Variable Selection and Discrimination in Gene Expression Data by Genetic Algorithms

Rüdiger Krause[1] and Gerhard Tutz

Department of Statistics,
Ludwig-Maximilians University, Akademiestr.1, 80799 München, Germany

Summary. Gene expression datasets usually have thousends of explanatory variables which are observed on only few samples. Generally most variables of a dataset have no effect and one is interested in eliminating these irrelevant variables. In order to obtain a subset of relevant variables an appropriate selection procedure is necessary. In this paper we propose the selection of variables by use of genetic algorithms with the logistic regression as underlying modelling procedure. The selection procedure aims at minimizing information criteria like AIC or BIC. It is demonstrated that selection of variables by genetic algorithms yields models which compete well with the best available classification procedures in terms of test misclassification error.

Keywords

Genetic algorithm, Variable selection, Logistic regression, AIC, BIC.

1 Introduction

The problem of variable selection (or subset selection) arises when the relationship between a response variable and a subset of potential explanatory variables is to be modelled, but there is substantial uncertainty about the relevance of the variables. Most datasets contain explanatory variables which are redundant or irrelevant and the objective is to identify the relevant ones.

A demanding challange for algorithms for variable selection is the analysis of gene expression data. Usually such studies consist of many thousends of genes but only of few samples. For a detailed presentation of microarray technology as well as approaches to the extraction of gene expression data we refer e.g. to Hamadeh & Afshari (2000). In the literature several algorithms for variable selection have been proposed (see e.g. Miller (2002)). In this paper we propose variable selection in gene expression data by application of genetic algorithms.

The paper is organized as follows: in the next section we describe the logistic regression procedure and some information criteria for model selection. In section 3 we present the genetic algorithm used for variable selection. Finally section 4 compares our approach with other methods proposed in the literature by two microarray datasets.

[1] krause@stat.uni-muenchen.de

2 Logistic Discrimination and Information Criteria for Model Selection

Let the observations be given by $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, where $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip})$ is a vector of explanatory variables and $y_i \in \{0, 1\}$ is a binary observation indicating membership of class. The most wickly used binary regression model is the *logistic model* (or *logit model*)

$$log \frac{\pi(\mathbf{x}_i)}{1 - \pi(\mathbf{x}_i)} = \mathbf{x}_i^T \beta,$$

where $\mathbf{x}_i = P(y_i = 1|\mathbf{x}_i)$ denotes the conditional probability of observing $y_i = 1$. The use of the logistic model for predicting y_i is often referred to as *logistic discrimination.*

Maximum likelihood estimates for the logistic model are obtained by maximizing the log-likelihood function

$$l = \sum_{i=1}^{n} y_i \, log(\pi_i) + \sum_{i=1}^{n} (1 - y_i) \, log(1 - \pi_i) \quad .$$

For further details of the algorithm see e.g. Fahrmeir & Tutz (2001).

When using the logit model on gene expression data, it is impossible to include thousands of explanatory variables. Maximum likelihood estimates asymptotically $(n \to \infty)$ exist under weak conditions. But if the number of explanatory variables p is large as compared to the sample size n maximum likelihood estimates do not longer exist because of total separation. A criterion for the existence of the maximum likelihood estimates is the lack of overlap between observations with $y_i = 0$ and $y_i = 1$ (compare Albert & Anderson (1984), Santner & Duffy (1986), Christmann & Rousseeuw (2001)). Thus the used limit for the existence of maximum likelihood estimates is needed if p equals n.

The problem becomes even harder if interactions are included in the model. Since one knows that only four of the variables corresponding to the thousands of genes are relevant, drastic variable selection is necessary. In the following we will consider the linear logistic model for a subset of covariates

$$log \frac{\pi(\mathbf{x}_i)}{1 - \pi(\mathbf{x}_i)} = \mathbf{x}_{i,s}^T \beta,$$

where $\mathbf{x}_{i,s}$ is a vector of variables from subset $S \subset \{1, \ldots, p\}$, i.e. $\mathbf{x}_{i,s}$ contains the selection $x_{ij}, j \in S$ from the total vector $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip})$. In addition we consider models that allow the form

$$log \frac{\pi(\mathbf{x}_{i,s})}{1 - \pi(\mathbf{x}_i)} = \sum_{j \in S} x_{ij} + \sum_{(s,t) \in J} x_{is} x_{it},$$

where $J \subset S \times S$ denotes the index set for interactions between variables from S. Usually J is much smaller than $S \times S$.

To assess the appropriateness of a chosen set of variables information criteria are used. These criteria compare the error of a model with the model complexity (i.e. the number of parameters used). An additional parameter should be integrated into a model only if the value of the information criterion decreases. If we have several competing models (sets of variables and interactions), we choose that one with the lowest value of the information criterion. Two common used criteria are the *Akaike information criterion (AIC)* proposed by Akaike (1973)

$$AIC = -2l + 2q \quad ,$$

and the *Bayesian information criterion (BIC)* of Schwarz (1978)

$$BIC = -2l + q\,log(n) \quad ,$$

where l is the log-likelihood function and q denotes the number of paramters in the model which have to be estimated. Usually BIC leads to a stronger penalization of more complex models than AIC.

3 A Genetic Algorithm for Variable Selection

Genetic Algorithms (Holland (1975), Goldberg (1989)) are originally based on Darwin's theory of evolution which refers to the principle that better adapted (fitter) individuals win against their competitors under equal external conditions. As in the biological model, genetic algorithms use operators like selection, crossover, or mutation to model the natural phenomenon of genetic inheritance and Darwinian strife of survival. For some background on the biological processes of genetics and the origin of the terminology see Haupt and Haupt (1998) and Mitchell (1996).

