Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computational Statistics & Data Analysis 50 (2006) 783–802
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Identification of interaction patterns and classification with applications to microarray data

Anne-Laure Boulesteix*, Gerhard Tutz

Department of Statistics, University of Munich, Germany

Received 23 April 2004; received in revised form 11 October 2004; accepted 11 October 2004
Available online 4 November 2004

Abstract

Emerging patterns represent a class of interaction structures which has been recently proposed as a tool in data mining. A new and more general definition referring to underlying probabilities is proposed. The defined interaction patterns (IP) carry information about the relevance of combinations of variables for distinguishing between classes. Since they are formally quite similar to the leaves of a classification tree, a fast and simple method which is based on the CART algorithm is proposed to find the corresponding empirical patterns in data sets. In simulations, it can be shown that the method is quite effective in identifying patterns. In addition, the detected patterns can be used to define new variables for classification. Thus, a simple scheme to use the patterns to improve the performance of classification procedures is proposed. The method may also be seen as a scheme to improve the performance of CARTs concerning the identification of IP as well as the accuracy of prediction.

Keywords: Classification trees; Discrimination; Gene expression; Emerging patterns

1. Introduction

In classification interaction structures among predictors may be used explicitly or implicitly. In linear discriminant analysis or logistic regression a familiar way to exploit interactions is the incorporation of interaction terms into the linear predictor. Nonparametric classifiers like nearest neighborhood classifiers do not specify the interaction structure

* Corresponding author. Department of Statistics, Seminar of Applied Stochastics, Ludwig-Maxmilian University, Akademiestr. 1, 80799 Munich, Germany. Tel.: +49 89 2180 3466; fax: +49 89 2180 5308.
E-mail address: boulesteix@stat.uni-muenchen.de (A.-L. Boulesteix).

0167-9473/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2004.10.004

explicitly but rely on its implicit use. Tree based methods like CARTs (classification and regression trees, registered trademark by Salford Systems) as suggested by Breiman et al. (1984) make interaction structures the central issue. The same holds for early versions of trees, where the detection of interaction structures gave the algorithm its name, i.e. AID for automatic interaction detection (Morgan and Sonquist, 1963). More recently, specific interaction structures called emerging patterns have been introduced by Dong and Li (1999) and applied to high-dimensional gene expression analysis in Li and Wong (2003). An alternative concept which is related to interactions is the search for boxes in the feature space in which the response variable has a particular distribution. Bump hunting as suggested by Friedman and Fisher (1999) is a method to seek boxes in which the response is as high as possible. A short overview on bump hunting is given in Hastie et al. (2001). In the following we will consider simple interaction structures of the emerging pattern type which have the form

$$\{x_1 > \theta_1\} \cap \{x_2 \leqslant \theta_2\} \cap \cdots \cap \{x_d > \theta_d\},$$

where $x_1, \ldots, x_d$ are covariates and $\theta_1, \ldots, \theta_d$ are thresholds to be estimated. An interaction structure of this type will be called an interaction pattern (IP). For simplicity, it will be abbreviated by P. Emerging patterns as considered by Dong and Li (1999) are IPs which discriminate between two classes in a specific sense. Let $\mathbf{x}^{\mathrm{T}} = (x_1, \ldots, x_p)$ denote the random vector of covariates and Y the class indicator which can take the values 1 and 2. Let $n_{P,j}$ denote the number of observations from class j in P. According to the definition of Dong and Li (1999), a pattern P is a ρ-emerging pattern from class i to class j if the growth rate from i to j GR_{ij} is larger than ρ, where GR_{ij} is defined as

$$GR_{ij}(P) = \frac{n_{P,j}/n_j}{n_{P,i}/n_i}.$$

The definition is based on a heuristic rather than a statistical criterion. The focus in Dong and Li (1999) is on data mining and therefore on algorithms that find all the ρ-emerging patterns without regard to relevance. The problem of overfitting is neglected. By investigating a large number of possible patterns, it is always possible to find a large growth rate in the training data, but in an independent test data set, growth rates are usually much lower. Another drawback of Dong and Li's patterns is that the definition is restricted to the case $K = 2$.

In this paper, we suggest a more general definition of IPs which is based on the underlying probability and allows for more than two classes. In addition, a CART-based method is proposed to identify statistically relevant interactions in cases where many variables are potential candidates. In gene expression data where the expression levels of thousands of genes are measured simultaneously the challenge is the number of predictors. The objectives of our approach are identification of IPs as well as their use in classification. In the microarray framework, the detection of interactions aims at the analysis of gene expression profiles to uncover how combinations of genes are linked to specific diseases. The classification part aims at the improvement of classification rules.

Two main papers address the problem of the discovery of emerging patterns. While Dong and Li (1999) focus on an enumeration based algorithm to find all patterns with large empirical growth rates, Boulesteix et al. (2003) propose a CART-based method. Here, we suggest an improvement of the CART-based method developed in Boulesteix et al. (2003). The method allows to identify candidate patterns and only those which satisfy a statistical

criterion are selected as IPs. In addition, a pruning criterion is used to prevent too long and irrelevant IPs. A simpler version of the algorithm which is restricted to the case of two classes is given in Boulesteix et al. (2003). The present paper can be seen as an extension of Boulesteix et al. (2003) with respect to three important issues. First, the concept of IPs is mapped into a theoretical statistical framework. Second, various statistical aspects of IPs are investigated (e.g. receiver operating characteristic, length of the IPs, survival plot). Third, the concept of IPs as well as the discovering algorithm are adapted to handle multicategorical response variables: all the variables involved in the patterns are tested for relevance (not only the variable involved in the last splitting, as in Boulesteix et al., 2003).

The rest of the paper is organized as follows. In Section 2, we define IPs. In Section 3, we present the discovering method and algorithm to find IPs in data. Section 4 presents the results of the method for simulated data. In Section 5, we show how IPs can be used for classification and show the results obtained for very large data sets.

2. Definition of IPs

2.1. IPs for two classes

In this section, we first consider the binary case. For simplicity, the variables $x_1, \ldots, x_p$ are assumed to be metric, although the method is easily generalized to categorical variables. A pattern may be characterized as a collection of restrictions on a subset of variables $x_{j_1}, \ldots, x_{j_d}$. The restrictions have the simple form $x_j \leqslant \theta_j$ or $x_j > \theta_j$. Let I_j denote an interval of this type, then the restrictions are collected in

$$x_{j_1} \in I_1, \ldots, x_{j_d} \in I_d.$$

More formally, the restrictions may be represented as a subset of the observation space $\mathbb{R}^p$ or in terms of the underlying event. As subset of $\mathbb{R}^p$ they are given by

$$\left\{x | x_{j_1} \in I_1\right\} \cap \cdots \cap \left\{x | x_{j_d} \in I_d\right\}.$$

For random variables $x_1, \ldots, x_p$ the underlying event for pattern P is given by

$$P = A_{j_1} \cap \cdots \cap A_{j_d},$$

where $A_s = \{\omega | x_s(\omega) \in I_s\}$. The pattern P may be simply identified by the sequence of variables and corresponding intervals $\{(j_s, I_s), s = 1, \ldots, d\}$, where d is the order of the pattern. In addition, let $P_{\setminus j}$ denote the pattern where the restriction for variable j is omitted, i.e.

$$P_{\setminus j} = \bigcap_{i \in \{j_1, \ldots, j_d\} \setminus \{j\}} A_i.$$

The original pattern is easily obtained by $P = P_{\setminus j} \cap A_j$.

