Università degli Studi di Napoli
Federico II

Unobserved Heterogeneity in Structural Equation Models: a new approach to latent class detection in PLS Path Modeling

Laura Trinchera

Doctoral Thesis
in Statistics

XX Ciclo

Unobserved Heterogeneity in SEM: a new approach to latent class detection in PLS Path Modeling

Napoli, 30 november 2007

a Leo

«Life has a strange way of sneaking up on you »
I will never forgot you

Acknowledgments

LIFE HAS A STRANGE WAY OF SNEAKING UP ON YOU, *mai come negli ultimi mesi questa frase mi è apparsa vera in tutta la sua complessità. Ma per fortuna la vita sa veramente sorprenderti anche facendoti scoprire risorse che altrimenti non avresti mai pensato di avere.*

E così, in un periodo della mia vita personale che non è sicuramente dei migliori, sono arrivata alla fine di quest'esperienza. Tre anni vissuti intensamente e che mi hanno dato la possibilità di crescere e di imparare molte cose. E allora eccoci ai ringraziamenti.
Per prima cosa vorrei ringraziare la persona senza la quale oggi sicuramente non sarei (o almeno non proverei ad essere!!) una statistica. Il prof. Carlo Lauro che con il suo corso di Analisi Multivariata ogni anno conquista le menti (ed i cuori) di giovani studenti. I suoi insegnamenti umani e scientifici hanno contribuito a formare numerose persone di indubbio spessore.
Tra questi ci tengo a ringraziare il Prof. Vincenzo Esposito Vinzi per

essere stato il mio punto di rifermento in questi anni; è stato un piacere ed un onore lavorare con lui. Lui e la sua famiglia sono e spero resteranno un punto di riferimento non solo accademico ma anche morale.
I miei ringraziamenti vanno inoltre al Prof. Francesco Palumbo, che piu volte mi ha sopportato ed incoraggiato, alla dottoressa Rosaria Romano che, ben al di là dell'essere semplicemente la miglior collega con la quale lavorare, si è dimostrata essere un'amica presente e sincera, e al dottor Michelangelo Misuraca, che è stato il mio Virgilio (per non dire la mia Beatrice!!) in questi anni.
Vorrei inoltre ringraziare i miei colleghi dottorandi (passati e presenti) con i quali ho condiviso puri momenti di "follia scientifica", gli studenti della facoltà di Economia che con le loro domande mi hanno fatto capire che le cose piu semplici spesso sono anche quelle piu complesse, e tutte le persone del Dipartimento di Matematica e Statistica che in questi anni con il loro esempio, la loro presenza e il loro entusiasmo mi hanno fatto capire che vuol dire fare ricerca nell'università. I miei ringraziamenti vanno anche ai miei collaboratori di questi anni, e alle persone che con i loro suggerimenti hanno fatto si che il mio lavoro acquistasse uno spessore maggiore. In particolare, i miei ringraziamenti vanno al Prof. Michel Tenenhaus.

Beh, inutile dire che se in questi anni ho potuto lavorare e studiare serenamente gran parte è legato alla mia famiglia ed ai miei amici. E per questo che ne approfitto per ringraziare tutte le persone che in

questi anni mi sono state vicino. Tra di loro in particolare i miei zii, i miei cugini Leo, Anto, Vivi e Robi, mio fratello Lorenzo, ed i miei genitori che in questi mesi (molti dei quali passati fuori) non mi hanno mai fatta sentire sola.

Questa tesi è dedicata ad una persona che purtroppo non c'è più e che mi manca immensamente. Non c'è giorno che non pensi a lui. Sarebbe sicuramente stato orgoglioso di me. Come io lo ero di lui.
Cià dotto'....butta un occhio da lassu!

Laura

Contents

List of Figures

Notation and Abbreviations

Notations

Explanation of the main Symbols 1/2	
N	number of units or observations
i	generic unit or observation i, with $i = 1, \ldots, N$
M	number of exogenous latent variables
J	number of endogenous latent variables
Q	total number of latent variables (endogenous and exogenous ones), with $Q = M + J$
P	total number of manifest variables
P_q	number of manifest variables in the q-th block, with $\sum_{q=1}^{Q} P_q = P$
$\boldsymbol{\xi_q}$	generic latent variable
x_{pq}	generic manifest variable in the q-th block
$\boldsymbol{B}$	path coefficient matrix; the generic element is β_{mj}, i.e. the path coefficient linking the m-th exogenous latent variables to the j-th endogenous latent variable
$\boldsymbol{\Lambda}$	external loading matrix; the generic element is λ_{pq}, i.e. the loadings associated to the generic manifest variable $\boldsymbol{x}_{pq}$
$\boldsymbol{W}$	external weight matrix; the generic element is w_{pq}, i.e. the external weight associated to the p-th manifest variable in the q-th block
$\boldsymbol{E}$	matrix containing the errors $\boldsymbol{\epsilon}_{pq}$ associated to the generic manifest variable $\boldsymbol{x}_{pq}$ in a reflective measurement model
$\boldsymbol{\Delta}$	matrix containing the errors $\boldsymbol{\delta}_{pq}$ associated to the generic manifest variable $\boldsymbol{x}_{pq}$ in a formative measurement model
$\boldsymbol{H}$	matrix containing the errors $\boldsymbol{\zeta}_j$ associated to the j-th endogenous latent variable in the structural model
$\boldsymbol{\Phi}$	covariance matrix of the exogenous latent variables
$\boldsymbol{\Psi}$	covariance matrix of the inner residuals in the structural model
$\boldsymbol{\Theta}$	covariance matrix of the external residuals in a reflective measurement model
$\boldsymbol{\Omega}$	matrix containing all model parameters, i.e. $\boldsymbol{\Omega} = \{\boldsymbol{\Lambda}, \boldsymbol{B}, \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Theta}\}$
$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Omega}}\right)$	implied covariance matrix of the manifest variables
$\boldsymbol{S}$	sample covariance matrix of the manifest variables
$\boldsymbol{\Sigma}$	population covariance matrix of the manifest variables
K	number of latent classes
k	generic latent class
n_k	number of units or observations in the k-th latent class
β_{mjk}	path coefficient linking the m-th exogenous latent variables to the j-th endogenous latent variable in the k-th latent class
$\boldsymbol{Z}$	partition matrix of dimension $N \times K$
z_{ik}	generic element of $\boldsymbol{Z}$ matrix, i.e. categorical variable define the membership of the i-th unit to the k-th class
π_k	mixing proportion in Mixture Models, i.e. class size
ρ_{ik}	posterior probability for unit i to belong to the k-th latent class

Explanation of the main Symbols 2/2	
a_k	number of extracted component in a PLS Regression for the k-th class
$\boldsymbol{T_k}$	matrix containing the component scores for the PLS Regression model for the k-th class
y_j	generic endogenous variable in a PLS Regression, with $j = 1 \dots J$
r_{ijk}	residual for the i-th unit in the k-th latent class
	corresponding to the j-th endogenous variable in a PLS Regression model
$v_{ip_{j*}k}$	residual of the redundancy model for the i-th unit
	in the k-th latent class, corresponding to the $j*$-th target block in a PLS-PTM
e_{ipqk}	measurement residual for the i-th observation in the k-th latent class,
	corresponding to the p-th manifest variable in the q-th block,
	i.e. the communality residuals in REBUS-PLS
f_{ijk}	structural residual for the i-th observation in the k-th latent class,
	corresponding to the j-th endogenous block in REBUS-PLS

Abbreviations

- MLR: *Multiple Linear Regression*
- LS: *Least Squares*
- OLS: *Ordinary Least Squares*
- PLS: *Partial Least Squares*
- ALS: *Alternating Least Squares*
- GLS: *Generalized Least Squares*
- ULS: *Unweighted Least Squares*
- ADF: *Asymptotically Distribution Free*
- RMR: *Root Mean Residual*
- ML: *Maximum Likelihood*
- EM: *Expectation - Maximization*
- ECM: *Excpectation-Conditional Maximization*
- ACEM: *Alternative Excpectation-Conditional Maximization*
- LRT: *Likelihood Ratio Test*
- FCM: *Fuzzy C-Means*

- FCL: *Fuzzy C-Lines*
- FCV: *Fuzzy C-Varieties*
- FCR: *Fuzzy Clusterwise Regression*
- GoM: *fuzzy Grade Of Membership model*
- INDCLUS: *INDividual Difference CLUSter analysis*
- GENNCLUS: *GENeral Nonhierarcichal CLUStering analysis*
- CONCLUS: *CONstrained CLUster analysis*
- NN: *Neural Network*
- AID: *Automatic Interaction Detection*
- MAID: *Multivariate AID*
- CHAID: *CHi-squared Automatic Interaction Detection*
- CART: *Classification And Regression Trees*
- LCA: *Latent Class Analysis*
- AIC: *Akaike's Information Criteria*
- MAIC: *Modified Akaike's Information Criteria*
- CAIC: *Consistent Akaike's Information Criterion*
- BIC: *Bayesian Information Criterion*

- ICOMP: *Informational COMPlexity criterion*
- EN: *Entropy Index*
- NEC: *Normed Entropy Criterion*
- SEM: *Structural Equation Models*
- LV: *Latent Variable*
- MV: *Manifest Variable*
- SEM-ML: *Maximum Likelihood Approach to SEM*
- SEM-PLS: *Structural Equation Models with Partial Least Squares*
- LISREL: *LInear Structural RELations*
- PLS-PM: *Partial Least Squares Path Modeling*
- GSCA: *General Structured Component Analysis*
- GME: *Generalized Maximum Entropy*
- FIMIX-PLS: *FInite-MIXture PLS*
- FCGSCA: *Fuzzy Clusterwise GSCA*
- PLS-TR: *PLS Typological Regression*
- PLS-TPM: *PLS Typological Path Modeling*
- REBUS-PLS: *REsponse Based Unit Segmentation in PLS-PM*

- PLS-R: *PLS Regression*
- GoF: *Goodness of Fit index*
- AGFI: *Adjusted Goodness of Fit Index*
- NFI: *Normed Fit Index*
- NNFI: *Non-Normed Fit Index*
- TLI: *Tucker-Lewis Index*
- IFI: *Incremental Fit Index*
- BFI: *Bentler Fit Index*
- RNI: *Relative Noncentrality Index*
- CFI: *Bentler Comparative Fit Index*
- RMSEA: *Root Mean Square Error of Approximation*
- FIT: *FIT index*
- AFIT: *Adjusted FIT index*
- PLS-PMC: *PLS-PM based Clustering*
- PLS-GAS: *PLS Genetic Algorithm Segmentation*
- ECVI: *Expected Cross Validation Index*
- GFI: *Global Fit Index*

- SRMR: *Standardized Root Mean square Residuals*
- CM: *Closeness Measure*
- GQI: *Grouping Quality Index*

Chapter 1

Introduction

Heterogeneity among units is an important issue in statistical analysis. In statistical methods, treating the sample as homogeneous, when it is not, may seriously affect the results.
Since human behaviors are complex, looking at groups or classes of units having similar behaviors will be particularly hard. Heterogeneity can hardly be detected using external information, i.e. using *a priori* clustering approach, especially in social, economic and marketing areas. Moreover, in the marketing field, in particular, more attention is given to clustering methods which are able to obtain groups that are homogeneous in terms of their response [Wedel & Kamakura 2000]. Therefore, *response-based* clustering techniques, as particular cases of *post hoc* clustering approaches, will become more and more important in statistical literature.

Simple models are not suitable to model complex human behaviors because they only take into account a small number of relationships among the variables. This is the reason, along with computer-science development, for the increase in the use of Structural Equation Models (SEM) [Bollen 1989, Kaplan 2000]. As a matter of fact, in Structural Equation Models the real word complexity can be studied taking into account a whole number of causal relationships among latent concepts (i.e. the Latent Variables (LVs)), each measured by several observed indicators usually defined as Manifest Variables (MVs). Two different approaches exist to estimate model parameters in Structural Equation Models: the *covariance-based* techniques and the *component-based* techniques. The first approach refers to the methods aiming at reproducing the sample covariance matrix of the manifest variables by means of the model parameters. In *component-based* techniques, instead, latent variable estimation plays a main role. As a matter of fact, the aim of *component-based* methods is to provide an estimate of the latent variables in such a way that they are the most correlated with one another (according to the path diagram structure) and the most representative of each corresponding block of manifest variables. Nevertheless, whatever estimation technique is used, Structural Equation Models assume homogeneity over the observed set of units. In other words, all units are supposed to be well represented by a unique model estimated on the whole sample, i.e. the *global model*. If all the units are considered as belonging to a single class in Structural Equation Models when it is not true, i.e. if heterogeneity is not taken into

account, it may lead to biased results both in terms of model parameters and of validation indexes [Jedidi, Jagpal & De Sarbo 1997a, Jedidi, Jagpal & S. De Sarbo 1997b]. Usually heterogeneity in Structural Equation Models is handled by forming classes on the basis of such external variables or on the basis of such standard clustering techniques on manifest and/or latent variables, and then by using the multigroup structural equation modeling of Jöreskog [1971] and Sörbom [1974]. But very rarely, heterogeneity in the models may be captured by well-known observable variables playing the role of moderating variables [Hahn, Johnson, Herrmann & Huber 2002]. Moreover, *post-hoc* clustering techniques on manifest variables, or on latent variable scores, do not take into account in any way the model itself. Hence, while the local models obtained by cluster analysis on the latent variable scores will lead to differences in the group averages of the latent variables but not necessarily to different models, the same method performed on the manifest variables is unlikely to lead to different and well-separated models, both in terms of model parameters and of average latent variable scores. In addition, *a priori* unit clustering in Structural Equation Models is not conceptually acceptable since no causal structure among the variables is postulated: when information concerning the causal relationships among variables is available (as it is in the theoretical causal network of relationships), classes should be looked for while taking into account this important piece of information. In other words, even in the Structural Equation Models framework the need is pre-eminent for a *response-based* clustering method, where the ob-

tained classes are homogeneous with respect to the postulated model.

The aim of this thesis is to find an answer to this specific need by presenting a new technique able to provide *response-based* clustering in PLS Path Modeling (PLS-PM): the Response Based Unit Segmentation in PLS-PM (REBUS-PLS) [Trinchera 2007, Trinchera, Squillacciotti, Esposito Vinzi & Tenenhaus 2007, Trinchera, Romano & Esposito Vinzi 2007, Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Esposito Vinzi, Amato & Trinchera 2008].

To reach this objective, I first review the main clustering techniques (cf. chapter two). The methods used in clustering research can be classified according to two different aspects. First, they can be classified into *a priori* and *post hoc* approaches [Green 1977, Wind 1978]. A clustering approach is called *a priori* when the type and the number of segments are determined in advance by the research, usually on the basis of external information. In *post hoc* clustering, instead, both the type and the number of segments we are looking for are determined on the basis of the results of some data analysis. A hybrid procedure combining both the *a priori* and the *post hoc* approaches is also possible. Nevertheless, its effectiveness depends mainly on the *post hoc* procedure used in the second step [Wedel & Kamakura 2000]. Secondly, they can be classified according to whether *descriptive* or *predictive* statistical methods are applied. Of course, in a *descriptive* clustering method units are segmented looking at the associations be-

tween a set of variables with no difference between endogenous and exogenous variables. In a *predictive* approach, instead, unit clustering is accomplished by analyzing relationships between two sets of variables, one influencing the other.
A priori clustering methods will be discussed in section 2.2, while the *post hoc* approaches will be presented in section 2.3. In both cases, we first discuss the descriptive approaches and then the predictive ones. A detailed discussion on the Mixture Models for clustering will be provided at the end of the chapter (cf. section 2.4).

Successively, a detailed discussion on the estimation methods for the Structural Equation Models will be provided (cf. chapter three). Structural Equation Models [Bollen 1989, Kaplan 2000] include a number of statistical methodologies that allow us to estimate the causal relationships, defined according to a theoretical model, linking two or more latent complex concepts, each measured through a number of observable indicators. The Structural Equation Models notation and the specification of the model will be introduced in section 3.2.
Essentially developed in a social domain, Structural Equation Models were first introduced by Jöreskog [1970] as confirmatory models to assess cause-effect relations among two or more set of variables, based on the maximum likelihood (ML) estimation method (SEM-ML). This method, also known as LISREL (*LInear Structural RELations*), has been for many years the only estimation method for SEM. The term LISREL was initially used for the software implementing the method-

ology [Jöreskog & Sörbom 1996]. However, it had such a rapid development that the methodology and the software have been associated to each other. Furthermore, it is important to notice that other estimation techniques besides the maximum likelihood approach can be used to estimate Structural Equation Models, such as the Generalized Least Squares (GLS) or the Asymptotically Distribution Free (ADF). All these methods are usually referred to as LISREL-type estimation techniques. The factor common to all the LISREL-type estimation techniques is that they are so-called *covariance-based* methods. As a matter of fact, all these techniques aim at reproducing the sample covariance matrix of the manifest variables by means of the model parameters. The fundamental hypothesis underlining these approaches is that the implied covariance matrix of the manifest variables is a function of the model parameters. The *covariance-based* approaches will be discussed in section 3.3. Namely, in subsection 3.3.1 we will focus on the LISREL-type methods.

Subsequently, the *component-based* estimation techniques will be shown (cf. section 3.4). As already said, the aim of *component-based* methods is to provide an estimate of the latent variables in such a way that they are the most correlated with one another (according to the path diagram structure) and the most representative of each corresponding block of manifest variables. The most recognized methods among the *component based* approaches is the PLS approach to Structural Equation Models, also known as PLS Path Modeling (PLS-PM) [Wold 1975, Tenenhaus, Esposito Vinzi, Chatelin & Lauro 2005]. This

approach will be discussed in detail in subsection 3.4.1. More recently, other *component based* techniques have been presented. Namely, the Generalized Maximum Entropy (GME) by Al-Nasser [2003], discussed in subsection 3.4.3, and the Generalized Structured Component Analysis (GSCA) by Hwang & Takane [2004], shown in subsection 3.4.2. For each of the discussed approaches the estimation procedure used, as well as the different indexes to assess model quality, will be discussed.

In the fourth chapter I focus on techniques for detecting unit segments by *response-based* techniques in the case of unknown (latent) moderating effects, i.e. when both the number and the structure of the classes are not known *a priori*. Ways to handle unobserved heterogeneity in the different approaches to Structural Equation Models will be presented. Firstly methods allowing *response-based* clustering in LISREL-type Structural Equation Models will be shown (cf. section 4.2): the Structural Equation Finite Mixture Model (STEMM) by Jedidi et al. [1997a] and Jedidi et al. [1997b] (cf. subsection 4.2.1) and the Bayesian Finite Mixture SEM by Zhu & Lee [2001] (cf. subsection 4.2.2). Further, *response-based* techniques for clustering in the PLS-PM framework will be presented (cf. section 4.3). In this framework, several approaches will be described. Namely, the Finite Mixture PLS [Hahn et al. 2002, Ringle, Wende & Will 2008] (cf. subsection 4.3.1), the PLS Typological Path Model [Squillacciotti 2005, Trinchera, Squillacciotti & Esposito Vinzi 2006] (cf. subsection 4.3.3), the PATHMOX [Sanchez & Aluja 2006, Sanchez & Aluja 2007] (cf. subsec-

tion 4.3.2) and the PLS Path Model Clustering (PLS-PMC) [Ringle & Schlittgen 2007] (cf. subsection 4.3.4). To conclude, unobserved heterogeneity in GSCA will be investigated by the Fuzzy Clusterwise Generalized Structured Component Analysis of Hwang, De Sarbo & Takane [2007] (cf. subsection 4.4.1). Moreover, once the groups are identified, it is important to assess the differences (and similarities) between the detected classes of units. This essentially entails comparing the obtained local models to one another and with the global model. It is for this reason that the last section of the fourth chapter will be devoted to presenting the different techniques allowing us to compare local models (cf. section 4.5), with special regards to the model parameter comparison (cf. subsection 4.5.2), the latent variable scores comparison (cf. subsection 4.5.3), and the model quality comparison (cf. subsection 4.5.4).