The smallest units linked to relevant information of a genetic algorithm are called *genes*. The genes are either single units or short blocks of adjacent units and the information is coded in form of numbers, characters, or other symbols. Usually several genes are arranged in a linear succession which is called *string* (also *chromosome, individual*). The genetic algorithm always uses several strings as a potential solution of an optimization problem. This collection of strings is called *population*. If we apply operators to strings we generate a population with new different strings. This new population of strings is called *offspring*. We denote the particular populations as *generations*, or more precisely as parent- respectively offspring generation. The function to be optimized (e.g. AIC, BIC, see section 2) is denoted as fitness function.

The basis of the genetic algorithm for variable selection is an $0-1$ coding of strings. Suppose we have p metrical variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$. Then the coding of the inclusion of metrical variables is given by

$$\delta_j^x = \begin{cases} 1 & if\ variable\ \mathbf{x}_j\ is\ included \\ 0 & else \end{cases} \qquad j = 1, \ldots, p.$$

Interactions δ_{jk}^{xx} are coded in the same way by

$$\delta_{jk}^{xx} = \begin{cases} 1 & if\ the\ interaction\ between\ \mathbf{x}_j\ and\ \mathbf{x}_k\ is\ included \\ 0 & else \end{cases} \quad ,$$

where $j, k = 1, \ldots, p, j \neq k$.

For better interpretability we only consider hierarchical models, for which interactions are included only if the corresponding main effects are included. The restriction

$$\delta_{jk}^{xx} \leq \delta_j^x \delta_k^x \tag{1}$$

implies that an interaction can be only included if main effects of both variables $\mathbf{x}_j$ and $\mathbf{x}_k$ are included. This relation has always to be checked after application of crossover- and mutation operators to interaction indicators. The indicators are collected into one string

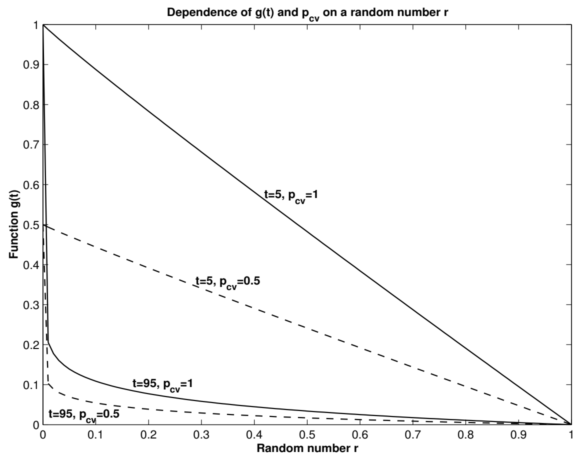

Figure 1. Function $g(t)$ is shown subject to a uniformly distributed random number r for two sizes of the generation number t and crossover probabilities $p_{cv} = 1$ respectively $p_{cv} = 0.5$. The user-dependent system parameter b is chosen as 1.

$$\delta = (\{\delta_j^x\}, \{\delta_{jk}^{xx}\})$$

which has values 0 or 1 as components.

For the design of a powerful genetic algorithm operators like crossover and mutation are important. Many authors (e.g. Oliveira, Benahmed, Sabourin, Bortolozzi & Suen (2001); Wallet, Marchette, Solka & Wegman (1996); Yang & Honavar (1997)) use operators which are constant during the whole process of the genetic algorithm. However, better results are obtained if different aspects of the search are differently weighted at various times: first we are generally interested in exploring the search space and aquire information about the nature of the space. Later we try to obtain information near the global optimum by local search. Therefore we propose adaptive and non-uniform operators:

(i) *Adaptive binary crossover (ABC)* operator: suppose we have two $0-1$ strings with indicator variables $\delta = (\delta_1 \ldots \delta_i \ldots \delta_k)$ and $\bar{\delta} = (\bar{\delta}_1 \ldots \bar{\delta}_i \ldots \bar{\delta}_k)$. A pair of bits $(\delta_i, \bar{\delta}_i)$ of the parent strings swap their places if we have a random number $r_i \in [0, 1]$ with
$$r_i < p_{cv}(\underbrace{1 - r^{(1-\frac{t}{T})^b}}_{\equiv g(t)}). \tag{2}$$
Here $r \in [0, 1]$ is a uniform random number equal for all bits of a string, p_{cv} is the crossover probability (of the variables), t is the number of the current generation, T is the maximum number of generations and b is a user-dependent system parameter which determines the degree of non-uniformity. Which strings are selected for crossover process is controlled by a similar expression as (2).