Definition 1. IPs for two classes

For $\eta > 1$, P is called a η-IP for class k_0 if

$$\frac{p(P|Y = k_0)}{p(P|Y \neq k_0)} > \eta \tag{2.1}$$

and for all $j \in \{j_1, \ldots, j_d\}$ the condition

$$\frac{p\left(P_{\setminus j}|Y = k_0\right)}{p\left(P_{\setminus j}|Y \neq k_0\right)} < \frac{p(P|Y = k_0)}{p(P|Y \neq k_0)} \tag{2.2}$$

holds.

In simple words, an IP is a condition on a collection of covariates for which the probability of occurrence is larger in one of the classes (Eq. (2.1)) and such that every involved covariate actually contributes to the ratio between the probabilities of occurrence within classes (Eq. (2.2)). The probabilities involved in the definition are unknown. Therefore, given a candidate pattern P, the data are used to decide if it is an IP fulfilling Eqs. (2.1) and (2.2). One option is to base the decision on a statistical test. For fixed k_0, condition (2.1) may be investigated by testing the hypothesis

$$\mathrm{H}_0^{(1)} : p(P|Y = k_0) \leqslant p(P|Y \neq k_0).$$

For simplicity $\eta = 1$ is used. Then testing of $\mathrm{H}_0^{(1)}$ is equivalent to one-sided independence testing in the following (2×2) contingency table with rows given by presence or non-presence of pattern P and columns defined by the classes.

	$Y = k_0$	$Y \neq k_0$	
P	n_{P,k_0}	$n_{P,\overline{k}_0}$	n_P
$\overline{P}$	$n_{\overline{P},k_0}$	$n_{\overline{P},\overline{k}_0}$	$n_{\overline{P}}$

In the contingency table P stands for presence of a specific pattern P and $\overline{P} = \mathbb{R}^p \setminus P$ denotes the non-presence of P. One can use for instance Fisher's exact test, which allows one-sided testing and is also valid for small numbers of observations. An overview on independence testing in contingency tables is given in Agresti (2002). The hypothesis $\mathrm{H}_0^{(1)}$ is rejected by the chosen independence test (for instance Fisher's test) to the significance level α_1 if $p^{(1)} < \alpha_1$, where $p^{(1)}$ denotes the p-value obtained by testing of $\mathrm{H}_0^{(1)}$. P is selected as an IP only if $p^{(1)} < \alpha_1$ holds. For the investigation of condition (2.2) it is useful to reformulate the condition. Since

$$\frac{p\left(P_{\setminus j}|Y = k_0\right)}{p\left(P_{\setminus j}|Y \neq k_0\right)} < \frac{p\left(P|Y = k_0\right)}{p\left(P|Y \neq k_0\right)}$$

is equivalent to

$$\frac{p\left(P_{\setminus j} \cap \overline{A}_j|Y = k_0\right)}{p\left(P_{\setminus j} \cap \overline{A}_j|Y \neq k_0\right)} < \frac{p(P|Y = k_0)}{p(P|Y \neq k_0)} \tag{2.3}$$

condition (2.2) may be investigated by one-sided independence testing in the following contingency table:

	$Y = k_0$	$Y \neq k_0$	
$P = P_{\setminus j} \cap A_j$	$n^{(j)}_{A,k_0}$	$n^{(j)}_{A,\overline{k}_0}$	n_P
$P_{\setminus j} \cap \overline{A}_j$	$n^{(j)}_{\overline{A},k_0}$	$n^{(j)}_{\overline{A},\overline{k}_0}$	$n_{P_{\setminus j}} - n_P$

Let $\gamma^{(j)}$ denote the associated odds ratio

$$\gamma^{(j)} = \frac{p\left(P \cap \{Y = k_0\}\right) / p\left(P \cap \{Y \neq k_0\}\right)}{p\left(P_{\setminus j} \cap \overline{A}_j \cap \{Y = k_0\}\right) / p\left(P_{\setminus j} \cap \overline{A}_j \cap \{Y \neq k_0\}\right)}.$$

Then, condition (2.3) can be reformulated as $\gamma^{(j)} > 1$. To investigate condition (2.3), one has to test for all j the hypothesis

$$\mathrm{H}_0^{(2,j)} : \gamma^{(j)} = 1 \quad \text{vs.} \quad \mathrm{H}_1^{(2,j)} : \gamma^{(j)} > 1.$$

An option is to use Fisher's one-sided independence test again. The hypothesis $\mathrm{H}_0^{(2,j)}$ is rejected by the chosen independence test to the significance level α_2 if $p^{(2,j)} < \alpha_2$, where $p^{(2,j)}$ denotes the p-value obtained by testing of $\mathrm{H}_0^{(2,j)}$. P is selected as an IP only if $\max_j p^{(2,j)} < \alpha_2$ holds, i.e. for all $j \in \{j_1, \ldots, j_d\}$, $\mathrm{H}_0^{(2,j)}$ has to be rejected.

The number of involved variables represents the order of the IP and is denoted by d. Patterns of order 1 are explicitly allowed. In the following, empirical IPs are simply denoted as IPs. The connection to emerging patterns is easily derived. In the emerging pattern literature which uses terminology from data mining the support is defined by $supp_k(P) = n_{P,k}/n_k$. This is an unbiased estimate of the probability $p(P|Y = k)$. The crucial difference between the present approach and the emerging pattern approach in data mining is that in the latter approach growth rates are simple descriptive tools and only condition (2.1) is investigated.

2.2. Generalization to multicategorical response

In practice, categorical variables often have more than two possible classes. In this section, we address the problem of multicategorical responses ($K > 2$) and propose a generalization of the definition of IPs.

