes for *Colon, Leukemia* and *NCI-60* datasets, respectively. The CN2 sets of rules finally selected by the wrapper procedure have, including the default 'class majority' rule, 9, 9 and 15 individual IF-THEN rules for the same datasets.

Table 3 shows, for each supervised algorithm and domain, the CPU time needed to estimate by LOOCV the accuracy of the supervised classifier when no FSS is applied and the time requirements of the whole SFS procedure. The accuracy improvements of the wrapper SFS approach are coupled with demanding computer-load necessities. As the no FSS approach works with the whole feature set, it also needs considerable computer resources for IB1 and CN2 classifiers. Due to the complex learning processes of C4.5 and CN2, they are the most time consuming algorithms for the SFS wrapper approach. The NB's excelent trade-off between accuracy percentage and CPU time necessities must be noted.

## 5. Summary and Future Work

Table 3: CPU times (in seconds) needed to estimate by LOOCV the accuracy of the classifier when no FSS is applied (first row) and the time requirements of the whole SFS wrapper procedure (second row)

| *Dataset* | IB1 | NB | C4.5 | CN2 |
|---|---|---|---|---|
| *Colon* | 2,385 | 58 | 672 | 12,654 |
| | 10,070 | 871 | 21,950 | 29,851 |
| *Leukemia* | 33,952 | 344 | 3277 | 153,403 |
| | 48,780 | 36,006 | 115,714 | 203,053 |
| *NCI-60* | 1,169 | 41 | 205 | 15,720 |
| | 10,023 | 3,634 | 56,791 | 100,136 |

In this paper, the cancer class prediction in three well known microarray datasets has been tackled from a ML perspective by means of four known classifiers with different biases to learning: IB1, NB, C4.5 and CN2. In huge dimensional tasks such as microarray domains, gene selection methods are crucial if the goal of the study is to obtain an accurate classification model and to identify genes whose expression patterns have meaningful and predictively accurate biological relationships to the class prediction task. The accuracy is notably improved and the number of genes of the classifiers is notably reduced with respect to the no gene selection approach by the sequential, hill-climbing SFS gene selection process, evaluating by a wrapper procedure the goodness of each proposed gene subset.

In this study, we have not attempted to discuss the biological significance of the specific genes selected by each algorithm, and the attainment of an accurate classification rule has been our central objective. The employment of a group of ML algorithms and its associated wrapper FSS procedure, so popular in other areas of the real world but barely used in microarray tasks, has guided our motivation.

In the future, we plan to use population-based, randomized search algorithms, such as Genetic Algorithms [25] or Estimation of Distribution Algorithms [5, 24] in DNA microarray tasks. While SFS returns a unique gene subset, the output of a population-based, randomized search algorithm can be interpreted as a group of different gene subsets, from which a 'consensed' final classifier and gene subset can be formed.

We also plan to study the biological significance of selected genes. As the microarray discipline is nowadays spectacularly growing, we would like to extend our experimentation to datasets that will certainly appear in the next years.

### Acknowledgments

This work is supported by the University of the Basque Country, by the Department of Education, University and Research of the Basque Government and the Ministry of Science and Technology of the Spanish Government under grants 9/UPV/EHU 00140.226-12084/2000, PI1999-40 and TIC2001-2973-C05-03, respectively.

## References

[1] D.W. Aha, D. Kibler, and M.K. Albert, 'Instance-based learning algorithms', *Machine Learning*, **6**, 37–66, (1991).

[2] V. Aris and M. Recce, ‘A Ranking Method to Improve Detection of Disease Using Selectively Expressed Genes in Microarray Data’, in *Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, (2000).

[3] M. Beibel, ‘Selection of Informative Genes in Gene Expression Based Diagnosis: A Nonparametric Approach’, in *Lecture Notes in Computer Sciences. Proceedings of the First International Symposium in Medical Data Analysis, ISMDA2000*, eds., R. Brause and E. Hanisch, volume 1933, pp. 300–307. Springer-Verlag, (2000).

[4] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, ‘Tissue Classification with Gene Expression Profiles’, *Journal of Computational Biology*, **7**(3-4), 559–584, (2000).

[5] R. Blanco, P. Larrañaga, I. Inza, and B. Sierra, ‘Selection of highly accurate genes for cancer classification by Estimation of Distribution Algorithms’, in *Workshop of Bayesian Models in Medicine. AIME 2001*, pp. 29–34, (2001).

[6] A. Brazma and J. Vilo, ‘Gene expression data analysis’, *Federation of European Biochemical Societies Letters*, **480**, 17–24, (2000).