In contrast to the conventional crossover operators the ABC operator considers the diverse objectives which have different relevance during the application of the genetic algorithm. We can distinguish between two extreme cases (compare also Figure 1): if t is small the exponent of $g(t)$ is close to zero and hence $g(t)$ is primarily influenced by a suitable choice of the random number r. Figure 1 illustrates this behaviour: for generation number $t = 5$ one obtains approximatively a straight line with slope -1. For $p_{cv} = 1$, many strings show swaps of corresponding bits (if r is small) during the crossover process. By suitable choice of p_{cv} the number of swaps between corrsponding bits can be varified. A small

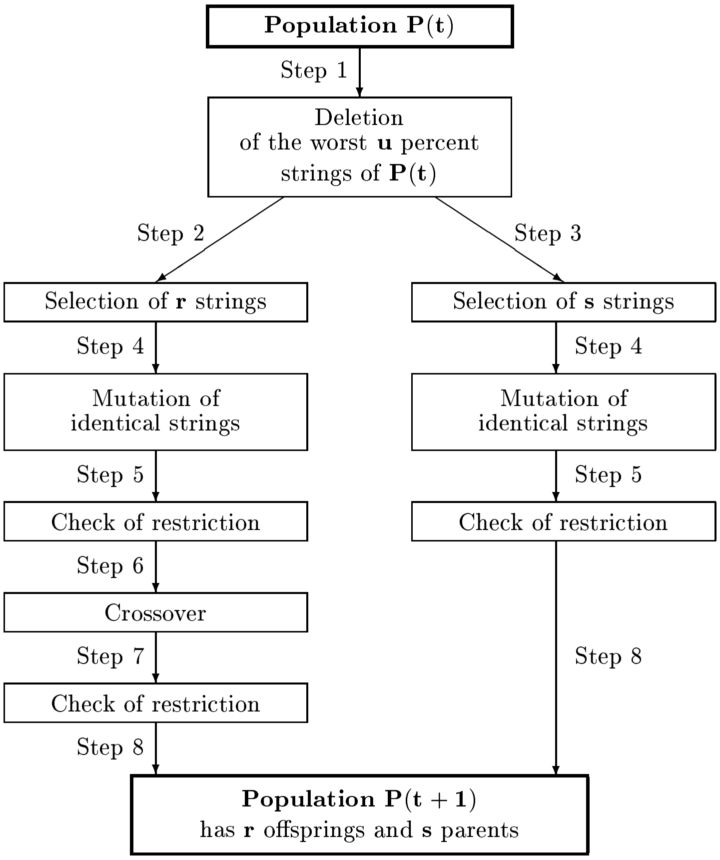

Figure 2. *Structure of the modified selection procedure (modSP) given as a flowchart. Details in the text.*

value of p_{cv} also decreases the number of swaps between two strings (e.g. a decrease of 0.5 reduces the number of swaps by a half during crossover process). If the generation number t is large, the exponent of $g(t)$ is close to zero and hence $g(t)$ also yields values close to zero for a wide range of random numbers r. This is illustrated in Figure 1 for $t = 95$. At the end of the genetic algorithm there are only few swaps between corresponding bits. In addition a small value of p_{cv} increases the effect.

(ii) *Adaptive binary mutation (ABM)* operator: for each bit of a string we generate a random number r_i and if

$$r_i < p_{mv}(1 - r^{(1-\frac{t}{T})^b}), \tag{3}$$

holds the bits mutate, i.e. 0 is changed to 1 and vice versa. Here p_{mv} is the mutation probability (of the variables). The idea and functionality of this operator are the same as described for the ABC operator.

The genetic algorithm for variable selection also uses the *modified selection procedure (modSP)* as introduced in Krause & Tutz (2003). The only difference is the check of available interactions after each crossover respectively mutation step. Hence the selection procedure consists of eight steps (compare Figure 2):

Step 1: *Suppose that a population $P(t)$ is generated in iteration step t. Then delete the worst u percent strings of $P(t)$.*

Step 2: *From the remaining strings of step 1 randomly select r strings, which do not necessarily have to be distinct.*

Step 3: *From the remaining strings of step 1 randomly select s parent strings. These have not to be distinct from the r selected strings in step 2.*

Step 4: *If a string has one or more further identical strings in the population (i.e. all genes of the strings are identical) the copies will be mutated. How many genes of a string are randomly selected and mutated is controlled by a random number (at least one gene is mutated). After mutation, there are r respectively s different strings. This operation will also be executed for the s parent strings.*

Step 5: *Check of the restriction $\delta_{jk}^{xx} \leq \delta_j^x \delta_k^x$ and deletion of illegal interactions.*

Step 6: *Controlled by the crossover probability p_{cv}, apply a crossover operator to the set of the r (distinct) strings and generate $i, 2 \leq i \leq r$ new strings.*

Step 7: *Check of the restriction $\delta_{jk}^{xx} \leq \delta_j^x \delta_k^x$ and deletion of illegal interactions.*

Step 8: *Let r offspring and s parent strings form the new population $P(t+1)$.*

4 Application to Gene Expression Data

In this paper we analyse two microarray datasets:

- The colon dataset contains expression levels of 2000 genes $x_{ij}, j = 1, \ldots, 2000$, and 62 observations $y_i, i = 1, \ldots, 62$. Here 40 observations descend from patients with tumor tissue and 20 observations descend from patients with normal tissue. The dataset is divided into a training dataset with 41 observations and a test dataset with 21 observations. We use 200 splits into training dataset and test dataset with randomly chosen observations. The remaining samples form the test datasets. For further details on the dataset see Alon, Barkai, Notterman, Gish, Ybarra, Mack & Levine (1999). The dataset can be downloaded from *http://microarray.princeton.edu/oncology*.
- The prostate dataset originally contains expression levels of 12600 genes and 102 observations (further informations and the dataset can be downloaded from *http://www-genome.wi.mit.edu/MPR/Prostate*). Hence 52 observations descend from patients with cancer tissue and 50 observations descend from patients with normal tissue. By a filtering step based on the Wilcoxon test (implemented in the software package R) the 2000 genes with the smallest p-values were chosen and used for subsequent analysis. The dataset is divided into a training dataset with 68 observations and a test dataset with 34 observations. We use 200 splits into training dataset and test dataset with randomly chosen observations.