Definition 2. IP for more than two classes

For $\eta > 1$, P is called a η-IP for the class k_0 if

$$\frac{p\left(P|Y = k_0\right)}{p\left(P|Y = k\right)} > \eta \tag{2.4}$$

holds for all k and for all j from $\{j_1, \ldots, j_d\}$ one has

$$\frac{p\left(P_{\setminus j}|Y = k_0\right)}{p\left(P_{\setminus j}|Y \neq k_0\right)} < \frac{p\left(P|Y = k_0\right)}{p\left(P|Y \neq k_0\right)}. \tag{2.5}$$

For fixed k_0, condition (2.4) may be investigated by testing the hypotheses

$$H_0^{(1,k)} : p(P|Y = k_0) \leqslant p(P|Y = k)$$

for all $k \neq k_0$. The hypothesis $H_0^{(1,k)}$ is rejected by the chosen independence test (for instance Fisher's test) to the significance level α_1 if $p^{(1,k)} < \alpha_1$, where $p^{(1,k)}$ denotes the p-value obtained by testing of $H_0^{(1,k)}$. For fixed α_1, P is selected as an IP if $\max_{k \neq k_0} p^{(1,k)} < \alpha_1$ holds.

Condition (2.5) can be investigated using the same procedure as for IPs for two classes.

3. Discovering IPs with trees

IPs and single leaves of classification trees have similar structures and properties. Thus, we propose to use the well-known and fast CART-algorithm proposed in Breiman et al. (1984) to discover IPs.

3.1. Tree methodology

Classification trees are an efficient exploratory tool to detect structures in data (Breiman et al., 1984). They are based on recursive partitioning whereby the measurement space $\mathbb{R}^p$ is successively split into subsets. Let $\mathbf{x}^T = (x_1, \ldots, x_p) \in \mathbb{R}^p$ denote the vector of covariates. If C is a subset of $\mathbb{R}^p$ (corresponding to the partitioning of $\mathbb{R}^p$ into C and $\overline{C} = \mathbb{R}^p \backslash C$), the split of C based on variable x_j divides C into

$$C_1(j, \theta) = \{\mathbf{x} \in C | x_j \leqslant \theta\},$$

$$C_2(j, \theta) = \{\mathbf{x} \in C | x_j > \theta\}.$$

Thus the subset C is split by use of one variable x_j, with the split simply depending on a threshold θ from the range of x_j. By starting with $C = \mathbb{R}^p$ and performing successive splittings one obtains a tree. After d splittings, one obtains subsets of $\mathbb{R}^p$ of the form

$$\{\mathbf{x}|x_{i_1} \leqslant \theta_1\} \cap \{\mathbf{x}|x_{i_2} > \theta_2\} \cap \cdots \cap \{\mathbf{x}|x_{i_d} \leqslant \theta_d\}.$$

A subset is identical to a pattern P given by the sequence $\{(j_s, I_s), s = 1, \ldots, d\}$ where j_s identifies the variable and I_s specifies the interval which in the simple case of binary splits has the form $I_s = (-\infty, \theta_s]$ or $I_s = (\theta_s, +\infty)$. The relationship between decision trees and patterns is simple: a pattern is equivalent to a leaf.

3.1.1. Splitting criterion

Given a pattern P of order d, an additional split in variable j at θ yields a $(d + 1)$-dimensional pattern. Let

$$P \cap A = P \cap \{\omega | x_j(\omega) \in I_j\}$$

denote the new pattern, where $I_j = (-\infty, \theta_j]$ or $I_j = (\theta_j, +\infty)$. Thus starting from P one obtains for the transition from P to $P \cap A$ the transition contingency table

	$Y=1$	...	$Y=K$
$P \cap A$	$n_{PA,1}$	...	$n_{PA,K}$
$P \cap \overline{A}$	$n_{P\overline{A},1}$	...	$n_{P\overline{A},K}$
	$n_{P,1}$	...	$n_{P,k}$

The margins $n_{P,k}$ for k from $\{1, \ldots, K\}$ represent the number of observations from class k in pattern P.

The new split is chosen to minimize a splitting criterion. One of the most common criteria is the deviance, also called cross-entropy, see Hastie et al. (2001). The deviance of a pattern P corresponds to the fit of the model

$$p(P|Y=1) = \cdots = p(P|Y=K).$$

Let n denote the total number of observations and n_k the number of observations from class k. The deviance has the form

$$\begin{aligned} D(P) &= 2 \sum_{k=1}^{K} \left\{ n_{P,k} \log \frac{n_{P,k}/n_k}{n_P/n} + n_{\overline{P},k} \log \frac{n_{\overline{P},k}/n_k}{n_{\overline{P}}/n} \right\} \\ &= 2 \sum_{k=1}^{K} \left\{ n_{P,k} \log \frac{\hat{p}(P|k)}{\hat{p}(P)} + n_{\overline{P},k} \log \frac{\hat{p}(\overline{P}|k)}{\hat{p}(\overline{P})} \right\} \\ &= 2 \sum_{k=1}^{K} n_k KL(\hat{p}(P|k), \hat{p}(P)), \end{aligned}$$

where $n_P = \sum_{k=1}^{K} n_{P,k}$, $\hat{p}(P|k) = \frac{n_{P,k}}{n_k}$, $\hat{p}(P) = \frac{n_P}{n}$, and KL stands for the Kullback–Leibler distance

$$KL(p,q) = p \log \frac{p}{q} + (q-p) \log \frac{1-p}{1-q}.$$

The new split which characterizes A is chosen to minimize the conditional deviance $D(P \cap A|P)$ given by

$$D(P \cap A|P) = D(P \cap A) - D(P)$$

and tests the hypothesis

$$p(P \cap A|Y=1) = \cdots = p(P \cap A|Y=K)$$

given $p(P|Y=1) = \cdots = p(P|Y=K)$. The conditional deviance can also be written as

$$D(P \cap A|P) = 2 \sum_{k=1}^{K} n_{P,k} KL(\hat{p}(P \cap A|k), \hat{p}(P \cap A)).$$

Various other splitting criteria have been used to grow trees, for instance the Gini-Index or the misclassification error, see Hastie et al. (2001).

3.1.2. Stopping criterion

The splitting criterion characterizes the way the tree is grown. In addition a stopping-criterion has to be chosen. In the tree literature, various stopping criteria have been proposed, for instance by Breiman et al. (1984). Let us consider a leaf P. One can decide not to split this leaf if its order exceeds a fixed number, if it contains less than a fixed number of observations or if the best split would yield at least one leaf with less than a fixed number of observations. Many other more sophisticated methods to limit the depth of trees such as cost-complexity pruning described in Hastie et al. (2001) have been investigated.

3.2. Discovering algorithm

When using trees for the detection of IPs the main problem is that trees are constructed by recursive partitioning. What is an advantage in terms of computation time and structuring turns into a disadvantage since the leaves share splits in the same variables. In particular, all leaves share the same root splitting. Patterns that do not involve the root splitting variable will never be found by a single tree. Therefore the proposed algorithm is based on the growing of several trees which use different sets of variables from which the splitting starts.