The original proposition will be made in chapter five. The Response Based Unit Segmentation (REBUS-PLS) algorithm [Trinchera 2007, Trinchera, Squillacciotti, Esposito Vinzi & Tenenhaus 2007, Trinchera, Romano & Esposito Vinzi 2007, Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Esposito Vinzi, Amato & Trinchera 2008], will be presented.
REBUS-PLS is an iterative algorithm allowing us to estimate at the same time both the memberships of units to latent classes and the parameters of the local models.
Coherent with PLS Path Modeling features, REBUS-PLS does not

require distributional hypotheses. Moreover, REBUS-PLS may lead to local models that are different in terms of both structural and measurement models. Furthermore, REBUS-PLS involves an error-based unit/model "distance" defined according to the Goodness of Fit (GoF) index structure (cf. subsection 3.4.1). This leads up to local models that fit better than the global model. To conclude, REBUS-PLS does not require external/concomitant variables to cluster the units. Nevertheless, external information (when available) can be used to characterize the latent classes identified by REBUS-PLS.

Simulation studies have been done to assess REBUS-PLS ability in detecting unobserved heterogeneity under different hypotheses. In particular, three different simulation schemes have been tested. In the first one, local models are different only as regards the path coefficients intensities, i.e. the structural parameters. In the second simulation scheme, local models are different only concerning the measurement model parameters. While the third scheme takes into account local models that are different as regards both the measurement and the structural parameters. The results of the simulation studies will be shown in section 6.1. To conclude, REBUS-PLS will be applied to a real dataset (cf. section 6.2) involving a customer preference study on the Benetton fashion firm.

The code for running REBUS-PLS algorithm in SAS-IML language (cf. appendix A.2) will be provided in the appendix.

It is important to underline that throughout this work the word *clustering* is to be referred to as unsupervised pattern recognition. Neither the number of classes, nor their composition is known at the beginning of the analysis. In marketing field, clustering, in the sense of unsupervised pattern recognition, is often referred to as segmentation. That is why throughout this work the word segmentation has to be considered equivalent to clustering.

Chapter 2

Clustering techniques

2.1 Introduction

Working with Unobserved Heterogeneity means finding groups of units or clusters having similar behaviors. This essentially entails determining both the number and the composition of classes. In Statistics we have to distinguish between *classification* and *clustering*. The idea behind *classification* is that units belong to a given group, and the aim of the several classification techniques is to assess a decision-rule in order to classify new units into the existing classes. From this point-of view *classification* has to be considered a taxonomic task. *Clustering*, instead, is essentially a grouping task, for which a large variety of methods are available. The aim common to all *clustering* methods is to find out class of units similar in such a way.
Classification and *clustering* are often referred to as supervised pat-

tern recognition and unsupervised pattern recognition. The word supervised and unsupervised refers to whether or not group membership from some training data is given, i.e. if a taxonomy of units is available.
Of course, working with Unobserved Heterogeneity promptly means working in the clustering field. As a matter of fact, no information about group membership is available in the case of Unobserved Heterogeneity.
Moreover, in the marketing field both *classification* and *clustering* are often referred to as segmentation. Nevertheless, throughout this work the word segmentation has to be considered equivalent to clustering.

The methods used in clustering research can be classified according to two different aspects. First, they can be classified into *a priori* and *post hoc* approaches [Green 1977, Wind 1978]. A clustering approach is called *a priori* when the type and the number of segments are determined in advance by researchs, usually on the basis of external information. In *post hoc* clustering, instead, both the type and the number of segments we are looking for are determined on the basis of the results of some data analysis. A hybrid procedure combining both the *a priori* and the *post hoc* approaches is also possible. Nevertheless, its effectiveness depends mainly on the *post hoc* procedure used in the second step [Wedel & Kamakura 2000].

A second way to classify clustering techniques is according to whether

descriptive or *predictive* statistical methods are applied. Of course, in a *descriptive* clustering method, units are segmented looking at the associations between a set of variables with no difference between endogenous and exogenous variables. In a *predictive* approach, instead, unit clustering is accomplished by analyzing relationships between two sets of variables, one influencing the other.

As stated by Gordon [1999], the principal outcome of a clustering study is to provide a *partition* of the units in a set of classes. Throughout this work the words class, group, segment and cluster are to be considered as synonyms.

In this chapter a review of the main clustering techniques will be provided. First *a priori* clustering techniques will be discussed (cf. section 2.2), then the *post hoc* approaches will be presented (cf. section 2.3). In both the cases, we first discuss the descriptive approaches and then the predictive ones. A detailed discussion on the Mixture Models for clustering will be provided at the end of the chapter (cf. section 2.4).

2.2 *A priori* clustering techniques

In the *a priori* approach the number and the type of classes are determined independently from the statistical method that will be later used to analyze the data. The main difference between an *a priori*

descriptive method and an *a priori* predictive method is that in the first case external information is used to obtain clusters of units homogeneous with regard to the variable used to cluster, while in the *a priori* predictive approach, once groups are obtained as regards to endogenous or exogenous variables, relations between the two sets of variables within the groups are studied.

2.2.1 *A priori* descriptive methods

The most popular approach to obtain an *a priori* clustering of units using a descriptive procedure is the cross-tabulation. This approach allows us to examine frequencies of units that belong to specific combinations of categories on more than one variable using a so-called cross-tabulation table displaying the joint distribution of two or more variables.

Several approaches exist to analyze cross-tabulation tables. Among them we have to distinguish between techniques to be used in the case of two variables and techniques allowing us to take into account interaction between more than two variables.
Without doubt, the most popular test for the significance of the relationship between categorical variables is the Pearson *chi-square*. This measure is based on the fact that we can compute the expected frequencies in a two-way table (i.e. frequencies that we would expect if there was no relationship between the variables). The *chi-square* test becomes increasingly significant as the observed frequencies deviate

further from the expected one. Discussion on the use of the *chi-square* test in cross-tabulation can be found in each statistical manual.
Among the techniques used to analyze multi-cross-tabulation tables, log-linear models provide a more sophisticated way of looking at cross-tabulation tables. Specifically, it is possible to test the different variables that are used in the cross-tabulation and their interactions for statistical significance. The term log-linear derives from the fact that in log-linear models logarithmic transformations allow us to restate the problem of analyzing multi-way frequency tables in terms that are very similar to ANOVA. In particular, it is possible to think of the multi-way frequency as a table to reflect various main effects and interaction effects that added together in a linear function, bring about the observed table of frequencies. Bishop, Fienberg & Holland [1975] provide details on how to derive log-linear equations to express the relationship among factors in a multi-way frequency table.

2.2.2 *A priori* predictive methods

In a *predictive* approach, instead, groups of units are obtained as regards only one of the two sets of endogenous and exogenous variables [Wilkie & Cohen 1977]. In a *forward* approach groups are formed by using exogenous variables, then the *a priori* classes are related to the set of endogenous variables. A *backward* procedure, instead, uses endogenous variables to define groups and then uses exogenous variables to describe the *a priori* obtained groups. A common method applied in a backward *a priori* predictive approach is discriminant analysis

[Fisher 1958, McLachlan 1992]. Nevertheless, this method is more useful to describe segments than a real clustering method aiming at identifying groups of units. In other words it is closer to a classification task than to clustering. In a forward perspective, tabulation appears to be the most popular method. Problems arise if more than two variables are taken into account. As Wildt & Mc Cann [1980] suggest, linear regression can overcome these difficulties by estimating both the effect of multiple segmentation variables and their partial contributions.

2.3 *Post hoc* clustering techniques

The *post hoc* methods define the number and the type of segments on the base of analysis' results. In a descriptive *post hoc* approach, groups are defined according to such measured characteristics, while in a predictive *post hoc* analysis, groups are obtained on the basis of the estimated relationships between the exogenous and the endogenous sets of variables. Therefore, segments obtained by a descriptive *post hoc* method are homogeneous as regards measured characteristics, while segments formed by a predictive *post hoc* method are homogeneous in the relationships between exogenous and endogenous variables.

A classification of *post hoc* clustering techniques can be obtained referring to the nature of the classes obtained. In this sense they can be distinguished in *nonoverlapping*, *overlapping* and *fuzzy* [Hruschka 1986]

clustering techniques. In *nonoverlapping* clustering methods each unit belongs to only one class. In an *overlapping* clustering method units can belong to multiple classes. Instead, in *fuzzy* clustering each unit is associated with a degree of membership to belong to a unique (nonoverlapping fuzzy) class or a multiple (overlapping fuzzy) class.
We can distinguish the three different types of clustering methods also according to the form of the partition matrix $\boldsymbol{Z}$. The partition matrix $\boldsymbol{Z}$ is an N by K matrix, where N is the number of units and K is the number of classes taken into account. The generic element of the $\boldsymbol{Z}$ matrix, z_{ik}, indicates the assignment of a unit to a class. Specifically, z_{ik} represents a membership-value that is equal to one if the i-the unit belongs to the k-th class otherwise it is equal to zero, i.e.:

$$z_{ik} = \begin{cases} 1 \text{ if } i \in k \\ 0 \text{ if } i \notin k \end{cases} \tag{2.1}$$

In clustering methods providing *nonoverlapping* clusters, units belong to a one and only class, therefore matrix $\boldsymbol{Z}$ has only one element in each row equal to one, the other elements being zero. In the situation of *overlapping* clusters units can belong to more than one class, consequently the rows of the matrix $\boldsymbol{Z}$ may have several elements equal to one. In *fuzzy* clustering procedures, units have partial memberships in more than one class. In this case z_{ik} indicates a membership value. Of course in this case z_{ik} is a nonnegative number bounded between zero and one and the sum of row values must be equal to one.

Two types of *fuzzy* clustering methods can be distinguished. The first one refers to the fuzzy set theory of Zadeh [1965], the second based on the idea that data arise from a mixture of distributions [McLachlan & Basford 1988]. In the first case, the traditional assumption that every unit is to be assigned to one cluster is replaced with the idea that units can belong to more than one class with a particular degree of membership z_{ik}.
The second type of *fuzzy* clustering techniques aims at estimating the probability of each unit of belonging to each segment. As already said, this approach is based on the idea that data arise from a mixture of distributions [McLachlan & Basford 1988]. A detailed discussion on Mixture Models will be given in subsection 2.4.
Both the approaches provide membership values z_{ik} that are bounded between zero and one. Nevertheless, in a "pure" fuzzy approach, the idea is that units really belong to different classes with a different degree of membership, while in a mixture approach the basic assumption is that units only belong to one class and the information in the data is insufficient to determine uniquely its assignation. In this last case z_{ik} is the probability of each unit of belonging to each class.

2.3.1 Descritive *post hoc* methods

Clustering procedures are the most widely-used tools to achieve *post hoc* descriptive clustering. Cluster analysis is not a single technique, but has to be considered as a variety of techniques that attempt to form classes with internal cohesion and external isolation [Gordon

1999]. Exhaustive introduction to cluster analysis overcomes the purpose of this section. A whole set of monographs was published on cluster analysis. For a review see for example, Everit [1992] and Gordon [1999]. In this work a classification of the different clustering procedures will be provided and the main features of the different approaches shortly analyzed.
The several descriptive *post hoc* methods will be discussed according to the nature of the partition matrix provided. The *nonoverlapping* methods will be presented first. Then the *overlapping* techniques will be shown and to conclude, the *fuzzy* descriptive *post hoc* methods will be discussed.

Nonoverlapping techniques can be distinguished essentially in hierarchical [Frank & Green 1968] and nonhierarchical methods.
The first ones provide a hierarchy of partitions. Different levels of aggregation, i.e. different number of classes, are investigated including the initial class formed by the whole sample and the N classes each formed by a single unit. Agglomerative hierarchical clustering methods start from N classes, each formed by a single unit, and arrive through successive steps at a unique class containing all the units. Divisive hierarchical methods, instead, start from the global class, i.e. the class containing all the units, and arrive at defining the N classes. The issue common to all clustering techniques is the definition of the dissimilarity (or similarity) criterion, which may be a dissimilarity measure, a distance measure or a ultrametric measure. Differences

arise also in the way used to compute the "distance" between a cluster and the units, or across classes, i.e. in the agglomerative criterion chosen. The final result common to all the nonoverlapping nonhierarchical clustering techniques is a dendogram, i.e. a tree structure representing all the hierarchy of partitions.

Nonhierarchical clustering methods, instead, provide a partition of the N units in a number of classes K defined *a priori*. They start from an initial partition of the units into K classes and move units from one class to another in successive steps by the optimization of a certain criterion of interest. Several nonhierarchical classification methods have been proposed, among them the Forgy [1965] and MacQueen [1967] methods. They differ according to the criterion optimized and the algorithm used in the optimization process. In particular, three characteristics distinguish the various classification methods: (1) the selection of seed points, i.e. of the starting points, (2) the type of cluster assignment process, (3) the statistical criterion used to assign the points to the clusters. The widely-used nonhierarchical method is the K-means algorithm presented by MacQueen [1967]. Several extensions have been proposed to the K-means algorithm such as the one of De Sarbo, Carroll & Clark [1984] that clusters units and simultaneously derives weights for the variables used to cluster units. A common problem of all nonhierarchical clustering procedures is the definition of the number of classes to be considered and the definition of the initial partition. The starting partition can be obtained in several ways, i.e. by randomly assigning units to clusters, on the basis of

external information or by performing a hierarchical clustering procedure. None of these procedures appears to be better than the others. As a matter of fact, a random partition may lead to local optimum, while using hierarchical clustering techniques needs to have a sufficient number of units.
As stated by Punj & Stewart [1983], nonhierarchical methods seem to perform better than hierarchical ones. As a matter of fact, they are more robust to outliers and to the presence of irrelevant attributes [Wedel & Kamakura 2000]. Nevertheless, hierarchical algorithms allow us to investigate different numbers of classes and do not require to define it *a priori* as nonhierarchical ones need to. In many applications external information may be used to select the number of classes to take into account, but if no information is available hierarchical clustering methods have to be preferred to nonhierarchical ones.

Overlapping clustering methods were first presented by Shepad and Araibe [Shepard & Arabie 1978]. Since then, a lot of different techniques have been proposed, such as the Individual Difference Cluster analysis (INDCLUS) by Carrol & Arabie [1983], the General Nonhierarcichal Clustering analysis (GENNCLUS) by De Sarbo [1982] and the Constrained Cluster analysis (CONCLUS) by De Sarbo & Mahajan [1984].

Two types of *fuzzy* clustering methods can be distinguished. The first one refers to the fuzzy set theory of Zadeh [1965], the second based on

the idea that data arise from a mixture of distributions [McLachlan & Basford 1988]. In the first case, the traditional assumption that every unit is to be assigned to one cluster is replaced with the idea that units can belong to more than one class with a particular degree of membership z_{ik}. The first authors who proposed applying the fuzzy set theory to clustering problems were Bezdek [1974] and Dunn [1974]. They developed the fuzzy c-means (FCM) algorithm. The FCM can be considered the fuzzy variant of K-means nonhierarchical algorithm. The idea is to classify the units in a pre-specified number of classes by minimizing a sum of squared errors, computed as the difference between each observed value and the center of each class. It is important to notice that in FCM since all units belong to all classes (according to fuzzy clustering logic) the centroid of a cluster is the mean of all units, weighted by their degree of belonging to the cluster. A generalization of FCM algorithm, the fuzzy c-lines (FCL) was developed by Bezdek *et al.* [Bezdek, Coray, Gunderson & Watson 1981*a*, Bezdek, Coray, Gunderson & Watson 1981*b*]. FCM and FCL, as well as fuzzy clusterwise regression (FCR) [Wedel & Steenkamp 1989, Wedel & Steenkamp 1991], and fuzzy grade of membership model, (GoM) [Manton, Woodnury & Tolley 1994] are part of a family of methods named fuzzy c-variates *FCV* [Bezdek et al. 1981*a*, Bezdek et al. 1981*b*], in which the prototypes are multidimensional linear varieties represented by some local principal component vectors. The FCV clustering algorithms can be regarded as a simultaneous algorithm of fuzzy clustering and principal component

analysis. This allows us to obtain not only round classes as in FCM, but also classes with linear configuration.
The second type of *fuzzy* clustering techniques aims at estimating the probability of each unit of belonging to each segment. As already said, this approach is based on the idea that data arise from a mixture of distributions [McLachlan & Basford 1988]. A detailed discussion on Mixture Models will be given in section 2.4.

2.3.2 Predictive *post hoc* methods

Post hoc predictive clustering methods allow us to obtain clusters of units homogeneous as regards the relationships in the model. Several techniques achieve this objective.
The traditional approach is the Automatic Interaction Detection (AID). Groups obtained by AID are maximally different according to an endogenous variable and are obtained on the basis of exogenous variables. Many extensions of the AID have been proposed to handle particular cases, such as the Multivariate AID (MAID) algorithm [MacLachlan & Johansson 1981] in the case of more than one dependent variable and the CHAID [Kass 1980] in the case of categorical dependent variables. A closely related technique is the so-called classification and regression trees (CART) presented by Breiman, Friedmabn, Olshen & Stone [1984]. More details on classification and regression trees can be found in Haughton & Oulabi [1993], and in Trasher [1991].
Even neural networks (NN) [Balakrrishnan, Cooper, Jacob & Lewis 1995] and extensions of conjoint analysis, such as componential clas-

sification [Green 1977, Green & De Sarbo 1979] have been used to obtain *post hoc* predictive clustering. Various hierarchical predictive clustering approaches have been presented by Christal [1968], Bottenberg & Christal [1968], Lutz [1977], Ogawa [1987] and Kamakura [1988]. The main drawback common to all of these methods is that misclassification at an early stage of the algorithm may carry on to higher levels [Wedel & Kamakura 2000].

Clusterwise regression [Späth 1979, Späth 1981, Späth 1982] tries to overcome this problem in a nonhierarchical way. The aim is to cluster units so as to optimize the fit of the regression within the classes. Since then, a number of extensions to the first method proposed by Späth have been developed. For example the De Sarbo, Oliver and Rangaswamy clusterwise regression algorithm to deal with overlapping classes and multiple dependent variables [De Sarbo, Oliver & Rangaswamy 1989] and the Wedel and Kistemaker algorithm to handle partial membership of units in the classes [Wedel & Kistemaker 1989]. Clusterwise regression can be considered a fuzzy approach since the algorithm provides a degree of membership of each unit to several classes. It is a powerful method to achieve *post hoc* predictive clustering since it combines clustering and prediction. Nevertheless, the properties of the estimators are not established and clusterwise results depend on subjective choices influencing the degree of separation of the classes.

An extension of clusterwise regression to the Structural Equation Model

context is the Fuzzy Clusterwise Generalized Structured Component Analysis [Hwang et al. 2007] (cf. subsection 4.4.1).

Latent Class (or Mixture) regression methods try to solve the drawbacks of the clusterwise regression. Mixture Regression methods allow simultaneous group units in latent (unobserved) classes and estimate regression models in each class [Wedel & De Sarbo 1994]. An extension of Mixture Regression models are the Mixture Models applied to Structural Equation Models (cf. subsection 4.2.1, subsection 4.2.2 and subsection 4.3.1). The biggest advantage of these methods is that they directly identify classes that are homogeneous in how they respond to the model.