The genetic algorithm proposed in section 3 is applied in two different versions:

(i) *Genetic algorithm without interaction*: Variable selection is executed on the 2000 genes $x_{ij}, j = 1, \ldots, 2000$, only. The initial population of the genetic algorithm is generated by calculating the fitness values for all subsets containing only one gene (hence we have 2000 fitness values). In the initial population we use the *popsize* $= 48$ strings with the best fitness values. Because of the small number of samples as well as the large number of genes in the dataset used, selection of too many genes leads to numerical instabilities during estimation. To prevent these problems we restrict the number of selected genes to maximal 10 genes (i.e. each string contains 10 genes at maximum).

(ii) *Genetic algorithm with interaction*: First calculate the fitness values for all subsets containing only one gene. Then we choose a default number $shrink = 40$ of the variables which yield the best fitness values. Hence the original number of 2000 genes is shrinked to $shrink$ genes which yield the pool for the subsequent variable selection. Furthermore the genetic algorithm can select interactions between any two genes of the pool (e.g. in case of $shrink = 40$ genes we have 780 interactions). The initial population is generated at random. Because of numerical instabilities during estimation the number of selected genes respectively interactions is restricted to maximal 10.

As default parameters of the genetic algorithm are used: population size ($popsize$) = 48 strings, crossover probability $p_{cv} = 1$, mutation probability $p_{mv} = 1$, deletion of $u = 60$ percent of the worst strings, selection of $r = 38$ and $s = 10$ strings, $\nu = 0.5$, $T = 1000$ and $b = 1$.

With prediction in mind the results of the genetic algorithms are compared to alternative approaches of classification which are known to perform well in high dimensional settings. These methods are:

(1) *Discrete AdaBoost*: The motivation for the discrete AdaBoost procedure (Friedman, Hastie & Tibshirani (2000)) was to combine the outputs of many "weak" classifiers to produce a powerful "committee". The algorithm works in the following way:

- Each observation of the training dataset has an initial weight $w_i = 1/m, i = 1, \ldots, m$.
- For $t = 1 : M$
  - Use a classifier $G_t(x)$ (e.g. CART, see below) and fit the classifier to the training data which use the weights w_i. The classifier $G_t(x)$ produces a prediction taking one of the two variables $\{-1, +1\}$, i.e. each element of the training sample is assigned a prediction $\in \{-1, +1\}$.
  - The resulting weighted error rate is computed by

$$err_t = \frac{\sum_{i=1}^{m} w_i I(y_i \neq G_t(x_i))}{\sum_{i=1}^{m} w_i} \quad ,$$

    i.e. in case that the true observation y_i and the prediction (produced by the classifier $G_t(x)$) are different, the error rate increases by a weighted amount.
  - Compute $\alpha_t = log((1 - err_t)/err_t)$ which weights the influence of the used classifier.
  - Set

$$w_i \equiv w_i \cdot e^{\alpha_t I(y_i \neq G_t(x_i))} \qquad , i = 1, \ldots, m \quad .$$

    The individual weight of each observation is updated for the next iteration. It is seen that the misclassified observations obtain larger weights that the correctly classified observations. The objective is that the next classifier $G_{t+1}(x)$ focusses on observations with larger weights.
- The predictions from all classifiers $G_t(x), t = 1, \ldots, M$, are combined through a weighted sum

$$G(x) = sign\left[\sum_{t=1}^{M} \alpha_t G_t(x)\right] \quad .$$

Software program	Average misclassification rate	Standard deviation
Genetic algorithm *without* interaction (AIC)	0.2317	0.0894
Genetic algorithm *without* interaction (BIC)	0.2224	0.0888
Genetic algorithm *with* interaction (AIC)	0.1776	0.0909
Genetic algorithm *with* interaction (BIC)	0.1700	0.0975
Discrete AdaBoost	0.1920	0.0720
CART	0.2930	0.0890
1-Nearest neighbour	0.2520	0.0860
5-Nearest neighbour	0.2140	0.0890

Table 1. *Average misclassification rate and standard deviation of the misclassification rates for 200 replications of the test dataset (colon dataset) are shown.*

In the colon dataset we have chosen the CART procedure (see below) as classifier. The number M of iterations is 50.

(2) *Nearest-Neighbour Method*: This method requires no model to be fit and works in the following way: given a point $\mathbf{x}_i^\star, i = 1, \ldots, n$, in the training dataset we choose the k nearest neighbour by the Euclidian distance

$$d_i = \|\mathbf{x}_i - \mathbf{x}_i^\star\| \quad .$$

An estimator $\hat{y}_i$ can be received by

$$\hat{y}_i = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x}_i^\star)} y_i \quad ,$$

where $N_k(\mathbf{x}_i^\star)$ is the neighbourhood of $\mathbf{x}_i^\star$ defined by the k closest points $\mathbf{x}_i$ in the training sample. If the estimator $\hat{y}_i$ takes a value > 0.5 we assign $\hat{y}_i$ the value 1 (i.e. the patient has tumor). Otherwise ($\hat{y}_i \leq 0.5$) the estimator is assigned the value 1 (i.e. the patient has no tumor). By comparison with the true observations y_i the number of errors can be determined. And this error rate has to be minimized. For application of the k-nearest-neighbour method to the colon dataset we choose $k = 1$ and $k = 5$.