The first stage is designed to find candidate patterns. Here candidate patterns are generated which are investigated in the following steps. The selection is directly based on classification trees. The iterative algorithm grows a tree on the available set of variables and then removes the variable that generates the first split from the available set of variables. Thus patterns result which include different sets of variables. In applications we use the CART-algorithm `tree` (Ripley, 1996) implemented in the `tree` library in `R` (R-Development-Core-Team, 2004) with the deviance as splitting criterion. As stopping criterion, we fix `mincut` (minimal number of observations to include in either child node) at 5, `minsize` (minimal allowed node size) at 10 and `mindev` (minimal ratio between within-node deviance and the root node deviance for the node to be split) at 0.01. These settings are the default values of the `R` program.

In a second stage, conditions (2.1) and (2.2) resp. (2.4) and (2.5) are tested for the selected candidates patterns. The significance levels for the tests (α_1 and α_2) as well as the test T to be used (e.g. Fisher's exact test) have to be specified as input. The whole procedure can be summarized by the following algorithm.

Stage 1: *candidate patterns*: Grow a classification tree. Store the obtained leaves and eliminate the variable defining the first splitting of the tree from the set of input variables. Repeat this procedure until there is no more variable in the input set. Define S as the set of all obtained leaves.

Stage 2: *relevance of candidate patterns*: For each leaf from S, define k_0 as the class that maximizes $\hat{p}(P|k)$.

(1) For each leaf, for all $k \neq k_0$, test $H_0^{(1,k)}$ with test T to the significance level α_1. Eliminate from S all the patterns for which $\max_{k \neq k_0} p^{(1,k)} > \alpha_1$. This step corresponds to the testing of condition (2.1) resp. (2.4).

(2) For all the remaining leaves from S, test $H_0^{(2,j)}$ for all j in $\{j_1, \ldots, j_d\}$ with test T to the significance level α_2. If $\max_j p^{(2,j)} > \alpha_2$, eliminate the variable for which $p^{(2,j)}$ is maximal from the IP. Repeat this procedure as long as variables are eliminated. This step corresponds to the testing of condition (2.2) resp. (2.5).
(3) Repeat step 1 for all the leaves that have be shortened in step 2. This step is necessary to ascertain that the shortened patterns still fulfill condition (2.1) resp. (2.4).
(4) Eliminate from S all the duplicated patterns.

The algorithm yields empirical IPs which are based on tests with significance levels α_1 and α_2. Since many tests are performed the question of the overall significance level arises. This might be controlled for the given set of candidate patterns. It is, however, hard to control for the total procedure. Approaches to control the level for trees by maximally selected rank statistics are found in Lausen and Schumacher (1992). Instead of performing multiple testing, which would be very difficult in this framework, we follow an alternative approach by defining receiver operating curves which capture the performance of the algorithms for varying significance levels. This topic is addressed in the following section, where it is shown that the algorithm can detect 'ideal' theoretical IPs with quite good accuracy.

3.3. Receiver operating characteristic

A popular method for summarizing the accuracy of a classification rule are receiver operating characteristic (ROC) curves. A ROC curve is a plot of the true-positive rates against the false-positive rates. In classification, curves result from the consideration of varying thresholds on the diagnostic scale. Let a disease be diagnosed if the diagnostic scale is larger than threshold γ. Then the true-positive and false-positive rates are functions of the threshold. The resulting ROC curve is convex under quite natural assumptions. A large body of literature deals with the concept and estimation of ROC curves. An early reference is Swets and Pickett (1982), more recent approaches to estimation are proposed in Lloyd (2000) and Venkatraman (2000). A version of the ROC curve is suggested here to illustrate the power of the method for detecting relevant interactions. The empirical ROC curve shows the hit rate HR (or sensitivity) against the false alarm rate FAR (or specifity), where HR and FAR depend on the parameters α_1 and α_2 and on the order of the IPs. Let, for example, the order of the IPs be fixed at $d = 2$, i.e. only pairs of variables are investigated. If p is the total number of variables, the total number of possible pairs of variables is $p(p-1)/2$. For each possible pair of variables, two binary variables are defined: r, which equals 1 if the pair forms a real IP of order 2 and 0 else, and d, which equals 1 if the pair is detected as an IP of order 2 by our method and 0 else. For each parameter setting (α_1 and α_2), we are interested in the following contingency table:

	d	$\overline{d}$	Σ
r	$n_{r,d}$	$n_{r,\overline{d}}$	n_{IP}
$\overline{r}$	$n_{\overline{r},d}$	$n_{\overline{r},\overline{d}}$	$p(p-1)/2 - n_{\mathrm{IP}}$

The hit rate (HR) is defined as the proportion of discovered IPs among the n_{IP} real IPs, i.e.

$$HR = \frac{n_{r,d}}{n_{\mathrm{IP}}}.$$

Similarly, the false alarm rate (FAR) is defined as the proportion of patterns which were detected as IPs among the non-IP patterns of the same order, i.e.

$$FAR = \frac{n_{\bar{r},d}}{p(p-1)/2 - n_{\mathrm{IP}}}.$$

3.4. Simulation study

3.4.1. Study design

In a simulation study it is investigated if the algorithm is able to detect simulated patterns. To make the problem more simple and reduce the number of parameters in the study, we consider only the case of two classes. Simulated data are obtained by the following procedure. The number of variables contained in the data set is fixed at $p = 50$ and the number of observations is varied ($n = 50, 80$). These sample sizes correspond roughly to the typical values found in real gene expression data sets. From the 50 variables, 20 variables form pairwise IPs (variable 1 forms an IP with variable 2, variable 3 with variable 4, and so on). The two threshold values defining each pattern are drawn randomly from the uniform distribution in $[0.25, 0.75]$. The type of inequality defining the pattern ($\leqslant$ or $>$) are also chosen randomly. Thus, various data configurations are obtained. In the subsets defined by the pattern and in its complement, the distribution is uniform. The rest of the 50 variables are generated randomly and independently of the class, following the uniform distribution in $[0, 1]$.

The simulation study is designed as follows. We generate 100 random data matrices following the procedure described above. Then the discovering algorithm is run on each data matrix with different values of α_1 and α_2. HR and FAR are estimated for each parameter setting from the contingency tables obtained for the 100 random data matrices. If an IP for class 1 is detected as an IP for class 2 or vice-versa, the IP is considered as false alarm. Finally the means across simulations are built.