2.4 Mixture Models for clustering

Mixtures of distributions have been largely used in the last years to solve some different issues in Statistics. One of the recent uses of Mixture Models is in clustering analysis [Everit & Hand 1981, Titterington, Smith & Makov 1985, McLachlan & Basford 1988, McLachlan & Peel 2000, Wedel & Kamakura 2000].
In this context it is assumed that data arise from a mixture of a specified number of populations (K) mixed in unknown proportions and each characterized by a specific density function. In such a sense, Mixture Models for clustering aim at estimating the unknown parameters of the K density functions and the posterior probability of group

membership for each unit.
The Mixture Models approach to clustering has to be considered a model-based clustering technique since the form of each density function has to be specified in advance. Several labels have been used to refer to Mixture Models for clustering, such as Latent Class Cluster Analysis [Vermunt & Magidson 2002], Mixture Likelihood Approach to Clustering [McLachlan & Basford 1988, Everit 1992], Unsupervised Learning [McLachlan & Peel 1996], and others, nevertheless the statistical techniques behind all of them are the same.
In Mixture Models, as well as in many other clustering techniques, the number of clusters, i.e. the number of components, has to be defined *a priori*. This problem will be explicitly discussed in subsection 2.4.3. Other issues related to the use of Mixture Models concern the estimation algorithm used to estimate the density function and its convergence in local optima, as well as the choice of starting values (cf. subsection 2.4.2).

2.4.1 A general definition of Mixture Models

Assuming that P variables have been measured on N units, $\boldsymbol{x}_i$ is a row vector of P dimensions containing the values of the P variables for the i-th unit. Under the Finite Mixture Models each $\boldsymbol{x}_i$ can be viewed as arising from a specific k-th population. The probability density function associated to the vector $\boldsymbol{x}_i$ can be represented by the general form $f_k\left(\boldsymbol{x}_i|\boldsymbol{\theta}_k\right)$, where $\boldsymbol{\theta}_k$ is the vector of all parameters associated with the specific form of the density function chosen for the k-th class.

Distributions of the exponential family, such as the normal, the Poisson, the exponential gamma and many others are usually used as density functions in a Mixture Model. As a matter of fact, all the distributions of the exponential family can be studied simultaneously, rather than as a collection of unrelated cases. Moreover, all these distributions are characterized by a mean value, μ_{pk} and possibly a dispersion parameter specific for each variable. In the case of normally distributed data, for example, $\boldsymbol{\theta}_k$ contains the means, μ_{pk}, and the variances, σ^2_{pk}, of the normal distribution within each class.

Once the conditional density function is defined as above, the unconditional density function can therefore be represented as a mixture of the conditional, i.e. class specific, density function:

$$f\left(\boldsymbol{x}_i|\boldsymbol{\phi}\right) = \sum_{k=1}^{K} \pi_k f_k\left(\boldsymbol{x}_i|\boldsymbol{\theta}_k\right) \tag{2.2}$$

where $\boldsymbol{\phi} = (\boldsymbol{\pi}, \boldsymbol{\theta})$ is the vector of all unknown parameters, and π_k's are the mixing proportions. The summation on the right-hand side indicates that the distribution of $\boldsymbol{x}_i$ is a weighted mean of the class specific distribution, where the mixing proportions serve as weights. The mixing proportions π_k are nonnegative quantities subject to the following constraints:

$$\pi_k \geq 0 \tag{2.3}$$

and

$$\sum_{k=1}^{K} \pi_k = 1 \tag{2.4}$$

The aim of the Mixture Models is to estimate the parameters of each density function and the mixing proportions. In other words, the goal is to estimate the parameters vector $\boldsymbol{\phi} = (\boldsymbol{\pi}, \boldsymbol{\theta})$ by a maximum likelihood approach.
In this sense, the likelihood function for $\boldsymbol{\phi}$ can be easily formulated as:

$$L(\boldsymbol{\phi}; \boldsymbol{X}) = \prod_{i=1}^{N} f(\boldsymbol{x}_i|\boldsymbol{\phi}). \tag{2.5}$$

This equation measures the likelihood that the parameters vector $\boldsymbol{\phi}$ could have produced the observed vector $\boldsymbol{x}_i$. The same results can be obtained by working on the log-likelihood function defined as:

$$\log L(\boldsymbol{\phi}; \boldsymbol{X}) = \sum_{i=1}^{N} \log f(\boldsymbol{x}_i|\boldsymbol{\phi}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}_i|\boldsymbol{\theta}_k) \right). \tag{2.6}$$

An estimate of $\boldsymbol{\phi}$ can be obtained as a solution of the following maximization problems:

$$\max L(\boldsymbol{\phi}; \boldsymbol{X}) = \prod_{i=1}^{N} f(\boldsymbol{x}_i|\boldsymbol{\phi}) \tag{2.7}$$
$$\text{subject to } \sum_{k=1}^{K} \pi_k = 1 \text{ and } \pi_k \geq 0$$

i.e. by maximizing the likelihood in equation 2.5, with respect to $\boldsymbol{\phi}$ and under the constraints in equations 2.3 and 2.4. The same can be obtained referring to the log-likelihood function expressed in 2.6.

The maximization problems in 2.7 can be resolved easily by means of standard optimization routines such as the Newton-Raphson method [McHugh 1956, McHugh 1958] or by using the Expectation-Maximization (EM) algorithm [Dempster, Laird & Rubin 1977].
The Newton-Raphson method does not always assure the convergence, but when the convergence is achieved it requires fewer iterations than the EM algorithm. The EM algorithm, instead, always assures the convergence at least in a local maximum. Moreover, the EM algorithm, thanks to its computational simplicity, is easily programmed. Although there is no evidence that the EM algorithm performs better than numerical optimization, it is preferred in general [Titterington 1990].
A detailed discussion on the EM algorithm and on its major drawbacks will be provided in subsection 2.4.2.

Once the parameters are estimated, i.e. once $\boldsymbol{\phi}$ is obtained, it is possible to estimate for each unit the posterior class membership probability. This provides a *fuzzy* clustering of units in K classes.
Each unit belongs to each class by a probability given by:

$$\rho_{ik} = P\left(k|\boldsymbol{x}_i\right) = \frac{P\left(k\right) \cdot P\left(\boldsymbol{x}_i|k\right)}{P\left(\boldsymbol{x}_i\right)} = \frac{\pi_k f_k\left(\boldsymbol{x}_i|\boldsymbol{\theta}_k\right)}{\sum_{k=1}^{K} \pi_k f_k\left(\boldsymbol{x}_i|\boldsymbol{\theta}_k\right)} \tag{2.8}$$

Where $P\left(k\right)$ is the probability of belonging to a class independently from the $\boldsymbol{x}_i$ values, i.e. the size of the class π_k; $P\left(\boldsymbol{x}_i|k\right)$ is the prior probability of membership; and $P\left(\boldsymbol{x}_i\right)$ is the probability to show specif $\boldsymbol{x}_i$ values independently from the class membership. That is finally an application of the Bayes theorem [Bayes 1763/1958].

A partition of the N units in *nonoverlapping* classes is possible by assigning each unit to the class to which it has the highest estimated posterior probability of membership.

2.4.2 The EM algorithm and the other estimation methods

The maximization problem expressed by equation 2.7 was first solved using the method of moments [Pearson 1894, Quandt & Ramsey 1978]. Nevertheless, nowadays it is usually solved by means of two main methods: the EM algorithm [Dempster et al. 1977] and the Newton-Raphson method [McHugh 1956, McHugh 1958]. These are both it-

erative procedures aiming at providing a numerical solution to the likelihood. The EM algorithm provides a solution to the maximum likelihood estimation in *incomplete-data* frameworks. The incomplete-data situation where the EM algorithm has been applied includes not only evident incomplete-data situations, such as the presence of missing values, but also many other situations where the incompleteness is not so evident, as namely in Mixture or Latent Classes.

The EM algorithm

The presentation of the EM algorithm is generally due to Dempster et al. [1977]. Nevertheless, before Dempster, Laird and Rubin's work many authors had proposed some methods that then turned out to be special applications of the EM algorithm, see for example Newcomb [1886], McKendrick [1926], Healy & Westmacott [1956] and Buck [1960].

In the case of the Mixture Model estimation, the aim of the maximization problem expressed in equation 2.7 is to estimate the model's parameters, $\boldsymbol{\theta}$ and the membership values $\boldsymbol{\pi}$, i.e. provide an estimate of $\boldsymbol{\phi}$, with $\boldsymbol{\phi} = (\boldsymbol{\pi}, \boldsymbol{\theta})$. The EM algorithm allows us to solve this problem by maximizing at each iteration a simplified function. This is obtained by associating to the incomplete-data problem a complete-data problem for which the ML is computationally easier to handle. This is achieved by adding at each iteration additional information that replaces unobserved data.

In the Mixture Models the unobserved data to be replaced is the mem-

bership value, π_{ik}, of the i-th unit to the k-th class. The additional information to be added is the expected membership, z_{ik}, of a unit to a class, given a set of preliminary estimates of the model's parameters.

Each iteration of the EM algorithm is composed of two steps, the E-step (Expectation Step) in which the expectations of the membership value, z_{ik}, are computed given a provisional estimate of $\boldsymbol{\theta}$, and the M-step in which the expectation of the log-likelihood obtained in E-step is maximized with respect to the parameters.
In more formal terms, in the first iteration let z_{ik} be one if the i-the unit belongs to the k-th class and zero otherwise, i.e.:

$$z_{ik} = \begin{cases} 1 \text{ if } i \in k \\ 0 \text{ if } i \notin k \end{cases} \tag{2.9}$$

The z_{ik} values are included in a $\boldsymbol{Z}$ matrix of dimensions N by K.
Once the data is "completed" by means of the z's values, the complete-data log-likelihood function defined in 2.6 can be rewritten as:

$$\log L(\boldsymbol{\phi}) = \sum_{k=1}^{K} \sum_{i=1}^{N} (z_{ik} \log f_k(\boldsymbol{x}_i) + z_{ik} \log \pi_k) \tag{2.10}$$

The first term of this equation, i.e. $\sum_{k=1}^{K} \sum_{i=1}^{N} z_{ik} \log f_k(\boldsymbol{x}_i)$, does not depend on $\boldsymbol{\phi}$, i.e. on $\boldsymbol{\pi}$. Since equation 2.10 is linear in z_{ik}, the E-step of the second iteration simply requires the calculation of the

expectation of z_{ik} given the observed data $\boldsymbol{x_i}$:

$$E\left(z_{ik}|\boldsymbol{x}_i\right) = z_{ik}^{(2)} = \rho_{ik} \tag{2.11}$$

The "new" value of z_{ik}, that can be shown to be equal to the posterior probability that unit i belongs to class k, is so used to obtain a provisional estimate of the completed log-likelihood function, i.e. $E\left(\log L\left(\boldsymbol{\phi}|\boldsymbol{xZ}\right)\right)$. This value is maximized in the M-step with respect to the π_{ik} to give:

$$\pi_k^{(3)} = \sum_{i=1}^{N} z_{ik}^{(2)}/N \tag{2.12}$$

Thus, the estimate of the prior probability at each step is the average of the posterior probability in each class: each unit contributes to the estimation of π_k according to its posterior probability of membership in the k-th class calculated in the previous iteration.
The two steps are alternated until there is convergence on the increase of the likelihood function value. It is important to notice that convergence is more a stopping rule than a real convergence. As a matter of fact, convergence of the EM algorithm is achieved when the likelihood function value does not increase noticeably from one step to another.

As said, one of the major reasons to use the EM algorithm is that transforming an incomplete-data situation to a complete-data one involves maximizing in the M-step a complete likelihood function that

is often computationally easier. If this is not the case, i.e.if even the complete likelihood function is still difficult to handle, then the EM algorithm is less attractive. Nevertheless, on several occasions the complete likelihood function is easily estimated under some conditions. Generalizations of the EM algorithm have recently been proposed to handle this kind of situation, such as the Expectation-Conditional Maximization (ECM) of Meng & Rubin [1993], the ECME algorithm [Liu & Rubin 1994] and the Alternative ECM (AECM) by Meng & van Dyk [1995].

The local maxima and others issues of the EM algorithm

Since Dempster et al. [1977] showed that the likelihood function value does not decrease after an EM iteration, and since the likelihood function value increases from one step to another of the EM algorithm, hence under fairly general conditions, the convergence is assured at least in a local maximum. In particular, in the case where the likelihood function $L(\boldsymbol{\phi})$ is unimodal, the EM sequence converges to the unique ML solution irrespective of its starting value. Otherwise, the convergence of the EM algorithm in a local or global maximum, and rarely in a saddle point, depends on the choice of the starting values, as related by McLachlan & Krishnan [1997].
The problem of multiple maxima in Mixture Models is well documented [Titterington et al. 1985]. In particular, conditions increasing the perils to converge to local optima are: a large number of parameters to be estimated, limited information on the units leading to no

correct starting values and groups' density functions which are not well separated. A great deal of advice has been proposed to solve the convergence in local maxima. Most advised to use different starting values for the EM algorithm, as well as using *a priori* descriptive clustering method to obtain an initial partition of the units, that in principle should be closer to the optimal solution than random starting values.
Another criticism of the EM algorithm is that it does not provide an estimate of the covariance matrix of the parameter estimates. Several developments of the EM algorithm have been presented to overcome this problem. Namely, the ones proposed by Louis [1982] and Meilijson [1989].

The Newton-Rahpson type methods

The Newton-Raphson [McHugh 1956, McHugh 1958] methods are numerical techniques allowing us to find zeros of a specified function, i.e. aiming to solve maximization problems. Three different approaches are included in this framework: the Newton-Raphson method, the quasi-Newton methods and the modified Newton methods. A detailed discussion on the Newton-Rahpson methods goes beyond the aim of this work. For a more detailed discussion please refer to Dennis & Schnabel [1983] and to Scales [1985].
The "pure" Newton-Raphson method uses a linear Taylor series expansion to find the zeros of the function. Usually, few iterations are required to converge. Nevertheless, convergence is not always as-

sured, namely in the case of the not concave log-likelihood function [McLachlan & Basford 1988]. Since the Newton-Raphson method requires the computation of the Information matrix (that is the negative of the Hessian matrix), it provides the asymptotical variance of the estimated parameters.

2.4.3 Select the number of classes in Mixture Models

A crucial problem in clustering by Mixture Models is the choice of the number of classes or groups to take into account, i.e. the number of components to include in the mixture. Two main approaches exist to determine the number of classes to be considered. The first one is based on a penalized form of the likelihood function, the other uses the likelihood ratio to perform a test. The two approaches are going to be briefly discussed. The most recent development is, however, the use of computationally intensive techniques like parametric bootstrap [McLachlan & Peel 1999] and Markov Chain Monte Carlo methods [Bensmail, Celeux, Raftery & Robert 1997].

Likelihood Ratio Test

The likelihood ratio test (LRT) appears to be the natural way to assess the number of classes to take into account. The LRT can be used to test the null hypothesis (H_0) of K classes against the alternative hypothesis (H_1) of K^* classes to consider, with $K^* > K$. Usually the

test is performed under the alternative hypothesis of $K + 1$ classes, i.e.:

$$H_0\text{: } k = K \tag{2.13}$$

versus

$$H_1\text{: } k = (K + 1) \tag{2.14}$$

This test is simply based on the difference between the maximized likelihood under H_0 and H_1, i.e.

$$-2 \log \lambda = 2 \left[\log L_{H_0} - \log L_{H_0}\right] \tag{2.15}$$

where λ is the likelihood ratio test statistic.
Given certain regularity conditions, the LRT statistic follows a *chi-square* distribution with degrees of freedom equal to the difference between the number of parameters under the null and the alternative hypothesis for nested models under the null hypothesis. Unfortunately, in the case of Mixture Models, the LRT is not asymptotically distributed as a *chi-square*. Because the H_0 corresponds to a boundary of the parameters space for H_1, one of the regularity conditions is broken [Böhning, Dietz, Schaub, Schlattmann & Lindsay 1994]. In particular, under H_0 the generalized likelihood ratio test statistic is not asymptotically a full rank quadratic form [Aitkin & Rubin 1985, Titterington 1990].
Several attempts have been made to propose different tests derived

from LRT and based on a Monte Carlo procedure [Aitkin, Anderson & Hinde 1981, Lachlan 1987, De Soete & De Sarbo 1991]. Nevertheless, all these proposed strategies are computationally hard. That is why nowadays other criteria, such as Information based criteria are preferred.

Information Criteria

As the likelihood increases with the addition of components to a Mixture Model, i.e. with a higher number of classes, some indexes for the assignment of the number of classes are based on a "penalization" of the likelihood. Usually a term taking into account the number of parameters in the model is subtracted from the likelihood, or log likelihood. This results in a penalized log likelihood yielding the so called Information Criteria for the choice of the number of classes in a Mixture Model approach to clustering.
Several Information Criteria are available. They are distinguished from one another on the basis of the "penalization" term to apply to log likelihood. These are heuristic criteria and it is not possible to perform any test. The "best" model, i.e. the number of groups to consider, is chosen comparing the criterion obtained for successive numbers of classes and the model for which the chosen criterion is the smallest is selected.
In a general form they are expressed by the equation:

$$C = -2 \log L + dt_K \tag{2.16}$$

where t_K is the number of parameters estimated and d is a constant. Different criteria impose different values on d.

The classical Information Criterion is the Akaike's one (AIC). It is characterized by a d value equal to 2:

$$AIC = -2 \log L + 2t_K \tag{2.17}$$

Bozdogan in 1987 proposed a modified Akaike Information Criteria, the MAIC, where $d = 3$ [Bozdogan 1987]:

$$MAIC = -2 \log L + 3t_K \tag{2.18}$$

Other criteria penalizing more the likelihood by means of a sample size penalty are: the Bayesian Information Criteria (BIC), proposed by Schwarz [1978]:

$$BIC = -2 \log L + \log (N)\, t_K \tag{2.19}$$

and the Consistent Akaike Information Criteria (CAIC):

$$CAIC = -2 \log L + \log (N + 1)\, t_K \tag{2.20}$$

Both are more conservative than the AIC and prefer models with fewer classes than models with more classes. More recently new criteria using the estimated Information matrix have been proposed. Among them, Bozdogan [1993] proposed the Informational Complexity crite-

rion (ICOMP). This is an extension of Akaike's Information Criterion obtained by adding a correction for model complexity that is measured by the complexity of the estimated inverse Information matrix.

$$ICOMP = -2\log(L) + t_K \log(tr\{\Sigma\}) - \log(|\Sigma|) \qquad (2.21)$$

where Σ is the covariance of the estimates obtained as the inverse of the Information matrix. This criterion considers the balance between improved fit with a more saturated model, i.e. a model with more classes, and the increased complexity of such a model.

2.4.4 Assessment of class separation

Mixture Models provide a fuzzy clustering of the data. Each unit belongs to each class with a probability value given in equation 2.8. Therefore, once the number of classes is chosen (cf. subsection 2.4.3), it is important to assess the class separation. As a matter of fact, it is necessary to ensure that class centroids of the conditional density are sufficiently separated for the selected number of classes.
The most simple way is just to look at the posterior probabilities. If units are in great measure associated to a class with probability value close to one, then we can conclude that classes are well separated.
Nevertheless, several indexes based on entropy have been proposed to investigate the degree of class separation. The Entropy index in equation 2.22 is an index bounded between zero and one. EN values close to one indicate that classes are well separated, while EN values

close to zero mean that there is no separation among classes. In fact, values close to zero indicate that the posterior probabilities are equal for each observation; this implies that the centroids of the conditional distributions are not sufficiently separated.

$$EN_K = 1 - \frac{\sum_{i=1}^{N}\sum_{k=1}^{K}\rho_{ik}\log\left(\rho_{ik}\right)}{N\log K} \tag{2.22}$$

where ρ_{ik} is the posterior probability of membership of unit i to belong to the k-th latent class (cf. equation 2.8). A modification to the EN was proposed by Celeux & Soromenho [1996]. They proposed a normed entropy criterion (NEC) defined as:

$$NEC_K = \frac{EN_K}{\log L\left(K\right) - \log L\left(1\right)} \tag{2.23}$$

where $\log L\left(K\right)$ and $\log L\left(1\right)$ are the values of the log-likelihood function in the case of K classes and in the case of a unique class.
A drawback of this index is that it is not defined in the case of $K = 1$. Usually the EN index is preferred to others to assess class separation.