(3) *CART*: Classification and regression trees (CART) have been developed by Breiman, Friedman, Olshen & Stone (1984). The idea is that the predictor space is successively divided and the resulting splits have to be heterogeneous as much as possible in respect of variable y. Otherwise the values have to be homogeneous within a split. For example let only one metrical variable x be given; thus we search for a cutting point c with the property: the split in sets $A_1 = \{x : x \leq c\}$ respective $A_2 = \{x : x > c\}$ lead to similar values within the sets but different values between A_1 and A_2 (i.e. $y = 0$ in A_1 and $y = 1$ in A_2 or vice versa). In case of the colon dataset in each iteration the optimal split is searched for each variable. The variable which yields the best split is selected. As maximal number of splits we choose 4, i.e. at most 5 variables were selected.

First the colon dataset is considered. Table 1 shows the average misclassification rate respectively the standard deviation of the misclassification rate for 200 replications of the test dataset. The average misclassification rate is defined as

$$\overline{err} = \frac{1}{N} \sum_{l=1}^{N} \left[\frac{1}{m} \sum_{i=1}^{m} I\left(\hat{y}_i, y_i\right) \right] \quad ,$$

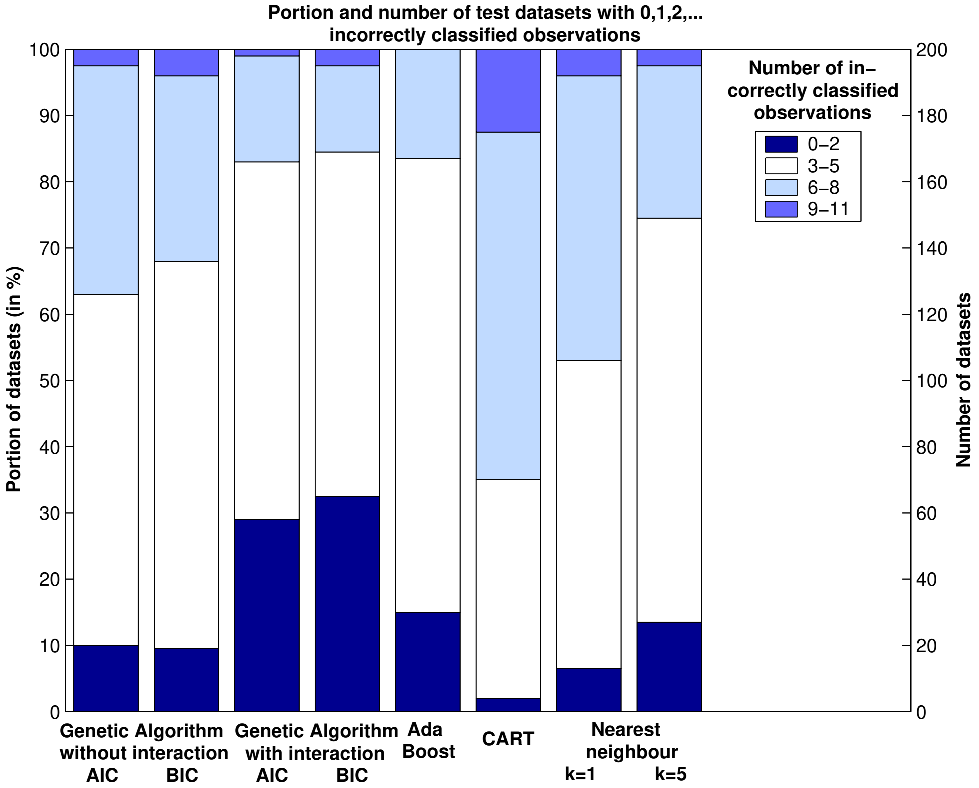

Figure 3. *Number of test datasets with* $0, 1, 2, \ldots$ *incorrectly classified observations for the considered classification approaches.*

where N is the number of replications of the test dataset (here $N = 200$), m is the number of samples in the test dataset ($m = 21$) and

$$I(\hat{y}_i, y_i) = \begin{cases} 0 & if \quad \hat{y}_i = y_i \\ 1 & if \quad \hat{y}_i \neq y_i \end{cases} .$$

It is seen from Table 1 that the misclassification rate takes the smallest value when the genetic algorithm with interaction is used. All other methods show distinctly worse results. While the nearest neighbour approach with $k = 5$ and the genetic algorithm without interaction (with BIC) yield similar results for the misclassification rate, AdaBoost approach performs better by approximately $10 - 14\%$. Higher misclassification rates are found by the nearest neighbour method with $k = 1$ and the CART approach. Except for discrete AdaBoost all methods show comparable standard deviations: the values differ between 0.0860 and 0.0975.

With respect to the different information criteria we detect smaller average misclassification rates for genetic algorithms with BIC. However, the differences are relatively slight: e.g. the average misclassification rate for the two genetic algorithms with interaction differ from each other by approximately 4%.