3.4.2. Simulation results

Fig. 1 displays the estimated ROCs for two values of n ($n = 50$ and 80): the hit rate is represented against the false alarm rate for different values of α_1 (ranging from 10^{-20} to 10^{-2}) and α_2 (ranging from 10^{-14} to 10^{-4}). It is seen that for decreasing significance levels the ROCs rather soon become horizontal, signaling a stable level of detection rate with the level depending on sample size. Within this stable level the increase of significance levels only increases the false alarm rate. Fig. 2 displays the boxplots of the hit rate and false alarm rate for $n = 50$ and 80, for three parameter settings which correspond to different zones of each ROC curve. As can be seen from Fig. 2, the variance of the hit rate and false alarm rate across the 100 simulations is quite low, although not negligible.

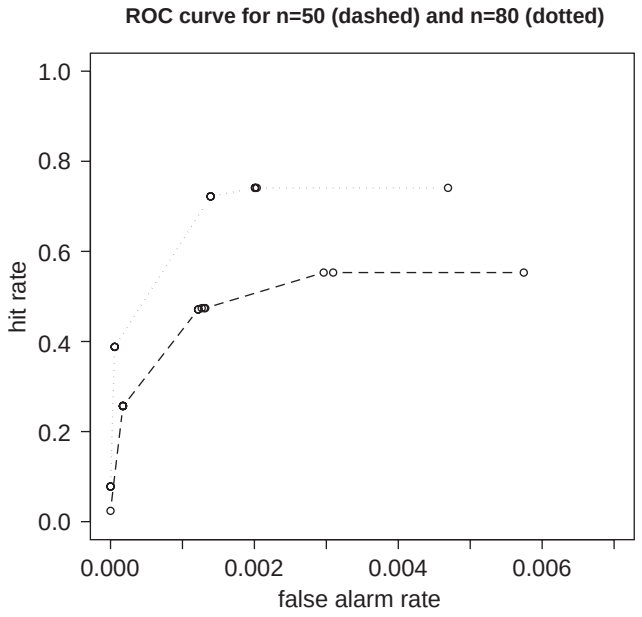

Fig. 1. ROC curve for $n = 50$ and 80 (simulated data).

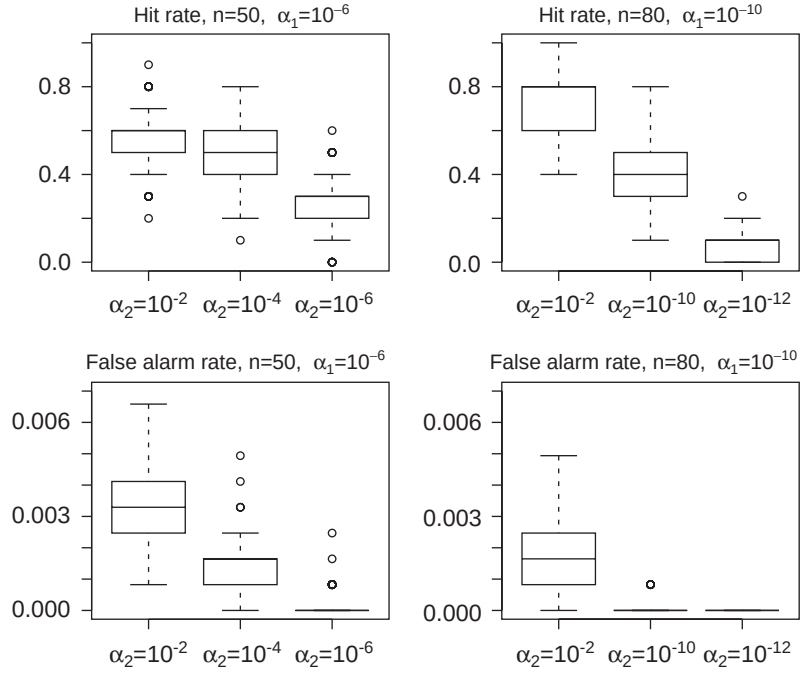

Fig. 2. Boxplot of the hit rate (top) and false alarm rate (bottom) for $n = 50$ (left) and 80 (right), for different values of α_1 and α_2.

4. Classification based on IPs

4.1. Method

As can be seen from their definition, IPs might be useful to define predictors for classification. An inconvenience of the CART approach for data sets with many variables and few observations is that the tree often consists of few splittings. If one stops growing the tree too late, then some splittings might be statistically irrelevant. And if the growing is stopped too early, the decision rule depends on very few variables, and does not use most of the potentially interesting variables from the data set. By using IPs instead of tree leaves as a basis for the decision rule, one avoids a major problem: the decision rule uses much more information from the data set than a single tree does. In the following, a simple method to use IPs for classification is proposed. It is particularly suited for data sets with many (metric or categorical) variables and few observations. It can also be used for data sets with fewer variables, however without spectacular gain in accuracy.

From now on, we suppose that we have a learning data set $\mathscr{L}$ and a test data set $\mathscr{T}$. To predict the class of the observations from $\mathscr{T}$, we proceed as follows: First, IPs are found by applying the discovering algorithm on the training set $\mathscr{L}$. Second, m new binary covariates $Z_1, \ldots, Z_m$ are defined, where m denotes the number of found IPs. The variables

$$Z_j = \begin{cases} 1 & \text{for the } j\text{th IP}, \\ 0 & \text{otherwise}, \end{cases}$$

indicate if the considered observation fulfills the conditions defining the considered IP. One obtains a transformed learning data set and a transformed test data set. Then virtually any supervised learning method can be applied to these data matrices, for instance linear discriminant analysis (with Bayes or maximum-likelihood rule), nearest neighborhood, logistic regression (if m is not too large), etc.

4.2. Study design

Fifty random partitions into a learning data set $\mathscr{L}$ (containing $n - 10$ observations) and a test data set $\mathscr{T}$ (containing 10 observations) are generated. For each partition, we proceed as follows. If the number of variables is high, a prescreening step is necessary. It is done by selecting the $\tilde{p}$ variables with lowest p-value for Wilcoxon's test testing the equality of the median in two classes, using only $\mathscr{L}$, as described in Dettling and Bühlmann (2003). If the number of classes K is greater than 2, the procedure is repeated K times: for the K classes successively, one tests the equality of the medians in the considered class and in all the other classes together. Then K groups of variables are selected. An alternative, which might seem more appropriate for multiclass problems, is to use the Kruskal–Wallis statistic. One applies the Kruskal–Wallis test to all genes and selects the $\tilde{p}$ genes with lowest p-values. However, the results obtained with this method are worse than with our procedure. One possible explanation is that the variables selected by the Kruskal–Wallis statistic do not necessarily separate well all K classes.

A prescreening is performed for three of the four investigated data sets: the leukemia, the colon and the SRBCT data sets, which are described in the following subsection. For each

data set, the number of selected variables is fixed successively at $\tilde{p} = 50$, 100, 200 and 300. These values have been chosen, because for greater values of $\tilde{p}$, the discovering algorithm is computationally very intensive and for lower values of $\tilde{p}$, the number of found IPs is too low (or even zero for some of the partitions).

