Università degli Studi di Napoli
Federico II

3Way Classification and Regression Trees

Methods, Computations and Applications

Valerio Aniello Tutore

Tesi di Dottorato di Ricerca in Statistica
XX Ciclo

FRIDERICUS·SECUNDUS·ROMANORUM·IMPERATOR·SEMPER·AUGUST

Dipartimento
di Matematica e Statistica
Università degli Studi di Napoli "Federico II"
via Cintia, Monte Sant'Angelo – 80126 Napoli

Contents

List of Tables

List of Figures

Ringraziamenti

Sono molte le persone che sento di dover ringraziare per l'aiuto ricevuto in questi tre anni di ricerca. Tre anni intensi sia sotto il profilo scientifico, ma, soprattutto, formativi da un punto di vista umano.
Innanzitutto voglio esprimere la mia profonda gratitudine alla prof. Roberta Siciliano, per l'appoggio costante che mi ha sostenuto in questo percorso e per l'attenzione con cui mi ha sempre ascoltato.
Un pensiero particolare va al prof. Carlo Lauro, per i preziosi consigli che nel corso di questi tre anni non ha mai mancato di fornirmi.
Un grazie va al dott. Massimo Aria per gli innumerevoli aiuti che da lui ho ricevuto, al prof. Franco Mola per avermi fatto capire che con la statistica ci si può anche divertire e al prof. Antonio Mango per i costanti inviti a cercare soluzioni alternative attraverso il confronto con gli altri.
Molte sarebbero le persone da citare per il periodo trascorso a Leiden nei Paesi Bassi: mi limito a ricordare il prof. Ab Mooijaart con cui ho potuto a lungo lavorare su temi presenti anche in questo lavoro e il prof. Willem Heiser per la sua capacità di passare in pochi passi dal particolare al generale.
Grazie a tutti i membri del Dipartimento di Matematica e Statistica e agli amici dottorandi con cui ho condiviso momenti davvero intensi.
Un grazie dal profondo del cuore va alla mia famiglia, i miei genitori e le mie sorelle, per aver sopportato i miei continui sbalzi di umore e,

soprattutto, a Stefania che mi è sempre stata vicina anche quando ci separavano duemila chilometri.
Grazie a tutti
Valerio

Introduction

This thesis is about tree methods. A tree is an oriented graph formed by a finite number of *nodes* departing from the so-called *root node* of the tree structure. That can be distinguished between *nonterminal nodes* in circles and the *terminal nodes* in squares. In binary trees, each parent node is linked to only two children nodes, namely the left node and the right node as in figure (1). Each node has a number such that node t generates the left node $2t$ and the right node $2t + 1$. In this way, it is always possible to recognize the position of each node, given its number, deriving the path from the node to the root node and viceversa. As an example, in figure 1, the node 10 is the left node of its parent node 5 which is the right node of its parent node 2 which is the left node of the root node. A *branch* of the tree is a subtree obtained pruning the tree at a given internal node.

The first utilization of binary trees goes back to AID (Automatic Interaction Detector) software proposed by Morgan and Sonquist in 1963 [81], where the split criterion is to maximize the Between Sum of Squares (BSS) and to minimize the Within Sum of Squares (WSS). The CHAID procedure, presented by Kass (1980), is referred to the case in which the response variable is a two or more nominal classes. A turning point in history of binary trees is represented by CART (Classification and Regression Trees) proposed by Breiman, Freidman, Olshen and Stone in 1984 [15]. This technique has two goals: the pre-

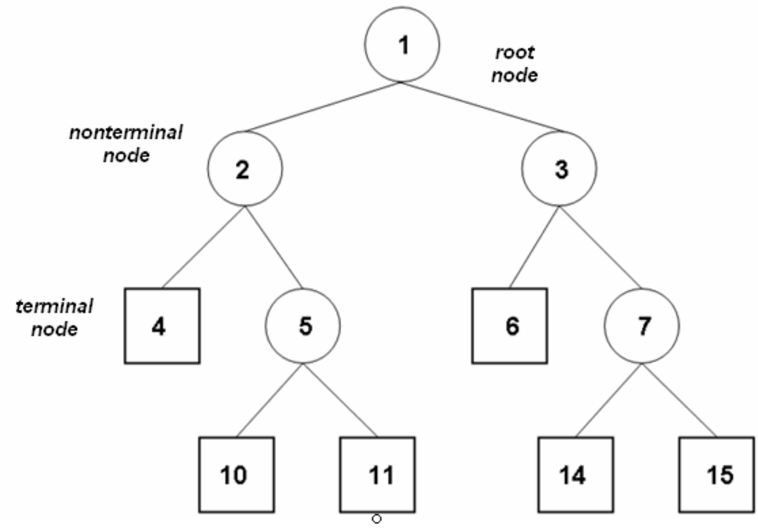

Figure 1: A tree structure

diction of a categorical variable *Classification Tree* and the prediction of a continuous variable *Regression Tree.* The CART procedure introduces several innovations: a split criterion based on impurity, the cross-validation, the pruning, the possibility to tract together nominal and continuous predictors, the processing of missing values.
C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan [86]. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often considered as a statistical classifier.

Several contributions on tree-based methods have been further developed in literature, among them this thesis takes inspiration in particular from the studies of the research group of Naples starting from the TWO-STAGE methodology of Mola and Siciliano introduced in 1992 and considering the further results also provided by other researchers, namely Conversano, Cappelli and Aria. Main mission of this group is to provide alternative methodological and computational solutions to classification and regression trees for the analysis of complex data structures. The research work resulted in several papers, computational routines and specialized software platforms, the recent one is Tree Harvest Software, developed in MATLAB environment.

In the following, the methodological resarch point of view considered by this research group will be described and the structure of this thesis with a summary of the main contributions will be presented.

Trees can be fruitfully used in modern statistics characterized by data analysis under complexity. The latter can be viewed in terms of *Data* (i.e., large sets, longitudinal sets, multivariate, etc), *Information* (i.e., data structure assumption, use of stratifying variables, constraints, missing data, etc.), *Knowledge Extraction* (i.e., modelling, procedures, strategies, etc.).

Modern statistic is funded on the paradigm of *learning from data* to provide information which is *statistically reliable* and *with an added value* in terms of problem solving and knowledge discovery. Actually, the cycle path connecting data, information and knowledge describes *the knowledge discovery process* whose aim is to determine proper actions for decision-making in business management as well as in noprofit organizations.

In this framework, segmentation procedures and trees can be fruitfully used either stand-alone or in combination with other methods in order to satisfy specific purposes of the knowledge discovery process when dealing with real world problems. Typically, two kinds of questions can be posed, namely to explore data (i.e., the path from data to information) as well as to model data (i.e., the path from information to knowledge).

On the one hand, exploratory trees provide a partition of objects into internally homogeneous groups with respect to a given response variable. The oriented tree graph simplifies a lot the interpretation of the dependence relations of the response variable on the given predictors showing special patterns, paths and typologies. On the other hand, decision trees, as a predicting method, allow, for a non parametric approach to classification and regression modeling, to assign a response class/value to a new object for which only measurements of the predictors are known. Tree-based decision rules have been recently applied also in the field of data editing when dealing with large databases, in particular as a non-parametric missing data imputation and an automated procedure for data validation.

Exploratory trees can be grown out of TWO-STAGE splitting criteria ([79]) which optimize first a global impurity reduction factor with respect to the candidate predictors and then a local impurity reduction factor with respect to the candidate splitting variables. In this way, a global role is played by the predictors in explaining the dependence relations with the response variable. As an alternative to CART-based trees, a Fast splitting Algorithm for Splitting Trees (FAST, [77]) will greatly decrease the computational cost of the partitioning procedure, in terms of number of splits to be tried out for find the best one in each node of the tree.

In this context, it is necessary to consider decision trees derived from the application of a tree induction procedure aiming to identify a decision rule for new objects. A common approach is to cut off the not significant and weakest branches of the tree in terms of goodness of fit by pruning the tree according to either a top-down or a bottom-up algorithm. Then a model selection criterion allows to choose the final decision tree. A recent approach is to find a compromise of various decision rules applying either boosting or bagging. These approaches will be also presented.

The thesis is structured in six chapters. In first chapter will be presented a synthetic review of classical regression and classification methods with supervised approach in explorative field. The analyzed methodologies are CART [15], the others tree-based models, in particular, the proposed TWO-STAGE and FAST to reduce computational cost of the analysis and the proposed TS-Dis and MBL respectively to reduce the dimensionality of the data and to treat classification problems with multi-class response variable. The goal of this review is to give a complete methodological description relative to the explorative analysis in the study of large datasets.

Second chapter will treat decision trees. Explorative trees can be used to analyze the structure of data, but they cannot be used in inductive aim as they do not allow to suitably classify new observations. The goal is, therefore, to simplify decisional trees maintaining the underlying accuracy, so the explorative trees become a main step for the induction of the decisional ones. The chapter will expose different approaches that allow to reduce such dimensionality (Pruning) without losing accuracy of these results and, in this latter area, it becomes most important to study techniques that allow to have more robust structures (Ensemble methods).

Third chapter is relative to a new method called ”Optimal scaling trees” to treat e special problem in data mining: the presence of variables correlated each others in groups. In presence of these complex relations, standard tree-procedures offer unstable and not interpretable solutions. It becomes fundamental to have a priori treatment of data through some explorative techniques based on algorithms to treat large datasets and nonlinear relations. Segmentation methods seems to be a valid instrument for the analysis of this kind of data. In particular situations it becomes very important to investigate the rule that every single variable plays in explaining the response variable. Therefore, in presence of structures of complex data, characterized by groups of variable internally correlated and hierarchically connected to the structure of synthesis, we propose to create nodes from splits obtained by using Nonlinear Canonical Correlation Analysis (NLCCA), a generalization of linear CCA.

In fourth chapter, we will introduce the concept of 3-way matrix. These matrices are characterized by the presence of a instrumental variable that allows to divide individuals with respect to categories of the variable itself. Moreover, it will be also shown a different vision of the above matrices: in fact, according to a different point of view, we will not reason only in terms of groups of individuals, but also in terms of blocks of predictors, that by their nature, are correlated.

Fifth chapter is relevant to the proposal so-called ”Multiple Discriminant Trees” that presents an innovative method in the analysis of 3-way matrices with different blocks of variables. This method allows to replace these blocks with one of their linear combination applying Linear Discriminant Analysis. In particular, this analysis is applied in two different moments: firstly to synthesize each block of variables and then to find a compromise between all the blocks of variables previously synthesized. In this approach it is possible to measure the strength of each group of variables in the building of this compromise.

Sixth chapter presents the method called ”Partial Predictability Tree” that takes into consideration groups of individuals discriminated by the presence of an instrumental variable. It is considered the two-stage split criterion based on Goodman and Kruskal prediction index. In the first stage the best predictor is found maximizing the global prediction respect to response variable; in the second stage the best split of the best predictor is found maximizing the local prediction. Later this criterion is extended to consider the power of prediction explained by every predictor (or every split) with respect to the response variable conditioned by instrumental variable. For this reason the indexes of Gray and Williams are used.

Chapter 1

Tree partitioning

1.1 Introduction

A lot of nonparametric methods such as also segmentation procedures have been stimulated by real problems of data analysis. The most appealing aspect for the user of segmentation is the final tree that provides a comprehensive description of the phenomenon in different contexts of application such as marketing, credit scoring, finance, medical diagnosis, etc. As a matter of fact, users very often accept statistical results only if these confirm theoretical hypotheses on the phenomenon derived from prior knowledge. Thus, several open questions arise using such heuristic tools. In segmentation the most difficult ones to answer are which is the tree to consider for explanation of the dependence data structure and how to evaluate the accuracy of the final tree classifier/predictor if this is extended to unseen objects without considering any “inferential dogma”. This latter aspect makes segmentation methodology to be considered not only as an exploratory tool but also as a confirmatory nonparametric model, also known as decision rule. A distinction is made between the two problems of investigation of the data sets, namely whether to explore dependency or to predict and decide about future responses on the basis of the selected predictors. *Exploration* can be obtained by performing a segmentation of the objects until a given stopping rule defines the final partition of the objects to interpret. *Confirmation* is a completely different problem that requires definition of decision rules, usually obtained performing a pruning procedure right after a segmentation procedure. Pruning consists in simplifying trees in order to remove the most unreliable branches and improve the accuracy of the rule for classifying fresh cases. Unfortunately, a weak point of constructing decision trees is given by the sensitivity of the classification/prediction rules because of the size of the tree and its accuracy for the type of dataset and for the adopted pruning procedure. In other words, the ability of a decision tree to detect cases and take right decisions can be not only evaluated by a statistical index but it requires a more sophisticated type of analysis, known as the choice of the most honest tree.

As an example, for a problem of credit scoring there is the necessity to classify a firm into one of two classes, "admitted to the bank financing" and "not admitted" on the basis of various business indicators. Using a sample of firms on which business indicators are measured as also the class is known it is possible to build up a tree structure. In this case segmentation aims to an exploratory goal, that is to understand which indicators are discriminant and which are the most important interactions among such indicators. Exploratory trees provide a hierarchy of importance of the predictors. In addition, the tree structure could be also employed to classify a new firm of which the class is not known but the business indicators have been measured. For that reason a decision tree needs to be identified such that the answer is not influenced by the specific sample that has been considered to build up the tree structure. In this chapter exploratory trees will be treated whereas decision trees will be described in the next chapter.

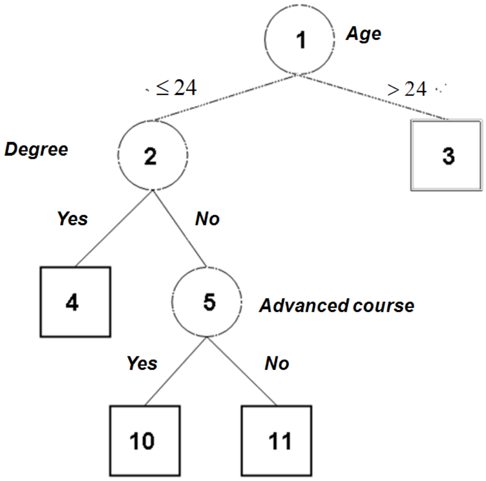

Figure 1.1: Some examples of *splits* in binary segmentation

1.2 Tree growing

1.2.1 Main steps

The idea of segmentation is to use a set of predictors (of categorical and/or numerical type) to partition recursively a sample of units into groups which are internally homogeneous and externally heterogeneous with respect to a response variable distribution. This is obtained maximizing the decrease of impurity at each node of the tree [15], where impurity can be evaluated by heterogeneity for classification tree (if the response is of categorical type) and by variation for regression trees (if the response is of numerical type).

As a result, the sample of objects in the root node is finally partitioned into a set of disjoint and exhaustive groups represented by the terminal nodes of the tree, each of them is labeled with either a response value (for regression trees) or a class (for classification trees). By definition, the terminal nodes present a low degree of impurity compared to the root node.
In tree growing predictors generate candidate partitions (or splits) at each internal node of the tree, so that a suitable criterion needs to be defined to choose the best partition (or the best split) of the objects. In the tree structure as in figure (1.1) it is possible to read the conditional interactions among the predictors to explain the behavior of the response variable.

Any segmentation methodology is defined by the following steps:

- *the partitioning criterion* to define the optimal function choosing the best partition (or split) of the objects into homogeneous subgroups;

- *the stopping rule* to arrest the growing procedure to build up the tree;

- *the assignment rule* to identify either a class or a value as label of each terminal node.

1.2.2 The standard procedure

Let $(Y, \mathbf{X})$ be a multivariate random variable where $\mathbf{X}$ is the vector of M predictors $(X_1, \ldots, X_m, \ldots, X_M)$ (nominal, ordinal or numerical) and Y is the criterion variable taking values either in the set of prior classes $C = 1, \ldots, j, \ldots, J$ (if categorical) or the real space (if numerical). The former case will be referred to classification trees and the latter to regression trees.

On the basis of a sample of N observations $C = \{(y_n, \mathbf{x}_n); n = 1, \ldots, N\}$ taken from the distribution of $(Y, \mathbf{X})$ a simple goal of exploratory trees is to uncover the predictive structure of the problem, understanding which predictors and which interactions of predictors are the most significant to explain the response variable.

Tree methods consist of a recursive partitioning procedure of observations into K disjoint classes such that observations are internally homogeneous within the classes and externally heterogeneous among the classes with respect to the response variable Y. The heterogeneity at any node t is evaluated in terms of an *impurity measure* $i_Y(t)$. In classification trees the impurity can be expressed by the following measures:

1. *the misclassification error*

$$i_Y(t) = 1 - max_j p(j|t) \tag{1.1}$$

2. *the Gini index of heterogeneity*

$$i_Y(t) = 1 - \sum_j p(j|t)^2 \tag{1.2}$$

3. *the entropy measure*

$$i_Y(t) = -\sum_j p(j|t) logp(j|t) \tag{1.3}$$

where $p(j|t)$ is the number of observations in node t that belongs to the class j. In regression trees the impurity is expressed in terms of variation or deviation of the response variable for the observations falling into the node t

$$i_Y(t) = \sum_{\mathbf{x}_n \in t} (y_n - \bar{y}(t))^2 \tag{1.4}$$

where $\bar{y}(t)$ is the mean response on the basis of the observations falling in node t, i.e., $\mathbf{x}_n \in t$. The *total impurity of any tree* T is defined as follows

$$I_Y(T) = \sum_{t \in \tilde{T}} I_Y(t) = \sum_{t \in \tilde{T}} i_Y(t)p(t) \tag{1.5}$$

where $I_Y(t)$ is the weighted impurity of node t being $p(t) = N(t)/N$ the weight of the node t for $N(t)$ the number of observations falling in node t, and $\tilde{T}$ is the set of terminal nodes of the tree T.

The total impurity of any tree is reduced by finding at each node of the tree the best partition p^* of the observations into K disjoint classes such that it induces the highest decrease in the impurity of the response variable Y when passing from the node t to the K descendant nodes t_k:

$$max_{p \in P} \Delta i_Y(t,p) = max_p \{ i_Y(t) - \sum_k i_Y(t_k) p(t_k|t) \} \tag{1.6}$$

where the $p(t_k|t)$ is the proportion of observations in node t falling into the k-th its descendant.

It is possible to show that the following relation holds:

$$I_Y(T) = \sum_{h \in H} \Delta I_Y(t,p) = \sum_{h \in \tilde{H}} \Delta i_Y(h,p)p(t) \tag{1.7}$$

where H is the set of nonterminal nodes of the tree.
From the computational point of view, the best partition at each node is found among all candidate partitions that can be generated by the set of predictors. How to determine the set P of candidate partitions? This is defined considering all possible ways to partition into K groups the modalities of each predictor. In most applications, binary trees are grown so that at each node a split into two disjoint classes is determined, i.e., $K = 2$. A numerical or ordinal predictor having G distinct modalities generates $G - 1$ possible splits, whereas a nominal predictor having G categories yields to $2^{G-1} - 1$ splits. For binary trees, the goodness of split can be defined as

$$max_{s \in S} \Delta i_Y(t,s) = max_s\{i_Y(t) - p_l i_Y(t_l) - p_r i_Y(t_r)\} \tag{1.8}$$

where $s \in S$ includes the set of splits generated by all predictors. The criterion (1.8) is substantially present in most of the tree-growing procedures implemented in the specialized software; for instance, CART [15], ID3 and CN4.5 [86].

1.2.3 Stopping rules

Tree growing can be arrested considering a suitable combination of the following conditions:

a) *The decrease of impurity.* A node can be declared to be "terminal" if the impurity reduction due to the further partition of the node is lower than a fixed threshold; indeed, it is not recommended to grow further branches which do not contribute significatively to the total impurity reduction of the tree;

b) *The number of observations.* It could be fixed the maximum percentage of the sample of objects which can fall into a node to declare it as a "terminal" one; it is in fact useless to keep growing nodes with a small sample size;

c) *The size of the tree.* A further condition could be based either on the total number of terminal nodes or on the number of levels of the tree to limit its expansion.

The above mentioned stopping rules represent an empirical system to define a tree structure which can be used for exploratory purposes. On the contrary, the choice of the most honest size tree for decision-making on new observations for which the response class/value is not known requires a suitable induction procedure. In this case, the aim is to identify a parsimonious and accurate decision tree[1].

1.3 TWO-STAGE partitioning criteria

1.3.1 The basics

Two-stage segmentation of Mola and Siciliano ([79], [78], [76], [98]) is funded on the concept that any predictor X is not merely used as generator of partitions but it plays a global role in the analysis. Main issue is to evaluate the *global effect* of X on the response variable Y as well as the *local effect* of any partition p generated by X. Such effect can be understood in terms of prediction or predictability power of the predictor X as well as of the partition p (or the split s) on the response Y. At any node t the two stages can be defined as:

[1] The next chapter will also treat of *pruning* methods and re-sampling procedures ([14], [36]) as well as statistical testing procedures ([21],[19])

- *stage I: global selection*; one or more predictors are chosen as the most predictive for the response variable according to a given criterion; the selected predictors are used to generate the set of partitions or splits;
- *stage II: local selection*; the best partition is selected as the most predictive and discriminant for the subgroups according to a given rule.

The choice of the criteria used in the two stages depends on the nature of the variables, the tool of interpretation and the desired description in the final output.