In addition Figure 3 shows the number of datasets with $0, 1, 2, \ldots$ incorrectly classified observations. Here the number of errors has been divided into four classes, in fact $0 - 2$, $3 - 5$, $6 - 8$, and $9 - 11$ misclassified observations per test dataset. In accordance with Table 1 also Figure 3 shows that the genetic algorithms with interaction has conspicuously fewer incorrectly classified observations in the test datasets compared to all other approaches. For example the genetic algorithm with

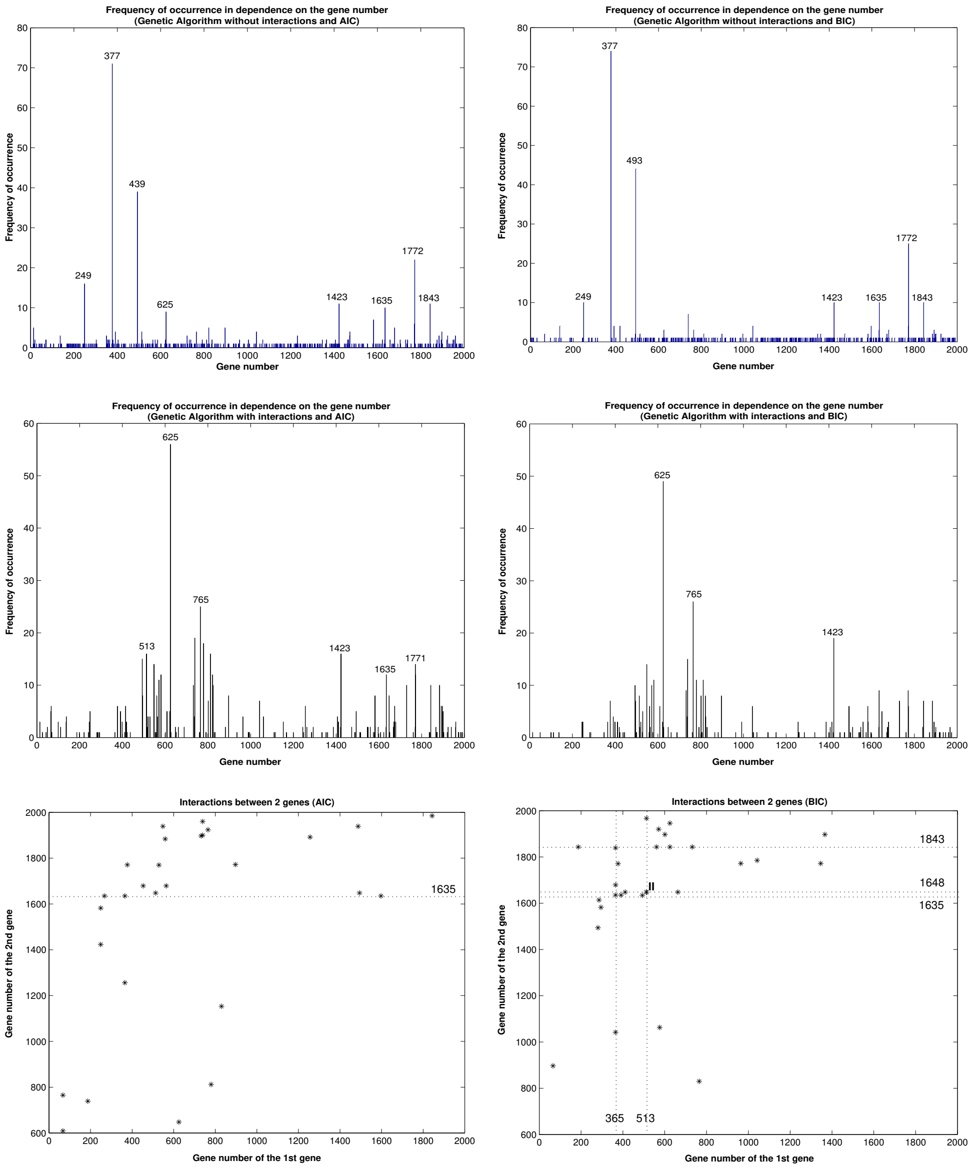

Figure 4. *Frequency of occurrence of genes for the genetic algorithms without (panels in the first row) respectively with (panels in the second row) interaction. The panels at the bottom show the interactions which occur in the genetic algorithm with interactions.*

interaction yields in approximately 30% of the test datasets $0-2$ incorrectly classified observations. The best competitor (discrete AdaBoost) has $0-2$ incorrectly classified observations in only 15% of the test datasets.

An important question in the analysis of gene expression data is the relevance of genes. One indicator for the relevance of genes is the predictive power of genes (in combination with other genes). Thus it is useful to investigate which variables (genes) have been used in prediction across the various splits into learning and test data.

Figure 4 yields the frequency of occurrence of genes for the genetic algorithms without and with interaction. The left panel of Figure 4 shows the results for the

Software program	Average misclassification rate	Standard deviation
Genetic algorithm *without* interaction (AIC)	0.0994	0.0538
Genetic algorithm *without* interaction (BIC)	0.0962	0.0564
Genetic algorithm *with* interaction (AIC)	0.0946	0.0442
Genetic algorithm *with* interaction (BIC)	0.0877	0.0396
Discrete AdaBoost	0.0703	0.0370
CART	0.1551	0.0560
1-Nearest neighbour	0.1587	0.0539
5-Nearest neighbour	0.1387	0.0537

Table 2. *Average misclassification rate and standard deviation of the misclassification rate for 200 replications of the test dataset (prostata data).*

genetic algorithm with AIC and the right panel shows the results for the genetic algorithm with BIC.

If no interaction is allowed it is seen that there is not much difference between AIC or BIC based approaches. There is a small number of genes which are chosen in many of the splits. The pictures changes slightly if interactions are included. Although most of the genes which are used in the higher approach are chosen again, the relevance has changed for some genes. For example gene 377 is no longer a favourite while the relevance of gene 625 has strongly increased. It should be noted that because of the stronger penalization of the BIC usually a smaller number of relevant genes is chosen from the 200 training datasets (i.e. for example many genes chosen by the genetic algorithm are not included in case of BIC).