We run the discovering algorithm to find IPs, with different values for the parameter α_1 and $\tilde{p}$. To reduce the number of parameters, α_2 was fixed at 10^{-4}. α_1 is chosen on a heuristic basis. It is chosen so that the number of found IPs is non zero and smaller than, say 200 for all the partitions. For the tree topology parameters, the default values of the R program as described in Section 3 are used.

Once the IPs are found, the new covariates are determined for all observations from $\mathscr{L}$ and $\mathscr{T}$. Then classification is carried out, either with nearest neighborhood classification based on 5 nearest neighbors (5-NN) or with linear discriminant analysis. Since the results were slightly better with 5-NN, the results with linear discriminant analysis are not shown. For the nearest neighborhood classification, the Euclidean distance was used.

Mean error rates over the 50 *partitions*: For each parameter combination, the mean error rate over the 50 random partitions (i.e. the mean proportion of observations from the test set that were misclassified) is computed. The results are summarized in a table. For comparison, we also show the mean error rate obtained with classical CART, using the same R program as in the discovering algorithm, and with 5-NN applied directly on the $\tilde{p}$ genes. The latter is known to be one of the best performing discrimination methods for microarray data (Dudoit et al., 2002).

Observation-wise error rate: For each parameter combination and for each single observation, the proportion of times it was misclassified (out of the runs in which it was in the test set) is recorded. We summarize the results by means of survival plots as described in Dudoit et al. (2000): the proportion of observations classified correctly in at least V% of the runs is represented against V. The results are shown only for the best parameter combination for each data set.

Variables involved in IPs: An interesting issue is whether the variables involved in the IPs also perform good individually. To answer this question, we first rank the variables according to the Wilcoxon-statistic using all the observations. Then we represent the proportion of runs in which the variables were selected against their rank. We show the results for the colon data and the leukemia data with $\tilde{p} = 300$ and $p_G = 10^{-6}$ (for colon) and $p_G = 10^{-10}$ (for leukemia).

Number of IPs: The number of found IPs depends highly on the parameters. Typically, it increases with $\tilde{p}$ and α_1. The number of found IPs of each order is stored each time the discovering algorithm is run. The results are summarized by plotting the mean number of found IPs of each order over the 50 random partitions, for each data set and for different values of α_1. For the 3 gene expression data sets (leukemia, colon, SRBCT), we show only the results for $\tilde{p} = 300$. For smaller values of $\tilde{p}$ the plots show similar patterns, but the absolute numbers of IPs are lower.

4.3. Data sets

Leukemia data: This data set was introduced in Golub et al. (1999) and contains the expression levels of 7129 genes for 47 ALL-leukemia patients and 25 AML-leukemia pa-

tients. It is included in the R library golubEsets. After data preprocessing following the procedure described in Dudoit et al. (2002), only 3571 variables remain. It is easy to achieve excellent classification accuracy on this data set, even with quite trivial methods as described in the original paper (Golub et al., 1999). Indeed, we found out that it is possible to find many IPs even if α_1 is very low. Thus, we set α_1 to $\alpha_1 = 10^{-10}$, 10^{-12} and 10^{-14} successively in our study.

Colon microarray data: The colon data set is a publicly available 'benchmark' gene expression data set which is extensively described in Alon et al. (1999). It can be downloaded from the web page http://microarray.princeton.edu/oncology/affydata/. The data set contains the expression levels of $p = 2000$ genes for $n = 62$ patients from two classes. Twenty two patients are healthy patients and 40 have colon cancer. This data set is not as 'easy' as the leukemia data set. The classification accuracy is usually much lower, for instance using Support Vector Machines as described in Furey et al. (2000). It is also more difficult to find good IPs: α_1 was set heuristically to $\alpha_1 = 10^{-6}$, 10^{-8} and 10^{-10}. Note that it is also possible to run the algorithm with $\alpha_1 = 10^{-12}$ and 10^{-14} as for the leukemia data set, but with such values for α_1, no IP would be found.

SRBCT microarray data: This gene expression data set is presented in Kahn et al. (2001). It can be downloaded from http://www.thep.lu.se/pub/Preprints/01lu_tp_01_06_supp.html.

It contains the expression levels of 2308 genes for 83 Small Round Blue Cells Tumor (SRBCT) patients belonging to one of the 4 tumor classes: Ewing family of tumors (EWS), non-Hodgkin lymphoma (BL), neuroblastoma (NB) and rhabdomyosarcoma (RMS). For this data set, α_1 was set to $\alpha_1 = 10^{-3}$, 10^{-4}, 10^{-5} and 10^{-6}. These values are considerably higher than for the leukemia and colon data sets. One of the possible explanations is that to be selected as an IP of type k ($k \in \{1, 2, 3, 4\}$), a pattern must have higher frequency in class k than in all three other classes, which is a stronger requirement than for the two-classes case.

Iris data: The famous (Fisher's and Anderson's) iris data set is included in the R library MASS. It gives 4 different measurements (sepal length and width, petal length and width) for 150 flowers from each of the three species (class labels) setosa, versicolor, virginica. α_1 was set successively to $\alpha_1 = 10^{-4}$, 10^{-8} and 10^{-12}.

4.4. Results

Mean error rate: The mean error rates for different values of the parameters are shown in Table 1 for the four data sets. For all four data sets, the new method performs much better than CART and is comparable to nearest neighborhood classification. Thus it is a competitor to one of the best classification procedures in microarray data with the advantage of providing information on the relevance of variables and IPs. Surprisingly, the number of variables as well as the significance level α_1 do not seem to have strong influence on the results, provided IPs are found. For the case of two classes the method may be compared to the method suggested in Boulesteix et al. (2003). It turns out that the classification results with the new method are as good as with the former method for the colon data and better for the leukemia data.