Chapter 3

Structural Equation Models: several estimation techniques for a unique model

3.1 Introduction

Modeling the real world is a fundamental task in Statistics. Models are built for describing, understanding, estimating, reproducing and inspecting real phenomena [Piccolo 1998]. As well-known, a model is an exemplification of reality. The basic aim is to explain the complexity inside a system by studying the relationships among variables observed over statistical units.
Structural Equation Models (SEM) [Bollen 1989, Kaplan 2000] include a number of statistical methodologies that allow us to estimate the

causal relationships, defined according to a theoretical model, linking two or more latent complex concepts, each measured through a number of observable indicators.

The basic idea is that complexity inside a system can be studied taking into account a whole of causal relationships among latent concepts, called Latent Variables (LV), each measured by several observed indicators usually defined as Manifest Variables (MV). It is in this sense that, Structural Equation Models represent a joint-point between the path analysis [Tukey 1964, Alwin & Hauser 1975] and the Confirmatory Factor Analysis [Thurstone 1931].

As a matter of fact, factor analysis presumes that a number of factors (i.e. the latent variables) smaller than the number of observed variables are responsible for the shared variance-covariance among the observed variables. Hence, SEM receive from Confirmatory Factor Analysis the idea that different subsets or blocks of variables are expression of different concepts. Moreover, path models are a logical extension of regression models as they involve the analysis of simultaneous multiple regression equations. More specifically, a path model is a relational model with direct and indirect effects among observed variables, while multiple-multivariate regression models being additive by definition, only take into account direct relationships between the independent variables and the dependent variables.

When the variables inside the path model are latent variables whose measure is inferred by a set of observed indicators, path analysis is termed Structural Equation Modeling.

Since the 1970s, when two seminal papers were published approaching SEM from two different perspectives, until now, several authors have been interested in Structural Equation Models, approaching the model in very different ways and dealing with very different kinds of problems within. A non-exhaustive list of the main works in Structural Equation domain is given: [Bollen 1989, Hoyle 1995, Jöreskog & Sörbom 1979, Kaplan 2000, Lohmöller 1989, Chin 1998, Fornell & Bookstein 1982, Tenenhaus et al. 2005].

Essentially developed in a social domain, Structural Equation Models were first introduced by Jöreskog [1970] as confirmatory models to assess cause-effect relations among two or more set of variables, based on the maximum likelihood (ML) estimation method (SEM-ML). This method, also known as LISREL (*LInear Structural RELations*), has been for many years the only estimation method for SEM. The term LISREL was initially used for the software implementing the methodology [Jöreskog & Sörbom 1996]. However, it had such a rapid development that the methodology and the software have been associated to each other. Furthermore, it is important to notice that other estimation techniques rather than the maximum likelihood approach can be used to estimate Structural Equation Models, such as the Generalized Least Squares (GLS) or the Asymptotically distribution free (ADF). All these methods are usually referred to as LISREL-type estimation techniques. The factor common to all the LISREL-type

estimation techniques is that they are the so-called *covariance-based* methods. As a matter of fact, all these techniques aim at reproducing the sample covariance matrix of the manifest variables by means of the model parameters. The fundamental hypothesis underlining these approaches is that the implied covariance matrix of the manifest variables is a function of the model parameters.

In 1975, Wold [1975] finalized a *soft modeling* approach to the analysis of the relations among several blocks of variables observed on the same statistical units. This method, known as PLS approach to SEM (SEM-PLS) or as PLS Path Modeling (PLS-PM), is a distribution-free approach that was developed as a flexible technique for handling a huge amount of data characterized by missing values, strongly correlated variables and a small sample size as compared to the number of variables.
Several authors have compared the two approaches over the years; see, for example, Jöreskog & Wold [1982], Fornell & Bookstein [1982], Djkstra [1983]. The two approaches differ in the objectives of the analysis, the statistical assumptions, the estimation procedures and the related outputs.

New estimation techniques for Structural Equation Model estimation have been presented recently. Namely, in 2003 Al-Nasser proposed to extend Information theory knowledge at Structural Equation Models context via a new technique called Generalized Maximum Entropy

(GME) [Al-Nasser 2003].
More recently, instead, Hwang & Takane [2004] presented the Generalized Structured Component Analysis (GSCA). These new estimation techniques remain in the optic of PLS approach to SEM since no distributional assumptions are required. Moreover, the same problems characterizing the PLS-PM, namely the lack of a global optimizing criterion, have yet to be successfully solved.
All these approaches to Structural Equation Models have to be considered as *component-based* estimation techniques. As a matter of fact, in all these techniques the latent variable estimation plays a central role.

In this chapter first we introduce the SEM notation and the specification of the model, then the four estimation techniques (the LISREL-type approach to SEM, the PLS approach, the GSCA and the GME) will be discussed in the details. For each of these approaches the estimation procedure used, as well as the different indexes to assess the model quality, will be discussed.

3.2 SEM: the bases

Structural Equation Models adhere to certain common drawing conventions (cf. figure 3.1). Specifically, ellipses or circles represent the latent variables and rectangles or squares refer to the manifest variables. Arrows showing causations among the variables (either latent

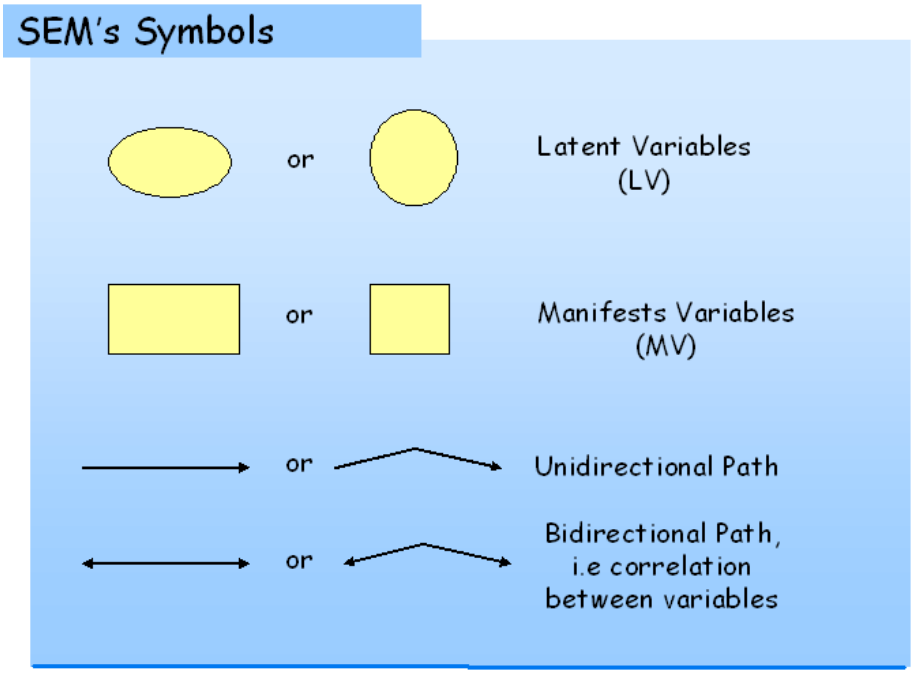

Figure 3.1: *Commonly used symbols in Structural Equation Models*

or manifest), and the direction of the array define the direction of the relation, i.e. variables receiving the array are to be considered as endogenous variables in the specific relationship.
Moreover, each Structural Equation Model is composed of two submodels: the measurement or outer model and the structural or inner model (cf. figure 3.2).

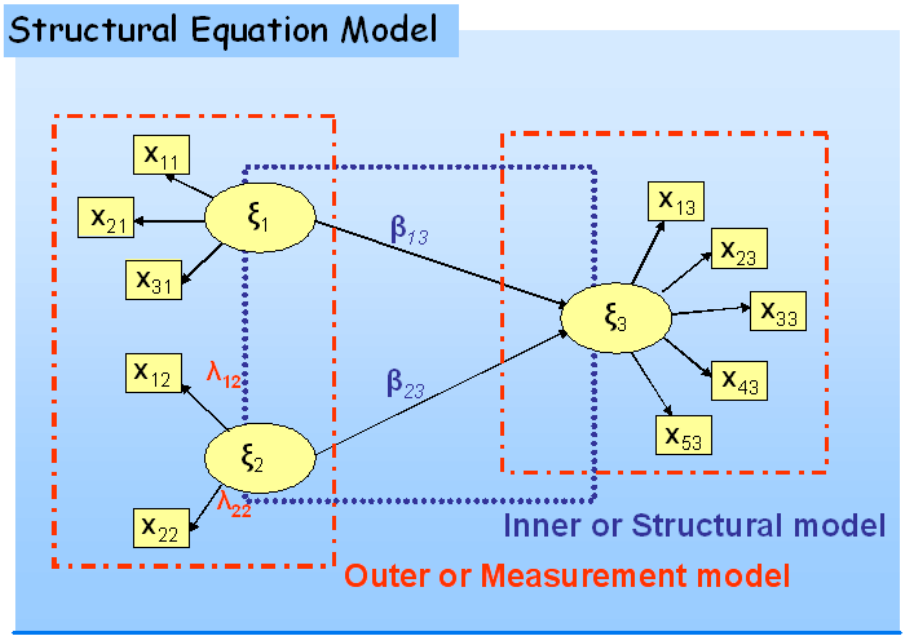

Figure 3.2: *Structural Equation Model representation*

The measurement model takes into account the way which the manifest variables are linked to the corresponding latent variable. Three different types of measurement model are available in Structural Equation Models: the *formative scheme*, the *reflective scheme* and the *MIMIC mode* (cf. figure 3.3). In a *reflective scheme* the set of manifest variables are assumed to measure a unique underlying concept. Each manifest variable reflects the corresponding latent variable and plays a role of endogenous variable in the block specific measurement model.

In the reflective measurement model, indicators linked to the same latent variables should covary: changes in one indicator imply changes in the others. Moreover, internal consistency has to be checked, i.e. each block needs to be unidimensional. It is important to notice that for the *reflective schemes*, the measurement model reproduces exactly the factor analysis model, in which each variable is function of the underlining factor.

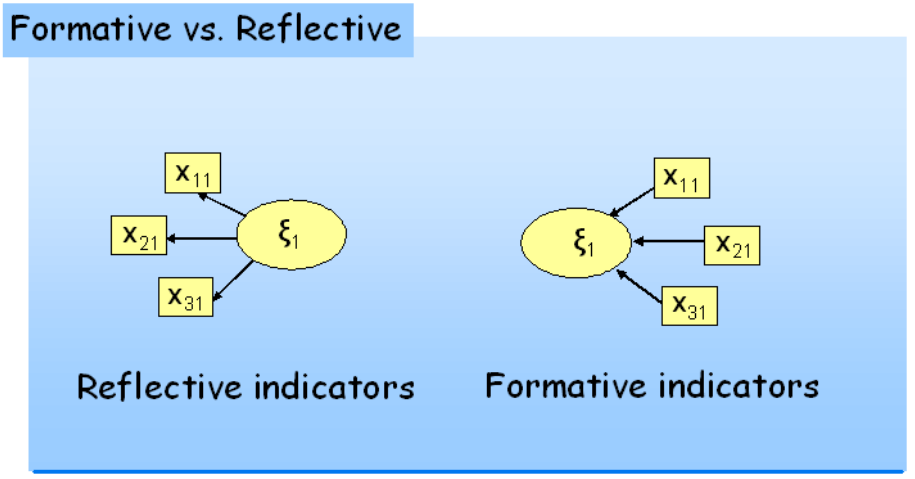

Figure 3.3: *Formative and Reflective Indicators*

In the *formative scheme*, each manifest variable or each sub-block of manifest variables represents different dimensions of the underlying concept. The latent variable is obtained as a linear combination of the corresponding manifest variables, thus each manifest variable is

an endogenous variable in the measurement model. These indicators need not to covary: changes in one indicator do not imply changes in the others. Measures of internal consistency are not necessary.
The *MIMIC scheme* is a mixture of the reflective and formative schemes. For each block, the same manifest variables are considered to be linked to the latent variable following a formative scheme and others following a reflective scheme.
Whatever scheme is used to build the measurement model, the parameters to be estimated are the so-called external or outer weights (w_{pq}) and the loadings (λ_{pq}).

The structural model, instead considers the relationship among the latent variables. Parameters to be estimated in the inner model are the path coefficients (β_{mj}), i.e. the regression coefficients linking the latent variables of each other, as well as the error terms for each regression in the structural model.

In SEM literature there is no agreement on the notation used to define latent variables and all the other parameters of the models. As a matter of fact, in *covariance-based* estimation techniques endogenous and exogenous latent variables, as well as the related manifest variables and parameters, are notated differently, while in *component-based* estimation techniques, especially in PLS-PM, all latent variables are notated in the same way regardless of their role in the regression-like relationships.

In this work, differently from the previous literature, we decide to use a unique notation for all the approaches to Structural Equation Models. In particular, we decide to use the same symbol to identify the latent variables regardless of whether they are endogenous or exogenous variables. Nevertheless, a prime will distinguish between endogenous ($^{(J)}$) and exogenous ($^{(M)}$) latent variables if necessary.
The same logic will be applied to all the elements of the model.
A list of the symbols used in this work will be found at the beginning of the thesis. Nevertheless, the following is a summarization of the main symbols used in this chapter:

- the generic manifest variable in the q-th block will be indicated by $\boldsymbol{x}_{pq}$, and $\boldsymbol{X}_q$ is the matrix containing all the manifest variables of the q-th block;

- the generic latent variable will be indicated by $\boldsymbol{\xi}_q$, and $\boldsymbol{\Xi}$ is the matrix containing all the latent variables;

- the generic outer weight linking the p-th manifest variables to the corresponding latent one will be w_{pq}, and $\boldsymbol{W}$ is all the matrix of the outer weights in the model;

- the generic loading associated to the p-th manifest variable in the q-th block will be indicated by λ_{pq}, and $\boldsymbol{\Lambda}$ is the matrix containing all the loadings in the model;

- the generic path coefficient linking the m-th exogenous latent variable to the j-th endogenous latent variable will be noted as

β_{mj}, and $\boldsymbol{B}$ is the matrix of all the path coefficients in the model.

- the generic measurement residuals associated to the generic manifest variable $\boldsymbol{x}_{pq}$ in a reflective scheme will be indicated by $\boldsymbol{\epsilon}_{pq}$, and the corresponding matrix containing all these measurement residuals will be $\boldsymbol{E}$;

- the generic measurement residuals associated to the generic manifest variable $\boldsymbol{x}_{pq}$ in a formative scheme will be indicated by $\boldsymbol{\delta}_{pq}$, and the corresponding matrix containing all these measurement residuals will be $\boldsymbol{\Delta}$;

- the generic structural residuals associated to the j-th endogenous latent variable, will be indicated by $\boldsymbol{\zeta}_j$, and the matrix containing all the structural residuals will be $\boldsymbol{H}$;

Taking the notation as listed above, Structural Equation Models can be described in more formal terms as composed of two different models: the measurement model and the structural model.

If differences among endogenous and exogenous latent variables are taken into account (like in the LISREL-type methods), the structural model describing the causations among the latent variables can be written for each unit in the model as:

$$\boldsymbol{\xi}_i^{(J)} = \boldsymbol{B}^{(J)}\boldsymbol{\xi}_i^{(J)} + \boldsymbol{B}^{(M)}\boldsymbol{\xi}_i^{(M)} + \boldsymbol{\zeta}_i \tag{3.1}$$

or as:

$$\left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right) \boldsymbol{\xi}_i^{(J)} = \boldsymbol{B}^{(M)} \boldsymbol{\xi}_i^{(M)} + \boldsymbol{\zeta}_i \tag{3.2}$$

where $\boldsymbol{\xi}^{(J)}$ are the endogenous latent variables in the model and $\boldsymbol{\xi}^{(M)}$ are the exogenous ones.
If no differences among endogenous and exogenous latent variables are taken into account (like in PLS-PM) equations 3.1 and 3.2 can be rewritten as:

$$\boldsymbol{\xi}_i = \boldsymbol{B}\boldsymbol{\xi}_i + \boldsymbol{\zeta}_i \tag{3.3}$$

Of course, the matrix $\boldsymbol{B}$ in equation 3.3 contains both the path coefficients ($\boldsymbol{B}^{(J)}$) interrelating the endogenous latent variables and the path coefficients ($\boldsymbol{B}^{(M)}$) relating the exogenous latent variables to the endogenous ones, i.e.:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}^{(J)} \\ \boldsymbol{B}^{(M)} \end{bmatrix} \tag{3.4}$$

Both the equations 3.1 and 3.3 perfectly parallel the multiples-multivariate regression. As a matter of fact, all the path coefficients, regardless of whether they refer to endogenous or exogenous latent variables, are to be considered equal to regression coefficients.

Different ways exist to formalize the measurement model according to the type of relations supposed to link the manifest variables to the

corresponding latent variable. In particular, as already said the relationship between latent and manifest variables could be formative or reflective. These two schemes suppose a different conception of the latent variable.
As a matter of fact, in a reflective scheme each manifest variable reflects the corresponding latent variable, thus it is related to the latent variable by a simple regression model:

$$\boldsymbol{x}_{pq} = \lambda_{pq}\boldsymbol{\xi}_q + \boldsymbol{\epsilon}_{pq} \tag{3.5}$$

The error term $\boldsymbol{\epsilon}_{pq}$ represents the imprecision in the measurement process. Furthermore, as the *reflective* block reflects the (unique) latent construct, it should be *unidimensional*. Hence, the set of manifest variables are assumed to measure the same unique underlying concept. There exist several tools for checking the unidimensionality of a block:

a) *Cronbach's alpha*: a block is considered unidimensional if this index is larger than 0.7

$$\alpha = \frac{\sum_{p\neq p'} cor(\boldsymbol{x}_{pq}, \boldsymbol{x}_{p'q})}{P_q + \sum_{p\neq p'} cor(\boldsymbol{x}_{pq}, \boldsymbol{x}_{p'q})} \times \frac{P_q}{P_q - 1} \tag{3.6}$$

where P_q is the number of manifest variables in the q-th block.

b) *Dillon-Goldstein's rho* (or *Jöreskog's*): a block is considered unidimensional if this index is larger than 0.7

$$\rho = \frac{(\sum_{p=1}^{P_q} \lambda_{pq})^2}{(\sum_{p=1}^{P_q} \lambda_{pq})^2 + \sum_{p=1}^{P_q}(1 - \lambda_{pq}^2)} \tag{3.7}$$

c) *Principal component analysis of a block*: a block is considered unidimensional if the first eigenvalue of the correlation matrix is higher than 1, while the others are smaller.