1.3.2 Global Impurity Proportional Reduction

Let define the index $\gamma_{Y|X}(t)$ to evaluate the *Global Impurity Proportional Reduction*) (Global IPR) of the response variable Y at node t, due to the predictor X:

$$\gamma_{Y|X}(t) = \frac{i_Y(t) - \sum_i p(g|t) i_Y(g|t)}{i_Y(t)} \tag{1.9}$$

where the g denotes the modalities of the predictor X for $(g = 1, ..., G)^2$, the $p(j|g,t)$ is the conditional proportion of objects falling in class j of Y given that they belong to the modality g of X. The index $\gamma_{Y|X}(t)$ measures the degree of dependency of Y on the predictor X when globally considered. It is a normalized measure taking values in $[0, 1]$. Depending on the choice of the impurity measure it is possible to derive well-known measures of predictability as special cases of (1.9). For classification trees, using the Gini index of heterogeneity it yields to the predictability τ index of Goodman and Kruskal:

[2] The modality can be either the category of a qualitative variables or the distinct number of a numerical variable.

$$\tau_{Y|X}(t) = \frac{\sum_g \sum_j p^2(g,j|t)/p(g|t) - \sum_j p(j|t)^2}{1 - \sum_j p(j|t)^2} \tag{1.10}$$

whereas using the entropy measure it gives the Shannon's conditional entropy index

$$H_{Y|X}(t) = -\frac{\sum_g \sum_j p(g,j|t) \log \frac{p(g,j|t)}{p(g|t)p(j|t)}}{\sum_j p(j|t) \log p(j|t)} \tag{1.11}$$

1.3.3 Local Impurity Proportional Reduction

At each node of the recursive partitioning any predictor X generates a set of candidate partitions $p \in P$ of the objects into K disjoint groups. Let define the index $\gamma_{Y|p}$ as the **Local Impurity Proportional Reduction** (Local IPR) of the response Y due to the partition p generated by the predictor X:

$$\gamma_{Y|p}(t) = \frac{i_Y(t) - \sum_k i_Y(t_k) p(t_k|t)}{i_Y(t)} \tag{1.12}$$

where the $i_Y(t_k)$ for $(k = 1, ..., K)$ is the impurity of the kth child node, the $p_(t_k|t)$ is the proportion of objects of the node t falling into the child node t_k on the basis of the partition p. For $K = 2$ the local IPR becomes

$$\gamma_{Y|s}(t) = \frac{i_Y(t) - (p_l i_Y(t_l) + p_r i_Y(t_r))}{i_Y(t)} \tag{1.13}$$

where p_l and p_r are respectively the proportion of cases into the left node t_l and the right node t_r. The numerator of (1.13) is equivalent to the well known decrease in impurity (1.8) defined in CART [15], thus provides a normalized measure which takes values in $[0, 1]$.

It is possible to show that the total impurity (1.5) can be expressed in terms of proportional reduction of impurity:

$$I_Y(T) = \sum_{h \in H} \gamma_Y(h,p) i_Y(h) p(h) \tag{1.14}$$

where H is the set of non-terminal nodes of the tree T. In this expression the total impurity is understood as a combination of impurity proportional reduction at each node weighted by the proportion of objects and by the node impurity.

1.3.4 Two-stage algorithms

For binary segmentation, two-stage partitioning considers the global effect of the predictor by using the values of (1.9) for each predictor. The *standard two-stage splitting criterion* consists of the following steps [79][3]:
stage I: select the best predictor $X^*(t)$ at node t by maximizing the (1.9) for any predictor in the set $m \in M$;
stage II: select the best split $s^*(t)$ at node t by maximizing the (1.13) for all splits of $X^*(t)$, i.e., $s \in S$.

Alternatively, we can apply a *modified two-stage splitting criterion* in order to consider that more than one predictor might have high and very similar value of (1.9) [78]. At any node t we order the predictors with respect to the values of (1.9), i.e., $(X_{(1)}, \dots, X_{(m)}, \dots, X_{(M)})$ where $X_{(m)}$ is the predictor with the m-th highest value of (1.9), and we select the first $K < M$ predictors to generate the set S of splits. In this way, we ensure that a sample fluctuation does not influence the choice of the best split. Using simulation studies we have verified that for $J = 2$ the best split according to CART splitting criterion is

[3] It is worth noting that not all the M predictors are present in the data matrix at node t.

generated by one of the first three ordered predictors with probability near to 1, and by one of the first two ordered predictors with probability near to 0.95 [76]. In general, a predictor with high predictability power on the response variable has high probability to generate the best split in CART.

Similar splitting criteria can be applied in regression trees by considering the Pearson's square correlation ratio η^2 [97].

Let $TSS_Y(t)$ be the total sum of squares of the numerical response variable Y and let $BSS_{Y|X}(t)$ be the between group sum of squares due to the predictor X. The η^2 of Y due to the predictor X is given by:

$$\eta^2_{Y|X}(t) = \frac{BSS_{Y|X}(t)}{TSS_Y(t)} \tag{1.15}$$

which has values in $[0,1]$ and gives the proportion of the variation of the response variable Y due to the predictability power of X globally considered. Similarly, we can consider the η^2 of the response variable Y given a split s:

$$\eta^2_{Y|s}(t) = \frac{BSS_{Y|s}(t)}{TSS_Y(t)} \tag{1.16}$$

where $BSS_{Y|s}(t)$ is the between group sum of squares of Y due to the split s.

1.4 Fast Algorithm for Segmentation of Trees

The popularity of segmentation procedures increased together with the improvement of computational capability. CART methodology can be fruitfully applied on large data sets providing very interesting results in the form of a tree structure. Nevertheless, the performance of recursive partitioning algorithms needs to be further investigated, especially when dealing with huge data sets and with decision tree selection. As it will be shown in the next chapter, for the choice of the final tree it is necessary to derive a set of tree structures repeating in this way the tree growing procedure a certain number of times.

From the computational point of view, the best partition could be found by minimizing the second term of (1.6), i.e., the *local impurity reduction factor* at node t:

$$\omega_{Y|p}(t) = \sum_{k} i_Y(t_k)p(t_k|t) \tag{1.17}$$

for $p \in P$, thus minimizing the total impurity tree. When applying the two-stage criterion, the best predictor could be found minimizing the *global impurity reduction factor* due to any predictor X:

$$\omega_{Y|X}(t) = \sum_{g \in G} i_{Y|g}(t)p(g|t) \tag{1.18}$$

where $i_{Y|g}(t)$ is the impurity of the conditional distribution of Y given the g-th modality of any predictor X for G the number of modalities of X.

The two-stage splitting criterion *sic et simpliciter* minimizes, in the first stage, the global impurity reduction factor (1.18) with respect to all predictors, that is

$$min_{m \in M} \omega_{Y|X_m}(t) \tag{1.19}$$

and, in the second stage, the local impurity reduction factor (1.17) with respect to all partitions derived from the best predictor, that is

$$min_{p \in P} \omega_{Y|p}(t) \tag{1.20}$$

The modified two-stage criterion selects a set of best predictors in the first stage increasing the confidence to get the best partition among the set of partitions generated by the selected predictors. Various approaches can be considered to select the predictors such as also statistical modeling.

The computational time consuming spent by a segmentation procedure is crucial so that a fast algorithm is required. The idea is to accelerate the recursive partitioning by improving the selection of the best partition or split. Siciliano and Mola [77] introduced a *Fast Algorithm for Splitting Trees* (FAST) in order to get the same solution of CART methodology without trying out necessarily all candidate partitions. In the following, the approach is presented in general, not only for binary trees, such that the algorithm can be proposed for segmentation of trees.

Main issue of FAST is that the γ measure of impurity proportional reduction satisfies the following property:

$$\gamma_{Y|X}(t) \geq \gamma_{Y|p}(t) \tag{1.21}$$

for any partition $p \in P$ generated by the predictor X. This means that the Global IPR measure of a predictor X needs to be not lower than the Local IPR measure of any partition derived from the same predictor. The property is equivalent to the following one:

$$\omega_{Y|X}(t) \leq \omega_{Y|p}(t) \tag{1.22}$$

for any $p \in P$ of X, namely the global impurity reduction factor due to the predictor X can be shown to be not larger than the local impurity reduction factor due to any partition derived from the same predictor.

FAST consists of two basic rules:

- iterate the two-stage partitioning criterion using (1.18) and (1.17) selecting one predictor at a time and each time considering the predictors that have not been selected previously;

- stop the iterations when the current best predictor in the order $X_{(v)}$ at iteration v does not satisfy the condition $\omega_{Y|X_{(v)}}(t) \leq \omega_{Y|p^*_{(v-1)}}$ where $p^*_{(v-1)}$ is the best partition at iteration $(v-1)$.

The fast partitioning algorithm finds the optimal solution but with substantial time savings in terms of the reduced number of partitions or splits to be tried out at each node of the tree. Simulation studies show that in binary trees the relative reduction in the average number of splits analyzed by the fast algorithm with respect to the standard approach increases as the number of predictor modalities and the number of observations at a given node increase. Further theoretical results about the computational efficiency of the fast-like algorithms can be found in [61].

1.5 Two-Stage segmentation via DIScriminant analysis

A typical data mining problem is to deal with large sets of within-groups correlated inputs compared to the number of observed objects. Main task is to define few typological predictors. As an example in marketing, questionnaires in survey analysis are often structured into distinct parts, each one dedicated to a particular subject of interest. As a result, the number of inputs can be very large with respect to the number of interviews so that any standard procedure might yield to spurious interactions among different types of inputs, describing relations among predictors which might be not logically related so that the final interpretation becomes a hard job. A variable reduction criterion needs to be applied in the pre-processing: inputs of a given subject can be combined into one typological predictor describing a given part of the questionnaire. Other examples might concern medical data sets (i.e., the gene expression data) where inputs are partitioned into distinct groups on the basis of its own characteristics. Inputs are correlated within each group and not necessarily correlated among the groups. Furthermore, it might be interesting to analyze how this partition or stratification influences the final outcome. Finally, another typical data mining problem is to deal with more data marts within a data warehouse. Each data mart includes several within-group correlated inputs which are internally logically related and together externally related to inputs of other data marts. Any statistical analysis might relate typological predictors belonging to different data marts in order to apply statistical tools for the data warehousing. More generally, in classification problems, every time we analyze a complex and large data set, the objective is not only to classify but to interpret, too. Standard tree-based procedures in this type of data sets do not work fine for two main reasons. Firstly, the interpretation of the final decision tree can be very poor: a small sample size implies a very short tree with a very few splits deduced from predictors belonging to completely different subjects of interest, limiting thereby the interpretation of the variable interactions in the tree. Secondly, standard tree-based procedures offer unstable solutions especially in case of complex relationships. Indeed, small samples requires cross-validation estimates of the prediction errors. This approach, in presence of too many inputs compared to the sample size, yields to two drawbacks: an unstable selection of the splitting inputs and a computationally expensive greedy searching procedure for the best split to be repeated many times. A possible way to overcome the first of the two drawbacks is bagging estimation procedure, namely an averaging of unstable solutions provides an even better final estimate of the prediction error.

1.5.1 The Key Idea

A standard binary segmentation procedure aims to find at each node the best split of objects into two sub-groups which are internally the most homogeneous and externally the most heterogeneous with respect to the given output. The best split is found among all (or a sufficient set) possible splits that can be derived from the given inputs, namely partitioning the modalities of the input into two sub-groups in order to provide the corresponding binary split of the objects. A similar approach is considered in r-way partitioning procedures (also knows as multiway splits) where the internal homogeneity within the r sub-groups is maximized. Any recursive partitioning procedure is able to deal with large datasets and is particularly suitable for data mining tasks. Some alternatives strategies should be considered in order to deal with large sets of within-groups correlated inputs compared to the sample size.

The key idea is to approach this problem using an inductive method. Without loss of generality, we consider the case of binary splits although generalizations of the proposed approach can be derived for r-way or multiway splits. Firstly, we define the optimal partition of the objects into two subgroups which are the most internally homogeneous with respect to the given output without considering the inputs. Then, we look at the observed candidate partitions of the input features (and their combinations) which provide some alternative solutions that best approximate the optimal one. In other words, known the optimal solution we look for the most suitable combination of inputs which has the highest chance to provide nearly the best partition of the objects. Despite the optimal partition is found in spite of the inputs and it can be just theoretical (in the sense that there might be no input which ensures that partition of the objects into two subgroups), the observed one is found considering the candidate inputs (or a combination of them) aiming to approximate the optimal solution.

1.5.2 Notation and Definitions

Let Y be the output and let $\mathbf{X}_g = (X_{1g}, ..., X_{d_g g})$ denote the g-th set of inputs, for $g = 1, \ldots, G$ groups and $D = \sum_g d_g$ total inputs. Denote by $\mathcal{L} = \{y_n, \mathbf{x}_n; n = 1, \ldots, N\}$ the training sample of objects in which are measured the output and the G sets of inputs, being the row-vector $\mathbf{x}_n = (\mathbf{x}_{1n}, \ldots, \mathbf{x}_{Gn})$ formed by juxtaposing the G sets of input measurements on the n-the object. Furthermore, assume that within each set, the inputs are strongly correlated. Inputs and outputs are numerically defined in the real space. Any binary segmentation procedure can be defined as a recursive partition of the objects into two subgroups such that at each node the best split of the input features (yielding to the binary partition of the objects) maximizes the between group deviation of the output Y, or minimizes the within groups deviations of the output Y in the two subgroups. A greedy searching procedure is applied to look for the best split among all possible (or a suitable subset) splits that can be deduced from the inputs.

We distinguish between prospective and retrospective splits of the objects at a given node:
Definition 1: Any split s of the objects induced by splitting the input features is named *prospective split*. As an example, an object goes either to the left sub-node if $X \leq c$ or to the right sub-node if $X > c$. Standard tree-growing procedure adopts prospective splits. Let S denote the set of prospective splits.
Definition 2: We define a *retrospective split* any split k of the objects induced by splitting the output: being Y numerical any cut point of the real interval in which the Y is defined yields a retrospective split. Note that in this definition the inputs do not play a role. Let K denote the set of retrospective splits.
This terminology is motivated as follows: a prospective split of the objects is deduced by *looking forward* to the splitting of the input features, whereas a retrospective split of the objects requires *to look*

backward to which observed split of the input features might induce that partition of the objects, so that an inductive approach must be considered.

Property 1: It is worth noting that the set S of prospective splits do not necessarily coincide with the set K of retrospective splits. It can be shown that $S \subseteq K$.

Property 2: There can be retrospective splits which are not admissible, in the sense that for a split of the objects based only on the Y there cannot be found any split of the input features yielding to the same partition of the objects. This distinction is important in the proposed methodology because it allows to define upper bounds for the optimality criteria that will be considered. Let denote $L \equiv Left$ and $R \equiv Right$ the sub-groups of any split k or of any split s. Given a retrospective split k we can calculate within each sub-group the sample mean of Y, denoted by $\bar{y}_k(L)$ and $\bar{y}_k(R)$, the within-group deviations, denoted by $Dev_k(W|L)$ and $Dev_k(W|R)$, or the between class deviation, denoted by $Dev_k(B)$. Similar notation is used for prospective splits.

Definition 3: We define the *optimal theoretical split* of the objects into two sub-groups the split that maximizes the between class deviation of Y over all possible retrospective splits in the set K:

$$k^* \equiv argmax_k\{Dev_k(B)\} \tag{1.23}$$

which yields the best discrimination of the objects belonging to the left sub- group of k^* (having an average $\bar{y}_{k^*}(L)$) from the objects belonging to the right sub-group of k^* (having an average $\bar{y}_{k^*}(R)$). This is a theoretical partition since it can be not necessarily produced by any observed split of the input features. Let $\tilde{Y}$ denote a dummy output which discriminates the two sub-groups of objects according to the optimal solution provided by the retrospective split (1.23).

Definition 4: We define the *best observed split* of the objects into two sub-groups the split s that maximizes the between class deviation of Y over all possible prospective splits in the set S, namely $s^* \equiv argmax_s\{Dev_s(B)\}$.

The set K of all possible retrospective splits can be reduced using the property related to the use of a numerical variable Y. Formally, if Y has N distinct ordered values then there are $N-1$ suitable cut points to divide the values (and thus the objects) into two sub-groups. Although the number of possible splits is in principle $2^{N-l}-1$, the number of suitable splits reduces to $N-1$ if we consider the ordinal scale of Y and the statistical properties of mean and deviation which is based on the optimality criterion. In other words, the best discrimination of the Y values must simply satisfy the ordering of the Y values by definition. For a node size constraint of say m objects, the cardinality of the set of candidate splits to find the optimal one reduces to $N-2(m-1)-1$.

Definition 5: The quantity $Dev_{k^*}(B)$ derived from (1.23) is the upper limit of the between-group deviation that can be found by any prospective split of the input features, i.e., $Dev_{k^*}(B) \leq Dev_s(B)$.

Definition 6: The ratio $Dev_{s^*}(B)/Dev_{k^*}(B)$, i.e., the between class deviation due to the best observed split over the between class deviation due to the optimal theoretical split, is an efficiency measure of the partition of the objects that is found at a given node. It says how good is the discrimination between the two sub-groups with respect to the given output ranging from zero and one by definition.

1.5.3 Two-Stage Segmentation

This methodology is inspired by two-stage segmentation and fast splitting algorithm. The general idea was to emphasize the role of the inputs to be globally considered before selecting the best split. According to the two-stage splitting criterion, firstly we find the best input that provides a good prediction of the given output in order to generate the set of candidate splits, then we find the best split that provides the best partition into two sub-groups. Several two-stage criteria have been proposed considering statistical modeling as also factorial methods for univariate and multivariate output. In the following, we provide an alternative two-stage splitting criterion to overcome the limits of standard procedures.

Our methodology can be viewed as a recursive partitioning which at each node applies the following two stages:

I. Factorial analysis: For each group of within group correlated inputs we find a factorial linear combination of inputs such to maximize the predictability power to get the optimal split of the objects;

II. Multiple splitting criterion: Among the prospective splits that can be deduced from the linear combinations determined in stage one we find the best factorial (multiple) split of the objects.

Stage I allows to reduce the dimensionality of the problem passing from $D = \sum_g d_g$ inputs to G linear combinations of inputs. Stage II provides to define automatically the factorial multiple split of the objects into two sub-groups. This proposed procedure is named TS-DIS (*Two-Stage segmentation via DIScriminant analysis* [99]). In general, main advantage of tree-based models with splits based on factorial linear combinations is to better provide prediction accuracy and shorter trees. With respect to the CART use of discriminant analysis, the approach deals with numerical rather than a dummy output.

1.5.4 Linear Discriminant Functions of Within-Groups Correlated Inputs

Factorial discriminant analysis is applied in stage I. We consider as output the dummy variable Y which summarizes the optimal split of the objects. For each set of inputs, i.e., $\mathbf{X}_g = (X_{1g}, \ldots, X_{d_g g})$ with $g = 1, \ldots, G$, we calculate the within group deviation $\mathbf{W}_g$ and the between class deviation $\mathbf{B}_g$ of the inputs in the g-th group. For each group, we find the linear discriminant variable such that the between class deviation is maximized with reference to the within group deviation:

$$Z_g = \sum_{j}^{d_g} \alpha_j X_{jg} \tag{1.24}$$

where α_j are the values of the eigenvector associated to the largest eigenvalue of the matrix ${\mathbf{W}_g}^{-1}\mathbf{B}_g$. The (5.1) is the g-th linear combination of the inputs belonging to the g-th group with weights given by the first eigenvector values. It is obtained maximizing the predictability power of the d_g inputs to explain the optimal split as summarized by the output $\tilde{Y}$. Moreover, the Z_g variables are all normalized such to have mean equal to zero and variance equal to one. In this respect, they will play the same role in the greedy selection of the best split in stage II, thus producing unbiased splits in the procedure.

It is worth noting that the discriminant analysis is applied considering the dummy output $\tilde{Y}$ and each set of inputs separately. In this way, we find the best linear combination within each group of internally correlated inputs.

1.5.5 Multiple Split Selection

The selection of the best split of the objects into two sub-groups is done in stage II. The linear combinations $Z_l, \ldots, Z_G$ are the candidate splitting variables which generate the set S of prospective splits. These can be interpreted as multiple splits being defined on the basis of a combination of inputs. The best multiple split is found maximizing the between class deviation of the output:

$$s^* \equiv argmax_s\{Dev_s(B)\} \tag{1.25}$$

for any split s in the set S.

We can also calculate the efficiency measure based on the ratio between the between class deviation due to the best observed split, i.e., $Dev_{s^*}(B)$, and the between class deviation due to the optimal split, i.e., $Dev_{k^*}(B)$. This measure could be also used as a stopping rule for the tree-growing recursive procedure.

1.5.6 The Recursive Algorithm

In order to summarize the proposed procedure we outline the main steps of the recursive splitting algorithm at any node of the tree:

1. Find the optimal retrospective split of the objects maximizing the between class deviation of the Y and define the dummy output $\tilde{Y}$;

2. Find the discriminant variables $(Z_l, \ldots, Z_G)$ for the G groups according to (1.23) maximizing the between class deviation of the inputs with respect to the optimal split summarized by $\tilde{Y}$;

3. Using the set of prospective splits generated by the discriminant variables find the best observed split s^* maximizing the between class deviation of the output Y;

4. Calculate statistical measures within each sub-node, providing interpretation aids and visualization of the splitting process through a factorial axis description.

1.5.7 The Empirical Evidence

1.5.8 A Simulation Study

Aim of our simulation study was to analyze the performance of the proposed procedure TS-DIS compared to the standard CART procedure.