[7] B. Cestnik, ‘Estimating probabilities: a crucial task in machine learning’, in *Proceedings of the European Conference on Artificial Intelligence*, pp. 147–149, (1990).

[8] P. Clark and T. Nibblet, ‘The CN2 induction algorithm’, *Machine Learning*, **3**(4), 261–283, (1989).

[9] T.G. Dietterich, ‘Approximate statistical tests for comparing supervised learning algorithms’, *Neural Computation*, **10**(7), 1895–1924, (1998).

[10] C.H.Q. Ding, ‘Tumor Tissue Classification Using Support vector Machines and K-Nearest Neighbor Methods’, in *Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, (2000).

[11] J. Doak, ‘An evaluation of feature selection methods and their application to computer security’, Technical Report CSE-92-18, University of California at Davis, (1992).

[12] W. Dubitzky, M. Granzow, D. Berrar, S. Bulashevska, C. Conrad, D. Gerlich, and R. Eils, ‘Symbolic and Subsymbolic Machine Learning Approaches for Molecular Classification of Cancer and Ranking of Genes’, in *Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, (2000).

[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[14] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caliguri, C.D. Bloomfield, and E.S. Lander, ‘Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring’, *Science*, **286**, 531–537, (1999).

[15] K.B. Hwang, D.Y. Cho, S.W. Wook Park, S.D. Kim, and B.Y. Zhang, ‘Applying Machine Learning Techniques to Analysis of Gene Expression Data: Cancer Diagnosis’, in *Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, (2000).

[16] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra, 'Feature Subset Selection by Bayesian network-based optimization', *Artificial Intelligence*, **123**(1-2), 157–184, (2000).

[17] I. Inza, M. Merino, P. Larrañaga, J. Quiroga, B. Sierra, and M. Girala, 'Feature Subset Selection by Genetic Algorithms and Estimation of Distribution Algorithms. A case study in the survival of cirrhotic patients treated with TIPS', *Artificial Intelligence in Medicine*, **23**(2), 187–205, (2001).

[18] J. Kittler, 'Feature set search algorithms', in *Pattern Recognition and Signal Processing*, ed., C.H. Chen, pp. 41–60. Sithoff and Noordhoff, (1978).

[19] R. Kohavi, 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in *Proceedings of the International Joint Conference on Artificial Intelligence*, eds., N. Lavrac and S. Wrobel, (1995).

[20] R. Kohavi, *Wrappers for performance enhancement and oblivious decision graphs*, Ph.D. Thesis, Stanford University, 1995.

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

[22] R. Kohavi, D. Sommerfield, and J. Dougherty, 'Data mining using MLC++, a Machine Learning library in C++', *International Journal of Artificial Intelligence Tools*, **6**, 537–566, (1997).

[23] P. Langley and S. Sage, 'Induction of selective Bayesian classifiers', in *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 399–406, (1994).

[24] P. Larrañaga and J.A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer Academic Press, 2001.

[25] L. Li, L.G. Pedersen, T.A. Darden, and C. Weinberg, 'Computational Analysis of Leukemia Microarray Expression Data Using the GA/KNN Method', in *Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, (2000).

[26] W. Li and Y. Yang, 'How many genes are needed for a discriminant microarray data analysis?', in *Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, (2000).

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

[28] A. Mateos, J. Herrero, J. Tamames, and J. Dopazo, 'Supervised and hierarchical unsupervised neural networks for clustering both gene expression profiles and genes', in *Proceedings of the Second Conference on Critical Assessment of Microarray Data Analysis, CAMDA2001*, (2001).

[29] D. Michie, 'Personal models of rationality', *Journal of Statistical Planning and Inference*, **25**, 381–399, (1990).

[30] T.M. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[31] P. Pudil, J. Novovicova, and J. Kittler, 'Floating search methods in feature selection', *Pattern Recognition Letters*, **15**(1), 1119–1125, (1994).

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

[33] D.T. Ross, U. Scherf, M.B. Eisen, C.M. Perou, C. Rees, P. Spellman, V. Iyer, S.S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J.C.F. Lee, D. Lashkari, D. Shalon, T.G. Myers, J.N. Weinstein, D. Botstein, and P.O. Brown, 'Systematic variation in gene expression patterns in human cancer cell lines', *Nature Genetics*, **24**(3), 227–234, (2000).

[34] W. Siedelecky and J. Sklansky, 'On automatic feature selection', *International Journal of Pattern Recognition and Artificial Intelligence*, **2**, 197–220, (1988).

[35] E.P. Xing, M.I. Jordan, and R.M. Karp, 'Feature Selection for High-Dimensional Genomic Microarray Data', in *Proceedings of the Eighteenth International Conference in Machine Learning, ICML2001*, pp. 601–608, (2001).