According to Chin [1998] the *Dillon-Goldstein's rho* is considered to be a better indicator of the unidimensionality of a block than the *Cronbach's alpha*.
In a formative scheme, instead, each latent variable is obtained as the linear combination of the manifest variables of the block, thus the measurement model can be expressed as:

$$\boldsymbol{\xi}_q = \sum_{p=1}^{P_q} w_{pq}\boldsymbol{x}_{pq} + \boldsymbol{\delta}_{pq} \tag{3.8}$$

The error term $\boldsymbol{\delta}_{pq}$ represents the fraction of the corresponding latent variable not accounted for by the manifest variables. Relationships among manifest and latent variables can be specified either in a series of equations, one for each observed (in reflective scheme) or latent (in formative scheme) variable, as done in equations 3.5 and 3.8, or in

matrix form, i.e. as:

$$\boldsymbol{X} = \boldsymbol{\Xi\Lambda} + \boldsymbol{E} \tag{3.9}$$

for the reflective scheme, where $\boldsymbol{\Xi}$ is the N by Q matrix containing the latent variable scores, $\boldsymbol{\Lambda}$ is the Q by P matrix containing the loadings and $\boldsymbol{E}$ is the N by P matrix containing the external residuals (or specification errors). And as:

$$\boldsymbol{\Xi} = \boldsymbol{XW} + \boldsymbol{\Delta} \tag{3.10}$$

for the formative scheme, where $\boldsymbol{W}$ is a P by Q matrix containing the external weights linking each manifest variable to the corresponding latent variable, and $\boldsymbol{\Delta}$ is the N by Q matrix containing the external errors associated to each latent variable.
Moreover, the equations 3.9 and 3.10 , as well as the equations 3.5 and 3.8, can be rewritten considering distinctly the measurement model concerning the endogenous blocks and the measurement model related to the exogenous blocks.

Both the measurement and the structural models, as well as the way to estimate the model coefficients (i.e. the path coefficients, the loadings and the external weights) in Structural Equation Models will be presented in detail according to the chosen estimation techniques, i.e. in LISREL-type models (cf. subsection 3.3.1), in PLS-PM (cf. subsection 3.4.1), in GSCA (cf. subsection 3.4.2) and in GME (cf. subsection

3.4.3).

The several estimation techniques listed above are grouped into two different approaches to Structural Equation Model estimation: the *covariance-based* approach to Structural Equation Models and the *component-based* approach to Structural Equation Models. The aim of *covariance-based* techniques is to estimate model parameters in such a way that the model becomes capable of "emulating" the analyzed sample covariance (or correlation) matrix. In *component-based* estimation methods, instead, a key role is played by the estimation of the latent variables in the model. In other words, the main aim of *component-based* methods is to provide an estimation of the latent variables in such a way that they are the most correlated with one another (according to the path diagram structure) and the most representative of each corresponding block of manifest variables. In the section 3.3, the *covariance-based* estimation techniques, such as the SEM-ML, will be shown in detail. In the section 3.4 the PLS Path Modeling and the other component-based estimation techniques will be discussed in depth.

3.3 Covariance-based Structural Equation Modeling

The aim of the *covariance-based* techniques is to reproduce the sample covariance matrix by the model. In other words, model coefficients

are estimated in such a way as to reproduce the sample covariance matrix. These techniques can be considered as a generalization of the Confirmatory Factor Analysis to the case of multi-tables data linked to one another. Therefore, in the *covariance-based* approach the measurement model is only considered as reflective. Formative indicators (i.e. formative manifest variables) are not allowed.
Different estimation techniques exist in a *covariance-based* approach to Structural Equation Models. The first methods proposed by Jöreskog [1970] to estimate Structural Equation Models is based on the maximum likelihood (ML) estimation method (SEM-ML). Since then, several estimation techniques have been applied in Structural Equation Models frameworks, keeping the aim of reproducing the sample covariance matrix. All these techniques are commonly referred to as LISREL-type techniques. As a matter of fact, for a long time the LISREL (*LInear Structural RELations*) software [Jöreskog & Sörbom 1996] was the main (and unique!) reference for Structural Equation Models in *covariance-based* framework. To be point that the word LISREL overlapped the more correct SEM-ML.
In reality, LISREL has to be used only to refer to the software, while LISREL-type methods have to be used to refer to the classical methods which allow us to estimate Structural Equation Models in a *covariance-based* framework, such as the SEM-ML. Moreover the expression "LISREL-type" has to be considered equivalent to *covariance-based* approach to Structural Equation Models.

3.3.1 The LISREL-type Structural Equation Models

Structural Equation Models were first introduced by Jöreskog [1970] as confirmatory models to assess cause-effect relations between two or more set of variables, based on the maximum likelihood (ML) estimation method (SEM-ML). This method, also known as LISREL (*LInear Structural RELations*), has been for many years the only estimation method for SEM. As already said, the term LISREL was initially used for the software implementing the SEM-ML. However, it had such a rapid development that the methodology and software have been associated to each other.
Furthermore, since the 70's many other estimation techniques besides the maximum likelihood approach, have been presented, such as Generalized Least Squares (GLS) or the Asymptotically Distribution Free (ADF).
Here, we first introduce the model specification of the LISREL-type Structural Equation Models, as well as the model identifiability, and all the other issues common to all estimation techniques. Then, the different estimation techniques will be discussed. Finally, the quality indexes used in a LISREL-type framework will be presented.

The LISREL-type model specification and other issues

Traditionally, in the LISREL-type Structural Equation Models the endogenous and the exogenous latent variables (as well as all the corresponding parameters in the models) are indicated differently. As a

matter of fact, in Jöreskog's notation the Greek letter $\boldsymbol{\eta}$ refers to endogenous latent variables while $\boldsymbol{\xi}$ refers to exogenous latent variables. Here, the author prefers to use the same symbol for all the latent variables and the corresponding parameters, regardless of whether they are endogenous or exogenous. Therefore, the symbol $\boldsymbol{\xi}$ refers to a generic latent variable.
Nevertheless, a prime will be used to distinguish between endogenous ($\boldsymbol{\xi}^{(J)}$) and exogenous ($\boldsymbol{\xi}^{(M)}$) latent variables as well as for all the other parameters in the model.

Let $\boldsymbol{S}$ be the sample (i.e. observed) covariance matrix associated with the manifest variables and $\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\Omega}}\right)$ be the predicted (i.e. implied) covariance matrix obtained by estimating model parameters ($\boldsymbol{\Omega}$). Since the *covariance-based* approaches to Structural Equation Models aim at reproducing the sample covariances matrix, then it is possible to identify a so-called *discrepancy function* F as some differences between the sample covariance matrix and the implied covariance matrix:

$$F = f\left(\boldsymbol{S}, \boldsymbol{\Sigma}\left(\boldsymbol{\Omega}\right)\right) \tag{3.11}$$

The *discrepancy function* F assumes different forms with regards to the estimation technique used to estimate the model parameters. Nevertheless, regardless of the estimation technique used, the *discrepancy function* F must have the following properties:

1. F is a scalar;

2. $F \geq 0$

3. $F = 0$ if and only if $\boldsymbol{\Sigma}(\boldsymbol{\Omega}) = \boldsymbol{S}$

4. F is continuous in $\boldsymbol{S}$ and in $\boldsymbol{\Sigma}(\boldsymbol{\Omega})$.

Minimizing the *discrepancy function* that satisfies these condition leads to consistent estimators of the model parameters ($\widehat{\boldsymbol{\Omega}}$) [Brown 1984]. The several estimation techniques available to estimate model parameters in the LISREL-type Structural Equation Models will be discussed afterward (cf. subsection 3.3.1). Here, we are interested in defining the implied covariance matrix using model parameters.

According to equation 3.1, the structural model for LISREL-type models is:

$$\boldsymbol{\xi}_i^{(J)} = \boldsymbol{B}^{(J)}\boldsymbol{\xi}_i^{(J)} + \boldsymbol{B}^{(M)}\boldsymbol{\xi}_i^{(M)} + \boldsymbol{\zeta}_i \tag{3.12}$$

Moving all the endogenous latent variables to the left side of the equation 3.12 yields an alternative form of the structural model equation:

$$\left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)\boldsymbol{\xi}_i^{(J)} = \boldsymbol{B}^{(M)}\boldsymbol{\xi}_i^{(M)} + \boldsymbol{\zeta}_i \tag{3.13}$$

and to:

$$\boldsymbol{\xi}_i^{(J)} = \left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)^{-1}\boldsymbol{B}^{(M)}\boldsymbol{\xi}_i^{(M)} + \boldsymbol{\zeta}_i \tag{3.14}$$

The equations 3.12, 3.13 and 3.14 are equivalent. They represent different forms to express the structural model in LISREL-type Structural Equation Models. This last reformulation of the structural model is useful for expressing the structural model in terms of covariances. As a matter of fact, equation 3.14 is similar to the Confirmatory Factor Analysis model. Therefore, the covariance matrix of the endogenous latent variables can be written as:

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{\xi}_i^{(J)}\boldsymbol{\xi}_i^{(J)^T}} = & \left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1}\boldsymbol{B}^{(M)}E\left(\boldsymbol{\xi}_i^{(M)}\boldsymbol{\xi}_i^{(M)^T}\right)\boldsymbol{B}^{(M)^T}\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1^T}+ \\ & +\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1}E\left(\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^T\right)\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1^T} \end{aligned} \tag{3.15}$$

Let $\boldsymbol{\Phi}$ be the covariance matrix of the exogenous latent variables, i.e.:

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_i^{(M)}\boldsymbol{\xi}_i^{(M)^T}} = \boldsymbol{\Phi} = E\left(\boldsymbol{\xi}_i^{(M)}\boldsymbol{\xi}_i^{(M)^T}\right) \tag{3.16}$$

and let $\boldsymbol{\Psi}$ be the covariance matrix of the structural residuals, i.e.:

$$\boldsymbol{\Psi} = E\left(\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^T\right) \tag{3.17}$$

then the equation 3.15 can be rewritten as:

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{\xi}_i^{(J)}\boldsymbol{\xi}_i^{(J)^T}} = & \left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1}\boldsymbol{B}^{(M)}\boldsymbol{\Phi}\boldsymbol{B}^{(M)^T}\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1^T}+ \\ & +\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1}\boldsymbol{\Psi}\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1^T} \end{aligned} \tag{3.18}$$

while the covariance matrix associated to the exogenous latent variable is expressed in equation 3.16.

In LISREL-type Structural Equation Models the measurement model is only reflective. No formative indicators are allowed in *covariance-based* approaches. Moreover, two different measurement models are identified, one for the manifest variables related to the exogenous latent variables and the other for the manifest variables related to the endogenous latent variables. For these reasons, and to recall the equation 3.5, the measurement models for LISREL-type Structural Equation Models can be written, for each unit in the model, as:

$$\boldsymbol{x}_i^{(M)} = \boldsymbol{\Lambda}^{(M)}\boldsymbol{\xi}_i^{(M)} + \boldsymbol{\epsilon}_i^{(M)} \tag{3.19}$$

for the exogenous blocks, and as:

$$\boldsymbol{x}_i^{(J)} = \boldsymbol{\Lambda}^{(J)}\boldsymbol{\xi}_i^{(J)} + \boldsymbol{\epsilon}_i^{(J)} \tag{3.20}$$

for the endogenous blocks.
The two lambda matrices ($\boldsymbol{\Lambda}^{(M)}$) and $\boldsymbol{\Lambda}^{(J)}$ contain the external loadings, while the vectors $\boldsymbol{\epsilon}_i^{(M)}$ and $\boldsymbol{\epsilon}_i^{(J)}$ are the residuals associated to the manifest variables.
As in Confirmatory Factor Analysis, the implied covariance matrix associated to the manifest variables is obtained for the exogenous block

as:

$$\boldsymbol{\Sigma}_{X^{(M)}X^{(M)^T}} = \boldsymbol{\Lambda}^{(M)} E\left(\boldsymbol{\xi}^{(M)}\boldsymbol{\xi}^{(M)^T}\right)\boldsymbol{\Lambda}^{(M)^T} + E\left(\boldsymbol{\epsilon}^{(M)}\boldsymbol{\epsilon}^{(M)^T}\right) \quad (3.21)$$

Using $\boldsymbol{\Theta}^{(M)}$ to represent the residual covariance matrix, i.e.:

$$\boldsymbol{\Theta}^{(M)} = E\left(\boldsymbol{\epsilon}^{(M)}\boldsymbol{\epsilon}^{(M)^T}\right) \quad (3.22)$$

and being $\boldsymbol{\Phi}^{(M)}$ the covariance matrix of the exogenous latent variables, then the equation 3.21 can be rewritten as:

$$\boldsymbol{\Sigma}_{X^{(M)}X^{(M)^T}} = \boldsymbol{\Lambda}^{(M)}\boldsymbol{\Phi}\boldsymbol{\Lambda}^{(M)^T} + \boldsymbol{\Theta}^{(M)} \quad (3.23)$$

For the manifest variables of the endogenous blocks, instead, the covariance matrix is:

$$\boldsymbol{\Sigma}_{X^{(J)}X^{(J)^T}} = \boldsymbol{\Lambda}^{(J)} E\left(\boldsymbol{\xi}^{(J)}\boldsymbol{\xi}^{(J)^T}\right)\boldsymbol{\Lambda}^{(J)^T} + \boldsymbol{\Theta}^{(J)} \quad (3.24)$$

where the expected value of $\boldsymbol{\xi}^{(J)}\boldsymbol{\xi}^{(J)^T}$ cannot be immediately expressed, being function of the structural model parameters (c.f. equation 3.18). Replace equation 3.18 in equation 3.24, yields to:

$$\begin{aligned}\boldsymbol{\Sigma}_{X^{(J)}X^{(J)^T}} &= \boldsymbol{\Lambda}^{(J)}\left[\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1}\boldsymbol{B}^{(M)}\boldsymbol{\Phi}\boldsymbol{B}^{(M)^T}\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1^T}\right]\boldsymbol{\Lambda}^{(J)^T} + \\ &+\boldsymbol{\Lambda}^{(J)}\left[\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1}\boldsymbol{\Psi}\left(\boldsymbol{I}-\boldsymbol{B}^{(J)}\right)^{-1^T}\right]\boldsymbol{\Lambda}^{(J)^T} + \boldsymbol{\Theta}^{(J)} \quad (3.25)\end{aligned}$$

that can be rewritten in a more compact way as:

$$\begin{aligned}\boldsymbol{\Sigma}_{X^{(J)}X^{(J)^T}} = \; & \boldsymbol{\Lambda}^{(J)} \left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)^{-1} \left(\boldsymbol{B}^{(M)} \boldsymbol{\Phi} \boldsymbol{B}^{(M)^T} + \boldsymbol{\Psi}\right) \left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)^{-1^T} \boldsymbol{\Lambda}^{(J)'} + \\ & + \boldsymbol{\Theta}^{(J)} \end{aligned} \tag{3.26}$$

Assuming that the vector $\boldsymbol{\xi}_i^{(M)}$, is uncorrelated with all the errors in the model ($\boldsymbol{\zeta}_i$, $\boldsymbol{\epsilon}_i^{(M)}$ and $\boldsymbol{\epsilon}_i^{(J)}$), i.e.:

$$E\left(\boldsymbol{\xi}_i^{(M)}, \boldsymbol{\epsilon}_i^{(M)^T}\right) = E\left(\boldsymbol{\epsilon}_i^{(M)}, \boldsymbol{\xi}_i^{(M)^T}\right) = 0 \tag{3.27}$$

$$E\left(\boldsymbol{\xi}_i^{(M)}, \boldsymbol{\epsilon}_i^{(J)^T}\right) = E\left(\boldsymbol{\epsilon}_i^{(J)}, \boldsymbol{\xi}_i^{(M)^T}\right) = 0 \tag{3.28}$$

$$E\left(\boldsymbol{\xi}_i^{(M)}, \boldsymbol{\zeta}_i^{T}\right) = E\left(\boldsymbol{\zeta}_i, \boldsymbol{\xi}_i^{(M)^T}\right) = 0 \tag{3.29}$$

and assuming that the errors are uncorrelated with one another (i.e. that the error covariance matrices $\boldsymbol{\Psi}$, $\boldsymbol{\Theta}^{(J)}$ and $\boldsymbol{\Theta}^{(J)}$ are diagonal matrices) then, the covariance matrices between the endogenous and the exogenous manifest variables are:

$$\boldsymbol{\Sigma}_{X^{(M)}X^{(J)}} = \boldsymbol{\Lambda}^{(M)} \boldsymbol{\Phi} \boldsymbol{B}^{(M)^T} \left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)^{-1^T} \boldsymbol{\Lambda}^{(J)^T} \tag{3.30}$$

and

$$\boldsymbol{\Sigma}_{X^{(J)}X^{(M)}} = \boldsymbol{\Lambda}^{(J)} \left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)^{-1} \boldsymbol{B}^{(M)} \boldsymbol{\Phi}^T \boldsymbol{\Lambda}^{(M)^T} \tag{3.31}$$

with $\boldsymbol{\Sigma}_{X^{(M)}X^{(J)}} = \left(\boldsymbol{\Sigma}_{X^{(J)}X^{(M)}}\right)^T$.

The decomposition of the implied covariance matrix among the man-

ifest variables is:

$$\boldsymbol{\Sigma}\left(\boldsymbol{\Omega}\right) = \begin{bmatrix} \boldsymbol{\Sigma}_{X^{(M)}X^{(M)^T}} & \boldsymbol{\Sigma}_{X^{(M)}X^{(J)}} \\ \boldsymbol{\Sigma}_{X^{(J)}X^{(M)}} & \boldsymbol{\Sigma}_{X^{(J)}X^{(J)}} \end{bmatrix} \tag{3.32}$$

where the matrix elements are obtained according to the equations 3.23, 3.26, 3.30, 3.31.

Several techniques has been developed to estimate the model parameters (i.e.: $\boldsymbol{\Lambda}^{(M)}$, $\boldsymbol{\Lambda}^{(J)}$, $\boldsymbol{B}^{(M)}$, $\boldsymbol{B}^{(J)}$, $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Theta}^{(M)}$, $\boldsymbol{\Theta}^{(J)}$) in order to obtain the implied covariance matrix according to equation 3.32. These will be discussed afterward in this section.

Regardless of the estimation method used, the model needs to be *identifiable* in order to be estimated. A model is identifiable if the covariance matrix can be uniquely decomposed in function of the model parameters. This entail that the number of covariances among the manifest variables must be larger than the number of parameters to be estimated. Therefore, the degrees of freedom (df) of a model are obtained as the difference between the number of available covariances and the number of model parameters:

$$df = \frac{P\left(P+1\right)}{2} - t \tag{3.33}$$

where P is the number of manifest variables in the model and t is the number of parameters to be estimated.

A statistical model is perfectly identified if the information available implies that there is one best value for each parameter in the model. The perfectly identified models are models showing 0 degrees of freedom, that is the reason for which they are called *saturated model*. Moreover, perfectly identified models yield a trivially perfect fit, making the test of fit uninteresting. On the contrary, a model is overidentified if there are more knowns than unknowns. Overidentified models may not fit well and this is their interesting feature. They are characterized by positive degrees of freedom.
Nevertheless, having a positive degrees of freedom is only a necessary condition for a model to be identified, not a sufficient one. It is for this reason that several methods have been proposed to determine model identification. For further information on model identification in Structural Equation framework please refer to Bollen [1989].

The estimation techniques for the LISREL-type model

Several estimation techniques have been applied to the LISREL-type Structural Equation Models. For all of these techniques the aim is to minimize the *discrepancy function* F, in such a way as to obtain an implied covariance matrix ($\widehat{\mathbf{\Sigma}}$), function of the estimated parameters, that is as close as possible to the sample covariance matrix. Where $\widehat{\mathbf{\Sigma}}$ stands for the implied covariance matrix for a specific estimate of the models parameters $\widehat{\mathbf{\Omega}}$, i.e. :

$$\widehat{\mathbf{\Sigma}} = \mathbf{\Sigma}\left(\widehat{\mathbf{\Omega}}\right) \tag{3.34}$$

Once the *discrepancy function* defined as:

$$F = \boldsymbol{S} - \widehat{\boldsymbol{\Sigma}} \tag{3.35}$$

the various estimation techniques are different as regards the form of the *discrepancy function* used.