The planning of our simulation study was the following. We fixed $G = 10$ the number of groups. The experimental design was based on the following parameters: the *sample size*, with levels 100, 500, 1000, 10000, the *number of inputs*, given by 20, 50, 100 partitioned into $G = 10$ groups, the *variance of the inputs and of the output* considering two cases. In the first, the variables were generated from normal distributed functions having mean equal to zero and variance respectively equal to 1, 10, 100, whereas in the second case, the variables were generated from uniform distribution ranging from zero and 10, 100, 1000 respectively. In total, we have generated 36 data sets for normal distributed inputs and 36 for uniform distributed inputs. In order to stress our procedure we have considered the worst conditions assuming within group uncorrelated inputs and checking in particular the performance in the root node of the tree.

We present the results of our simulation study in table 1 and in table 2. For sake of brevity, we have omitted to report the results concerning the last level of the sample size and the first level of the variance. In the first three columns we indicate the parameters of the experimental design. In the subsequent columns, we report blocks of results concerning respectively the optimal split solution (the optimal retrospective split), the TS-DIS best (observed) solution, the CART best (observed) solution. For each split, we give the average and the standard deviation of the output Y in the left and right sub-nodes, and for the observed splits, in addition, we give the percentage of errors. The latter is calculated considering the cross-classification of the dummy variable $\tilde{Y}$ with the best observed split of the objects: in this way, we can calculate how many objects were misclassified by the best observed split with respect to the optimal one.

Although we have randomly generated the variables without assuming a dependency data structure the proposed procedure offers better solutions in terms of both misclassification error and within class homogeneity.

1.5.9 The Real World Applications

Our methodology has been experienced in some applications for market basket analysis aiming to identify associations between a large number of products bought by different consumers in a specific location, such as a hypermarket ([4][10]). TS-DIS has allowed to define hierarchies of the most typological baskets of products which determine high monetary values spent on specific target products. This becomes particularly useful, from a promotional viewpoint: if two products resulted sequentially associated in the final tree, it is sufficient to promote only one to increase sales of both. At the same time, from a merchandising viewpoint, the products type should be allocated on the same shelf in the layout of a supermarket.

1.5.10 Concluding Remarks

This part of the thesis has provided a recursive partitioning procedure for a particular data mining problem, that is to find a tree-based model for a numerical output explained by large set of within group correlated inputs using a small training sample compared to the number of inputs. The procedure is based on two-stage splitting criterion employing linear discriminant analysis and defining factorial multiple splits. New concepts of retrospective and prospective splits were defined and an upper bound of the optimality splitting criterion was in this way defined. The results of our simulation study as well as of real world applications have been very promising, showing that our methodology works much better than CART standard procedure under the above conditions.

1.6 Partitioning criterion: Multi-Class Budget

The *Multi-Class Budget* is a two stage partitioning criterion that uses conditioned latent balance models to determine the best split in classification problems where the response variable is multi-class [9].

The idea is to select, at every node of the tree, more predictive predictor or subgroup of predictors respect to Y variable and to use conditioned the latent balance model to find the best partition of units in K groups where with K the number of considered latent balances is indicated. The choice of the K depends by different strategies of analysis. The first strategy is to choose a priori the number of balances equal to $K = 2$ or $K = 3$ to obtain a binary or ternary tree. A second way to proceed could be to not fix K a priori, but to choose step by step in every node the most parsimonious model from the considered data. In both situations the procedure comes the construction of a multi-class balances classification tree characterized by a sequence of latent balances models assigned recursively to internal nodes of the tree

The MCB criterion follows a two stages idea coming to the selection of predictor (or predictors) and successively to the generation of the split of units. This is a synthetic schema of the recursive algorithm:

Step 1. **Selection of the predictor**
From set X the best predictor, or a subgroup of the bests, is

selected by relative impurity global index calculated for each of them.

Step 2. **Definition of the split**
A latent balances model conditioned on selected predictors is applied and the best is select by a measure of goodness of fit. The parameters of the selected model are used to define the split and the response class of the child-nodes (from the latent components).

When in the first step of the procedure in the explication of the response variable not a single predictor is selected, but a group of significant predictors, then arises the problem to choose which of them to consider in definition of the split. This happens computing a number of latent budget models equal to selected predictors to choose the best by goodness of fit of the same model.

The used measure is the *dissimilarity index* by Clogg and Shihadeh [27]

$$D = \sum_j \sum_i \frac{p_{i.}}{2} |\pi_{j|i} - p_{j|i}| \tag{1.26}$$

that measures how the model is distant from the observed balances. The advantage of this index is that no distributional hypothesis is necessary and the index is normalized: for this reason it is possible to compare different models.

The obtained estimates from LBM model can be explained as conditioned probabilities allowing to assign a fundamental rule to the parameters on the definition and interpretation of the partition. Regarding binary tree, the N_t units belonging to t node can be partitioned in K subgroups with $K = 2$ with the estimates of the parameters of mix of the selected model. For A matrix of $(I \times 2)$ dimensions with $\sum_{k=1}^{2} \pi_{i|k} = 1$, I categories of X^* predictor are synthesized in 2 latent balances (the child-nodes). So the i-th category is assigned to k-th latent balance that presents the higher parameter of mix, the most high conditioned probability. This means that the units fall in left node when they have categories of the predictor which $\pi_{i|k=1} \geq 0.5$ parameter is associated whereas the others will fall in right-node:

$$\text{Split}\begin{cases} \pi_{i|k=1} \geq 0.5 \rightarrow t_{left} \\ \pi_{i|k=2} > 0.5 \rightarrow t_{right} \end{cases} \tag{1.27}$$

per $(i = 1, ..., I)$.

Chapter 2

Tree Induction

2.1 Decision trees

Exploratory trees can be used to investigate the structure of data whereas for induction purposes they cannot be used in a direct way. The main reason is that exploratory trees are accurate and effective with respect to the *training data set* used for growing the tree but they might perform poorly when applied for classifying/predicting fresh observations which have not been used in the growing phase.

A step further is required by considering the *tree induction*, whose aim is to define the structural part of the tree model reducing the size of the exploratory tree while retaining its accuracy. Tree induction relies on the hypothesis of *uncertainty* in the data due to *noise* and *residual variation* [74]. Simplifying trees is necessary for two main purposes: *understandability* - the tree structure for induction needs to be simple and not so large[1] -, and *identifiability* - on one hand, terminal branches of the expanded tree reflect particular features of

[1] This is a difficult task especially for binary trees since a predictor may reappear (eventhough in a restricted form) many times down a branch

the training set causing *overfitting*, on the other hand, overpruned trees do not necessarily allow to identify all the response classes/values (*underfitting*).

The goal of simplification for decision trees is thus inferential, i.e., to define the structural part of the tree model, reducing the size of the tree while retaining its accuracy. Basically, the idea of Mingers [74] that the simplification method performance in terms of accuracy is independent from the partitioning criterion used in the tree growing procedure has been confuted by Buntine and Niblett [18]. The choice of the most suitable method for simplifying trees depends not only on the partitioning criterion, and thus on the expanded tree from which to start simplifying, but also on the objective and the kind of data sets. Therefore, exploratory trees becomes an important preliminary step for decision trees induction. In simplification procedures it is worthwhile to distinguish between *optimality criteria for pruning* the tree and *criteria for selecting* the best decision tree. Such two moments do not necessarily coincide and often require independent data sets (*training set* and *test set*). In addition, a *validation data set* can be required to assess the quality of the final decision rule ([51]). In this respect, segmentation with pruning and assessment can be viewed as stages of any computational model building process based on a supervised learning algorithm like expert systems and neural networks.

2.2 Pruning and selection

Pruning trees is necessary to remove the most unreliable branches and improve understandability. Several strategies can be considered upon definition of the pruning criterion, the type of algorithm and the sample to be used. The result is either *a set of optimally pruned trees* (on which base the most honest tree is found) or just *one best pruned tree* (which represents in this way the final rule).

The pioneer approach to simplification was based on arresting the recursive partitioning procedure according to some stopping rule (prepruning). This is the case of the **critical-value pruning** of Mingers [74]: on the basis of an independent set this method specifies a critical value for the measure used in the partitioning criterion and prunes those nodes which do not reach the critical value for any node within their branch. The larger the critical value selected, the greater the degree of pruning and the smaller the resulting pruned tree; a set of optimally pruned trees can be generated by increasing critical values.

A more recent approach consists in growing the totally expanded tree and removing retrospectively some of the branches (**postpruning**). This can be done working from the bottom of the tree to the top (*down-top algorithm*) or viceversa (*top-down algorithm*). The training set is often used for pruning, whereas the test set is used for selection of the final decision rule; this is the case of the error-complexity pruning of CART. Nevertheless, some methods require only the training set such as the pessimistic error pruning and the error based pruning ([86],[85]) as well as the minimum error pruning [23] and the cross-validation method of CART, and other methods only the test set such as the reduced error pruning [85].

For the definition of the pruning criterion it is necessary to introduce a measure $R^*(.)$ that depends on the size (number of terminal nodes) and the accuracy (error rate, mean square error, etc.) both. In particular, let T_t be the branch departing from the node t having $|\bar{T}_t|$ terminal nodes. The criterion is such that prune node t if

$$R^*(t) \leq R^*(T_t) \tag{2.1}$$

For sake of brevity, the attention is hereby restricted to the classification problem. In CART, the following **error-complexity measure** is respectively defined for the node t and for the branch T_t as

$$R_\alpha(t) = r(t)p(t) + \alpha, \tag{2.2}$$

$$R_\alpha(T_t) = \sum_{h \in |\bar{T}_t|} r(h)p(h) + \alpha|\bar{T}_t|, \tag{2.3}$$

where α is the *penalty for complexity* due to one extra terminal node in the tree, $r(t)$ is the error rate (the proportion of cases in node t which are misclassified), $p(t)$ is the proportion of cases in node t and $|\bar{T}_t|$ is the number of terminal nodes of T_t. Basically, the branch T_t should be pruned if

$$R_\alpha(t) \leq R_\alpha(T_t) \tag{2.4}$$

Thus, using a down-top algorithm and a *training set* the criterion is to prune each time the branch T_t that provides the lowest reduction in error per terminal node (i.e., the *weakest link*) as measured by

$$\alpha_t = \frac{R(t) - R(T_t)}{|\bar{T}_t| - 1} \tag{2.5}$$

On the basis of the error-complexity measure $R_\alpha(.)$ a sequence of nested optimally pruned trees is generated pruning at each step the subtree with the minimum value of α_t. In the same framework of CART, Gelfand *et al.* [43] provides an alternative procedure which optimizes iteratively the tree-growing and the pruning for classification trees. Also related to CART, Cappelli, Mola and Siciliano [22] provides a pruning algorithm based on the **impurity-complexity measure** as an alternative to the error-complexity measure of CART. In particular, the error rate can be replaced by any impurity measure which takes account of the number of classes and the distribution of the cases over the classes. This approach might be viewed as a sort of critical-value pruning based on a more general *accuracy-complexity measure.*

Variants to the CART pruning have been proposed in different contexts such as expert systems and artificial intelligence. In particular, Quinlan ([85], [86]) has developed some pruning methods for classification trees. The **reduced error pruning** employes directly and exclusively the test set to produce a sequence of pruned trees. The criterion is always to prune the node t if

$$R^{ts}(t) \leq R^{ts}(T_t) \tag{2.6}$$

where the subscribe ts refers to the test set, choosing at each step as branch to prune the one with the largest difference. The down-top algorithm continues until no further pruning is possible as it increases the error rate, and it ends with the smallest subtree with the minimum error rate with respect to the test set.

Instead, the **pessimistic error pruning** uses a top-down pruning algorithm and produces only *one pruned tree* on the basis of the training set. The idea is to worsen the estimate of the error rate on the training set by applying the continuity correction for the Binomial distribution given by 0.5. This results in the *corrected* error rate

$$R^{*}(t) = r(t)p(t) + \frac{0.5}{n(t)} \tag{2.7}$$

for the node t and similarly defined $R^{*}(T_t)$ for the branch T_t using CART-like notation. Again, the branch T_t should be pruned if $R^{*}(t) \leq R^{*}(T_t)$ or alternatively if it is less than one standard error more than the corrected measure for its branch such as for the **error based pruning** of C4.5 [86]. It is worth noting that the criterion employed in the pessimistic-error pruning can be viewed as a special case of the error-complexity pruning criterion when α is fixed to be equal to $0.5/n(t)$. Because of the top-down type algorithm only one pruned tree is identified whereas using a down-top algorithm with the above criterion a sequence of optimally pruned trees can be instead defined. Anyway, the continuity correction appears to be suitable only for the two class problem; a more general correction which is based on the number J of response classes is given by $(J-1)/J$.

In decision tree induction accuracy refers to the predictive ability of the decision tree to classify/predict an independent set of test data. In the particular case of classification trees, the error rate, as measured by the number of incorrect classifications that a tree makes on the test data, is a crude measure since it does not reflect the accuracy of predictions for different classes within the data. In other words, classes are not equally likely, and those with few cases are usually predicted badly.

2.3 Statistical Testing Pruning

In the following, we demonstrate that, under the condition that the impurity measure used to grow the tree is also used to prune the tree, the complexity parameter can be expressed as follows:

$$\alpha_t = \frac{1}{\left|\tilde{L}_t\right|} \sum_{l \in L_t} \Delta I\left(s^*, l\right), \tag{2.8}$$

i.e., as the average of the reduction in impurity induced by the best split s* of each internal node l of the branch T_t, in the set L_t which cardinality is $\left|\tilde{L}_t\right| = \left|\tilde{T}_t\right| - 1$ (this relation holds for strictly binary trees where the number of internal nodes is equal to the number of leaves minus one). It is worthwhile to notice that in the following we denote the complexity parameter to be indexed by t, i.e., α_t, in order to emphasise that α_t follows a conditional distribution given the observations from the node t. In order to prove the equivalence between the (2.8) and the (2.9) notice that for any node t it holds:

$$I\left(t\right)-I\left(T_t\right)=I\left(t\right)-I\left(T_{2t}\right)-I\left(T_{2t+1}\right) \tag{2.9}$$

where node $2t$ and $2t+1$ denote the left and the right daughter nodes of node t respectively according to a common formalism in binary trees. The equivalence is then verified by definition when $\left|\tilde{T}_t\right|=2$, being the (2.9) equal to $\Delta I\left(s^*,t\right).$ Now, let $\left|\tilde{T}_t\right|=3$ and without loss of generality, $\left|\tilde{T}_{2t}\right|=2$ (i.e., node $2t+1$ is terminal), it can be easily verified that:

$$I\left(T_{2t}\right)=I\left(2t\right)-\Delta I\left(s^*,2t\right) \tag{2.10}$$

replacing the (2.10) in the (2.9) yields:

$$I\left(t\right)-I\left(T_t\right)=\Delta I\left(s^*,t\right)+\Delta I\left(s^*,2t\right)=\sum_{I\in L_t}\Delta I\left(s^*,l\right) \tag{2.11}$$

where, again, L_t includes the internal nodes of the subtree T_t namely t and $2t$. By induction the (2.11) follows for any $\left|\tilde{T}_t\right|$. This result is the starting point of the definition of a statistical testing procedure as the third stage in tree growing approach aimed to validate the pruning process, i.e., to find the reliable honest tree, either in classification or in regression.

2.4 Statistical testing for growing reliable honest trees

2.4.1 The statistical test for classification trees

In classification one possible impurity measure used to grow the tree is the Gini's index of heterogeneity, defined at any node t as: $i\left(t\right) = 1 - \sum_{j=1}^{J} p^2\left(j\left|t\right.\right)$, where $p\left(j\left|t\right.\right)$ is the proportion of observations in node t belonging to class j. On the other hand, the error measure used to prune the tree is the weighted misclassification rate $R\left(t\right) = r\left(t\right)p\left(t\right)$, where $r\left(t\right) = 1 - \max_j p\left(j\left|t\right.\right)$. This is a rough measure that does not take into account the distribution of observations among classes as well as the number of response classes; at an extreme, misclassification rate might either be insensitive to changes of the distribution of observations or face the multi-class problem in the same way of the two class problem. Replacing the misclassification rate $r(t)$ with the Gini index $i(t)$ in the pruning process allows us to relate the complexity parameter α_t as defined in (2.8) to the χ^2 distribution. Indeed, the weighted decrease in the impurity measure in (2.8) is given by the definition of Breiman [15] about the total impurity of a tree T[2], i.e., $\Delta I\left(s^*, t\right) = \Delta I\left(s^*, t\right)p\left(t\right)$. Mola and Siciliano ([79], [78]) have shown that $\Delta I\left(s^*, t\right)$ is equivalent to the numerator of the sample version of the predictability index τ of Goodman and Kruskal [47], denoted by R^2 - which is considered by Light and Margolin [68] for the definition of the CATANOVA statistic - proving that:

$$\frac{\Delta I\left(s^*, t\right)}{i\left(t\right)} = R^2, \tag{2.12}$$

and that, under the usual probability assumptions,

[2] $I\left(T\right) = \sum\limits_{t \in \tilde{T}} I\left(t\right) = \sum\limits_{t \in \tilde{T}} i\left(t\right)p\left(t\right)$,, where $\tilde{T}$ is the set of terminal nodes

$$R^2 \left(N\left(t\right)-1\right)\left(J-1\right) \sim \chi^2_{J-1}. \tag{2.13}$$

Replacing the (2.12) in the (2.13) gives:

$$\frac{\Delta I\left(s^*,t\right)}{i\left(t\right)} = \left(N\left(t\right)-1\right)\left(J-1\right) \sim \chi^2_{J-1}. \tag{2.14}$$

Multiplying both terms by $N\left(t\right)/N\left(t\right)-1$ it results:

$$\Delta I\left(s^*,t\right) N\left(t\right) \sim i\left(t\right) \frac{N\left(t\right)}{N\left(t\right)-1} \frac{\chi^2_{J-1}}{J-1} \tag{2.15}$$

Replacing $i(t)$ with its maximum equal to $(J-1)/J$ (which is the maximum of the Gini's index) and approximating $N\left(t\right)/N\left(t\right)-1$ to 1, results:

$$\Delta I\left(s^*,t\right) N\left(t\right) \sim \frac{1}{J}\chi^2_{J-1}. \tag{2.16}$$

Summing over the set L_t of internal nodes of the branch T_t:

$$\sum_{l \in L_t} \Delta I\left(s^*,l\right) N\left(l\right) \sim \left|\tilde{L}_t\right| \frac{1}{J}\chi^2_{J-1}. \tag{2.17}$$

multiplying both terms by $1\Big/\left|\tilde{L}_t\right| N$ and considering the (2.8) yields:

$$NJ\alpha_t \sim \chi^2_{J-1}. \tag{2.18}$$

As a result, upon the choice of a significance level the (2.18) can be fruitfully used to verify for each complexity parameter in the sequence, whether the corresponding weakest link induces with its branch a significant reduction in impurity or not, i.e. whether the branch should be kept or pruned. In order to satisfy the independence hypothesis, the testing considers an independent test set on which observations the complexity parameters are computed. While on the training set the complexity parameter values increase at each step, on the test set they do not form necessarily an increasing sequence so the result is a single final tree, which might not correspond to any tree in the CART sequence. This tree will be the most pure respect to the impurity measure employed and, at the same time, statistically reliable, retaining, for the testing, only those splits which induces a significant reduction in impurity.

2.4.2 The statistical test for regression trees

The definition of a statistical testing procedure in regression tree pruning is simpler than in classification because the same impurity measure already used to grow the tree is also used to prune it, i.e., $I(t) = R(t)$. This measure is given at any node t by the sum of squares divided by N. Siciliano and Mola [97] have shown that the decrease in impurity induced by splitting node t multiplied by the constant factor N can be viewed as the between groups sum of squares:

$$N\Delta R(s^*,t) = TSS_Y(t) - WSS_{Y|s^*}(t) = BSS_{Y|s^*}(t), \tag{2.19}$$

where $TSS_Y(t)$ represents the total sum of squares at node t, $WSS_{Y||s^*}(t)$ represents the within groups (i.e. nodes) sum of squares and $BSS_{Y||s^*}(t)$ represents the between groups sum of squares (induced by splitting t into its daughter nodes). The (2.8) can be thus written as:

$$\alpha_t = \frac{1}{\left|\tilde{L}_t\right| N} \sum_{l \in L_t} BSS_{Y|s^*}(l) \tag{2.20}$$

so that, for any internal node t the complexity parameter can be viewed as the average of the between groups sums of squares arising from splitting node t (and its non terminal daughter nodes), divided by the constant factor N. Each complexity parameter, i.e., each pruning operation, can be then tested by applying the analysis of variance testing procedure which compares for any internal node t and its branch, the variance between the groups resulting by splitting t (and eventually its non terminal descendants) with the variance within the groups:

$$F_e = \frac{\sum_{l \in L_t} BSS_{Y|s}\left(l\right)}{WSS_{Y|s}\left(t\right)} \times \frac{N\left(t\right) - \left[\left|\tilde{L}_t\right| + 1\right]}{\left|\tilde{L}_t\right|}. \tag{2.21}$$

This statistic, under the usual probability assumptions, has a Snedecor and Fisher distribution with $\left|\tilde{L}_t\right|$ and $N\left(t\right) - \left[\left|\tilde{L}_t\right| + 1\right] = N\left(t\right) - \left|\tilde{T}_t\right|$ degrees of freedom. Concerning the probability assumptions, the data set should be large, so that the underlying hypothesis of multinormality may be supposed satisfied, and to assure the independence, the testing produce is made, as in classification, using the observations of a separate test set. The testing works as follows: for a fixed significance level, it verifies, by means of the F statistic, whether the branch to be pruned induces a significant increase in the variance between groups (it should be retained) or not (it should be cut). Again, the result of the testing procedure will be a single tree that is likely not coincident with any subtree of the sequence.