The two panels at the bottom of Figure 4 show the interactions between two genes which were used in the genetic algorithm with interaction. For better illustration genes participating in interactions of at least 3 datasets are marked by a dotted line. It is seen the selection of genes by the genetic algorithm based on BIC shows lower variability. Several combinations of genes have been chosen in more than one split of the observations (e.g. gene with the number 513 or 1635). For the genetic algorithm based on AIC only interactions with gene number 1635 have been used more often than once.

The analysis of the second dataset (prostate data) yields similar results. However the differences between genetic algorithm with and without interaction is not as distinct as in the colon dataset. Table 2 shows the average misclassification rates and standard deviations for $N = 200$ replications of the test dataset. The number of samples in the test dataset is $m = 34$. Now the best performer is discrete AdaBoost. The genetic algorithms have slightly increased error rates but distinctly outperform all other approaches. Figure 5 shows that with the exception of discrete AdaBoost the genetic algorithm with interaction has conspicuously fewer incorrectly classified observations compared with all other approaches.

In the same way as for the colon dataset the frequency of the occurrence of genes and interactions is investigated. Here we restrict the presentation to the AIC. The top panels of Figure 6 show the frequency of occurrence of genes for the genetic algorithms without and with interaction. However one has to keep in mind that for this dataset the 2000 genes have been sorted: genes with low p-values (based on the Wilcoxon test) have low gene number and vice versa. The panels demonstrate that the genetic algorithm, without and with interaction, prefer genes with samll p-values. Especially the first gene (lowest p-value) is contained in almost every dataset. Therefore the exact frequency of occurance is omitted in the plots because of scaling

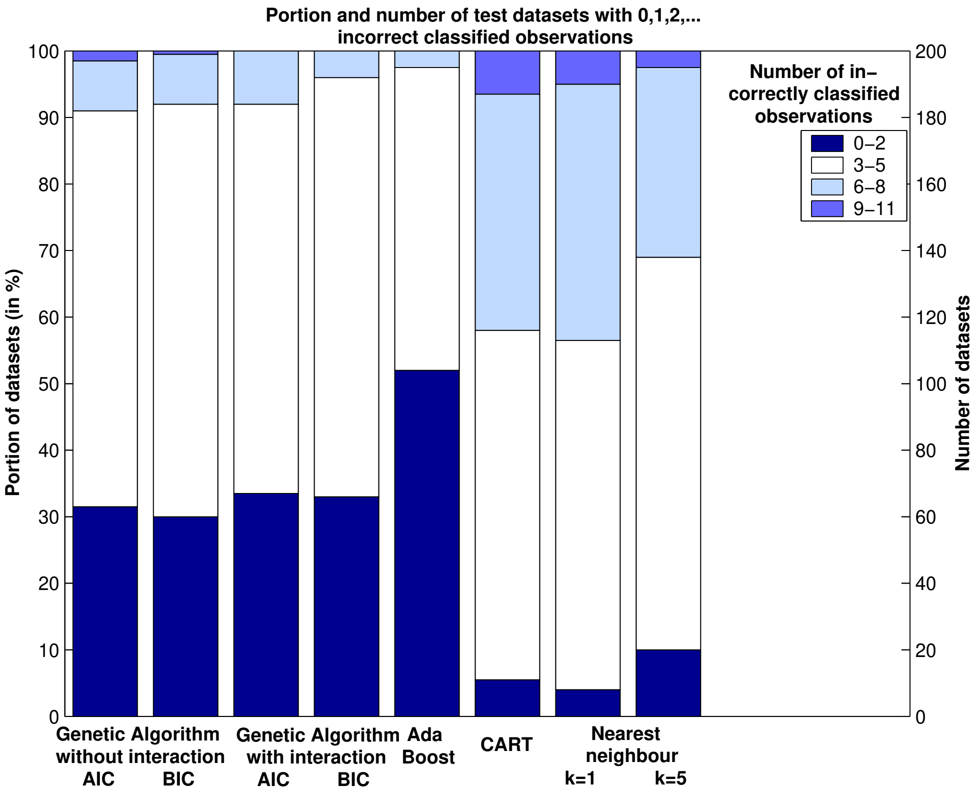

Figure 5. *Number of test datasets (prostate dataset) with* $0, 1, 2, \ldots$ *incorrectly classified observations. The number of errors has been divided into four classes:* $0-2$, $3-5$, $6-8$, *and* $9-11$ *misclassified observations per test dataset.*

effects. For this dataset many genes occur in only one dataset. The number of genes contained in one dataset is conspicuously lower when the genetic algorithm with interaction is used.

The panel at the bottom of Figure 6 shows the frequency of interactions between two genes. It is seen that many datasets use genes with low p-values for interactions (e.g. the gene number of the first gene differs between a gene number of 1 and 150). It should be noticed that application of BIC leads to only 6 interactions. That is the reason why the genetic algorithm with interaction leads to little improvements in the misclassification rate (compare Table 2). Application of the genetic algorithm with interaction to the full prostate dataset (i.e. with 12600 genes) certainly would yield much more interactions.