The most widely used *discrepancy function* for LISREL-type Structural Equation Models is the Maximum Likelihood (ML) function. Following this approach, the *discrepancy function* to be minimized is:

$$F_{ML} = \ln \left|\widehat{\boldsymbol{\Sigma}}\right| - \ln |\boldsymbol{S}| + tr\left(\boldsymbol{S}\widehat{\boldsymbol{\Sigma}}^{-1}\right) - P \tag{3.36}$$

where tr is the trace of a matrix, i.e. the sum of the diagonal elements of a matrix. As $\widehat{\boldsymbol{\Sigma}}$ converge to $\boldsymbol{S}$, $\widehat{\boldsymbol{\Sigma}}$ inverse will approximate $\boldsymbol{S}$ inverse and $\boldsymbol{S}\widehat{\boldsymbol{\Sigma}}^{-1}$ will approximate an identity matrix ($\boldsymbol{S}\boldsymbol{S}^{-1}$). Because an identity matrix has ones on the diagonal, the trace of $\boldsymbol{S}\widehat{\boldsymbol{\Sigma}}^{-1}$ will be equal to the matrix size, i.e. to P. Thus, if the model is able to reproduce exactly the sample covariance matrix, then the F expressed in equation 3.36 will be equal to zero.
The use of the *discrepancy function* defined in equation 3.36 is based on the assumption that the manifest variables have a multinormal distribution or that the sample covariance matrix $\boldsymbol{S}$ has a Wishart distribution. Moreover, we have to assume that both the implied and the sample covariance matrices are positive-definite, which means that they are non-singular.

Since the *discrepancy function* expressed in equation 3.36 is usually a complicated function, numerical iteration algorithms, such as the EM algorithm [Dempster et al. 1977] (cf. subsection 2.4.2), are used to find zeros of the *discrepancy function*.
ML estimators are widely used thanks to their several asymptotic proprieties. As a matter of fact, for large samples, ML estimators are asymptotic unbiased, consistent and asymptotically efficient. Moreover, the distribution of a ML estimator approximates a normal distribution as sample size increases. This implies that, for large samples, the ratio of the estimated parameter and its standard errors should approximate a standardized normal distribution.
To conclude, F_{ML} is usually scale invariant and scale free. The scale invariance properties implies that the value of the *discrepancy function* is the same using the correlation or the covariance matrices (or more generally it is the same for any change of scale). The scale freeness, instead, implies that changing the measurement units of one or more of the observed variables (or more in general, applying a linear transformation on the manifest variables) leads to obtaining new estimates of the model parameters that are simply related to the ones obtained for the non-transformed manifest variables.

One application of the OLS (Ordinary Least Squares) principle in a Structural Equation Model framework is the use of an Unweighted Least Squares (ULS) estimation procedure. As OLS estimation technique allows us to obtain model parameters by minimizing the sum

of squares of the residual term in a regression model, ULS allows us to obtain model parameters by minimizing one-half of the sum of the squares of each element in the residual matrix. The main difference between the two approaches is that in a OLS optic, differences are computed for individual observations, while ULS focuses on covariance matrices. As a matter of fact, the *discrepancy function* to be minimized in a ULS optic is defined as:

$$F_{ULS} = \frac{1}{2} tr \left[\left(\boldsymbol{S} - \widehat{\boldsymbol{\Sigma}} \right)^2 \right] \tag{3.37}$$

Finding zeros of the function expressed in equation 3.37 could be a difficult task. Once again, iterative numerical techniques may help to solve the minimizing problem involving the function 3.37.
Even if, in the case of a big sample size ULS often provide estimates close to the ML's ones, it does not lead to the asymptotically most efficient estimators for the model parameters (because the ML estimators are more efficients). Nevertheless, in the case of big sample size ULS provides consistent estimators without the need to make assumptions on the manifest variables distribution. To conclude, differently from F_{ML}, the ULS *discrepancy function* is not scale invariant, nor is it scale free. Using covariance matrices or correlation matrices will lead to different F_{ULS} values that are not linked to each other. In other words, it is not possible to obtain the parameters estimated by using the covariance matrix from the one obtained by using the correlation matrix (or vice versa).

The ULS estimation technique implicitly weights all the elements of the *discrepancy function* as if they have the same variances and covariances. This is exactly the same problem arising in classical regression problems when OLS estimators are applied in the case of heteroschedasticity of the errors. And, exactly as in a regression framework, this limitation is overcome by using Generalized Least Squares (GLS) estimators, i.e. by adding a weight matrix ($\boldsymbol{D}$) to the ULS *discrepancy function*. A general formulation of the GLS *discrepancy function* is:

$$F_{GLS} = \frac{1}{2} tr \left[\boldsymbol{D}^{-1} \left(\boldsymbol{S} - \widehat{\boldsymbol{\Sigma}} \right)^2 \right] \tag{3.38}$$

where the weights matrix for the residuals ($\boldsymbol{D}$) is either a random matrix that converges in probability to a positive definite matrix as $N \to \infty$, or it is a positive definite matrix of constants. It is easy to notice that ULS is a particular case of GLS, when the weights matrix is equal to the identity matrix.

Usually, estimators obtained by means of GLS are consistent estimators and their distributions approximate normal distributions as sample size increases. Nevertheless, these proprieties depend on the choice of $\boldsymbol{D}$. As a matter of fact, using $\boldsymbol{D} = \boldsymbol{I}$ leads to obtaining ULS estimators that, as already said do not have these proprieties. In order to assure proprieties to the GLS estimators, the weights matrix have to be chosen under two assumptions on the element of the sample covariance matrix:

1. the elements of $\boldsymbol{S}$ are unbiased estimators of the corresponding variance/covariance, i.e. $E(s_{ij}) = \sigma_{ij}$.

2. the elements of $\boldsymbol{S}$ are asymptotically multinormal distributed with means equals to σ_{ij}, and asymptotic covariance between s_{ij} and s_{gh} equal to $N^{-1}(\sigma_{ig}\sigma_{jh} + \sigma_{ih}\sigma_{jg})$.

This last assumption requires that the units are *i.i.d.* and that the fourth-order moments of the manifest variables exist. Moreover, in order to obtain asymptotic covariance equal to $N^{-1}(\sigma_{ig}\sigma_{jh} + \sigma_{ih}\sigma_{jg})$, the manifest variables need to be multinormallly distributed, or at least following other distributions without excessive kurtosis.
Under these assumptions, and choosing a weights matrix obtained so that $\boldsymbol{D}^{-1} = c\boldsymbol{\Sigma}^{-1}$, the GLS estimators have an asymptotic multinormal distribution and are asymptotically efficient. Since no information about the population covariance matrix ($\boldsymbol{\Sigma}$) is available, the sample covariance matrix is the most used consistent estimator of $\boldsymbol{\Sigma}$. Moreover, usually $c = 1$. This leads to using $\boldsymbol{S}^{-1}$ as weight matrix. Thus, the GLS *discrepancy function* expressed by equation 3.38 can be rewritten as:

$$F_{GLS} = \frac{1}{2} tr\left[\boldsymbol{S}^{-1}\left(\boldsymbol{S} - \widehat{\boldsymbol{\Sigma}}\right)^2\right] \tag{3.39}$$

To conclude, GLS estimators are scale invariants and scale free. Nevertheless, they require more assumption than ML ones. Among them, the most restrictive assumption is the one on the asymptotic covariance of the elements of $\boldsymbol{S}$. As a matter of fact, as underlined

by Bollen [1989], if the manifest variables have very "fat" or "thin" tails, the asymptotic covariance between s_{ij} and s_{gh} may deviate from $N^{-1}(\sigma_{ig}\sigma_{jh} + \sigma_{ih}\sigma_{jg})$.

Other standard estimation techniques can be used to estimate model parameters in the LISREL-type Structural Equation Models, such as the Asymptotically Distribution Free estimation technique. A detailed discussion on all these techniques goes further than the aim of this work. Nevertheless, the author wishes to discuss a new estimation technique recently proposed by McDonald [1996].
This method is based on the ULS estimation technique. Nevertheless, McDonald imposes as zero the measurement error covariance matrices, i.e. $\boldsymbol{\Theta}^{(M)} = 0$ and $\boldsymbol{\Theta}^{(J)} = 0$

$$F_{McDonalds} = \left\| \boldsymbol{S} - \widehat{\boldsymbol{\Sigma}} \right\|^2 \tag{3.40}$$

Being a generalization of the Principal Component Analysis, this technique can be used even if the sample covariance matrix $\boldsymbol{S}$ is not of full rank and the sample size is small.

The Quality indexes

Since in *covariance-based* approaches the aim is to reproduce the sample covariance matrix, the goodness of fit is related to the ability of the model to reproduce the sample covariance matrix. As a matter of fact, the differences between the implied covariance matrix computed by the model ($\widehat{\boldsymbol{\Sigma}}$) and the sample covariance matrix ($\boldsymbol{S}$) can be con-

sidered as a measure of fit.

Let F be the computed minimum value of the fit function (i.e. the discrepancy function) obtained by means of one of the estimation techniques discussed above, e.g. ML (see equation 3.36) or GLS (see equation 3.38).

Overall fit is assessed by a *chi-square* goodness of fit test based on the F value:

$$\chi^2 = (N-1)\,F \tag{3.41}$$

Under the null hypothesis of perfecting fitting (i.e. $F = 0$), the χ^2 expressed in equation 3.41 follows a *chi-square* distribution with degrees of freedom (*df*) equal to the difference between the number of covariates and the number of parameters in the model. The null hypothesis is rejected (i.e. the model is considered not fit to the data) when the p-value associated to the tested model is smaller than a certain significance value, usually 0.05.

If for perfectly fitting models sample size has no effect on the χ^2 statistic, for imperfectly fitting models, the higher the sample size is, the higher the χ^2 value is, regardless of the model fit (i.e. the F value). Moreover, since the degrees of freedom remain the same regardless of the sample size then the reference *chi-square* distribution against which the χ^2 is judged for significance also remains the same. This implies that, with a very large sample size, there is a spurious tendency to obtain large values of χ^2, which tend to be associated to small p-values. Consequently, for very large samples there will be an

artificial tendency to reject the model, even if the model fits the data well (i.e. even if the F value is close to zero). On the contrary, very small samples are more easily associated to small χ^2 values, and are more easily accepted as good models.

Other indexes based on the discrepancy between the implied covariance matrix and the sample covariance matrix have been proposed to overcome this problem. Among them, the *Root Mean Residual* (RMR) that is simply the square root of the mean of the squared discrepancy between all the elements of the implied covariances matrix and the sample covariances matrix:

$$RMR = \sqrt{2 \sum_{p=1}^{P} \sum_{r=1}^{p} \frac{\left(s_{pr} - \widehat{\sigma}_{pr}\right)^2}{P\left(P+1\right)}} \tag{3.42}$$

(where, P is the total number of manifest variables, s_{pr} is the generic element of the sample covariance matrix, and $\widehat{\sigma}_{pr}$ is the generic element of the implied covariance matrix) and the *Goodness of Fit Index* (GFI) expressed as:

$$GFI_{ML} = 1 - \frac{tr\left[\left(\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{S} - \boldsymbol{I}\right)^2\right]}{tr\left[\left(\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{S}\right)^2\right]} \tag{3.43}$$

if the Maximum Likelihood estimators are used, and as:

$$GFI_{GLS} = 1 - \frac{tr\left[\left(\boldsymbol{I} - \widehat{\boldsymbol{\Sigma}}\boldsymbol{S}^{-1}\right)^2\right]}{P} \tag{3.44}$$

and

$$GFI_{ULS} = 1 - \frac{tr\left[\left(\boldsymbol{S} - \widehat{\boldsymbol{\Sigma}}\right)^2\right]}{tr\left(\boldsymbol{S}^2\right)} \tag{3.45}$$

if the model is fitted respectively by GLS or ULS. In all the cases $\boldsymbol{I}$ is an identity matrix.

The GFI assess the relative amount of the variances and covariances jointly accounted for by the model (similar to the R^2 in a regression analysis). The GFI was initially devised by Jöreskog & Sörbom [1996] for ML, GLS and ULS estimation. Since then, it has been generalized to other estimation criteria.

Moreover, the GFI does not take into account the complexity of the model. That is why, Jöreskog & Sörbom [1996] proposed also a modified version of the GFI considering the number of parameters in the model: the *Adjusted Goodness of Fit Index* ($AGFI$):

Like the GFI, also the $AGFI$ formulation changes according to the estimation technique used. In particular, the $AGFI$ changes since the GFI uses changes:

$$AGFI = 1 - \left[\frac{P\,(p+1)}{2df}\right][1 - GFI] \tag{3.46}$$

where *df* are the degrees of freedom of the model and GFI is the Goodness of Fit Index computed according to the estimation technique used.
Both the GFI and the $AGFI$ are bounded between 0 and 1. Values close to 1 are usually associated with well-fitting models. Moreover, the calculation of both the GFI and the $AGFI$ is not affected by the sample size. Nevertheless, simulation study performed by Anderson & Gerbing [1984] suggest that the means of the sampling distribution of GFI_{ML} and $AGFI_{ML}$ tend to increase as sample size increases, while they tend to decrease as the number of manifest variables in each block or the number of latent variables increases.

Tests presented above assume that the closer the implied covariance matrix is to the sample covariance matrix, the better the model fits. Nevertheless, the null hypothesis of perfecting fit is too "restrictive". As a matter of fact, a model is used to analyze certain phenomena if it should represent a useful simplification and approximation of the reality rather than a precise replica of it. Following this idea, the null hypothesis of perfecting fit is still not interesting. A weaker hypothesis to be tested can be detected. That is why, other tests to assess model quality have been presented, the so-called *tests of close-fit*.
Several fit indexes comparing the performance of the model to be tested with a so-called *null model* have been presented. The *null model* represents the extreme case of no relationships among the manifest variables, so a less restrictive null hypothesis than the perfect fit one.

In other words, all the manifest variables in the model are supposed to be independent of one another. Only the elements on the diagonal of the implied covariance matrix, i.e. the variable variances, are different from zero. The *null model* is the "worst fitting" model. Comparing the fit function value obtained for the *null model* (F_0) to the one obtained for the proposed model or the χ^2 value obtained for the *null model* (χ_0^2) to the one obtained for the proposed model, allows us to assess model quality.

The first index comparing the tested model performance to the *null model* to be proposed was the *Normed Fit Index* (NFI) by Bentler & Bonett [1980], defined as:

$$\begin{aligned} NFI &= \frac{F_0 - F}{F_0} \\ &\text{or} \\ NFI &= \frac{\chi_0^2 - \chi^2}{\chi_0^2} \end{aligned} \tag{3.47}$$

In its original version this index allows us to compare the performance of two alternative models rather than the performance of one model against the *null model*, i.e. another F value can be used instead of the F_0. This is an index bounded between 0 and 1. Bentler and Bonett suggested accepting the model if NFI is greater than 0.90.

Nevertheless, the NFI does not take into account the complexity of the model. That is why Bentler and Bonett also proposed a modified

version of the NFI: the *Non Normed Fit Index* ($NNFI$).

$$\begin{aligned} NNFI = & \quad \frac{F_0/df_0 - F/df}{F_0/df_0 - 1/(n-1)} \\ & \text{or} \\ NNFI = & \quad \frac{\chi_0^2/df_0 - \chi^2/df}{\chi_0^2/df_0 - 1} \end{aligned} \tag{3.48}$$

The $NNFI$ is a simple variant of the NFI that takes into account the degrees of freedom of the tested model. Once again it could be used to test two alternative models rather than the proposed model against the *null model*. If one of the two tested models is the *null model* as expressed in equation 3.48, then the $NNFI$ is exactly the Tucker-Lewis index (TLI) [Tucker & Lewis 1973].
The $NNFI$ is robust across sample size changes [Hu & Bentler 1995, Marsh, Balla & McDonald 1988], but it is not bounded between 0 and 1.
Another index based on the comparison between the proposed model and the *null model* is the *Incremental Fit Index* (IFI) by Bollen [1989]:

$$IFI = \frac{\chi_0^2 - \chi^2}{\chi_0^2 - df} \tag{3.49}$$

where df (the degrees of freedom of the proposed model) is the expected value of the χ^2 obtained for the proposed model.

Further, other indexes have been developed to handle the case of non-central *chi-square* distribution. Among them both the *Bentler Fit In-*

dex (BFI) proposed by Bentler [1990], and the *Relative Noncentrality Index* (RNI) by McDonald & Marsh [247–255]. These two indexes are the same, and they are expressed as:

$$RNI \text{ or } BFI = \frac{[\chi_0^2 - df_0] - [\chi^2 - df]}{\chi_0^2 - df_0} \tag{3.50}$$

This index is not bounded between 0 and 1. A modified version of the BFI bounded between 0 and 1 was proposed by Bentler in 1990: the *Bentler Comparative Fit Index* (CFI).
All these indexes can be used to test nested models, as is the case for the *null model* against the proposed model. The implicit answer to the question *"how well does the model do compared with several or a unique alternative model with the same data?"* is obtained by comparing the results obtained for the model with the ones obtained for the (nested) alternative models.

Models that differ as regards the relationships in the model cannot be compared using the indexes discussed above. As a matter of fact, in the case of non-nested models the simple way to compare the models' performance is to compare absolute fit indexes such as the χ^2 value or the GFI. Nevertheless, direct comparison is complicated because no direct statistical comparison is possible. For such models, other fit indexes based on the Information Theory could be used. The Information based indexes do not have ideal values to attain but provide a relative ordering of different models estimated on the same sample.
Among them the most popular are the LISREL-type SEM version of

the Akaike Information Criteria (AIC) (cf. subsection 2.4.3) as presented by Jöreskog:

$$AIC_{(Jöreskog)} = \chi^2 - 2df \tag{3.51}$$

and as presented by Tanaka:

$$AIC_{(Tanaka)} = \chi^2 + 2fp \tag{3.52}$$

where fp is the number of free parameters in the model.
Other fit indexes based on the information criteria and modified in order to be applied to the LISREL-type Structural Equations Models are the modified version of the AIC, i.e.:

$$CAIC = \chi^2 - \ln(1 + N)\, df \tag{3.53}$$

and the expected cross validation index ($ECVI$) by Browne & Cudeck [1993]:

$$ECVI = \frac{\chi^2}{N} + 2\frac{fp}{N} \tag{3.54}$$

where, as above, fp is the number of free parameters in the model.

The following tests, instead, are based on the discrepancy between the implied covariance matrix and the population covariance matrix ($\mathbf{\Sigma}$).
The Root Mean Square Error of Approximation (RMSEA) by Steiger

& Lind [1980] is defined as:

$$RMSEA = \sqrt{\frac{F_0}{df}} \tag{3.55}$$

where F_0 is the value assumed by the fit function for $\boldsymbol{S} = \boldsymbol{\Sigma}$, i.e.:

$$F_0 = \boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}, \tag{3.56}$$

so assuming that we are comparing the implied covariance matrix with the population covariance matrix, and *df* are the degrees of freedom as defined above (i.e. as the differences between the number of covariates and the number of parameters in the model).
Since no information about the $\boldsymbol{\Sigma}$ value is available, the RMSEA expressed by equation 3.55 is estimated using the sample covariance matrix as:

$$RMSEA_{estimated} = \sqrt{\frac{F}{df} - \frac{1}{N-1}} \tag{3.57}$$

where F as usual is the obtained value of the fit function:

$$F = \boldsymbol{S} - \widehat{\boldsymbol{\Sigma}}$$

3.4 Component-based Structural Equation Modeling

In this section the *component-based* estimation techniques will be discussed in detail. As already said, the aim of *component-based* methods is to provide an estimate of the latent variables in such a way that they are the most correlated with one another (according to the path diagram structure) and the most representative of each corresponding block of manifest variables. These techniques are to be considered as a generalization of Principal Component Analysis to multi-tables data linked to one another. In the *component-based* approaches the measurement model can be both reflective and formative.
The most recognized estimation technique among the *component based* methods is the PLS Path Modeling [Wold 1975, Tenenhaus et al. 2005] (cf. subsection 3.4.1). More recently, other *component based* techniques have been presented. Namely, the Generalized Maximum Entropy (GME) by Al-Nasser [2003] (cf. subsection 3.4.3) and the Generalized Structured Component Analysis (GSCA) by Hwang & Takane [2004] (cf. subsection 3.4.2).