2.4.3 The Statistical Testing Procedure

Statistical testing for pruning classification trees has been previously proposed by Mingers [74], with the so called critical value pruning. This pruning method does not address the problem of statistical testing exclusively, in fact it simply consists in fixing a critical value for the goodness of split measure and pruning those nodes that do not reach this value. Obviously, if a probability measure has been used in

creating the totally expanded tree the critical value will be a probability measure and pruning will result from fixing a significance level for the measure. Similarly Zhang [112] has proposed to grow a very large tree, to assign a statistic to each internal node from the bottom up and then select a threshold and change an internal node to a terminal node if its statistic is less than the threshold level. Concerning regression, the F statistic has been proposed by Lanubile and Malerba [64] as a top down stopping rule to stop growing one depth branch. As it will be shown, our approach differs from these since the pruning algorithm mimics the CART one in order to identify the weakest links and then on the basis of these weakest links it validates, on a separate test sample, the corresponding complexity parameters which, both for classification and regression have been proved to be related to a statistical distribution.
On the basis of the statistical tests that can be done in classification and regression trees we propose to add a third stage in CART methodology yielding to apply the following three procedures:

1. Splitting procedure to grow the maximal expanded tree using the training set;

2. Pruning procedure to identify a sequence of nested pruned (honest) trees using the training set; it is worth noting that in the classification case the error measure used in the CART pruning algorithm is the Gini index instead of the error rate;

3. Testing procedure to find the reliable honest tree using the test set.

Hereby we describe the main steps of the algorithm for implementing the Statistical Testing Procedure (STP) in CART methodology.

2.4.4 The algorithm for Statistical Testing Procedure (STP)

Consider the sequence of nested pruned trees $T_1 \supset T_2 \supset ... \supset T_q \supset ... \supset \{t_1\}$ obtained by applying the CART pruning procedure to the maximum expanded tree using the training set. For any T_q is associated the qth weakest link, i.e., the node numbered by d_q to which it corresponds the minimum complexity parameter α_{t}. The following STP procedure will recursively define non-nested subtrees $(D_1, D_2...)$ (which do not necessarily correspond to the CART sequence) ending with the final reliable honest tree.

Step 1 *Initialize*

$k \leftarrow 1, D_k \leftarrow T_1$, fix a significance level

Step 2 *Test the weakest link using the test set*

1. $q \leftarrow k$
2. Test α_{q} corresponding to the node d_{q}
3. If it is significant then go to 4.
   Otherwise
   - 3.1 Prune the branch T_{d_q} departing from the node number d_{q}
   - 3.2 $k \leftarrow k+1, D_k \leftarrow D_{k-1} - \left\{T_{d_q}\right\}$
4. If $d_{\mathrm{q}} \equiv \{t_1\}$ then stop being the current D_k the reliable honest tree.
   Otherwise continue.
5. $q \leftarrow q+1$ and go to 2.

It is worth noting that as in any statistical testing procedure the choice of the significance level will affect the final decision, namely, the degree of pruning. This choice is usually made upon the type of domain. Moreover, by varying the significance level it is possible to repeat the STP procedure in order to define a sequence of reliable honest trees, to each one being associated a confidence level.

2.5 Model Tree Selection

Tree growing approach should be based on:

1. any splitting procedure, to grow the exploratory tree;
2. more pruning methods to define alternative decision trees;
3. a further step to choose one decision tree to be used for future predictive purposes.

Concerning the third phase, since the pruning methods yield to different optimal subtrees a selection is required. To this aim, we propose some alternative strategies that can be adopted, which result either in the selection of a method among the others or in a sort of compromise among them.

Selecting a method

A simple selection procedure for choosing the best decision tree is based on the misclassification rate to be minimized. Let $T_1^*, \dots, T_q^*, \dots, T_Q^*$ be the decision trees resulting from Q different pruning methods. The best decision tree is $T_{q^*}^*$ where:

$$R(T_{q^*}^*) = \ min_{q \in Q} \ R(T_q^*) \tag{2.22}$$

A sophisticated procedure consists in evaluating the predictability power of the decision trees produced by each pruning method, using statistical indexes to be applied to the table which cross-classifies the leaves of each pruned tree with the prior response classes. In such a table, each row reports the distribution of cases among classes at the given leaf. We suggest using the corrected Akaike criterion defined for the q-th decision tree T_q^* as:

$$\overline{AIC}(T_q^*) = G^2(T_q^*) - 2(degrees\ of\ freedom) \qquad (2.23)$$

where $G^2(T_q^*)$ is the likelihood ratio statistic for testing the hypothesis of independence calculated on the table which cross-classifies the leaves of the decision tree T_q^* with the prior classes. This index takes into account the number of degrees of freedom and thus it allows comparisons among tables with a different number of rows (the number of columns is constant, since it is equal to the number of response classes). The higher the value of the index the better the predictive power of the partition given by the terminal nodes of the decision tree, therefore the best decision tree is $T_{q^*}^*$ such that

$$\overline{AIC}(T_{q^*}^*) = max_{q \in Q}\ \overline{AIC}(T_q^*) \qquad (2.24)$$

Table 1.4 shows the values of the corrected Akaike criterion and of the error rate on the test set for each data set and each pruning method considered. From the values in table 2.1 the best decision tree according to the different selection procedures can be deduced yielding to a different choice.

Looking for a compromise

An alternative strategy we propose consists in creating a sort of compromise among the various decision trees. When a decision tree is used to classify a case, a path of conditions is followed from the root node to one of its leaves. A condition at node t is due to a splitting variable s_t of a given predictor X, that is a dummy variable with value 0 or 1 associated to a particular question. For example, if X is nominal a splitting variable is a question of the form “Is X in B or in the complement of B?” where B and $\bar{B}$ are two disjoint subgroups of

	Graduates		Banking		Archeology	
	$\overline{AIC}$	$R(T)$	$\overline{AIC}$	$R(T)$	$\overline{AIC}$	$R(T)$
$0-SE$	60.8	33.7	243.3	41.2	47.2	41.9
$1-SE$	23.4	34.7	233.7	47.3	54.5	50.0
$PEP_{1/2}$	55.9	40.0	240.4	44.7	57.9	48.3
$PEP_{(J-1)/J}$	55.9	40.0	230.5	42.5	47.2	27.61
REP	58.7	30.5	237.5	36.5	62.9	41.4

Table 2.1: Corrected AIC and error rate of different decision trees

categories; if X is ordinal a splitting variable is a question of the form "Is X below a cutpoint modality c?". As a result, at node t, cases for that the splitting variable has value 0 go to the left descendant and the others go to the right descendant. A path of such conditions can be reguarded as a *production rule* defined as the *conjunction* of splitting variables of the form

$$if\ \bar{s}_{t_1} \wedge s_{t_2} \wedge \bar{s}_{t_5} \wedge s_{t_10}\ then\ class\ j \tag{2.25}$$

where the generic s_t is the splitting variable at node t. The *disjunction* of the production rules which provide the same class j, for $j = 1, \dots, J$, define the *classification rule* for class j. Therefore, a decision tree can be reguarded as a set of classification rules.

2.6 Ensemble methods

The general idea is to use a combination of multiple models in order to achieve better prediction performance. The setting is quite general, which means that the building blocks can be obtained by modifying any of the three levels of the model building process: model class, variable selection and transformation and model parameters. In the following we will describe some ensemble methods in order to illustrate the various possibilities.

2.6.1 Bagging

Bagging (bootstrap aggregating) was introduced by Breiman [14] to improve the performance of any individual predictor. The idea is to take bootstrap sample S_i of the data in each step i and obtain a model $\hat{f}_i$ fitted to S_i. After N steps we obtain models $\hat{f}_i, ..., \hat{f}_N$. The new predictor $\hat{g}$ is then created by aggregating models as follows: in a regression setting, the predicted value for an observation x is:

$$\hat{g}(x) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}_i(x) \tag{2.26}$$

in the classification setting the predicted class is determined by plurality vote among the classes $c_i \in C$, i.e. the class label most frequently predicted by the models $\hat{f}_i$ is chosen:

$$\hat{g}(x) = \arg\max_{c \in C} \sum_{i=1}^{N} I(\hat{f}_i(x) = c). \tag{2.27}$$

The bagged predictor delivers much improved prediction results if the base predictor, that is the model used to fit the data in each step, is unstable. This means that predictors that are barely affected by changes in the data are not likely to be good candidates for bagging. The trees are in most cases unstable predictors, because small changes in the data can cause changes in top nodes, leading to cascading effects. Therefore trees are often used as base predictors for bagging.
The number of necessary steps N in the bagging process is usually lower fore regression problems than for classification problems. Prediction error rate estimates of the bagged predictor can be used to

check whether additional steps are necessary.
One potential drawback of the bagged classifier is the fact that in general it cannot be expressed in terms of a single model of the base class. This prevents any interpretation or plausibility check of the model. However due to the fact that the bagged predictor is a combination of individual models which are directly based on the data, it is possible to gather and analyze information about the data from the intermediate models.

2.6.2 Boosting

Bagging achieves by exploiting the instability of the predictor and consulting all predictors on the way. Boosting, however, uses weights to emphasize unusual observations, that is incorrectly classified cases, in order to improve the model by forcing it to correct itself. Boosting tries to steer the model building in an intelligent way, based on the data. The original *AdaBoost* algorithm for classification tasks was developed by Freund and Schapire [40] and can be described as follows. Every case x_i has a weight ${}_iw_t$ attached, where t denotes the number of trial. The weights ${}_iw_1$ for the first trial are set to $\frac{1}{N}$ for all i. For each case trial the following steps are taken: a model M_t is constructed using the training cases weight w_t. The error rate estimate e_t of the model is calculated as

$$e_t = \sum_{i \in M} {}_iw_t, \tag{2.28}$$

where M is the set of all incorrectly classified cases. If $e_t = 0$ or $e_t \geq 1/2$ the process terminates. Otherwise the weights w_{t+1} are set according to

$$
{}_i w_{t+1} = \frac{{}_i w_t}{c_i}, c_i = \begin{cases} 2e_t \\ \text{if } M_t \text{ misclassifies case } i \\ 2(1-e_t) \\ \text{otherwise} \end{cases} \tag{2.29}
$$

so the error rate of M_t under weights $w_t + 1$ is exactly 1/2. The final classifier is obtained by voting. For a case x_i each M_t adds $\log((1-e_t)/e_t)$ to the vote for the category it classifies x_i into. The final class for that case x_i is the one with the highest total vote. The advantage of boosting compared to previous methods is that it continuously adjusts weights to learn more from cases that tend to be misclassified. This is accompanied by higher computing costs.

The AdaBoost algorithm was the first step in the development of more general boosting method. Originally the term boosting referred to the AdaBoost algorithm, but recently more profound theory on methods implicitly used in AdaBoost was developed. Boosting in general can be seen as a stage-wise gradient procedure in an exponential cost function and can be defined for a variety of problems, including regression and unsupervised learning.

2.6.3 Random forests

Bagging and boosting work in general for a large class of predictors. Random forests are an example of an ensemble method which is tuned to a specific model class, here classification and regression trees. The idea is to modify the actual model building algorithm in order to obtain an improved predictor based on the combination of models. This approach allows the use of models which are not necessary optimal.

Random forests represent a general framework that uses randomization of the input for model fitting. The general idea is to randomly sample, permute or modify the data before constructing the tree model. In the sens bagging is merely one form of a random forest when used with classification or regression trees. The randomization is not limited to the global input, but it can be applied during the model construction process. The most well known type of random forest is the $randominputforest$. In each node F variables are selected at random from the set of explanatory covariates, then the locally optimal split is chosen from the F variables. Multiple trees are constructed using the procedure. The resulting trees are not pruned and the final predictor is constructed by aggregation as in the case of bagging. Surprisingly the choice of F as low as 1 also yields good results.

2.6.4 Bayesian model averaging

In a Bayesian framework we have a model $p\left(y|\theta\right)$ and a prior distribution for the parameters $\pi\left(\theta\right)$ reflecting the knowledge about *theta* before the data y are taken into account. Then we can compute the posterior distribution on the parameters *theta*, given the data:

$$p\left(y|\theta\right) = \frac{p\left(y|\theta\right)\pi\left(\theta\right)}{\int p\left(y|\theta\right)\pi\left(\theta\right)dy} \tag{2.30}$$

This distribution describes the updated knowledge about *theta* after the data were taken into account. In order to select one model it would be sufficient to pick parameters θ^* with the highest posterior probability $p\left(\theta^*|y\right)$. This would mean that we assume that $\theta^*$is the true parameter.

Instead of using a single model, the posterior distribution allows us to include our uncertainty in estimating *theta* in the prediction. Given a new observation z, we can obtain the prediction based on the predictive distribution:

$$p\left(z|y\right) = \int p\left(z|\theta\right)\cdot p\left(\theta|y\right)d\theta \tag{2.31}$$

Although the idea seems to be very appealing, the actual implementation can be fairly complex and is highly dependent on the models used. In practice it is sufficient to concentrate on models and *theta* with sufficiently high posterior probabilities. Furthermore the choice of the prior $\pi(\theta)$ ha significant influence on the results. In some cases it is a positive property, but in other cases it may mask the actual information in the data.

Chapter 3

Optimal Scaling Trees

3.1 Introduction

The framework of this chapter is supervised statistical learning in data mining. A typical data-mining problem is to deal with large sets of within-groups correlated inputs compared to the number of observed objects. In case of complex relationships standard tree-based procedures offer unstable and not ever interpretable solutions. For that multiple splits defined upon a suitable combination of inputs are required. In data mining it becomes central to have a prior treatment of the data through some exploratory tools based on feasible algorithms to be implemented even for huge data sets characterized by complex relations and so reduce the dimensionality of the problem could to be a basic goal ([53]). Segmentation methods have proved to be a powerful and effective nonparametric tool for high-dimensional data analysis. In certain context it results to be very important to investigate the role that each single variable plays in explaining the response. For example, when in presence of complex data structures, characterized by groups of co-variates internally correlated with each-others and hierarchically connected to a synthesis framework the need of a better interpretative value is particularly felt. The idea is to build up a tree-based model which nodes splitting are given by splits obtained from Nonlinear canonical correlation analysis's object scores. This method with k sets of variables is a generalization of linear CCA with k sets. The generalizations are used depends on subjective choices.

3.2 The methodology

3.2.1 Binary segmentation

A standard binary segmentation procedure aims to find at each node the best split of objects into two sub-groups which are internally the most homogeneous and externally the most heterogeneous with respect to the given output. The best split is found among all possible splits that can be derived from the given inputs, namely partitioning the categories of the input into two sub-groups so to provide the corresponding binary split of the objects. A similar approach is considered in multiway splits where the internal homogeneity within the r sub-groups is maximized.
The key idea of this part is to approach this problem using an inductive method. Without loss of generality, we consider the case of binary splits although generalizations of the proposed approach can be derived far r-way or multiway splits. First, we define the optimal partition of the objects into two subgroups which are the most internally homogeneous with respect to the given output without considering the inputs. Then, we look at the observed candidate partitions of the input features and their combinations which provide some alternative solutions that best approximate the optimal one. In other words, known the optimal solution we look far the most suitable combination of inputs which has the highest chance to provide nearly the best

partition of the objects.

3.2.2 The key idea

The proposed methodology is inspired by Two-stage segmentation and $FAST$ splitting algorithm. The TWO STAGE criteria ([79]) defines the tree structure through an approach organized in two main steps: first of all a sub set of original predictors that better explain the response variable, has to be identified. Then the best binary partition within those who were generated by the subset of predictors, has to be identified. Starting from that idea, in literature, has been proposed several two-stage approaches dealing with partitioning of non standard data structure, such as: $FAST$ ([77]) to reduce computational cost of analysis of huge datasets; $TS-DIS$ ([80], [99]) which uses linear discriminant functions to define a multivariate splitting criterion; Multi-Class Budget Tree ([7]) based on a latent budget partitioning algorithm introduced to analyze fuzzy data.
The idea consist on the definition of a splitting criteria with optimization criterion, using Nonlinear Canonical Correlation Analysis, allows to reduce the dimensionality of the analysis, shifting the attention towards a set of latent predictors synthesis of the original variables. The new latent variables will be the object scores extracted by Overals method. In the second step the methodology identify the best split of latent variables respect to the response. The second step, inspired by $TS-DIS$ idea, can be justified for the presence of a latent and discriminant variable for each group that is the nonlinear combination of other variables, allows to generate binary splits to which every predictor into the group contributes at the same time.
As already seen in the previous chapters we can obtain any split of the objects induced by splitting the predictor's modalities (Prospective split) or by splitting the response's modalities without caring for the predictors (Retrospective split). This method introduces the Op-

timal retrospective split maximizes the heterogeneity of the response variable across the two sub-nodes, thus providing the optimal discrimination of the objects, to define the theoretical (dummy) response variable Y^*. So the standard trees use the best prospective split such to minimize the heterogeneity of the observed response variable in the two sub-nodes. In our method we consider the optimal prospective split such to minimize the heterogeneity of the theoretical response variable in the two sub-nodes and the advantage is the heterogeneity of the theoretical response variable is lower than the one of the observed variable, thus the overall splitting procedure provides to reduce the misclassification error in few iterations.

3.2.3 Multiple split selection

Two sets Canonical Correlation Analysis is a technique that computes linear combinations of sets of variables which correlate in a optimal way. Generalized CCA does the same for k sets. Nonlinear CCA relates sets of nonlinearly transformed variables in a optimal way.
Several authors apply nonlinear transformations of multivariate techniques. This can be done in the form of optimal scaling. Optimal scaling means that for each categorical variable a nonlinear transformation is permitted, such that it maximizes the analysis criterion. Naturally the transformations are restricted by measurements constraints. Thus combining the CCA problem with measurement restriction gives CCA with optimal scaling.
The optimal scaling is included in k-sets CCA in the following manner [108]. Instead of using the original variables h (columns of H_1,...,H_k), transformed variables q (columns of Q_1,...,Q_k) are used, which are optimally scaled. The matrices Q_t and H_t have the same size. Geometrically this means that instead of considering a variable as a vector, a variable is considered as a cone of vectors of which one is chosen. Satisfying measurement restrictions for numerical variables means that q

is a linear transformation of h. For ordinal variables it means that q is a monotone transformation of h [63] and for nominal variables it means that q is equivalent with h.

Nonlinear canonical correlation analysis ([32], [108]) is applied in stage I. We consider as output the variable Y which summarizes the optimal split of the objects. We find the NLCCA's object scores minimizing

$$\sum\nolimits_{t=1}^{k} tr(X - Q_t A_t)'(X - Q_t A_t) \tag{3.1}$$

subject to the conditions that

$\mathbf{X}'\mathbf{X} = n\mathbf{I}$, $\mathbf{u}'\mathbf{X} = 0$ and $q = f(h)$ with $f \in C(h)$

where X are object scores, Q_t are the transformed variables from original variable matrix H and A_t are the collection of multiple and single category quantifications across variables and sets. As for each variable only one transformation is employed, defines k-sets CCA with single transformations.

Considering a variable as a collection of category scores, which means automatically that variables are supposed to be discrete, make measurement restrictions perhaps more clear. When transformations are defined with respect to categories, ties are automatically maintained. Then nominal transformations do not employ additional restrictions, ordinal transformations require the category quantifications to be a monotone transformation of the original category scores, and numerical transformations yield a linear transformation of the original category scores.

The selection of the best split of the objects into two sub-groups is done in stage II. The linear combinations Z_1,...,Z_G are the candidate splitting variables which generate the set S of prospective splits. These can be interpreted as multiple splits being defined on the basis of a combination of inputs. The best multiple split is found maximizing the between class deviation of the output:

$$s \equiv \arg max_s \{Dev_s(B)\} \tag{3.2}$$

for any split s in the set S. We can also calculate the efficiency measure based on the ratio between the between class deviation due to the best observed split, i.e., $Dev_s(B)$, and the between class deviation due to the optimal split, i.e., $Dev_k(B)$.

3.3 Customer Satisfaction Case Study

We use a dataset about a Customer satisfaction on Public Transport Service in Naples in 2005 (see table 3.1) with 1405 objects. In the case study we have 12 predictors divided in four groups and every group has three variables non linear correlated (see figure 3.1).

Node	n(t)	perc n(t)	Gini	Group	error rate
Node 1	1405	100,00	0,6362	1	58,51
Node 2	627	44,63	0,2983	1	8,93
Node 3	778	55,37	0,3212	1	24,55
Node 4	571	40,64	Terminal node (d)		0,000
Node 5	56	3,99	Terminal node (e)		0,000
Node 6	756	54,66	0,0854	2	22,35
Node 7	22	1,57	Terminal node (a)		0,000
Node 12	597	42,49	Terminal node (c)		0,000
Node 13	159	11,32	Terminal node (b)		0,000

Table 3.1: Customer Satisfaction study: main results

The response variable has five levels of global satisfaction according to Likert scale (a → totally unsatisfied; e → totally satisfied) and

ordinal predictors have 5 levels of satisfaction too. In the table 1 are illustrated the most important results. With n(t) we indicate the number of objects in the relative node, with perc n(t) the percentage of observations in node t, with Gini the node's impurity from Gini formula, with Group the latent variable selected to split the node and with error rate the number of misclassified in node t over the number of observations falling into node t.
In the case study we have chosen a response variable with more than two categories because we suppose that the proposed procedure could be an alternative to CART with categorical predictors and multi-class response variable (typical scenario in customer satisfaction analysis). From the results it underlines that after four split, with more than a thousand subjects, a perfect division appears. We have five terminal nodes and each terminal node has only subjects with the same category.