Table 1
Mean error rate over 50 random partitions

Colon data	$\alpha_1 = 10^{-6}$	$\alpha_1 = 10^{-8}$	$\alpha_1 = 10^{-10}$	tree	5–*NN*
50 variables	0.16	0.17	0.19	0.30	0.16
100 variables	0.14	0.14	0.16	0.30	0.14
200 variables	0.15	0.15	0.15	0.29	0.15
300 variables	0.15	0.15	0.15	0.29	0.15
Leukemia data	$\alpha_1 = 10^{-10}$	$\alpha_1 = 10^{-12}$	$\alpha_1 = 10^{-14}$	tree	5–*NN*
50 variables	0.042	0.042	0.042	0.15	0.042
100 variables	0.025	0.025	0.025	0.15	0.025
200 variables	0.016	0.016	0.016	0.15	0.016
300 variables	0.016	0.016	0.016	0.15	0.016
SRBCT data	$\alpha_1 = 10^{-4}$	$\alpha_1 = 10^{-5}$	$\alpha_1 = 10^{-6}$	tree	5–*NN*
20 variables	0.0077	0.0077	0.0080	0.25	0.0077
50 variables	0.0046	0.0046	0.0048	0.25	0.0046
Iris data	$\alpha_1 = 10^{-4}$	$\alpha_1 = 10^{-8}$	$\alpha_1 = 10^{-12}$	tree	5–*NN*
	0.035	0.035	0.035	0.059	0.035

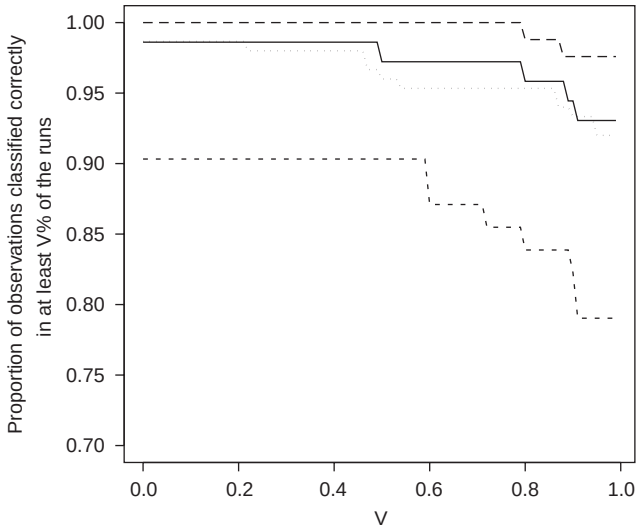

Fig. 3. Survival plot for leukemia (solid), colon (dashed), SRBCT (longdash) and iris (dotted).

Observation-wise error rate: As can be seen from the survival plot depicted in Fig. 3, a large part of the error rate is due to observations that are misclassified each time they are included in the test data set. Indeed, even for small *V*, the proportion of observations classified correctly in at least *V*% of the runs is not 1, and it decreases slowly for large *V*.

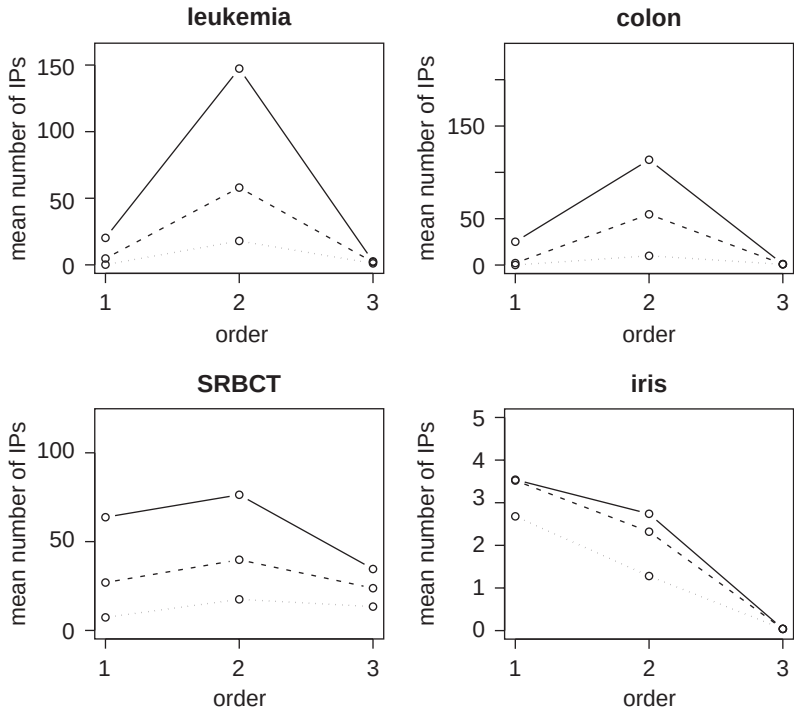

Fig. 4. Number of IPs of each order. Leukemia: $\tilde{p}=300$ and $\alpha_1=10^{-10}$ (solid), $\alpha_1=10^{-12}$ (dashed), $\alpha_1=10^{-14}$ (dotted). Colon: $\tilde{p}=300$ and $\alpha_1=10^{-6}$ (solid), $\alpha_1=10^{-8}$ (dashed), $\alpha_1=10^{-10}$ (dotted). SRBCT: $\tilde{p}=50$ and $\alpha_1=10^{-4}$ (solid), $\alpha_1=10^{-5}$ (dashed), $\alpha_1=10^{-6}$ (dotted). Iris: $\alpha_1=10^{-4}$ (solid), $\alpha_1=10^{-8}$ (dashed), $\alpha_1=10^{-12}$ (dotted).

We found out that most of the 'problematic' observations are also misclassified by other classification methods (data not shown).

Number of IPs: As can be seen from Fig. 4, the most frequent IPs are IPs of order 2. We did not find any IP of order 4, and few IPs of order 3. If the data sets contained more observations, it would certainly be possible to find more IPs of order 3 and 4 (or more). IPs of order 1 are quite frequent and correspond to variables that can separate the classes well. Unsurprisingly, the number of found IPs increases with α_1. An important fact which cannot be seen in the figure is the high variability of the numbers of IPs over the random partitions: like CART, our learning method is not very robust, which can be seen as a drawback from the statistical point of view.

Variables involved in IPs: As can be seen from Fig. 5 (for the colon and the leukemia data sets), most of the 'best' variables appear in at least one IP in most runs. But some 'less relevant' variables are involved in IPs in many runs as well, thus showing that variables that perform poorly individually might be interesting in association with others. On the whole, there seems to be a weak linear dependence between the variable rank and the frequency of selection. Separate analysis for IPs of order $1, 2, 3$ would probably show

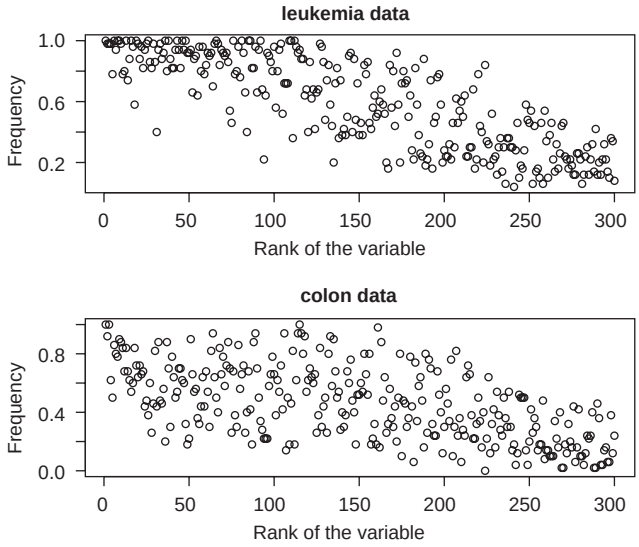

Fig. 5. Proportion of runs in which the variable is involved in at least one IP.

stronger dependence for IPs of order 1 than for IPs of order 2 and 3. In the next section, we give an example on how interactions patterns can be interpreted in practice.