3.4.1 The PLS Path Modeling

The PLS (Partial Least Squares) approach to Structural Equation Models, also known as PLS Path Modeling (PLS-PM) has been proposed as an alternative estimation procedure to the LISREL-type approach to Structural Equation Models (cf. section 3.3). In Wold's

[1975] seminal paper the main principles of *partial least squares* for the *principal component analysis* [Wold 1966], were extended to situations with more blocks of variables. The first presentation of the PLS Path Modeling is given in Wold [1979], and the algorithm is described in Wold [1982] and in Wold [1985]. An extensive review on PLS approach to Structural Equation Models is given in Chin [1998] and in Tenenhaus et al. [2005].

As all the *component-based* estimation techniques, also PLS Path Modeling is an estimation method based on components. It is an iterative algorithm that separately estimates the several blocks of the measurement model and then, in a second step, estimates the structural model coefficients. Differently from LISREL-type estimation techniques, PLS Path Modeling aims at explaining at best the residual variance of the latent variables and, potentially, also of the manifest variables in any regression run in the model [Fornell & Bookstein 1982]. That is why PLS Path Modeling is considered more an explorative approach than a confermative one: it does not aim at reproducing the sample covariance matrix.

Moreover, differently from LISREL-type estimation techniques, the PLS Path Modeling is a completely free approach that does not require any distributional assumptions. For this reason the PLS-PM is considered as a *soft modeling* approach: no strong assumptions (with respect to the distributions, the sample size and the measurement scale) have to be made. Nevertheless, PLS-PM does not seem to optimize a well identified global scalar function. Until now convergence is

proved only for path diagram with one or two blocks [Lyttkens, Areskoug & Wold 1975]. Researches on this topic are on going.
Further, PLS Path Modeling provides a direct estimate of the latent variable scores.

The Algorithm

PLS Path Modeling aims to estimate the relationships among Q blocks of variables, which are expression of unobservable constructs. Specifically, PLS-PM estimates through a system of interdependent equations based on simple and multiple regressions, the network of relations among the manifest variables and their own latent variables, and among the latent variables inside the model.
Formally, let us as usual assume P variables observed on N units $(i = 1, \ldots, N)$. The resulting data x_{npq} are collected in a partitioned table of standardized data $\boldsymbol{X}$:

$$\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_q, \ldots, \boldsymbol{X}_Q],$$

where $\boldsymbol{X}_q$ is the generic q-th block.
Let the measurement and structural models be defined as in equations 3.5, 3.8 and 3.1. And since PLS approach to SEM does not need to distinguish between endogenous and exogenous latent variables, at least at the inner and outer estimation level, the structural model can be rewritten as:

$$\boldsymbol{\xi}_q = \boldsymbol{B}\boldsymbol{\xi}_q + \boldsymbol{\zeta}_q \tag{3.58}$$

The measurement model for the reflective scheme, as well as for the formative scheme, is the same as the one expressed in equation 3.5 and in equation 3.8. As a matter of fact, in a reflective scheme each manifest is related to the latent variable by a simple regression model, i.e:

$$\boldsymbol{x}_{pq} = \lambda_{pq}\boldsymbol{\xi}_q + \boldsymbol{\epsilon}_{pq} \tag{3.59}$$

An assumption behind this model is that the residual $\boldsymbol{\varepsilon}_{pq}$ has a zero mean and is uncorrelated with the latent variable of the same block:

$$E(\boldsymbol{x}_{pq}|\boldsymbol{\xi}_q) = \lambda_{pq}\boldsymbol{\xi}_q \tag{3.60}$$

This assumption defined *predictor specification* assures desirable estimation properties in OLS modeling.
In a formative scheme, instead, each latent variable is obtained as a linear combination of the manifest variables of the block. Thus the measurement model can be expressed as:

$$\boldsymbol{\xi}_q = \sum_{p=1}^{P_q} w_{pq}\boldsymbol{x}_{pq} + \boldsymbol{\delta}_{pq} \tag{3.61}$$

The error term $\boldsymbol{\delta}_{pq}$ represents the fraction of the corresponding latent variable not accounted for by the manifest variables. The assumption

behind this model is the following *predictor specification*:

$$E(\boldsymbol{\xi}_q|\boldsymbol{x}_{pq}) = \sum_{p=1}^{P_q} w_{pq}\boldsymbol{x}_{pq} \tag{3.62}$$

In PLS Path Modeling an iterative procedure allows us to estimate the model parameters, i.e the outer weights (w_{pq}) and the latent variable scores ($\boldsymbol{\xi}_q$). The estimation procedure is named *partial* since it solves blocks one at a time by means of alternating single and multiple linear regressions. The path coefficients (β_{mj}) come afterwards from a regular regression between the estimated latent variable scores.
The estimation of the latent variable scores are obtained through the alternation of the *outer* and the *inner* estimations, iterating till convergence. It is important to underline that no formal proof of convergence has been provided until now. As a matter of fact, until now convergence is proved only for path diagram with one or two blocks [Lyttkens et al. 1975]. Nevertheless, empirical convergence is always assured.
The procedure starts by choosing arbitrary weights w_{pq}. Then, in the external estimation, each latent variable is estimated as a linear combination of its own manifest variables:

$$\boldsymbol{\nu}_q \propto \sum_{p=1}^{P_q} w_{pq}\boldsymbol{x}_{pq} = \boldsymbol{X}_q\boldsymbol{w}_q \tag{3.63}$$

where $\boldsymbol{\nu}_q$ is the standardized outer estimation of the q-th latent variable $\boldsymbol{\xi}_q$ and the symbol $\propto$ means that the left side of the equation corresponds to the standardized right side.
In the internal estimation, each latent variable is estimated by considering its links with the other Q' adjacent latent variables:

$$\boldsymbol{\vartheta}_q \propto \sum_{q'=1}^{Q'} e_{qq'} \boldsymbol{\nu}_q \tag{3.64}$$

where $\boldsymbol{\vartheta}_q$ is the standardized inner estimation of the q-th latent variable $\boldsymbol{\xi}_q$ and the inner weights ($e_{qq'}$) are equal (in a centroid scheme) to the signs of the correlations between the q-th latent variable $\boldsymbol{\nu}_q$ and the $\boldsymbol{\nu}_{q'}$s connected with $\boldsymbol{\nu}_q$. Inner weights can be obtained following other schemes rather than the centroid one. Namely, the inner weights can be equal to:

1. the signs of the correlations between the q-th latent variable $\boldsymbol{\nu}_q$ and the $\boldsymbol{\nu}_{q'}$s connected with $\boldsymbol{\nu}_q$ in the centroid scheme (the Wold's original scheme)

2. the correlations between the q-th latent variable $\boldsymbol{\nu}_q$ and the $\boldsymbol{\nu}_{q'}$s connected with $\boldsymbol{\nu}_q$ in the factorial scheme (the Löhmoller scheme)

3. the multiple regression coefficient of $\boldsymbol{\nu}_q$ and the $\boldsymbol{\nu}_{q'}$s connected with $\boldsymbol{\nu}_q$, if the $\boldsymbol{\nu}_q$ is the inner estimation of an endogenous latent variables, or the correlations coefficient for exogenous latent variables in structural scheme.

Once a first estimation of the latent variables is obtained, the algorithm goes on by updating the outer weights w_{pq}.
Two different ways are available to update the outer weights usually related to the two different kinds of measurement model (i.e. the *formative* or the *reflective* scheme) expressed in section 3.1:

- *Mode A*: each outer weight w_{pq} is the regression coefficient in the simple regression of the p-th manifest variable of the q-th block ($\boldsymbol{x}_{pq}$) on the inner estimate of the q-th latent variable $\boldsymbol{\vartheta}_q$. As a matter of fact, since the latent variable score $\boldsymbol{x}_{pq}$ is standardized, the generic outer weight w_{pq} is obtained as:

$$w_{pq} = cov\left(\boldsymbol{x}_{pq}, \boldsymbol{\vartheta}_q\right) \tag{3.65}$$

  i.e. as the covariance between each manifest variable and the corresponding inner estimate of the latent variable.

- *Mode B*: the vector $\boldsymbol{w}_q$ of the weights w_{pq} associated to the manifest variables of the q-th block is the regression coefficient vector in the multiple regression of the inner estimate of the q-th latent variable $\boldsymbol{\vartheta}_q$ on its centered manifest variables $\boldsymbol{X}_q$:

$$\boldsymbol{w}_q = \left(\boldsymbol{X}'_q \boldsymbol{X}_q\right)^{-1} \boldsymbol{X}'_q \boldsymbol{\vartheta}_q \tag{3.66}$$

As already said, the choice of the external weight estimation mode is strictly related to the nature of the model. For a *reflective model* the *Mode A* is more appropriate, while *Mode B* is better for the *formative*

model.
Furthermore, *Mode A* is suggested for endogenous latent variables, while *Mode B* for the exogenous ones.
It is worth noticing that *Mode B* is affected by multicollinearity. In such a situation, PLS regression may be used as a valuable alternative to OLS regression to obtain the external weights according to equation 3.66.
The algorithm is iterated till convergence, which is demonstrated to be reached for one and two-block models. However, for multi-block models, convergence is always verified in practice.

After convergence, structural (or path) coefficients are estimated through an OLS multiple regression among the estimated latent variable scores.

Wold's original algorithm has been further developed [Lohmöller 1987, Lohmöller 1989]. In particular, new options for computing both inner and outer estimations have been implemented together with a specific treatment for missing data and multicollinearity [Tenenhaus & Esposito Vinzi 2005].
As regards this last point, in the case of multicollinearity among the estimated latent variables, PLS regression can be used to obtain path coefficient estimates instead of OLS regression.

Here, a schematic description of the original PLS Path Modeling Wold's algorithm is given:

Algorithm 1 PLS Path Modeling Wold's algorithm

Input: $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_q, \ldots, \boldsymbol{X}_Q]$ standardized MV's;
Output: $\boldsymbol{\beta}_j$, $\boldsymbol{w}_q$, $\boldsymbol{\xi}_q$;

1: **for all** $q = 1, \ldots, Q$ **do**
2: initialize $\boldsymbol{w}_q$
3: $\boldsymbol{\nu}_q \propto \sum_{p=1}^{P_q} w_{pq}\boldsymbol{x}_{pq} = \boldsymbol{X}_q\boldsymbol{w}_q$
4: $e_{qq'} = sign\left[cor\left(\boldsymbol{\nu}_q, \boldsymbol{\nu}_{q'}\right)\right]$ following the centroid scheme
5: $\boldsymbol{\vartheta}_q \propto \sum_{q'=1}^{Q'} e_{qq'}\boldsymbol{\nu}_{q'}$
6: update $\boldsymbol{w}_q : w_{pq} = cor(\boldsymbol{x}_{pq}, \boldsymbol{\vartheta}_q)$ or $\boldsymbol{w}_q = (\boldsymbol{X}'_q\boldsymbol{X}_q)^{-1}\boldsymbol{X}'_q\boldsymbol{\vartheta}_q$
7: **end for**
8: **Steps 1 to 7 are repeated until convergence** on a specific latent variable is achieved, i.e. until:

$$\boldsymbol{\nu}_{q^*} = \boldsymbol{\vartheta}_{q^*}$$

9: **Once the convergence is assured**:

(i) for each block the latent variable scores are computed as:

$$\boldsymbol{\xi}_q \propto \boldsymbol{X}_q\boldsymbol{w}_q,$$

(ii) for each endogenous latent variable $\boldsymbol{\xi}_j^{(J)}$, the vector of the path coefficients is obtained as:

$$\boldsymbol{\beta}_j = \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}\boldsymbol{\xi}_j^{(J)},$$

where $\boldsymbol{\Xi}$ includes the exogenous latent variables scores of the latent variables connected to the j-th endogenous latent variable $\boldsymbol{\xi}_j^{(J)}$.

The Quality indexes

PLS Path Modeling lacks a well identified global optimization criterion so that there is no *global fitting function* to be evaluated to determine the goodness of the model. Furthermore, it is a variance-based model strongly oriented to prediction. Thus, model validation focuses on the model predictive capability. According to PLS-PM structure, each part of the model needs to be validated: the *measurement model*, the *structural model* and the overall model. That is why, PLS Path Modeling provides three different fit indexes: the *communality* index, the *redundancy* index and the *Goodness of Fit* (GoF) index.

For each q-th block in the model the quality of the measurement model is measured by means of the *communality* index measure:

$$Com_q = \frac{1}{P_q} \sum_{p=1}^{P_q} cor^2 \left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q \right) \tag{3.67}$$

This index measures how much of the manifest variable variability in the q-th block is explained by its own latent variable $\boldsymbol{\xi}_q$. That means how well the manifest variables describe the related latent variable. Moreover, the communality index for the q-th block is nothing but the average of the squared correlation between each manifest variable in the q-th block and the q-th latent variable.
It is possible to measure the quality of the whole measurement model

by means of the *average communality* index, i.e:

$$\overline{Com} = \frac{1}{P}\sum_{q=1}^{Q} P_q Com_q \tag{3.68}$$

This is a weighted average of all the Q block-specific *communality* indexes (see equation 3.67) with weights equal to the number of manifest variables in each block. Moreover, since the *communality* index for the q-th block is nothing but the average of the squared correlation in the block, then the *average communality* is the average of all the squared correlations between each manifest variable and the corresponding latent variable in the model, i.e.:

$$\overline{Com} = \frac{1}{P}\sum_{q=1}^{Q}\sum_{p=1}^{P_q} cor^2\left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q\right) \tag{3.69}$$

Although the quality of each structural equation is measured by a simple evaluation of the R^2 fit index, this is not sufficient to evaluate the whole structural model. Specifically, since the structural equations are estimated once the convergence is assured, i.e. once the latent variable scores are estimated, then the R^2 values only take into account the fit of each regression in the structural model. That is why a new index is computed for each endogenous block in addition to the R^2 value in order to take into account also the measurement model: the *redundancy* index.

The *redundancy* index computed for the j-th block, measures the portion of variability of the manifest variables connected to the j-th endogenous latent variable explained by the latent variables indirectly connected to the block, i.e.:

$$Red_j = Com_j \times R^2 \left(\boldsymbol{\xi}_j^{(J)}, \{ \boldsymbol{\xi}_q \text{'s explaining } \boldsymbol{\xi}_j^{(J)} \} \right) \tag{3.70}$$

A global quality measure of the structural model is also provided by the *average redundancy* index, computed as:

$$\overline{Red} = \frac{1}{J} \sum_{j=1}^{J} Red_j \tag{3.71}$$

where J is the total number of endogenous latent variables in the model.

As aforementioned, there is no overall fit index in PLS Path Modeling. Nevertheless, a global criterion of goodness of fit has been recently proposed by Amato, Esposito Vinzi & Tenenhaus [2005]: the GoF index.
Such index has been developed in order to take into account the model performance in both the measurement and the structural model. For this reason the GoF index is obtained as the geometric mean of the *average communality* index and the average R^2 value:

$$GoF = \sqrt{\overline{Com} \times \overline{R^2}} \tag{3.72}$$

where the average R^2 value is obtained as:

$$\overline{R^2} = \frac{1}{J} R^2 \left(\boldsymbol{\xi}_j^{(J)}, \left\{ \boldsymbol{\xi}_q \text{'s explaining } \boldsymbol{\xi}_j^{(J)} \right\} \right) \tag{3.73}$$

According to equations 3.67 and 3.73 the GoF index can be rewritten as:

$$GoF = \sqrt{\frac{\sum_{q=1}^{Q} \sum_{p=1}^{P_q} Cor^2 \left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q \right)}{P} \times \frac{\sum_{j=1}^{J} R^2 \left(\boldsymbol{\xi}_j^{(J)}, \left\{ \boldsymbol{\xi}_q \text{'s explaining } \boldsymbol{\xi}_j^{(J)} \right\} \right)}{J}} \tag{3.74}$$

As PLS Path Modeling is a *soft modeling* approach with no distributional assumptions, it is possible to estimate the significance of the parameters based on cross-validation methods like jack-knife and bootstrap [Efron & Tibshirani 1993].
It is also possible to build a cross-validated version of all the quality indexes (i.e. of the *communality* index, of the *redundancy* index, and of the GoF index) by means of a *blindfolding* procedure. For more details on the *blindfolding* procedure please refers to Tenenhaus et al. [2005].

A normalized version of the GoF has been presented by Tenenhaus, Amato & Esposito Vinzi [2004].
This index is obtained by relating each term in equation 3.72 to the corresponding maximum value.
In particular, it is well known that in principal component analysis the best rank one approximation of a set of variables $\boldsymbol{X}$ is given by the

eigenvector associated to the largest eigenvalue λ of the $\boldsymbol{X^T X}$ matrix. Furthermore, the sum of the squared correlation between each variable and the first principal component of $\boldsymbol{X}$ is a maximum.
Therefore, if data are mean centered and with unit variance, the first term in equation 3.77 is such that $\sum_{p=1}^{P_q} cor^2\left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q\right) \leq \lambda_q$. Thus, the normalized version of the first term of the *GoF* is obtained as:

$$T_1 = \frac{1}{P} \sum_{q=1}^{Q} \frac{\sum_{p=1}^{P_q} cor^2\left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q\right)}{\lambda_q} \tag{3.75}$$

In other words, here the sum of the communalities in each block is divided by the first eigenvalue of the block.
As concerning the second term of the equation 3.77, the normalized version is obtained as:

$$T_2 = \frac{1}{J} \sum_{j=1}^{J} \frac{R^2\left(\boldsymbol{\xi}_j^{(J)}, \left\{\boldsymbol{\xi}_q \text{'s explaining } \boldsymbol{\xi}_j^{(J)}\right\}\right)}{\rho_j^2} \tag{3.76}$$

where ρ_j is the first canonical correlation of the canonical analysis of matrices $\boldsymbol{X}_j$ containing the manifest variables associated to the j-th endogenous latent variable, and $\boldsymbol{X}_q$ containing the manifest variables associated to the exogenous latent variables explaining $\boldsymbol{\xi}_q$.
Thus, according to equations 3.75, 3.76 and 3.72, the relative *GoF* index is:

$$GoF = \sqrt{\frac{1}{P} \sum_{q=1}^{Q} \frac{\sum_{p=1}^{P_q} Cor^2\left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q\right)}{\lambda_q} \times \frac{1}{J} \sum_{j=1}^{J} \frac{R^2\left(\boldsymbol{\xi}_j^{(J)}, \left\{\boldsymbol{\xi}_q \text{'s explaining } \boldsymbol{\xi}_j^{(J)}\right\}\right)}{\rho_j^2}} \tag{3.77}$$

This index, is bounded between 0 and 1. Both the GoF and the relative GoF are descriptive indexes, i.e. there is no inference-based threshold to judge their values. Nonetheless, the higher their value is, the best the model performance is. As a rule of thumb, a value of the relative GoF equal to or higher than 0.9 clearly speaks in favor of the model.