3.4 Concluding Remarks

This chapter has provided a partitioning procedure for a data mining problem, that is to find a tree-based model when there is a large set of within group correlated predictors. The goal is to treat directly the group as variable to split the node to understand the relevance of each group and not of the single original variable. The procedure is based on two-stage splitting criterion employing Nonlinear canonical correlation analysis and defining factorial multiple splits. The idea of using object scores' Overals is born because it isn't ever usual to have a linear relationship between the variables. The most important remarks obtained by this chapter are:

- The method improves the overall misclassification error rate at each node of the splitting procedure.

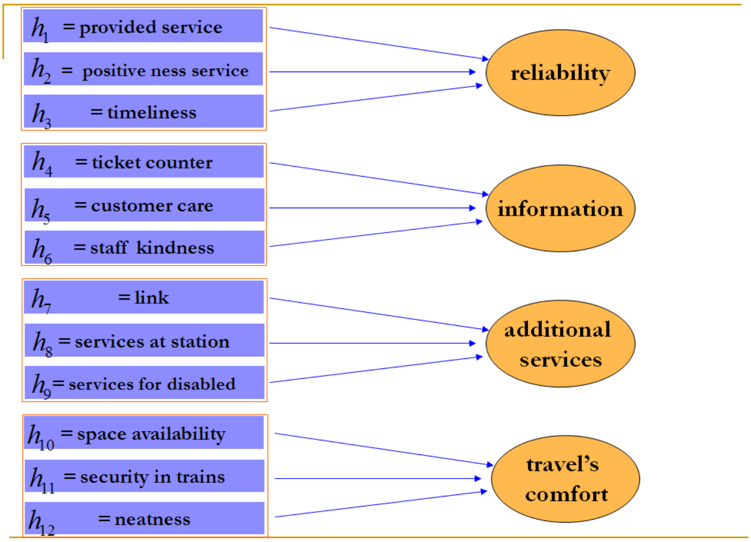

Figure 3.1: Customer Satisfaction schema

- At each node an efficiency measure allows to evaluate the gain as well as the distance toward to optimal tree.

- The monotone transformation of the predictors according the NLCCA provides to consider an optimal quantification of ordinal predictors.

Chapter 4

3Way Data

4.1 Introduction

So far segmentation methods for classification and regression trees have been proposed as supervised approach to analyse data sets where a response variable (of numerical or categorical type) and a set of predictors (of any type) are measured on a sample of objects or cases [53]. In binary segmentation, the aim is to find the best split of a predictor to split the cases into two sub-groups such to reduce the impurity of the response within each sub-group. The recursive splitting of the cases yields a tree structure. Pruning algorithms or ensemble methods allows to define a decision tree model to classify/predict new cases of unknown response on the basis of the measured predictors. To deal with more response variables it is possible to define multivariate trees [98].

Following the pioneer work [105], this chapter provides the methodological framework for the analysis of three-way data sets through tree-based models. Such data sets can be described by a cube, namely a set of variables (including both predictors and responses) is measured on a sample of units in a number of distinct situations, also called occasions. Each slide of the cube is a two-way data matrix,i.e. units times variables. Typically, the occasions are associated to modalities of a categorical variable. Alternatively, a time variable could be also considered as well. Main idea is to provide suitable classification and regression tree methods for the analysis of 3-way data sets.

4.2 The data set

The three ways of the data set are cases, attributes and situations, respectively. Let $\mathbf{D}$ be the three-way data matrix of dimensions N, V, Q, where N is the number of cases, objects or units, V is the number of variables, Q is the number of situations. Assume that the V variables can be distinguished into two groups, namely there are M predictor variables $X_1, \ldots, X_m, \ldots, X_M$ and C response variables $Y_1, \ldots, Y_c, \ldots, Y_C$ where $M + C = V$. The Q situations refer to modalities of a stratifying variable, which is called *instrumental variable.* Alternatively a time variable can be also considered for longitudinal data analysis. Predictors can be of categorical and/or numerical type whereas responses can be either categorical or numerical, thus a distinction can be made between a classification problem and a regression problem respectively.

4.3 Notation and definitions

4.3.1 Segmentation and decision trees

Tree-based methodology can have two mail goals: exploratory analysis, to describe the relationship between the predictors and the responses observed on a given sample, as well as confirmatory analysis,

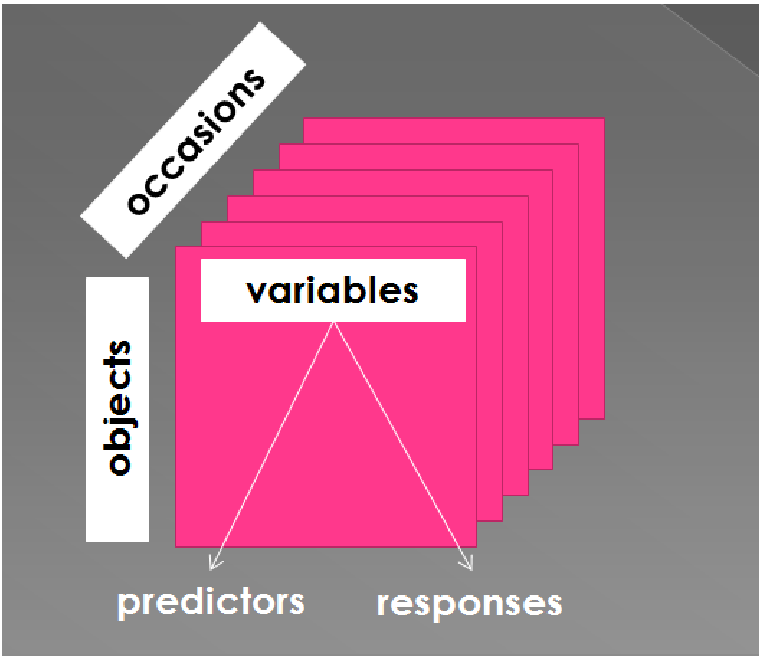

Figure 4.1: The cube: case, attributes and situations

to predict the unknown responses on the basis of the observed predictors measured on the given sample. Usually, for exploratory analysis a segmentation procedure is considered whereas for confirmatory analysis a segmentation procedure together with a decision rule induction is required. For *segmentation procedure* it is intended a recursive k-ary partitioning of a sample of units on the basis of the highest predictability power of M predictors on C responses in Q occasions. As a result, a k-ary tree is obtained describing the hierarchy role of the predictors in explaining the distribution of the responses, namely it allows to use tree-based model for statistical learning. For *decision rule induction* it is intended a tree-based model for each occasion to assign an accurate response class/value for each response variable on the basis of measurements of predictors on a new sample of units. A decision rule allows to use tree-based model for predictive learning.

4.3.2 Univariate and multivariate k-ary trees

Typically, both segmentation procedures and decision rule induction consider a learning sample where one response variable and a set of predictors are observed. Thus, for $C = 1$, namely one response variable, we define *univariate trees*. Instead, for $C > 1$, namely more response variables, we define *multivariate trees*. Furthermore, any partitioning procedure requires the definition of the number of groups of the partition. Binary segmentation is usually considered, where the partition of any tree-node is into two subnodes. More in general, ternary trees could be also considered, as well as *k-ary trees*, where the partition of any tree-node is into k subnodes, for the Q situations. The partitioning criterion can be based either on a predictability statistical index (i.e., Goodman and Kruskal tau index, η^2 Pearson coefficient) or on a statistical model (i.e., factorial methods, logistic regression, linear regression, etc.). Main issue is to evaluate the most predictive partition of the predictor modalities to determine the partition of the node sample of units. The partition of the node sample into k subsamples is obtained on the basis of the k-ary partition of the predictor space. This type of partition is defined as a *prospective partition*. Among all possible prospective partitions, the best one is such that the response variables are internally the most homogeneous and externally the most heterogeneous. Heterogeneity is measured in terms of variation in case of numerical responses (i.e., regression trees) and in terms of entropy or mutability in case of categorical responses (i.e., classification trees). It is also possible to define a *retrospective partition*, namely the theoretical partition of the node sample of units into k subsamples such that the response variables are internally the most homogeneous without caring for the predictors. This is a theoretical partition of units to which it does not correspond necessarily a partition of the predictor modalities. In this respect, it is a benchmarking for the best prospective partition. In other words, the degree of internal homogeneity of the responses induced by the theoretical partition is the upper bound for the degree of internal homogeneity induced by the best prospective partition.

4.3.3 Tree-based methods for two-way univariate trees

Assume the three-way **D** matrix with the special constraints $Q = 1$ and $C = 1$, thus one occasion and one response variable. Indeed, the cube matrix reduces to a two-way data matrix of standard type, where M predictors and one response variable are measured on a sample of units. Tree-based methods discussed in chapter one are suitable for analyzing such data set. They yield to two-way univariate trees.

4.3.4 Two-stage partitioning methods for two-way multivariate trees

Assume the three-way **D** matrix with $Q = 1$, thus only one occasion. The cube matrix reduces to a two-way data matrix where responses are more than one, i.e. $C > 1$, as well as also predictors are more than one, i.e. $M > 1$. Some of the methods discussed in chapter one can be extended to deal with more responses. This special case turns to yield two-way multivariate trees.

As an example, the two-stage splitting criterion for two-way multivariate binary trees is as follows:

$$\max_{m}\sum_{c}\gamma_{(Y_c|X_m)}(t)p_{Y_c}(t) \tag{4.1}$$

$$\max_{s}\sum_{c}\gamma_{(Y_c|s)}(t)p_{Y_c}(t) \tag{4.2}$$

for $c = 1, \ldots, C$ (i.e. responses), $m = 1, \ldots, M$ (i.e. predictors), $s = 1, \ldots, S$ (i.e. splitting variables), with $\sum_c p_{Y_c}(t) = 1$. The global impurity proportional reduction measure is defined as a weighted average of the measures calculated for each given response. A suitable weighting system $p_{Y_c}(t)$ can be given by the percentage of the total impurity due to each response. Anagously, it can be defined the local impurity proportional reduction measure due to each splitting variable. This criterion assumes that the response variables are independent, so that the best predictor as well as the best split is selected according to the best compromise of the predictability measures. On the basis of the type of response variables, we can choose a suitable impurity measure for classification trees as well as for regression trees.

4.4 3-way segmentation

4.4.1 3-way univariate and multivariate trees

In general, 3-way segmentation can be defined when the number of occasions in the three-way **D** matrix is larger than one, i.e., $Q > 1$. Furthermore, any 3-way approach yields to either univariate or multivariate trees as soon as $C = 1$ or $C > 1$. It is interesting to consider constrained versions of the data matrix to take into account special hypotheses of analysis. In the following, two main constrained versions are described for univariate trees. The extension to multivariate trees can be also developed.

4.4.2 Constrained version: multi-group of units

Assume the three-way **D** matrix with $C = 1$, but $M > 1$ and $Q > 1$. Consider an instrumental variable Z_o having Q modalities to distinguish Q subsamples of units or objects. A binary segmentation procedure can be understood as a recursive splitting of the sample and its subsamples such that the response variable is the most homogeneous within each subsample and the most heterogeneous across the distinct subsamples simultaneously. Thus, the best split of the sample and its subsamples is based on the available predictors. The two-stage splitting criterion can be defined as follows:

$$\max_m \sum_q \gamma_Y(t|_q X_m) p_Y(t|q) \tag{4.3}$$

$$\max_s \sum_q \gamma_Y(t|s) p_Y(t|q) \tag{4.4}$$

for $q = 1, \ldots, Q$ (i.e. subsamples), $m = 1, \ldots, M$ (i.e. predictors), $s = 1, \ldots, S$ (i.e. splitting variables), with $\sum_q p_Y(t|q) = 1$. The global impurity proportional reduction measure is defined as a weighted average of the measures calculated across the Q occasions. A suitable weighting system $p_Y(t|q)$ can be given by the percentage of the total impurity of the response in each subsample. Anagously, it can be defined the local impurity proportional reduction measure due to each splitting variable. On the basis of the type of response variables, we can choose a suitable impurity measure for classification trees as well as for regression trees.

4.4.3 Constrained version: multi-block of predictors

Assume the three-way **D** matrix with $C = 1$, but $M > 1$ and $Q > 1$. Consider an instrumental variable Z_p having Q modalities to distinguish Q blocks of predictors. A binary segmentation procedure can be understood as a recursive multiple splitting of the sample such that the response variable is internally the most homogeneous and externally the most heterogeneous where the multiple split summarizes the predictability of the distinct blocks of predictors. Statistical models such as discriminant analysis or factorial analysis can be considered to summarize the information provided by each block of predictors into a multiple split.

Chapter 5

Multiple Discriminant Trees

5.1 Introduction

The framework of this part is supervised learning using classification trees. Two types of variables play a role in the definition of the classification rule, namely a response variable and a set of predictors. The tree classifier is built up by a recursive partitioning of the prediction space such to provide internally homogeneous groups of objects with respect to the response classes. In the following, we consider the role played by an instrumental variable to stratify either the variables or the objects. This yields to introduce a tree-based methodology for conditional classification. Two special cases will be discussed to grow *multiple discriminant trees* and *partial predictability trees.* These approaches use discriminant analysis and predictability measures respectively. Empirical evidence of their usefulness will be shown in real case studies.

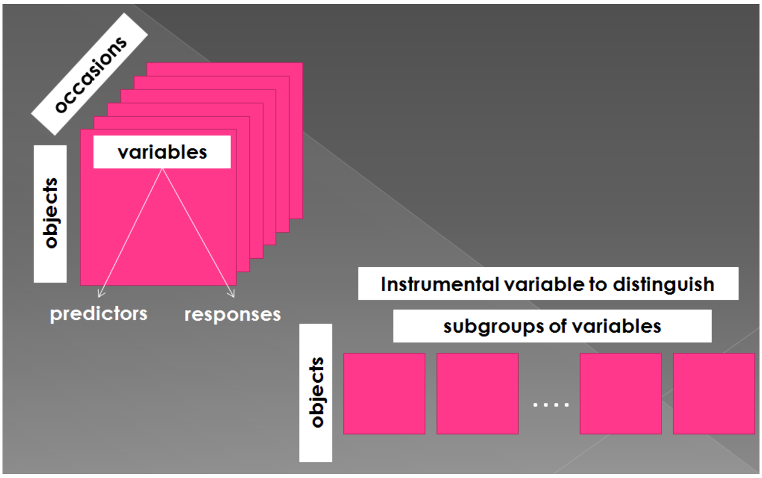

Figure 5.1: Constrained version 1: ”Multiple”

5.1.1 Nowdays data analysis

Understanding complex data structures in large databases is the new challenge for statisticians working in a variety of fields such as biology, finance, marketing, public governance, chemistry and so on. Complexity often refers to both the high dimensionality of units and/or variables and the specific constraints among the variables. One approach is Data Mining [52], namely the science of extracting useful information from large data sets by means of a strategy of analysis considering data preprocessing and statistical methods. Another approach is Machine Learning that combines data-driven procedures with computational intensive methods by exploiting the information technology such to obtain a comprehensive and detailed explanation of the phenomenon under analysis. Turning data into information and then information into knowledge are the main steps of the knowledge discovery process of statistical learning [53] as well as of intelligent data analysis [52]. Key questions in the choice of the best strategy of analysis refer to the type of output (i.e., regression or classification), the type of variables (i.e., numerical and/or categorical), the role played by the variables (i.e., dependent or explanatory), the type of statistical units (i.e., observational or longitudinal data), the type of modeling (i.e., parametric or nonparametric).

5.1.2 Binary segmentation

In this framework, segmentation methods have proved to be a powerful and effective nonparametric tool for high-dimensional data analysis. A tree-based partitioning algorithm of the predictor space allows to identify homogeneous sub-populations of statistical units with respect to a response variable. A tree path describe the dependence relationship among the variables explaining the posterior classification/prediction of units. Any induction procedure allows to classify/predict new cases of unknown response [96]. In this part, we refer, obviously, to CART methodology for classification trees [15] and some advancements provided by two-stage segmentation [79] and the fast algorithm [77]. At each node of the tree, a binary split of sample units is selected such to maximize the decrease of impurity of the response variables when passing from the top node to the two children nodes. This objective function can be shown to be equivalent to maximizing the predictability measure of the splitting variable to explain the response variable. Typically, candidate splits are all possible dichotomous variables that can be generated by all predictor variables, but the fast algorithm allows to find the optimal solution of CART without trying out all possible splits.

5.1.3 The genesis of the idea

As a matter of fact, when dealing with complex relations among the variables, any CART-based approach offers unstable and not interpretable solutions. An alternative in case of within-groups correlated inputs (of numerical type) is two-stage discriminant trees, where splitting variables are linear combinations of each group of inputs derived by the discriminant factorial analysis [80].
This work aims to define a segmentation methodology for three-way data matrix starting from some recent results [105]. A three-way data matrix consists of measurements of a response variable, a set of predictors, and in addition a stratifying or descriptor variable (of categorical type). The latter play the role of conditional variable for either the predictor variables or the objects (or statistical units). Two basic methods will be discussed in details providing some justification for the concept of supervised conditional classification in tree-based methodology. Our proposed segmentation methods for complex data structures are all introduced in the Tree-Harvest Software [9] [95] using MATLAB.

5.2 Multiple discriminant trees

5.2.1 Notation and definition

Let Y be the output, namely the response variable, and let $\mathbf{X} = \{X_1, \ldots, X_M\}$ be the set of M inputs, namely the predictor variables. In addition, let Z_P be the stratifying predictor variable with G categories. The response variable is a nominal variable with J classes and the M predictors are covariates, thus of numerical type. The input variables are stratified into G groups, on the basis of the instrumental variable Z_P. The g-th block of input variables includes m_g input variables $\mathbf{X}_g = ({}_gX_1, \ldots, {}_gX_{m_g})$ for $g = 1, \ldots, G$.

5.2.2 The multiple method

The proposed method aims to replace blocks of covariates by their linear combinations applying the factorial linear discriminant analysis. In particular, the discriminant analysis is applied twice, first to summarize each block of input variables (within-block latent compromise) and then to find a compromise of all blocks (across-blocks latent compromise). In the first stage, the attention is shifted from the G sets of original covariates to G latent variables, obtained searching for those linear combinations of each block of original variables which summarize the relationships among the covariates with respect to a grouping variable. In our approach, the role of grouping variable is played by the response variable. In the second stage, the algorithm runs to the creation of one global latent variable, synthesis of the previous discriminant functions obtained in the first step. On the basis of such compromise the best split will be found.

5.2.3 Within-block latent compromises

The process is to divide up a m_g dimensional space into pieces such that the groups (identified by the response variable) are as distinct as possible. Let $\mathbf{B}_g$ be the between group deviation matrix of the inputs in the g-th block and $\mathbf{W}_g$ the (common) within group deviation matrix. The aim is to find the linear combination of the covariates:

$$\phi_g = \sum_{m=1}^{m_g} {}_g\alpha_m \cdot {}_gX_{mg} \tag{5.1}$$

where ${}_g\alpha_m$ are the values of the eigenvector associated to the largest eigenvalue of the matrix $\mathbf{W}_g{}^{-1}\mathbf{B}_g$. The (5.1) is the g-th linear combination of the inputs belonging to the g-th block with weights given by the first eigenvector values. It is obtained maximizing the predictability power of the m_g inputs to make the J classes as distinct as possible.

Moreover, the ϕ_g variables are all normalized such to have mean equal to zero and variance equal to one.

5.2.4 Across-block latent compromise

As second step, we find a compromise of the G blocks applying the linear discriminant analysis once again, thus obtaining:

$$\psi = \sum_{g=1}^{G} \beta_g \phi_g \tag{5.2}$$

where β_g are the values of the eigenvector associated to the the largest eigenvalue of the matrix $\mathbf{W}^{-1}\mathbf{B}$. These matrices refer to the between group deviation matrix of the discriminant functions ϕ_g for $g = 1, \ldots, G$.

5.2.5 Multiple factorial split

Finally, the best split will be selected among all possible dichotomizations of the ψ variable maximizing the decrease of impurity function. As a result, a multiple discriminant split is found where all covariates play a role that can be evaluated considering both the set of coefficients of the linear combination of the blocks and the set of coefficients of the linear combination of the covariates belonging to each block.

5.2.6 The computational steps

The three-steps procedure can be justified with a two-fold consideration: on one hand, a unique global latent and discriminant variable (i.e., the linear combination of within-block latent variables) allows to generate binary splits for that all predictors have contributed; on the other hand, taking into account the within-block latent variables, it is possible to calculate a set of coefficients that represent each the weight of the link among those, the predictors, the response variable and the global latent variable. In other words, if the conditions for the application are verified, the addition of a third stage allows a better interpretation for the explanation of the phenomenon, because all the variables act simultaneously at the same time the split is created, but it is possible to interpret the valence of each of those towards both response and dimensional latent variables.

5.2.7 Multiple discriminant trees: Local Transport Survey

Multiple discriminant tree method has been fruitfully applied for a Customer Satisfaction Analysis. On 2006, a survey of $N = 1290$ customers of a local public transport company in Naples has been collected measuring the level of global satisfaction and the level of satisfaction with respect to four dimensions of the service, each considering three aspects.