5 Conclusions

In this paper variable selection in gene expression datasets by genetic algorithms has been investigated. In addition to a genetic algorithm which only chooses main effects (i.e. genetic algorithm without interaction) we have also presented a genetic algorithm for the selection of main effects and their respective interactions. It has been shown that the resulting subsets yield classification procedures that perform very well. The error rates are well comparable with the error rates of discrete Ada-Boost which is one of the best classifiers around. Alternative procedures have been outperformed.

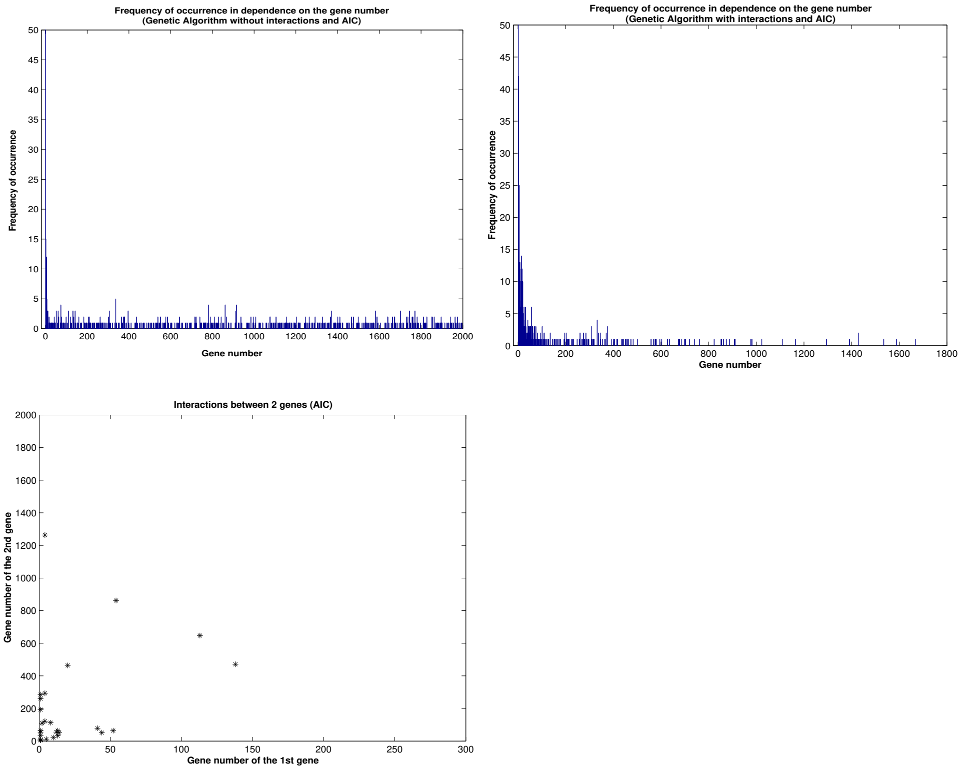

Figure 6. *Frequency of occurrence of genes for the genetic algorithms based on AIC without (top left panel) respectively with (top right panel) interaction. The lower panel shows the interactions which occur in the genetic algorithm with interactions.*

The success in prediction may be seen as a strong indication that the selection of variables by genetic algorithms identifies relevant variables. Otherwise performance would be worse. The stability of selected variables across the spits into learning and test dataset supports that the selection procedure works well.

Here classification error is mainly considered as a criterion for the successful selection of variables. From a different point of view one might also consider it as a way of constructing a classifier. If classification is the main purpose logistic discrimination with variables selected by genetic algorithms seems to compete well with other classifiers.

References

Akaike, H. (1973). A new look at statistical model identification. *IEEE Transactions on Automatic Control* **19**, 716–723.

Albert, A. and Anderson, J. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **71**, 1–10.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of National Academy of Science, Cell Biology* **96**, 6745–6750.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classi cation and Regression Trees.* Pacific Grove: Wadsworth and Brooks/Cole.

Christmann, A. and Rousseeuw, P. (2001). Measuring overlap in logistic regression. *Computational Statistics and Data Analysis* **37**, 65–75.

Fahrmeir, L. and Tutz, G. (2001). *Multivariate statistical modelling based on generalized linear models, 2nd edition.* New York: Springer Verlag.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *The Annals of Statistics 28*(2), 337–407.

Hamadeh, H. and Afshari, C. A. (2000). Gene chips and functional genomics. *American Scientist* **88**, 508–515.

Krause, R. and Tutz, G. (2003). Additive modelling with penalized regression splines and genetic algorithms. *Discussion Paper 312, SFB 386, University of Munich.*

Miller, A. (2002). *Subset Selection in Regression.* Boca Raton, London, New York: Chapman & Hall/CRC.

Oliveira, L. S., Benahmed, N., Sabourin, R., Bortolozzi, F., and Suen, C. Y. (2001). Feature subset selection using genetic algorithms for handwritten digit recognition. In *Proceedings of the 14^{th} Brazilian Symposium on Computer Graphics and Image Processing*, pp. 362–369. Floriano'polis-Brazil: IEEE Computer Society.

Santner, T. and Duffy, D. (1986). A note on a. albert and j.a. anderson's conditions for the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **73**, 755–758.

Schwarz, G. (1978). Estimating the dimensions of a model. *Annals of Statistics* **6**, 461–464.

Wallet, B. C., Marchette, D. J., Solka, J. L., and Wegman, E. J. (1996). A genetic algorithm for best subset selection in linear regression. In *Proceedings of the 28th Symposium on the Interface.*

Yang, J. and Honavar, V. (1997). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* **13**, 44–49.