4.5. An example

In this section, we illustrate the concept of IPs using a concrete example from the colon data. Since the goal is not the evaluation of the classification performance but the identification of relevant patterns, the discovering algorithm is run on the whole colon data set with $\alpha_1 = 10^{-10}$ and $\alpha_2 = 10^{-6}$. The discovering algorithm outputs a list of 9 IPs. For example, the genes R55310 and H72234 are found to form an IP for class 1 (normal tissue) which is defined by the restrictions

$$R55310 > 0.40$$

and

$$H72234 < -0.1$$

as depicted in Fig. 6. The corresponding biological hypothesis can be formulated as "in normal tissues, gene R55310 has a high expression level and gene H72234 has a low expression level". Hypotheses of this type might be used as a basis for the design of biological experiments.

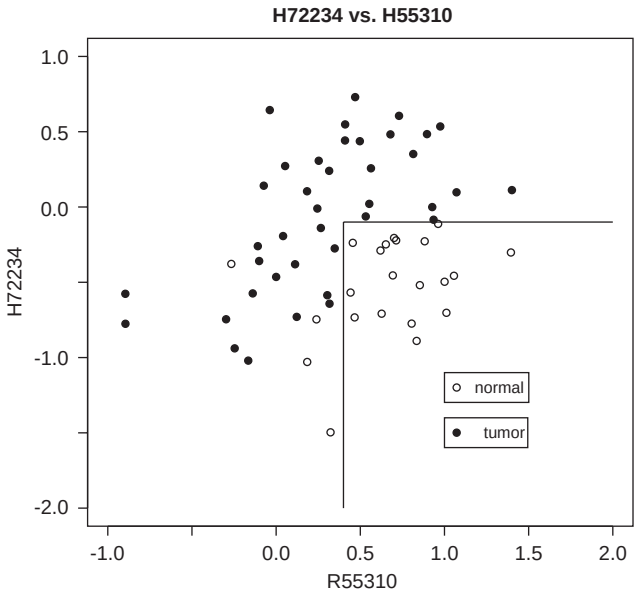

Fig. 6. An IP from the colon data.

5. Discussion

CART is one of the most popular classification methods in many application fields of statistics, for instance medicine. The main advantages that make it so popular are its simplicity and its interpretability. Moreover, scientists are often interested in the interaction structures implied by the CART decision rules. However, when the number of variables is high and the number of observations small, like in microarray data, CART usually performs poorly, because it uses only a very small part of the available information. Among the huge number of variables, it is often possible to find a few that separate the classes very good or even perfectly in the learning set. Thus, the obtained trees have very short branches and often perform poorly on new data sets. Modern methods based on aggregation of trees do improve the results a little as argued in Dudoit et al. (2002), but do not seem to overcome the problem completely. Instead of partitioning the input space like in CART, our method defines a wide collection of leaves with non-empty intersection, thus allowing more robust classification.

Another advantage of our classification method is its interpretability in terms of interaction structures. This is a very important issue for applied scientists, especially those working on gene expression data. Indeed, although it is almost certain that genes somehow interact, the challenging question of modelling these interactions remains partly unanswered. The proposed method can detect quite successfully IPs in simulated 'perfect' data.

The proposed approach differs significantly from Dong and Li's approach in several aspects. First, we use a statistical criterion to define the patterns instead of the heuristic growth rate. Second, while Dong and Li find patterns of high order, we argue that short

pattern involving only relevant variables are preferable, in order to avoid overfitting of the learning data. Therefore, condition (2.2) was added in the definition. Third, the method to detect the patterns is completely different: while Dong and Li perform a dramatic variable selection and enumerate all the possible patterns built with the selected variables, we use a CART-based algorithm which accelerates the search considerably and do not necessitate such a dramatic variable selection. The approach described in Boulesteix et al. (2003) may be seen as a simplification of the method for binary responses. The search algorithm is similar, but the testing of condition (2.2) is replaced by a pruning step while building the trees. Thus, only the variables involved in the subsequent splittings can be eliminated from a pattern. This approach is often appropriate for binary responses, since the successive splittings of the trees are chosen to minimize the deviance. However, it is too restrictive for multicategorical responses or for highly correlated predictors. The proposed definition and search algorithm overcome this inconvenience and generalize the framework developed in Boulesteix et al. (2003).

References

Agresti, A., 2002. Categorical Data Analysis, Wiley, New York.

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

Boulesteix, A.L., Tutz, G., Strimmer, K., 2003. A cart-based approach to discover emerging patterns in microarray data. Bioinformatics 19, 2465–2472.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, J.C., 1984. Classification and Regression Trees, Wadsworth, Monterey, CA.

Dettling, M., Bühlmann, P., 2003. Boosting for tumor classification with gene expression data. Bioinformatics 19, 1061–1069.

Dong, G., Li, J., 1999. Efficient mining of emerging patterns: discovering trends and differences. Proceedings of the SIGKDD (5th ACM International Conference on Knowledge Discovery and Data Mining), San Diego, USA, 43–52.

Dudoit, S., Fridlyand, J., Speed, T.P., 2000. Comparison of discrimination methods for the classification of tumors using gene expression data. Technical Report 576: http://www.stat.berkeley.edu/tech-reports/index.html.

Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77–87.

Friedman, J.H., Fisher, N., 1999. Bump hunting in high-dimensional data. Statistics and Computing 9, 123–143.

Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16, 906–914.

Golub, T., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537.

Hastie, T., Tibshirani, R., Friedman, J.H., 2001. The elements of statistical learning, Springer, New York.

Kahn, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., Meltzer, P.S., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat. Med. 7, 673–679.

Lausen, B., Schumacher, M., 1992. Maximally selected rank statistics. Biometrics 48, 73–85.

Li, J., Wong, L., 2003. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. Bioinformatics 19, 71–78.

Lloyd, C.J., 2000. Regression models for convex ROC curves. Biometrics 56, 562–567.

Morgan, J.N., Sonquist, J.A., 1963. Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association 58, 415–435.

R-Development-Core-Team, 2004. R: a language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria, iSBN 3-900051-00-3.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge.

Swets, J., Pickett, R., 1982. Evaluation of Diagnostic Systems; Methods from Signal Detection Theory, Academic Press, New York.

Venkatraman, E.S., 2000. A permutative test to compare receiver operating characteristic curves. Biometrics 56, 1134–1138.