The response variable has two classes distinguishing the satisfied and the unsatisfied customers. The strata of the instrumental variable Z_P are service's reliability, informations, additional services and travel's comfort, each is characterized by three ordinal predictors, where a Thurstone data transformation has allowed to treat them as numerical ones. Figure 3 describes the role played by the variables in discriminating between satisfied and unsatisfied customers. Table 4 provides summary information concerning the path yielding to a terminal node which label is satisfied customer, whereas Table 5 concerns the path yielding to a terminal node which label is unsatisfied customer. Table 4 describes for each node the number of individuals, the split-score, the BETA coefficients of each dimension and the ALPHA coefficients within each dimension. From the Table 4 it is clear

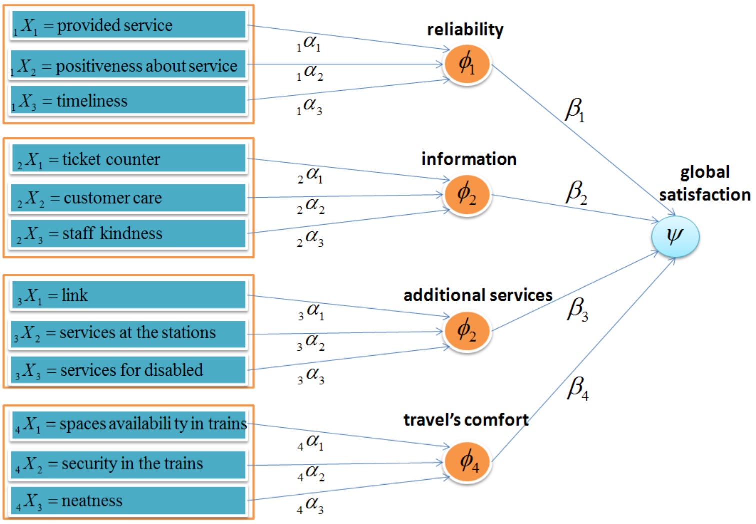

Figure 5.2: Discriminant Satisfaction Schema (Local Transport System Survey)

the high strength of dimension 1 in the split of node 1 and of node 2 as the BETA coefficient is relatively bigger for dimension 1 with respect to the others. Only in the split of node 4 of this path another dimension has the highest coefficient. In the Table 5 for every split the highest BETA coefficient is always in the dimension 1. It is evident that the satisfied customers are well discriminated considering just the dimension relative to reliability; instead for the unsatisfied customers are important all the dimensions of the service.

Node	**n**	**score**	**DIM**	1	2	3	4
node 1	1290	145.42	BETA	**0.59**	0.41	0.45	0.34
			ALPHA	10.53	10.50	5.68	9.70
				5.45	4.80	6.01	6.06
				8.80	6.93	10.35	7.30
node 2	473	106.07	BETA	**1.15**	1.01	1.02	0.94
			ALPHA	2.69	1.92	2.00	2.35
				0.41	2.31	1.47	1.31
				2.38	3.40	2.23	2.83
node 4	129	68.27	BETA	1.13	**1.25**	0.98	1.17
			ALPHA	1.73	2.07	0.53	1.00
				0.85	0.31	0.74	1.80
				2.50	2.09	2.97	0.46
node 8	51		*terminal node*				

Table 5.1: Path 1-8: Terminal label - Unsatisfied customers of Local Transport System

Node	**n**	**score**	**DIM**	1	2	3	4
node 1	1290	145.42	BETA	**0.59**	0.41	0.45	0.34
			ALPHA	10.53	10.50	5.68	9.70
				5.45	4.80	6.01	6.06
				8.80	6.93	10.35	7.30
node 3	817	117.71	BETA	**0.67**	0.24	0.43	0.37
			ALPHA	7.54	7.23	6.68	8.27
				6.85	4.27	3.52	6.07
				7.87	5.75	7.61	5.87
node 6	300	92.44	BETA	**1.01**	0.80	0.89	0.87
			ALPHA	1.19	1.29	1.72	2.56
				2.82	2.25	1.74	1.59
				4.51	0.81	1.66	3.18
node 13	210	46.74	BETA	**0.72**	0.04	0.36	0.28
			ALPHA	4.32	3.25	5.71	3.81
				3.59	1.51	0.00	3.31
				3.54	2.24	3.72	2.96
node 27	122	34.43	BETA	**0.77**	0.14	0.29	0.28
			ALPHA	2.52	1.36	4.31	2.67
				3.01	2.03	0.27	3.17
				3.21	1.52	2.51	1.64
node 55	51		*terminal node*				

Table 5.2: Path 1-55: Terminal label - Satisfied customers of Local Transport System

Chapter 6

Partial Predictability Trees

6.1 Notation and definition

Let Y be the output, namely the response variable, and let $\mathbf{X} = \{X_1, \ldots, X_M\}$ be the set of M inputs, namely the predictor variables. In addition, let Z_O be the stratifying object variable with K categories. The response variable is a nominal variable with J classes and the M predictors are all categorical variables (or categorized numerical variables). The sample is stratified according to the K categories of the instrumental variable Z_O.

6.2 The partial method

We consider the two-stage splitting criterion [79] based on the predictability τ index of Goodman and Kruskal [47] for two-way cross-classifications: in the first stage, the best predictor is found maximizing the global prediction with respect to the response variable; in the second stage, the best split of the best predictor is found maximizing the local prediction. It can be demonstrated that skipping the first

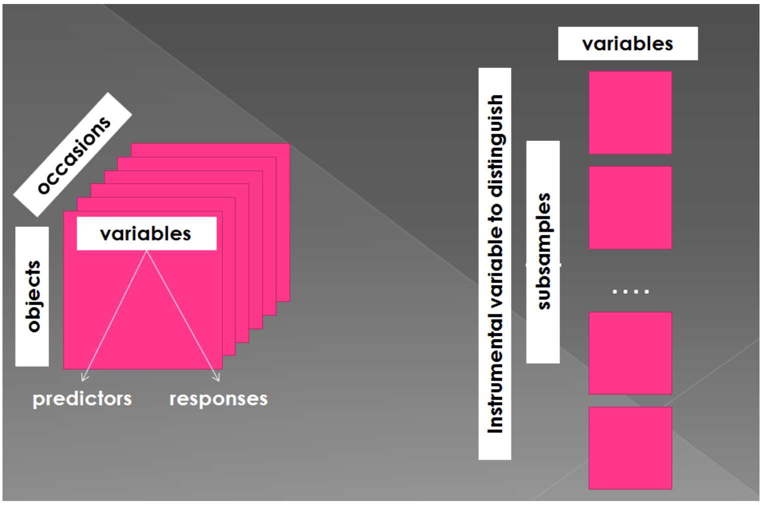

Figure 6.1: Constrained version 2: ”Partial”

stage maximizing the simple τ index is equivalent to maximizing the decrease of impurity in CART approach. In the following, we extend this criterion in order to consider the predictability power explained by each predictor/split with respect to the response variable conditioned by the instrumental variable Z_O. For that, we consider the predictability indexes used for three-way cross-classifications, namely the multiple τ_m and the partial τ_p predictability index of Gray and Williams [49], that are extensions of the Goodman and Kruskal τ_s index.

6.2.1 The splitting criterion

At each node, in the first stage, among all available predictors X_m for $m = 1, \ldots, M$, we maximize the partial index $\tau_p(Y|X_m, Z_O)$ to find the best predictor X^* conditioned by the instrumental variable Z_O:

$$\tau_p(Y|X_m, Z_O) = \frac{\tau_m(Y|X_m Z_O) - \tau_s(Y|Z_O)}{1 - \tau_s(Y|Z_O)} \tag{6.1}$$

where $\tau_m(Y|X_m Z_O)$ and $\tau_s(Y|Z_O)$ are the multiple and the simple predictability measures. In the second stage, we find the best split s^* of the best predictor X^* maximizing the partial index $\tau_s(Y|s, Z_O)$ among all possible splits of the best predictor. It can be possible to apply a CATANOVA testing procedure using the predictability indexes calculated on an indipendent test sample as stopping rule [98].

6.3 Applications

6.3.1 Partial predictability trees: German Credit Survey

There are several fields in which this methodology can be applied with good results. In this section, we present an application about credit leave in Germany. The data regard a survey collected by Professor Dr. Hans Hofmann, University of Hamburg, with $N = 2026$ [83]. Table 1 describes the predictors of German credit dataset. The response variable is a dummy variable, namely the good and the bad client of the bank.

Predictors	
1. Account Status	10. Present residence since
2. Credit history	11. Property
3. Purpose	12. Age
4. Duration	13. Other installment plans
5. Saving accounts/bond	14. Housing
6. Present employment since	15. Existing credits at this bank
7. Installment rate of disposable income	16. Job
8. Personal status and gender	17. People being liable to provide maintenance for
9. Others debitors/guarantors	18. Telephone

Table 6.1: Predictors in the German Credit Dataset

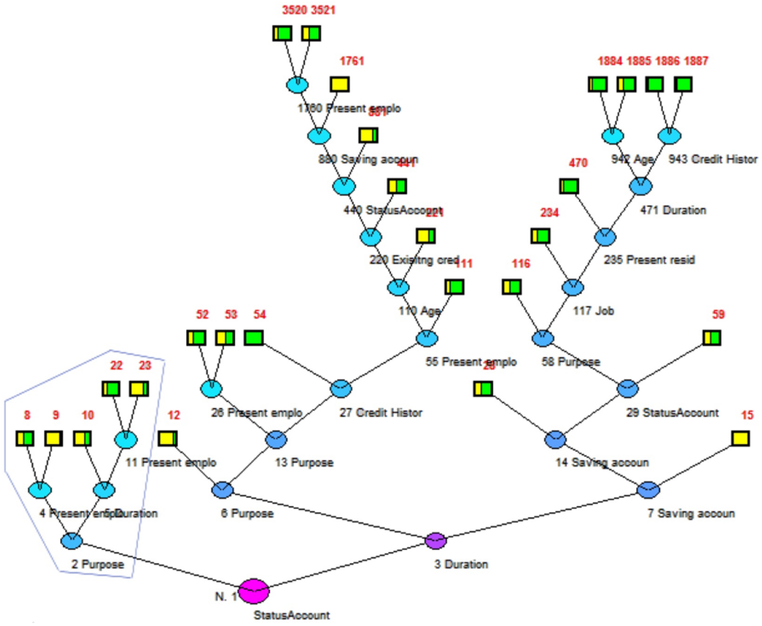

Figure 6.2: Partial Predictability Tree Graph (German Credit Survey)

A proper classification rule should consider the different typologies of bank customer. This can be considered as the instrumental Z_O having four strata ordered on the basis of the credit amount requested. Figure 1 shows the final binary tree with 26 terminal nodes, where nodes are numbered using the property that the node t generates the left subnode $2t$ and the right subnode $2t + 1$; we denote the predictor used for the split at each nonterminal node and by distinct color the response class distribution within each terminal node. The branch hold by node 2 is described in details in Figure 2. It is interesting to point out some useful information by interpreting this type of output results. Each node is divided in four parts (one for every

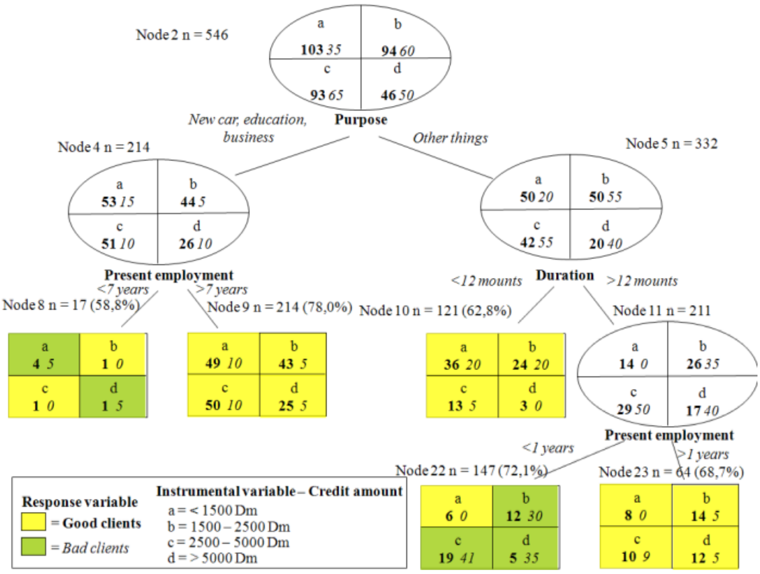

Figure 6.3: An example of data interpretation: branch of node 2 (German Credit Survey)

category of the instrumental variable) and close to the terminal nodes the percentage of good classified within each group is indicated. In addition, the predictor used for the splits of all strata of cases is also indicated. It can be noticed that the split at the node 2 is based on the credit purpose: in the left subnode (i.e., the node 4) there are relatively more good clients than bad clients as soon as the credit amount increases, their credit is for new car, education and business; in the right subnode (i.e., the node 5) there are relatively more bad clients and the good clients are identified in the further split on the basis of a duration of the credit lower than 12 months. Finally, clients employed by less than 1 year can be considered as a bad client as soon as the credit amount increases. As further examples of output, Table

2 provides summary information concerning the path yielding to the terminal node 23 which label is good client, whereas Table 3 concerns the path yielding to the terminal node 54 which label is bad client. In particular, in each table we report the response classes distribution of the objects within the four strata of Z_O, for the predictor selected in each nonterminal node. In addition, we give the Catanova test significance value.

As an example, we can see in Table 2 that the original scenario shows a bigger presence of bad clients than good clients in all four strata. After the first split, the situation changes in the first three strata, instead in the fourth there are more bad clients than good. Only after four splits in all strata there is a bigger presence of good clients although there are clear differences within each stratum.

Response Classes Distribution			z_1		z_2		z_3		z_4		
Node	**n**	**Predictor**	**G**	**B**	**G**	**B**	**G**	**B**	**G**	**B**	**C sign.**
1	2026	Account Status	205	415	171	295	199	375	91	275	0,0000
2	546	Purpose	103	35	94	60	93	65	46	50	0,0000
5	332	Duration	50	20	50	55	42	55	20	40	0,0000
11	211	Present employm. since	14	0	26	35	29	50	17	40	0,0000
23	64	*Terminal Node*	8	0	14	5	10	10	12	5	0,0080

Table 6.2: Path $1-23$: Terminal label - Good client - (German Credit Survey)

Response Classes Distribution			z_1		z_2		z_3		z_4		
Node	**n**	**Predictor**	**G**	**B**	**G**	**B**	**G**	**B**	**G**	**B**	**C sign.**
1	2026	Account Status	205	415	171	295	199	375	91	275	0,0000
3	1480	Duration	102	380	77	235	106	310	45	225	0,0031
6	720	Purpose	95	295	56	95	44	85	10	40	0,0004
13	698	Purpose	89	290	50	95	39	85	10	40	0,0027
27	491	Credit History	50	220	28	65	28	70	10	20	0,0028
54	71	*Terminal Node*	0	40	0	15	1	10	0	5	0,0195

Table 6.3: Path $1-54$: Terminal label - Bad client - (German Credit Survey)

6.4 Concluding remarks

This chapter has provided conditional classification trees using an instrumental variable. Two cases have been discussed. In the first, the response variable is a dummy variable, the predictors are numerical and the instrumental variable provides to distinguish them into a set of different blocks. Standard tree-based models would find at each node a split based on just one predictor regardless the multi-block structure of the predictors that can be internally correlated. We have introduced a method to grow *multiple discriminant trees* where the discriminant analysis is used to summarize the information within each block and a multiple splitting criterion has been defined accordingly. At each node of the tree, we are able to assign a coefficient of importance to each predictor within each block as well as to each block in the most suitable discrimination between the two response classes. A Customer Satisfaction Analysis based on a real data set has been briefly presented in order to show some issues in the interpretation of the results. The second case deals with all categorical variables and the instrumental variable provides to distinguish different subsamples of objects. A standard splitting criterion would divide the objects regardless their subsamples belonging. We have introduced a splitting criterion that finds the best split conditioned by the instrumental variable. This yields to grow *partial predictability trees* that can be understood as an extension of two-stage segmentation and to some extent of CART approach. An application on a well-known real data set has been briefly described in order to point out how the procedure gives, at each node, the set of response class distributions, one for each sub-sample. The results of several applications have been very promising for both methods, showing that our methodology works much better than CART standard procedure to explain the interior structure of tree models. Both the two procedures have been implemented in MATLAB environment enriching the Tree Harvest Software we are developing as an alternative to standard tree-based methods for special structures of data.

Conclusions

Tree-based methodology is becoming fundamental for data mining and knowledge discovery process in presence of huge data and complexity. It aims to a supervised learning from data, thus a target or response variable is required distinguishing classification trees when the response is qualitative and regression trees when it is numerical. A tree-based model is non parametric, it does not care for probabilistic distribution assumptions on the variables that can be of any type, it can be considered either for exploratory analysis or for decision-making. Main results in literature concern partitioning procedures to describe the hierarchical relationship of the response variable on the basis of the given predictors, as well as procedures to yield accurate and robust decision rules for new objects. A trade off between the easy interpretability of the tree graph and the accuracy of the decision rules is still considered a challenge of the statisticians working in this field. As a matter of fact, further improvements in the tree-based approach are required in presence of complex data structures. Real world applications often deal with data sets characterized by a great number of predictors and predictors often play a different role in the analysis. As an example, predictors can be structured into different groups, where, within each group, they are intrinsically related to each other; as a result, the splits of the partitioning procedure as well as the final decision rule can be built up using very few predictors and the overall result can be not robust.

This thesis has focalized the attention on the data and their structure, considering two main directions of research.

The first direction takes origin from the two-stage discriminant tree-based method that is used when the response variable, numerical or dummy variable, is explained by a set of internally correlated numerical predictors. The splitting criterion is based on discriminant analysis in order to summarize, in terms of predictability power, the information of each block of predictors so that a multiple (factorial) split is defined. In the present dissertation, an alternative method has been proposed. Main issue is to remove the assumption of linearity in the dependence relationship of the response variable on the predictors, thus considering a nonlinear multivariate approach. As a result, the optimal scaling tree method has been introduced, in particular a splitting criterion allows to define a multiple factorial split using an alternating least squares estimation algorithm. The proposed method can be fruitfully used for multi-class response variable and categorical predictors, such as for instance in customer satisfaction studies, where both predictors and responses are ordinal describing the level of satisfaction associated respectively to distinct dimensions of the satisfaction and the overall global satisfaction.

In the second direction of research, this thesis deals with three-way data where variables are measured on objects in distinct occasions or situations. Main idea is to introduce a stratifying variable in the analysis taking into account prior information through specific constraints upon either variables or objects. Two distinct partitioning procedures have been provided: the multiple discriminant tree and the partial predictability tree, the former for numerical variables and the latter for categorical ones. Multiple discriminant trees can be understood as a further extension of two-stage discriminant tree, enabling all predictors to provide a contribution to the definition of the split at each internal node of the tree. As a result, it is possible to assign an impact factor of each predictor as well as of each group of predictors in the definition of the partition of a node sample into two subgroups. In the latter, namely the partial predictability tree, the stratifying variable allows to distinguish groups of objects, thus a suitable splitting criterion has been defined to find the best simultaneous partition of the objects. Main issue is to provide distinct response distributions in each node sample and the best split is found as a compromise of the impurity reduction of all response class distributions in the distinct subsamples. The final result is to define a general methodology for 3-way data, offering a new bridge to the scientific contribution in the field of classification and regression trees.

Finally, the proposed methods have been implemented in MATLAB environment so that the corresponding routines can be inserted into Tree Harvest Software developed by the research group of Naples specialized in tree-based methods, providing a step further to the dissemination of nonstandard tree-based models to deal with complex data structures.

Appendix A

MatLab Code

A.1 Multiple Discriminant Trees

```
function [matriceX,sintpadre,sintfiglio,varmax,imp,decimp,tree,nodihandle,
sintesi2]=
threeLDAtree(X,Xlabel,gruppi,maxoss,decmin);

nodoL=1;
memnodo=[0,1];
lung=length(memnodo);
sizeX=size(X);
nodo1(1:sizeX(1),1)=1;
matriceX=[nodo1(:,1),X];

%%number of classes choice of Y
[newX,nclass,percenti]=twostageclassi(X(:,1),4);

percentili=percenti.variabile.perc;

it=0;it2=0;
tree.nodo(1).X=X;
tree.nodo(1).label=mean(X(:,1));
tree.nodo(1).impurita=std(X(:,1));
tree.nodo(1).term=0;
cont=0;
imp.decimpurita=0;
imp.nodo=0;
```

```
sintpadre.nodoL=0;
while memnodo(lung) ~= 0         % routine to grow tree
    it=it+1;
    sizeX=size(X);
    while sizeX(1) >= maxoss % routine for the split

[XSLL,XSRR,varmax,col,spli,tab,tab2,Bbest,BLDA,ef1,percmislda,lbaparcol,
indbBest,decimpBest,U,psi,score,correl]=threestageLDA(X,gruppi);

        impurita=decimpBest;

        if  impurita < decmin & nodoL>1
            nodoL
            tree.nodo(nodoL).decimp=impurita;
            tree.nodo(nodoL).term=1;
            'break'
            break
        end

        it2=it2+1;
        imp.decimpurita(it2)=impurita;
        imp.nodo(it2)=(nodoL/2);
        cont=cont+1;


        sintpadre.nodoL(cont)=nodoL;
        sintpadre.numnodo(cont)=length(X(:,1));
        sintpadre.col(cont)=col;
        sintpadre.split(cont)=spli;
        sintpadre.ef1(cont)=ef1;
        sintpadre.misclas(cont)=percmislda;
        sintpadre.tabmisclass(cont).tab=tab2;

% to assign a new notation
        nodoL=nodoL*2;
        nodoD=nodoL+1;

        tree.nodo(nodoL).X=XSLL;
        tree.nodo(nodoD).X=XSRR;
        tree.nodo(nodoL/2).X=[XSLL;XSRR];
        tree.nodo(nodoL/2).col=col;
        tree.nodo(nodoL/2).U=U;
        tree.nodo(nodoL/2).correl=correl;
        tree.nodo(nodoL/2).score=score;
        tree.nodo(nodoL/2).psi=psi;
        tree.nodo(nodoL/2).term=0;
        tree.nodo(nodoL/2).coeff=lbaparcol;
        tree.nodo(nodoL/2).split=spli;
        tree.nodo(nodoL/2).misclass=percmislda;
        tree.nodo(nodoL/2).tabmisclass=tab2;
        tree.nodo(nodoL/2).decimp=impurita;
% data child-nodes
        sintfiglio.destro.tab1(cont,:)=[nodoD tab(2,:)];
        sintfiglio.sinistro.tab1(cont,:)=[nodoL tab(1,:)];

        XSLL(:,1)=[];
        XSRR(:,1)=[];
        matriceX=joinnode(XSRR,XSLL,matriceX,nodoL);
        memnodo=[memnodo,nodoD,nodoL];
        X=tree.nodo(nodoL).X;
        lung=length(memnodo);
        memnodo(lung)=[];
        sizeX=size(X);

    end
    if sizeX(1) < maxoss
        tree.nodo(nodoL).term=1;
    end
    lung=length(memnodo);
    nodoL=memnodo(lung);

    if sizeX(1) < maxoss | impurita < decmin
        memnodo(lung)=[];
        lung=length(memnodo);
    end

    %matrix X;


    if nodoL > 1
        X=newmatrix(matriceX,nodoL);
    end

end

% compute goodness
noditot=[sintfiglio.destro.tab1(:,1:2) ; sintfiglio.sinistro.tab1(:,1:2)];
noditot2=[sintfiglio.destro.tab1 ; sintfiglio.sinistro.tab1];
noterm=sintpadre.nodoL';
noterm(1)=[];
noterm;
n=length(noterm);
m=length(noditot(:,1));
cont=0;

% definition terminal nodes
% to compute goodness function
for j=1:m
    term=1;
    for i=1:n
        if noterm(i) == noditot(j,1)
            term=0;
            i=n;
        end
    end
    if term==1
        cont=cont+1;
        sintesi(cont,1:2)=noditot(j,1:2);
        sintesi2(cont,1:4)=noditot2(j,:);
    end
end



sintesi=sortrows(sintesi,1);
decimp = goodness(matriceX,sintesi);


    %inputdlg('Print Tree? (1=Yes 0=No)    ')
   [tree,nodihandle]=tsdisrplot(Xlabel,sintpadre,sintfiglio,sintesi2,tree,nclass,
   percentili);
   %% Print tree
metodo=4;
cd results
save albero nclass percentili metodo
cd ..


tsdisrplot2(tree,sintpadre,sintfiglio,gruppi,sintesi2,decimp,0,percentili);

-------------------------------------------------------------------------------

function [XSLL,XSRR,varmaxG,colmaxG,spli,tab,tab2,Bbest,BLDA,ef1,
percmislda,lbaparcol,indbBest,decremento,U,psi,score,correl]=threestageLDA(X,gruppi);

lbapar=0;
numgruppi=length(gruppi);
[n,p]=size(X);
[X,b]=sortrows(X,1);

Y=X(:,1);

T=n*var(Y,1);
for i = 1:n-1
```

```
    B(i)=T-((length(Y(1:i))-1)*var(Y(1:i),1)+(length(Y(i+1:n))-1)*var(Y(i+1:n),1));
end
[Bbest,indbBest]=max(B);

% XS=X(b,:);
temp=[zeros(indbBest,1)+1; ones(n-indbBest,1)+1];
XS=[temp X];


n1=indbBest;n2=n-indbBest;


newmat=zeros(n,numgruppi);
k=2;
for i = 1:numgruppi
    [newmat(:,i),u]=lda([XS(:,1) XS(:,k+1:k+gruppi(i))]);
    k=k+gruppi(i);
    u=u';
    lbapar=[lbapar u];
end

newmat=[XS(:,2) newmat];

%%%% GLOBAL VARIABLE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[psi,U]=lda([XS(:,1) newmat(:,2:end)]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

newmat=[newmat psi];
%%%%%% Impurity and global variable split %%%%%%%%%%%%%%%%%%%%%%%
[s,ss]=sort(psi);
y=newmat(ss,1);
for k=1:n-1

    GG(k)=T-((length(y(1:k))-1)*var(y(1:k),1)+(length(y(k+1:n))-1)*var(y(k+1:n),1));
end
[varmaxG,colmaxG]=max(GG);
spli=colmaxG;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

psi=s;
score=s(spli);
correl=corr([y s]);
correl=correl(2);
XS=XS(ss,:);
XSLL=XS(1:spli,:);
XSRR=XS(spli+1:n,:);
```

```
tab(1,1)=length(XSLL(:,1));
tab(2,1)=length(XSRR(:,1));
tab(1,2)=mean(XSLL(:,2));
tab(2,2)=mean(XSRR(:,2));
tab(1,3)=std(XSLL(:,2));
tab(2,3)=std(XSRR(:,2));

tab2(1,1)=sum(XSLL(:,1)==1);
tab2(2,1)=sum(XSLL(:,1)==2);
tab2(1,2)=sum(XSRR(:,1)==1);
tab2(2,2)=sum(XSRR(:,1)==2);
BLDA=( mean(XSLL(:,1))^2*length(XSLL))+(mean(XSRR(:,1))^2*(length(XSRR)))-((mean(XS(:,2))^2)*n);
ef1=BLDA/Bbest;
percmislda=((tab2(1,2)+tab2(2,1))/sum(sum(tab2)))*100;


        devS=var(XSLL(:,2))*(length(XSLL(:,1))-1);
        devR=var(XSRR(:,2))*(length(XSRR(:,1))-1); % varianza interna nodi figli
        XS=[XSLL;XSRR]; % nodo padre
        devT=var(XS(:,2))*(length(XS(:,1))-1); % varianza nodo padre
        decremento=devT-(devS+devR); %decremento della varianza
        XSLL(:,1)=[];
        XSRR(:,1)=[];

        colmaxG=1;
```

A.2 Partial Predictability Trees

```
function [matriceX,sintpadre,sintfiglio,imp,tree,nodihandle,sintesi2]=
MCSc(X,tipoX,Xlabel,decmin,maxoss,numvar,tw,prun,Z);


nodoL=1;
memnodo=[0,1];
lung=length(memnodo);
sizeX=size(X);
nodo1(1:sizeX(1),1)=1;
matrice=[nodo1(:,1),X];


tabY=tabulate(X(:,1));
nclass=length(tabY(:,1));
percentili(1)=0;
for i=1:nclass
    percentili(i+1)=tabY(i,1);
end



it=0;it2=0;
tree.nodo(1).X=X;
tree.nodo(1).Z=Z;
[tree.nodo(1).crossZ tree.nodo(1).chi_pvalue tree.nodo(1).C]=
marginali(X(:,1), Z);

[tree.nodo(1).label,freqmoda]=moda(X(:,1));
tree.nodo(1).misclass=(sizeX(1)-freqmoda)/sizeX(1)*100;
tree.nodo(1).term=0;
tree.nodo(1).father=0;
tree.nodo(1).n=sizeX(1);
cont=0;
decremento=1;
imp.decimpurita=0;
imp.nodo=0;

while memnodo(lung) ~= 0        % routine to grow tree
    it=it+1;
    sizeX=size(X);
    while sizeX(1) >= maxoss % routine to split
        %nodoL
        tabX=tabulate(X(:,1));
        if max(tabX(:,3))==100
            decremento=0;
            tree.nodo(nodoL).decimp=decremento;
            tree.nodo(nodoL).term=1;
            break
        end
        [XSLL,XSRR,ZL,ZR,BestPred,Split,tab,tab2,decimpBest,percmis,Taos,TaoSplit,
        Tp,Ts]=MCScsplit(X,tipoX,tw,numvar,Z);
        decremento=decimpBest;

        if decremento <= decmin
            tree.nodo(nodoL).decimp=decremento;
            tree.nodo(nodoL).term=1;
            break
        end
        cont=cont+1;
% data for split node

        sintpadre.nodoL(cont)=nodoL;
        sintpadre.numnodo(cont)=length(X(:,1));
        sintpadre.col(cont)=BestPred;
        %sintpadre.split(cont)=Split;


        %%%%%%%%%%%%%%%%%%%%%%%
        if tipoX(BestPred)==0
            indiceL=find((matrice(:,BestPred+2)<=Split)& (matrice(:,1)==nodoL));
            matrice(indiceL,1)=nodoL*2;
            indiceR=find(matrice(:,1)==nodoL);
            matrice(indiceR,1)=nodoL*2+1;
        else
            II=length(Split(1,:));
            for ii=1:II
                indiceL=find((matrice(:,BestPred+2)==Split(ii))& (matrice(:,1)==nodoL));
                matrice(indiceL,1)=nodoL*2;
            end
            indiceR=find(matrice(:,1)==nodoL);
            matrice(indiceR,1)=nodoL*2+1;
        end
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        nodoL=nodoL*2;
        nodoD=nodoL+1;


        tree.nodo(nodoL).X=XSLL;
        tree.nodo(nodoL).Z=ZL;
        [tree.nodo(nodoL).crossZ tree.nodo(nodoL).chi_pvalue tree.nodo(nodoL).C]=
        marginali(XSLL(:,1), ZL);

        tree.nodo(nodoD).X=XSRR;
        [tree.nodo(nodoD).crossZ tree.nodo(nodoD).chi_pvalue tree.nodo(nodoD).C]=
        marginali(XSRR(:,1), ZR);

        tree.nodo(nodoD).Z=ZR;
        tree.nodo(nodoL/2).term=0;
        tree.nodo(nodoL/2).ordpred=Taos;
        tree.nodo(nodoL/2).Tp=Tp;
        tree.nodo(nodoL/2).Ts=Ts;
        tree.nodo(nodoL/2).decsplit=TaoSplit;

        tree.nodo(nodoL/2).numbsplit=nsplit(X,tipoX,Taos,BestPred);

        tree.nodo(nodoL).father=nodoL/2;
        tree.nodo(nodoD).father=nodoL/2;
        tree.nodo(nodoL/2).col=BestPred;
        tree.nodo(nodoL/2).TypeX=tipoX(BestPred);
        tree.nodo(nodoL/2).split=Split;
        tree.nodo(nodoL).label=tab(1,2);
        tree.nodo(nodoD).label=tab(2,2);
        tree.nodo(nodoL).impurita=tab(1,3);
        tree.nodo(nodoD).impurita=tab(2,3);
        tree.nodo(nodoL/2).decimp=decremento;
        tree.nodo(nodoL).misclass=percmis(1);
        tree.nodo(nodoD).misclass=percmis(2);
        tree.nodo(nodoL/2).tabmisclass=tab2;
        tree.nodo(nodoL).n=size(XSLL,1);
        tree.nodo(nodoD).n=size(XSRR,1);

% data child-nodes
        sintfiglio.destro.tab1(cont,:)=[nodoD tab(2,:)];
        sintfiglio.sinistro.tab1(cont,:)=[nodoL tab(1,:)];

        %matriceX=joinnode(XSRR,XSLL,matriceX,nodoL);
        memnodo=[memnodo,nodoD,nodoL];
        X=tree.nodo(nodoL).X;
        Z=tree.nodo(nodoL).Z;
        lung=length(memnodo);
        memnodo(lung)=[];
        sizeX=size(X);

% impurity
        it2=it2+1;
        imp.decimpurita(it2)=decremento;
        imp.nodo(it2)=(nodoL/2);

    end
    if sizeX(1) < maxoss
        tree.nodo(nodoL).term=1;
    end
    lung=length(memnodo);
    nodoL=memnodo(lung);

    if sizeX(1) < maxoss | decremento < decmin
        memnodo(lung)=[];
        lung=length(memnodo);
    end


    if nodoL > 1
       X=tree.nodo(nodoL).X;
       Z=tree.nodo(nodoL).Z;
    end

end
if tree.nodo(1).term==1
    noditot=0;
    noditot2=0;
    sintfiglio=0;
    sintpadre=0;
    sintesi=0;
    sintesi2=0;
    matriceX=0;
    decimp=0;
    nodihandle=0;
else
% compute goodness
    noditot=[sintfiglio.destro.tab1(:,1:2) ; sintfiglio.sinistro.tab1(:,1:2)];
    noditot2=[sintfiglio.destro.tab1 ; sintfiglio.sinistro.tab1];
    noterm=sintpadre.nodoL';
    noterm(1)=[];
    noterm;
    n=length(noterm);
    m=length(noditot(:,1));
    cont=0;

    for j=1:m
        term=1;
        for i=1:n
            if noterm(i) == noditot(j,1)
                term=0;
                i=n;
            end
        end
        if term==1
            cont=cont+1;
            sintesi(cont,1:2)=noditot(j,1:2);
            sintesi2(cont,1:4)=noditot2(j,:);
        end
    end

    sintesi=sortrows(sintesi,1);

    matriceX=matrice;

    decimp = goodnessc(matriceX,tree,sintesi);
    tree.nodo(1).tw=tw;
    %%% pruning %%%%%%%%%%%%%%%%%%%%%
    if prun==1
        load sottoalberi
        [Rt,Rtlearn,seq,y,treepruning]=pruning(tree,sintpadre,sintfiglio,
        sintesi2,Xtest,1);

        save sottoalberi Rt Rtlearn seq y treepruning Xtest
        decimp nclass percentili prun

[tree,nodihandle]=Fastcplot(treepruning(seq(1)).sintp,
treepruning(seq(1)).sintfiglio,
treepruning(seq(1)).sintesi2,tree,nclass,percentili,Xlabel,prun);

%% Print tree

        Fastcplot2(tree,treepruning(seq(1)).sintp,treepruning(seq(1)).sintfiglio,
        treepruning(seq(1)).sintesi2,decimp,Xlabel);
    else
        [tree,nodihandle]=Fastcplot(sintpadre,sintfiglio,sintesi2,tree,nclass,
        percentili,Xlabel,prun);
        Fastcplot2(tree,sintpadre,sintfiglio,sintesi2,decimp,Xlabel);
        %% Prin tree
    end

    metodo=1;
    cd results
    save albero nclass percentili metodo
    cd ..

end

--------------------------------------------------------------------------

function [XSLL,XSRR,ZL,ZR,BestPred,Split,tab,tab2,decimpBest,percmis,Taos,
TaoSplit,Tp,Ts]=MCScsplit(X,tipoX,tw,numvar,Z);

[n,p]=size(X);
```

```
%%Z instrumental variable

Y=X(:,1);
XS=X(:,2:p);
Xg=giust(XS,Z);
Ts=tao(Z,Y);  % simple Tao

tabY=tabulate(Y);
pp=0;
decimp=0;
k=0;
for i=1:p-1
    Xi=Xg(:,i);
    distXi=tabulate(Xi);

    if length(distXi(:,3))==1
        k=k+1;
        Tao(k,3)=1;
        Tao(k,2)=0;
        Tao(k,1)=i;
        pp=pp+1;
        if pp==p-1
            %disp('terminal node')
            XSLL=0;
            XSRR=0;
            ZL=0;
            ZR=0;
            decimpBest=0;
            Split=0;
            tab=0;
            tab2=0;
            percmis=0;
            BestPred=0;
            Taos=0;
            TaoSplit=0;
            return
        end
    else
    k=k+1;
    Tao(k,2)=tao(Xi,Y);
    Tao(k,1)=i;
    Tao(k,3)=1-Tao(k,2);
    end
end

Taos=sortrows(Tao,3);

%%%% compute partial tao
```

```
Tp=(Taos(:,2)-Ts)./(1-Ts);



%% Best split of best predictor
BestPred=Taos(1,1);

BestX=XS(:,BestPred);                       )
distBestX=tabulate(BestX);
% indice=find(distBestX(:,2)==0);
% distBestX(indice,:)=[];
YY=[Y XS Z];
[newY,ordine]=sortrows(YY,BestPred+1);
Zn=newY(:,end);
newY(:,end)=[];

if tipoX(BestPred)~=1
    for i=1:length(distBestX(:,2))-1


        nL=sum(distBestX(1:i,2));
        nR=n-nL;
        XF(1:nL,1)=1;
        XF(nL+1:n,1)=2;
        TaoSplit=tao2(XF,newY(:,1));
        decimp(i)=TaoSplit;
    end
    [decimpBest,Spli]=max(decimp);
    Split=distBestX(Spli,1);
    nL=sum(distBestX(1:Spli,2));
    nR=n-nL;
    %XF(1:nL,1)=1;
    %XF(nL+1:n,1)=2;
    XSLL=newY(1:nL,:);
    ZL=Zn(1:nL,:);
    XSRR=newY(nL+1:n,:);
    ZR=Zn(nL+1:n,:);
    %TaoSplit=tao2(XF,newY(:,1));
    TaoSplit=decimp(Spli);
else
    [decimpBest,XSLL,XSRR,combsplit,decimp,ZL,ZR]=MCScsplitnom(XS,Y,BestPred,Z);
    TaoSplit=decimpBest;
    Split=combsplit.L;
end

for p=2:length(Taos(:,1))          %% select a new predictor

    Taopred=Taos(p,2);
```

```
if TaoSplit >= Taopred

     break
end
if tw==1 & p==numvar+1
    %% tw=1 two-stage
    %% tw=0 fast
    decimpBest=(gini(Y)-((gini(XSLL(:,1))*length(XSLL(:,1))/n)+(
    gini(XSRR(:,1))*length(XSRR(:,1))/n)))/gini(Y);

    decimpBest=round(decimpBest*1000)/1000;
    tab(1,1)=length(XSLL(:,1));
    tab(2,1)=length(XSRR(:,1));
    [tab(1,2),freqL]=moda(XSLL(:,1));
    [tab(2,2),freqR]=moda(XSRR(:,1));
    tab(1,3)=gini(XSLL(:,1));
    tab(2,3)=gini(XSRR(:,1));

    tab2(1,1)=freqL;
    tab2(2,1)=tab(1,1)-freqL;
    tab2(1,2)=tab(2,1)-freqR;
    tab2(2,2)=freqR;

    percmis(1)=(tab(1,1)-freqL)/tab(1,1)*100;
    percmis(2)=(tab(2,1)-freqR)/tab(2,1)*100;
    return
end

BestPred2=Taos(p,1);
BestX=XS(:,BestPred2);
distBestX2=tabulate(BestX);



YY=[Y XS Z];
[newY,ordine]=sortrows(YY,BestPred2+1);
Zn=newY(:,end);
newY(:,end)=[];




if tipoX(BestPred2)~=1     % (0=ordinal or numerical 1=nominal)

    for i=1:length(distBestX2(:,2))-1

        nL=sum(distBestX2(1:i,2));
        nR=n-nL;
```

```
            XF(1:nL,1)=1;
            XF(nL+1:n,1)=2;
            TaoSplit=tao2(XF,newY(:,1));                 % local Tao
            decimp(i)=TaoSplit;                         % impurity
        end
        [decimpBest2,Spli]=max(decimp);
    else
        [decimpBest2,XSL,XSR,combsplit,ZL,ZR]=MCScsplitnom(XS,Y,BestPred2,Z);
    end

    if decimpBest < decimpBest2
        decimpBest=decimpBest2;
        BestPred=BestPred2;
        distBestX=distBestX2;
        if tipoX(BestPred)~=1
            Split=distBestX(Spli,1);
            nL=sum(distBestX(1:Spli,2));
            nR=n-nL;
            XF(1:nL,1)=1;
            XF(nL+1:n,1)=2;
            XSLL=newY(1:nL,:);
            ZL=Zn(1:nL,:);
            ZR=Z(nnL+1:n,:);
            XSRR=newY(nL+1:n,:);
            TaoSplit=tao2(XF,newY(:,1));
        else
            Split=combsplit.L;
            XSLL=XSL;
            XSRR=XSR;
            TaoSplit=decimpBest2;
        end
    end
end

decimpBest=(gini(Y)-((gini(XSLL(:,1))*length(XSLL(:,1))/n)+(gini(XSRR(:,1))*
*length(XSRR(:,1))/n)))/gini(Y);;
decimpBest=round(decimpBest*1000)/1000;
tab(1,1)=length(XSLL(:,1));
tab(2,1)=length(XSRR(:,1));
[tab(1,2),freqL]=moda(XSLL(:,1));
[tab(2,2),freqR]=moda(XSRR(:,1));
tab(1,3)=gini(XSLL(:,1));
tab(2,3)=gini(XSRR(:,1));

tab2(1,1)=freqL;
tab2(2,1)=tab(1,1)-freqL;
tab2(1,2)=tab(2,1)-freqR;
tab2(2,2)=freqR;
```

```
if tab(1,1)>0
    percmis(1)=(tab(1,1)-freqL)/tab(1,1)*100;
else
    percmis(1)=0;
end
if tab(2,1)>0
    percmis(2)=(tab(2,1)-freqR)/tab(2,1)*100;
else
    percmis(2)=0;
end

---------------------------------------------------------------------

function [decimpBest,XSLL,XSRR,combsplit,decimp,ZL,ZR]=
MCScsplitnom(XS,Y,BestPred,Z);

Xi=XS(:,BestPred);
tabXi=tabulate(Xi);
indice=find(tabXi(:,2)==0);
tabXi(indice,:)=[];
XX=[1:length(tabXi)];
[comb,numcomb]=splitcomb(tabXi(:,1));
n=length(Xi(:,1));

for i=1:numcomb
    indice=[];
    for j=1:length(comb.split(i).L)
        combsplit=tabXi(comb.split(i).L',1);
        indiceL=find(Xi(:,1)==combsplit(j));
        indice=[indice; indiceL];
    end
    XSLL=[Y XS];
    XSLL=XSLL(indice,:);
    XSRR=[Y XS];
    XSRR(indice,:)=[];
    XF(1:n,1)=2;
    XF(indice,1)=1;
    decimp(i)=tao(XF,Y);
    splitnum.split(i).XSLL=XSLL;
    splitnum.split(i).XSRR=XSRR;
    splitnum.split(i).L=tabXi(comb.split(i).L',1);
    splitnum.split(i).R=tabXi(comb.split(i).R',1);
    splitnum.split(i).indice=indice;
    XF=[];
    XSLL=[];
    XSRR=[];
end
[decimpBest,i]=max(decimp);
```

```
indice=splitnum.split(i).indice;
ZL=Z(indice,:);
ZR=Z;
ZR(indice,:)=[];
XSLL=splitnum.split(i).XSLL;
XSRR=splitnum.split(i).XSRR;
combsplit.L=splitnum.split(i).L';
combsplit.R=splitnum.split(i).R';
```

Bibliography

[1] Agrawal, R., Imielinski, T., Swami, A. (1993) Mining Associations between Sets of Items in Massive Databases, *Proc. of the ACM-SIGMOD 1993 Int'l Conference on Management of Data*, Washington D.C., 207-216.

[2] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A.I. (1995). Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, Chapter 12, pages 307–328, AAAI/MIT Press, Menlo Park, CA.

[3] Agrawal, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules, *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, Sept. 1994. Expanded version available as IBM Research Report RJ9839.

[4] Agresti (1990). *Categorical Data Ananlysis.* J. Wiley.

[5] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19, 716-723.

[6] Aluja-Banet, T., Nafria, E. (1998). Robust impurity measure in decision trees. In *Proceedings of the V IFCS Conference, Data Science, Classificationa and Related Methods*, pp 207-214, Physica Verlag.

[7] Aria, M. (2005). Multi-Class Budget Exploratory Trees. In *Studies in Classification, Data Analysis, and Knowledge Organization: New Developments in Classification and Data Analysis*, a cura di M. Vichi, P. Monari, S. Mignani, A. Montanari, ed. Springer-Verlag, pp. 3-8.

[8] Aria, M., Mooijaart, A., Siciliano, R., (2003). Neural Budget Networks of Sensorial Data, in M. Schader et al.: *Studies in Classification, Data Analysis, and Knowledge Organization, Between Data Science and Applied Data Analysis*, Springer-Verlag, XIII, 369-377.

[9] Aria, M., Siciliano, R. (2003). Learning from Trees: Two-Stage Enhancements. *In Proceedings of Classification and Data Analysis Group (CLADAG 2003)*, 22-24 Settembre, Bologna.

[10] Aria, M., Mola, F., Siciliano R. (2002). Growing and Visualizing Prediction Paths Trees in Market Basket Analysis, Wolfgang Härdle, Bernd Rönz (Eds.), *Proceedings in Computational Statistics: 15th Symposium Held in Berlin, Germany 2002 (COMPSTAT2002)*, pp.123-128, Physica-Verlag, Heidelberg.

[11] Benzecri, J.P., (1973). *L'Analyse des Données*, 2 Vols. Dunod, Paris, France.

[12] Bishop, C.M., (1995). *Neural network for Pattern Recognition*, Claredon Press, Oxford.

[13] Bolasco, S. (1997). *Analisi Multidimensionale dei Dati, Metodi, Strategie e Criteri di Interpretazione*. Carocci.

[14] Breiman, L. (1996). Bagging Predictors, *Machine Learning*, 26, 46-59.

[15] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984). *Classification and Regression Trees.* Wadsworth International Group, Belmont, California.

[16] Brin, S., Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Proceedings of the Seventh International World-Wide Web Conference*, Brisbane, Australia, pp. 107-117.

[17] Buja, A., Lee, Y.S. (1999). A data mining criteria for tree based regression and classification. Technical report, At & T Labs. www.research.att/andreas/papers/tree.ps.gz.

[18] Buntine, W.L., Niblett, T. 1992). *A Further Comparison of Splitting Rules for Decision-Tree Induction.* Machine Learning 8: 75-85.

[19] Cappelli, C., Mola, F., Siciliano, R. (2002). A statistical approach to growing an honest reliable tree. Computational Statistics and Data Analysis, 38, 285-299, Elsevier Science.

[20] Cappelli, C., Mola, F., Siciliano, R. (2000). Selecting Regression Tree Models: A Statistical Testing Procedure, in S. Borra, R. Rocci, M. Vichi, M. Schader (Eds.): *Advances in Classification and Data Analysis*, Berlin (D), Springer-Verlag, 249-256.

[21] Cappelli, C., Mola, F., Siciliano, R. (2000). A third stage in regression growing: searching for statistical reliability. In *Proceedings of the VII IFCS Conference, Data Science, Classification and Related Methods*, pp. 193-198, Springer Verlag.

[22] Cappelli, C., Mola, F., Siciliano, R. (1998). An alternative pruning procedure based on the impurity-complexity measure, in R. Payne, P. Green (Eds.): *Proceedings in Computational Statistics:*

13th Symposium of COMPSTAT (Bristol, August 24-28, 1998, Physica Verlag, Heidelberg (D), 221-226.

[23] Cestnik, G., Bratko, I., (1991). On estimating probabilities in tree pruning. *In Proceedings of the European Working Session on Learning*, Springer-Verlag, Berlin, pp. 138-150.

[24] Cherkassky V., Mulier F. (1998). *Learning from Data: concepts, theory, and methods.* John Wiley & Sons., New York, USA.

[25] Ciampi, A. (1994). *Classification and discrimination: the RECPAM approach*, COMPSTAT'94, Dutter R. and Grossmann W. eds, Phisica-Verlag, Heidelberg, 129-147.

[26] CISIA - CERESIA (2001), SPAD version 5.0, Manuel de Prise en Main, CISIA-CESTA, Montreuil, France.

[27] Clogg, C.C., Shihadeh E.S. (1994). *Statistical Models for Ordinal Variables*, Thousand Oaks, CA.: Sage Publications.

[28] Conversano, C., Mola, F., Siciliano, R. (2001). Partitioning and Combined Model Integration for Data Mining, presented at the Symposium on Data Mining and Statistics (Augsburg, November 2000), *Journal of Computational Statistics*, 16, 323-339, Physica Verlag, Heidelberg (D).

[29] Conversano, C., Mola, F., Siciliano, R. (2000). Generalized Additive Multi-Model for Classification and Prediction, in H.A.L. Kiers, J.P. Rasson, P.J.F. Groen, M. Shader (Eds.): *Data Analysis, Classification and Related Methods*, Springer Verlag, Berlin (D), 205-210.

[30] Conversano, C., Siciliano, R., Mola, F., (2000). Supervised Classifier Combination through Generalized Additive Multi-Model, in

F. Roli, J. Kittler (Eds.): *Proceedings of the First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, Physica Verlag, Heidelberg (D), 167-176.

[31] Cover, T., Thomas, J. (1991). *Elements of Information Theory.* Wiley, New York.

[32] De Leeuw, J., Young, F.W., Takane, Y. (1976). Additive structure in qualitative data: an alternating least square method with optimal scaling features. *Psychometrika.* Volume 31, pp. 33-42

[33] Duda, R., Hart, P., Stork, D. (2000). *Pattern Classification (Second Edition).* Wiley, New York.

[34] Efron, B., (1979). Bootstrap methods: Another look at the Jackknife, *Annals of Statistics*, 7, pp. 1-26.

[35] Efron, B., Tibshirani, R.J. (1993). *An Introduction to the Bootstrap.* Monographs on Statistics and Applied Probability 57. London: Chapman and Hall.

[36] Efron, B., Tibshirani, R.J. (1993). Statistical analysis in the computer age, *Science*, 253: 390-395.

[37] Esposito, F., Malerba, D., Semeraro, G. (1998). The effects of pruning methods on predictive accuracy of induced decision trees: A new experimentation with cross-validation. In *Proceedings VIII International Symposium on Applied Stochastic Models and Data Analysis*, pp 129-134, Rocco Curto.

[38] Fabbris, L. (1997). *Statistica Multivariata.* McGraw-Hill.

[39] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. (1996). From data mining to knowledge discovery: an overvirew. In *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G.

Piatetsky-Shapiro, P.Smyth, and R. Uthurusamy (eds.). Menlo Park, CA: AAAI Press. pp. 1-34.

[40] Freund, Y., Schapire, R.E. (1996). Experiments with a new boosting algorithm, *13th International Conference on Machine Learning.*

[41] Friedman, J.H.F., (1994). An overview of predictive learning and function approximation, in V.Cherkassy, J.Friedman, H.Wechsler (eds), *From Statistics to Neural Networks*, Vol.136 of NATO ISI Series F, Springer Verlag, New York.

[42] Friedman, J.H.F., Hastie, T., Tisbshirani, R. (2000). Additive logistic regression: a statistical view of boosting, *Annal of Statistics*, 28, 377-386.

[43] Gelfand, S.B., Ravishankar, C.S., Delp, E.J. (1991). *An Iterative Growing and Pruning Algorithm for Classification Tree Design.* IEEE Trans. Pattern Anal. Mach. Intell. 13(2): 163-174.

[44] Gelman, A., Carlin, J., Stern, H., Rubin, D. (1995). *Bayesian Data Analysis*, CRC Press, Boca Raton, FL.

[45] Goodman, L.A. (1985). The analysis of cross-classified data having ordered and/or unordered categories: Association models, correlation models, and asymmetry models for contigency tables with or without missing entries. *Annals of Statistics*, 13, 10-69.

[46] Goodman, L.A., Kruskal, W.H. (1954). Measures of association for cross-classification. *Journal of American Statistical Association*, 48, 732-762.

[47] Goodman, L.A., Kruskal, W.H. (1979). *Measures of association for cross classifications.* Springer.

[48] Gordon, A. (1999). *Classification (Second Edition)*, Chapman and Hall/CRC Press, London.

[49] Gray, L.N., Williams, J.S. (1975). Goodman and Kruskal's tau b: multiple and partial analogs. *Proceedings of the Americal Statistical Association*, 444-448.

[50] Hand, D. (1996). Classification and computers: Shifting the focus. In *Compstat96 Proceedings in Computational Statistics*, pp 77-88. Physica Verlag.

[51] Hand, D., (1998). Data Mining,: Statistics or more?. *Am.Statist.*, 52, 112-118.

[52] Hand.D., Mannila H., Smyth P. (2001). *Principles of Data Mining.* A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.

[53] Hastie, T.J., Tibshirani, R.J., Friedman, J.H. (2001). *The Elements of Statistical Learning.* Springer Verlag.

[54] Hofmann, H., Siebes, A., Wilhelm A. (2000). Visualising Association Rules. In *KDD-2000 - Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 227-235, ACM, New York.

[55] Hotelling, H. (1936). The relation between two sets of variates, *Biometrika*, 28, pp. 321-377.

[56] Houtsma M.,Swami, A. (1993). *Set-oriented mining of association rules.* Research Report RJ 9567, IBM Almaden Research Center, San Jose, CA.

[57] Jobson, J.D.,(1992). *Applied Multivariate Data Analysis Volume I: Regression and Experimental Design.* Springer-Verlag, New York.

[58] Jobson, J.D.,(1992). *Applied Multivariate Data Analysis Volume II: Categorical and Multivariate Methods.* Springer-Verlag, New York.

[59] Kearns, M., Vazirani, U. (1994). *An Introduction to Computational Learning Theory*, MIT Press.

[60] Kim, H., Loh, W.Y. (2001). Classification Trees with Unbiased Multiway Splits, *Journal of the American Statistical Association*, 96, 454, 589-604.

[61] Klaschka, J., Siciliano, R., Antoch, J., 1998. *Computational enhancements in tree-growing methods.* In: Rizzi, A., Vichi, M., Bock, H.H. (Eds.). *Advances in Data Science and Classification.* Physica Verlag, Heidelberg, pp. 295-302.

[62] Kruskal, J.B., (1984). Multilinear Methods. In Law, H.G. et al. (eds), *Research Methods for Multimode Data Analysis*, Praeger, New York, USA, pp 36-62.

[63] Kruskal, J.B., Shepard, R.N. (1972). A nonmetric variety of linear factor analysis. *Psychometrika*, volume 32 number 2, pp. 123-157.

[64] Lanubile A., Malerba D. (1998) Induction of Regression Trees with Regtree, *Classification and Data Analysis: Book of Short Papers*, 253-256, Meeting of the Italian Group of Classification, Pescara.

[65] Lauro, N.C., Siciliano, R. (1989). Exploratory methods and modelling for contigency tables analisys: an integrated approach. *Statistica Applicata*, 1.

[66] Lebart, L., Morineau, A., Fenelon, J.P. (1979). *Traitement des données statistiques (Méthodes et Programmes*, Dunod.

[67] Lebart, L., Morineau, A., Piron, M. (1995) *Statistique Exploratoire Multidimensionnelle.* Dunod.

[68] Light, R.J., Margolin, B.H. (1971). *An analysis of variance for categorical data.* J. Amer. Statist. Association, pp. 534-544

[69] Loh, W., Vanichsetakul, N. (1988). Tree-Structured Classification via Generalized Discriminant Analysis. *Journal of the American Statistical Association*, 83, 715-728.

[70] Lubinsky, D. J. (1995). Increasing the performance and consistency of classification trees by using the accuracy criterion at the leaves. *In Proceedings of the Twelth International Conference on Machine Learning*, 371-377 Taho City, Ca. Morgan Kaufmann.

[71] Marchand P., Holland O.T., (2003). *Graphics and GUIs with MatLab. Third Edition.* Chapman & Hall CRC, N.Y., USA.

[72] Martinez W.L., Martinez, A.R., (2002). *Computational Statistics Handbook with MatLab.* Chapman & Hall/CRC, Boca Raton, Florida.

[73] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). *Equations of state calculations by fast computing machines.* J Chem Phys, 21:1087-1091.

[74] Mingers J.(1989) *An Empirical Comparison of Pruning Methods for Decision Tree Induction*, Machine Learning, 4, 227-243.

[75] Mola, F. (1993). *Aspetti metodologici e computazionali delle tecniche di segmentazione binaria: Un contributo basato su una*

funzione di predizione. Tesi di Dottorato in Statistica Computazionale, Napoli.

[76] Mola, F., Siciliano, R. (1998). A general splitting criterion for classification trees, *Metron*, 56, 3-4.

[77] Mola, F., Siciliano, R. (1997). A Fast Splitting Procedure for Classification and Regression Trees, *Statistics and Computing*, 7, Chapman Hall, 208-216.

[78] Mola, F., Siciliano, R. (1994). Alternative strategies and CATANOVA testing in two-stage binary segmentation, in E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, B. Burtschy (Eds.): *New Approaches in Classification and Data Analysis: Proceedings of IFCS 93*, Springer Verlag, Heidelberg (D), 316-323.

[79] Mola, F., Siciliano, R. (1992). A two-stage predictive splitting algorithm in binary segmentation, in Y. Dodge, J. Whittaker. (Eds.): *Computational Statistics: COMPSTAT 92*, 1, Physica Verlag, Heidelberg (D), 179-184.

[80] Mola, F., Siciliano, R. (2002). Discriminant Analysis and Factorial Multiple Splits in Recursive Partitioning for Data Mining, in Roli, F., Kittler, J. (eds.): *Proceedings of International Conference on Multiple Classifier Systems* (Chia, June 24-26, 2002), 118-126, Lecture Notes in Computer Science, Springer, Heidelberg.

[81] Morgan J. N., Sonquist J. A., Problem in the Analysis of Survey Data, and a Proposal, *Journal of American Statistical Association*, 58, (1963), 415-434.

[82] Murphy, P. M. and Aha, D. W. (1993). UCI repository of machine learning databases. *Machine-readable data repository.* University

of California, Department of Information and Computer Science, Irvine, CA.

[83] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J. (1998). *UCI Repository of machine learning databases* [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[84] Piatetsky-Shapiro, G. (1999) The data-mining industry coming of age. *IEEE Expert*, 14(6), pp. 32-34.

[85] Quinlan, J.R. (1987). *Simplifying decision tree.* Internat. J. Man.Mach. Studies 27, 221-234.

[86] Quinlan, J. R. (1993). *C4.5: Programs For Machine Learning.* Morgan Kaufmann, Los Altos.

[87] Rao, C.R., (1980). Matrix Approximations and reduction of dimensionality in multivariate statistical analysis, P.R.Krishnaiah (Ed.), *Multivariate Analysis*, pp. 3-22, North Holland, Amsterdam.

[88] Rao, C.R., (1979). Separation theorems for singular values of matrices and their applications in multivariate analysis, *Journal of Multivariate Analysis*, 9, pp. 362-377.

[89] Rizzi, A. (1985). *Analisi dei Dati.* La Nuova Italia Scientifica.

[90] Shannon W.D., Banks, D. (1999). Combining classification trees using mle. *Statistical in Medicine*, 18:727-740.

[91] Siciliano, R. (1999). Latent budget trees for multiple classification, in M. Vichi, P. Optitz (Eds.): *Classification and Data Analysis: Theory and Application*, Springer Verlag, Heidelberg (D).

[92] Siciliano, R. (1998). Exploratory versus Decision Trees, invited lecture to COMPSTAT '98 (Bristol, August 24-28), in R. Payne, P. Green (Eds.): *Proceedings in Computational statistics: 13th Symposium of COMPSTAT*, Physica Verlag, Heidelberg (D).

[93] Siciliano, R., Aria, M., Conversano, C. (2004). Harvesting trees: methods, software and applications. In Proceedings in Computational Statistics: 16th Symposium of IASC, held Prague, August 23-27. 2004 (COMPSTAT2004), Eletronical Edition (CD) Physica-Verlag, Heidelberg.

[94] Siciliano, R., Conversano C. (2002). Tree-based Classifiers for Conditional Missing Data Incremental Imputation, *Proceedings of the International Conference on Data Clean* (Jyväskylä, May 29-31, 2002), University of Jyväskylä.

[95] Siciliano, R., Aria, M., Conversano, C. (2004). Harvesting trees: methods, software and applications. In *Proceedings in Computational Statistics: 16th Symposium of IASC*, held Prague, August 23-27. 2004 (COMPSTAT2004), Eletronical Edition (CD) Physica-Verlag, Heidelberg.

[96] Siciliano, R., Conversano, C., (2005). Decision Tree Induction, in Wang J. (eds.), *Encyclopedia of Data Warehousing and Data Mining*, IDEA Group. Inc., Hershey, USA, volume 2, 242-248.

[97] Siciliano, R., Mola F. (1996) A Fast Regression Tree Procedure, in: *Statistical Modelling: Proceedings of the 11th International Workshop on Statistical Modelling*, A. Forcina et al. eds, 332-340, Perugia: Graphos.

[98] Siciliano, R., Mola, F. (2000). Multivariate Data Analysis through Classification and Regression Trees, *Computational Statistics and Data Analysis*, 32, 285-301, Elsevier Science.

[99] Siciliano, R., Mola, F. (2002). Discriminant Analysis and Factorial Multiple Splits in Recursive Partitioning for Data Mining, in Roli, F., Kittler, J. (eds.): *Proceedings of International Conference on Multiple Classifier Systems* (Chia, June 24-26, 2002), 118-126, Lecture Notes in Computer Science, Springer, Heidelberg.

[100] Siciliano R., Tutore V.A., Aria M. (2007), 3Way Trees. *Classification and Data Analisys 2007*, Book of short papers (Macerata, September 12-14, 2007), EUM Macerata, pp 231-234.

[101] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions, *Journal of the Rojal Statistical Society*, Series B, Vol. 36, pp. 111-133.

[102] Takeuchi, K., Yanai, H., Mukherjee, B. (1982). *The Foundations of Multivariate Analysis*, Wiley Eastern, New Dehli.

[103] Taylor, P.C., Silverman, B.W. (1993). Block Diagrams and Splitting Criteria for Classification Trees. *Statitics and Computing*, 3, 147-161.

[104] Thisted, R.A., (1988). *Elements of Statistical Computing: Numerical Computation.* London, Chapman and Hall.

[105] Tutore, V.A., Siciliano, R., Aria, M. (2006). Three Way Segmentation in *Proceedings of Knowledge Extraction and Modelling (KNEMO06)* IASC INTERFACE IFCS Workshop, Capri, September 4th-6th 2006.

[106] Tutore V.A., Siciliano R., Aria M., (2007), Conditional Classification Trees using Instrumental Variables. *Advances in Intelligent Data Analysis*, Springer-Verlag, pp 163-173.

[107] Urbanek, S. (2002). Different ways to see a tree - KLIMT, in *Proc. of the 14th Conference on Computational Statistics*, (Compstat 2002), p303-308, Physica, Heidelberg.

[108] Van de Burg, E. (1988). *Nonlinear Canonical Correlation and some related techniques*, Leiden: DSWO Press.

[109] Vapnik, V. (1998). *Statistical Learning Theory.* Chichester, John Wiley & Sons, United Kingdom.

[110] Zani, S. (2000) *Analisi dei dati statistici, vol. II, Osservazioni multidimensionali*, Giuffré ed., Milano.

[111] Zani, S. (1998) *Analisi dei dati statistici, vol. I, Osservazioni in una e due dimensioni*, Giuffré ed., Milano.

[112] Zhang, H.P, Singer, B. (1999). *Recursive Partitioning in the Health Sciences.* Springer-Verlag.