3.4.2 The Generalized Structured Component Analysis

Generalized Structured Component Analysis is a method recently proposed by Hwang & Takane [2004] to estimate Structural Equation Models. As usual, Structural Equation Models can be formalized taking into account both the structural and the reflective measurement models as expressed by equations 3.1 and 3.5.
For the i-th unit the structural and measurement models can be rewritten as:

$$\boldsymbol{x}_i = \boldsymbol{\Lambda}\boldsymbol{\xi}_i + \boldsymbol{\epsilon}_i \tag{3.78}$$

and

$$\boldsymbol{\xi}_i = \boldsymbol{B}\boldsymbol{\xi}_i + \boldsymbol{\zeta}_i \tag{3.79}$$

where: $\boldsymbol{x}_i$ is a P by 1 vector containing all the manifest variables for unit i, $\boldsymbol{\xi}_i$ is the vector of dimension Q by 1 of all the latent variables

(both the J endogenous and the M exogenous ones) for the i-th unit, $\boldsymbol{\Lambda}$ is a P by Q matrix of the loadings, $\boldsymbol{B}$ is a square matrix Q by Q containing the path coefficients of the structural model (an element of $\boldsymbol{B}$ is equal to zero if the relationship is not included in the model), and $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\zeta}_i$ are the two vectors of the residuals in the structural and measurement models respectively.
GSCA integrates the two models expressed in equation 3.78 and in equation 3.79 in a unique formulation, i.e.:

$$\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{\xi}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{\xi}_i + \begin{bmatrix} \boldsymbol{\epsilon}_i \\ \boldsymbol{\zeta}_i \end{bmatrix} \tag{3.80}$$

Moreover, in GSCA the latent variables are defined as weighted components of the observed variables, i.e.:

$$\boldsymbol{\xi}_i = \boldsymbol{W}\boldsymbol{x}_i \tag{3.81}$$

where $\boldsymbol{W}$ is a Q by P matrix containing the component weights.
Then, the equation 3.80 can be rewritten as:

$$\begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{W} \end{bmatrix} \boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{W} \end{bmatrix} \boldsymbol{x}_i + \begin{bmatrix} \boldsymbol{\epsilon}_i \\ \boldsymbol{\zeta}_i \end{bmatrix} \tag{3.82}$$

where:$\boldsymbol{I}$ is an identity matrix of order P.
Moreover, defining $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix}$, $\boldsymbol{r}_i = \begin{bmatrix} \boldsymbol{\epsilon}_i \\ \boldsymbol{\zeta}_i \end{bmatrix}$ and $\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{W} \end{bmatrix} \boldsymbol{x}_i$,

the last equation can be rewritten as:

$$\boldsymbol{u}_i = \boldsymbol{A}\boldsymbol{u}_i + \boldsymbol{r}_i \tag{3.83}$$

As is easy to notice in GSCA all the manifest variables, as well as all the latent variables, are included in the supervector $\boldsymbol{u}_i$ of dimension $(P+Q)$ by 1. Moreover all the parameters of the model (i.e. the loadings and the path coefficients) are included in the matrix $\boldsymbol{A}$ of dimension $(P+Q)$ by Q. As the authors underlined *"differently from the PLS-PM, in GSCA the structural and the measurement models are not addressed separately, on the contrary they are combined in a unique algebraic formulation"* [Hwang & Takane 2004]. This allows the authors to identify a unique function to maximize.
Therefore, the parameters of GSGA ($\boldsymbol{W}$ and $\boldsymbol{A}$) are estimated so that the sum of the squares of all residuals $\boldsymbol{r}_i$ for the i-th unit is as small as possible.
In other words, the following least-squares criterion is minimized:

$$\vartheta = \sum_{i=1}^{N} \left(\boldsymbol{u}_i - \boldsymbol{A}\boldsymbol{u}_i\right)' \left(\boldsymbol{u}_i - \boldsymbol{A}\boldsymbol{u}_i\right) \tag{3.84}$$

with respect to $\boldsymbol{W}$ and $\boldsymbol{A}$ and under the constraint that the latent variable scores are normalized, i.e.: $\sum_{n=1}^{N} \xi_{iq}^2 = 1$.
This is equivalent to minimize:

$$\vartheta = SS\left(\boldsymbol{U} - \boldsymbol{U}\boldsymbol{A}\right) \tag{3.85}$$

where $SS(X) = trace(X'X)$.

An Alternating Least Squares (ALS) algorithm [de Leeuw, Young & Takane 1976] is used so as to minimize equation 3.84. ALS algorithm is an iterative procedure composed of two steps.
In the first step of ALS algorithm applied to GSCA $\boldsymbol{A}$ is update for a fixed $\boldsymbol{W}$. While, in the second step $\boldsymbol{W}$ is update for the value of $\boldsymbol{A}$ estimated in the first step. These two steps are alternate until convergence is assured, i.e. until the decrease in the function value falls below a certain threshold value. For more details on ALS please refer to de Leeuw et al. [1976].
Since ALS monotonically decrease the value of the chosen criterion the convergence is assured. Nevertheless, it is not assured that the convergence is reached in a global minimum. To overcome this problem different procedures are available, namely using such "good" initial values or running the algorithm with different starting values. In particular, Hwang & Takane [2004] suggest using a Constrained Component Analysis to obtain such "good" starting values for $\boldsymbol{W}$, and then simply obtain $\boldsymbol{A}$ as least square estimate given $\boldsymbol{W}$.

Generalized Structured Component Analysis can be performed with both formative and reflective manifest variables. Moreover, as for the PLS Path Modeling, GSCA includes a lot of existing standard multivariate techniques as special cases, for example the regression model, the ANOVA and the discriminant analysis.

The Quality indexes

Generalized Structured Component Analysis provides an overall measure of model fit based on the part of the endogenous variable variance explained by the model: the FIT index. This index is given by:

$$FIT = 1 - \frac{SS\left(\boldsymbol{U} - \boldsymbol{UA}\right)}{SS\left(\boldsymbol{U}\right)} \tag{3.86}$$

The FIT index is a function of the residuals from the model summarizing the discrepancy between the model and the data. The higher the residual variance is, the smaller the FIT index is. Models showing higher FIT index values are to be preferred to models showing lower FIT index values.
Furthermore, the FIT index is bounded between 0 and 1: models with FIT values close to one have to be considered "good" models, while models with FIT values close to zero have to be rejected.
Nevertheless, FIT index does not take into account model complexity. It is for this reason that more recently a new global quality index has been developed: the Adjusted FIT index.

$$AFIT = 1 - \left(1 - FIT\right)\frac{df_0}{df_1} \tag{3.87}$$

where $df_0 = NP$ is the number of degrees of freedom for the model for which $\boldsymbol{W} = 0$ and $\boldsymbol{A} = 0$, and $df_1 = NP - fp$ is the degrees for the model to test where fp is the number of free parameters.

Other existing indexes can be used in the GSCA framework to assess model quality. Namely, the GFI of Jöreskog & Sörbom [1996] for the unweighted least-squares, and the Standardized Root Mean square Residuals (SRMR). These two indexes are based on the discrepancy between the sample covariance matrix and the covariance matrix obtained by the model. Moreover, both the index values close to one have to be considered as associated to a good fitted model.

3.4.3 The Generalized Maximum Entropy Approach

In 2003 Al-Nasser proposed an alternative method to estimate Structural Equation Models in a distribution free optic: the Generalized Maximum Entropy Approach to SEM [Al-Nasser 2003]. This method is an extension of the Generalized Maximum Entropy (GME) procedure for general linear econometric model presented by Golan, Judge & Miller [1996]. In this subsection first a review of the GME procedure by Golan et al. [1996] will be done, then the GME approach to Structural Equation Models will be presented.

The Generalized Maximum Entropy procedure

The Generalized Maximum Entropy procedure (GME procedure) for the general linear econometric model presented by Golan *et al.* represents an estimation technique allowing us to obtain model parameter estimates when the underlining model is incompletely known and the

data are limited, partial or incomplete [Golan et al. 1996]. This estimation technique is based on the *Shannon's entropy-information measure* [Shannon 1948] and on the Maximum Entropy Principle introduced by Jaynes [1957*a*] [1957b].

Letting $\boldsymbol{x}$ a random variable observed on N unit, with possible outcome $x_1, \cdots, x_N$ whose probability of occurrence are $p_1, \cdots, p_N$, such that $\sum_{i=1}^{N} p_i = 1$. Shannon defined the *entropy* or the *information of entropy* of the distribution of $\boldsymbol{x}$ as:

$$H(\boldsymbol{x}) = -\sum_{i=1}^{N} p_i \ln(p_i) \tag{3.88}$$

where $(-ln(p_i))$ is the amount of the self-information of the event x_i and $0 \ln(0) = 0$. The average of the self-information is defined as the *entropy*. The function H reaches a maximum of $\ln(N)$ when $p_1 = p_2 = \cdots = p_N = 1/N$, i.e. when all the possible outcome $x_1, \cdots, x_N$ have the same probability of occurrence. While H is zero when $p_i = 1$ for the i-th unit and otherwise zero.

The GME procedure [Jaynes 1957*a*, Jaynes 1957*b*] allows us to obtain the distribution function of the random variable $\boldsymbol{x}$ by recovering the unknown probabilities p_i. To recover the unknown probability p's that characterize a given data set, Jaynes proposes maximizing entropy, subject to sample-moment information and adding up constraints on the probabilities [Jaynes 1957*a*, Jaynes 1957*b*]. The idea behind this is that, if sufficient information on the data is not available, the best estimation of the true distribution is obtained using the

frequency that maximizes the entropy. This principle was taken up by Golan et al. [1996] to obtain an alternative estimation technique to estimate regression model parameters in the case of an ill-posed problem. Further information on GME estimation procedure can be found in Skilling [1989] and Golan et al. [1996].

The Generalized Maximum Entropy procedure applied to model parameter estimation needs to express the model parameters, as well as the errors in the model, in terms of probability values. This is why the first step in all GME procedures is to convert the standard problem into a probability form. Moreover, the GME procedures needs, for each parameter and error, to specify the support spaces, i.e the ranges within which each estimated parameter and error lies. The support space is specified, based on prior knowledge. Once the re-parametrization of the model is obtained, the GME procedure can be seen as a non linear programming problem maximizing the *Shannon's Entropy measure* (cf. equation 3.88) solved by numerical methods.
The Generalized Maximum Entropy procedure applied to model parameter estimation can be summarized in the following steps:

1. re-parametrization of the unknown parameters and of the disturbance terms as a convex combination of the expected value of a discrete random variable;

2. rewriting the model with the new re-parametrization as a constraint:

3. formulation of the GME problem as non linear programming problem, i.e.:

   Objective function = Shannon's Entropy measure

   under:

   - the normalization constraints
   - the consistency constraints, which represent the new formulation of the model

4. solving the non linear programming problem by using numerical methods.

A reformulation of SEMs in a GME optic

Let $\boldsymbol{x}_i^{(M)}$ and $\boldsymbol{x}_i^{(J)}$ be the vectors of the manifest variables associated with the exogenous latent variables and with the endogenous latent variables observed for the i-th unit.
And let as usual, $\boldsymbol{\xi}_i^{(M)}$ and $\boldsymbol{\xi}_i^{(J)}$ be the vectors of the exogenous and endogenous latent variables for the i-th unit.
Then, in the case of reflective scheme and according to equation 3.5, the measurement model of a Structural Equation Model can be rewritten as:

$$\boldsymbol{x}_i^{(M)} = \boldsymbol{\Lambda}^{(M)}\boldsymbol{\xi}_i^{(M)} + \boldsymbol{\epsilon}_i^{(M)} \tag{3.89}$$

for the manifest variables associated to the exogenous latent variables, and as:

$$\boldsymbol{x}_i^{(J)} = \boldsymbol{\Lambda}^{(J)} \boldsymbol{\xi}_i^{(J)} + \boldsymbol{\epsilon}_i^{(J)} \tag{3.90}$$

for the manifest variables associated to the endogenous latent variables. Where $\boldsymbol{\Lambda}^{(M)}$ and $\boldsymbol{\Lambda}^{(J)}$ are the loadings associated to the manifest variables of type M and of type J.

Further, according to equation 3.1 the structural model is:

$$\left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right) \boldsymbol{\xi}_i^{(J)} = \boldsymbol{B}^{(M)} \boldsymbol{\xi}_i^{(M)} + \boldsymbol{\zeta}_i \tag{3.91}$$

The above equations can be mixed in a unique matrix formulation of the specified Structural Equation Model, i.e:

$$\boldsymbol{X}^{(J)} = \boldsymbol{\Lambda}^{(J)} \left(\boldsymbol{I} - \boldsymbol{B}^{(J)}\right)^{-1} \left[\boldsymbol{B}^{(M)} \left(\boldsymbol{\Lambda}^{(M)}\right)^{-1} \left(\boldsymbol{X}^{(M)} - \boldsymbol{E}^{(M)}\right) + \boldsymbol{H}\right] + \boldsymbol{E}^{(J)} \tag{3.92}$$

where $\boldsymbol{I}$ is an identity matrix, $\boldsymbol{E}^{(M)}$ and $\boldsymbol{Z}$ are the matrices containing the measurement and structural errors, and $\left(\boldsymbol{\Lambda}^{(M)}\right)^{-1}$ is the generalized inverse of $\boldsymbol{\Lambda}^{(M)}$.

As already said, GME needs a re-parametrization of the model in a probabilistic form. Therefore, in order to apply a GME procedure to Structural Equation Models and following Al-Nasser's [2003] work we need to rewrite the parameters of the model so as to have them expressed as probabilities.

With this intent the model parameters (i.e. the structural coefficients $\boldsymbol{B}^{(M)}$ and $\boldsymbol{B}^{(J)}$, the external loadings $\boldsymbol{\Lambda}^{(M)}$ and $\boldsymbol{\Lambda}^{(J)}$, as well as the errors terms $\boldsymbol{Z}$, $\boldsymbol{E}^{(M)}$ and $\boldsymbol{E}^{(J)}$), are re-parametrized as the expected values of discrete random variables with two or more sets of points. In other words, each model parameter is transformed according to this general formulation:

$$\theta = \textstyle\sum_{a=1}^{A} p_a \vartheta_a \quad \text{with} \sum_{a=1}^{A} \vartheta_a = 1$$

where A is the number of fixed points, θ is a generic parameter, ϑ_a is a generic fixed point and p_a is the probability associated to the a-th fixed point. The transformation expressed in equation 3.93 is applied to each element of the model parameter matrices. By way of example the generic element $\beta_{mj}^{(J)}$ of the matrix $\boldsymbol{B}^{(J)}$ containing the path coefficients associated to the endogenous latent variables is re-parametrized as:

$$\beta_{mj}^{(J)} = \sum_{a=1}^{A} p_{mja} b_{mja}^{(J)} \tag{3.93}$$

under the constraint that $\sum_{a=1}^{A} b_{mja}^{(J)} = 1$.

Once all the parameters are re-parametrized according to equation 3.93, the Structural Equation Model expressed in a matrix form as in

equation 3.92 can be rewritten as:

$$\boldsymbol{X}^{(J)} = \psi\left(b^{(M)}, b^{(J)}, l^{(M)}, l^{(J)}, e^{(M)}, e^{(J)}, z\right) \tag{3.94}$$

where $b^{(M)}$ and $b^{(J)}$ are the random variables for the path coefficients re-parametrization (according to equation 3.93), $l^{(M)}$ and $l^{(J)}$ are the random variables for the external loadings re-parametrization, $e^{(M)}$ and $e^{(J)}$ are the random variables for the external errors re-parametrization, and z is the random variable for the error term in the structural equations.
For more details on GME re-parametrization of Structural Equation Models please refers to Al-Nasser [2003].

According to the third step of the GME procedure, the parameter estimates are obtained by solving by means of numerical algorithm a non linear programming problem expressed as:

$$\max H\left(b^{(M)}, b^{(J)}, l^{(M)}, l^{(J)}, e^{(M)}, e^{(J)}, z\right) \tag{3.95}$$

subject to:

(i) the consistency constraints, i.e:

$$\boldsymbol{X}^{(J)} = \psi\left(b^{(M)}, b^{(J)}, l^{(M)}, l^{(J)}, e^{(M)}, e^{(J)}, z\right); \tag{3.96}$$

(ii) the normalization constraints, i.e:

$$\sum_{a=1}^{A} \vartheta_a = 1 \tag{3.97}$$

for all parameters in the models.

where H is the entropy function as defined in equation 3.88.
As already said, numerical optimization techniques are used to solve this system and to obtain parameter estimates.

In a simulation study, Ciavolino and Al-Nasser showed that the Generalized Maximum Entropy approach to Structural Equation Models seems to work better than PLS Path Modeling in the presence of outliers [Ciavolino, Al Nasser & D'Ambra 2006]. Nevertheless, in the case of high multicollinearity among manifest variables GME does not show better results than PLS Path Modeling at least for moderate sample sizes.
Further research on the GME approach to Structural Equation Models is required to specify better the capability and the drawbacks of the GME approach.

Chapter 4

Latent class detection in Structural Equation Models

4.1 Introduction

Traditionally, Structural Equation Models assume homogeneity over the observed set of units. In other words, all units are supposed to be well represented by a unique model estimated on all the units, i.e. the *global model*. This assumption may however often turn out to be false. In many cases it is reasonable to expect that different classes showing heterogeneous behaviors may exist in the observed set of units, and that treating all units as a single class may lead to biased results both in terms of model parameters and of validation indexes [Jedidi et al. 1997a, Jedidi et al. 1997b].

The traditional approach to clustering in Structural Equation Modeling consists in estimating separate models for unit segments obtained by external clustering techniques, either by assigning units to *a priori* classes on the basis of external variables such as demographic or consumption variables, or through cluster analysis. Concerning this last point, in Structural Equation Models, classes can be obtained by performing a cluster analysis either on the manifest or on the estimated latent variable scores.

In other words, usually heterogeneity in Structural Equation Models is handled by forming classes on the basis of such external variables or on the basis of such standard clustering techniques on manifest and/or latent variables, and then by using the standard multigroup structural equation modeling of Jöreskog [1971] and Sörbom [1974].

None of these *a priori* approaches, however, can be considered really satisfactory for several different reasons. Firstly, very rarely heterogeneity in the models may be captured by well-known observable variables playing the role of moderating variables [Hahn et al. 2002]. Moreover, clustering techniques on manifest variables or on latent variable scores do not take into account in any way the model itself. Hence, while the local models obtained by cluster analysis on the latent variable scores will lead to differences in the group averages of the latent variables but not necessarily to different models, the same method performed on the manifest variables is unlikely to lead to different and well-separated models, both in terms of model parameters and of average latent variable scores. Additionally, clustering procedures

may show some theoretical problems: traditional cluster analysis, in fact, assumes independence among variables, while Structural Equation Models are based on the assumption that variables (latent or manifest) are correlated [Jedidi et al. 1997b].

Apart from the methodological considerations, *a priori* unit clustering in Structural Equation Models is not conceptually acceptable since no causal structure among the variables is postulated: when information concerning the causal relationships among variables is available (as it is in the theoretical causal network of relationships), classes should be looked for while taking into account this relevant piece of information. In other words, a *response-based* clustering method should be used, where the obtained classes are homogeneous with respect to the postulated model.
This approach to clustering is opposed to the traditional *a priori* clustering, where classes are defined according to information which is not related to the existing model but depends on external criteria.

In this chapter we focus on techniques for detecting unit segments by *response-based* techniques in the case of unknown (latent) moderating effects, i.e. when both the number and the structure of the classes are not *a priori* known.
Ways to handle unobserved heterogeneity in three of the different approaches to SEM presented in chapter three will be presented. Firstly, methods allowing *response-based* clustering in LISREL-type Structural

Equation Models (cf. subsection 3.3.1) will be shown: the Structural Equation finite Mixture Model (STEMM) by Jedidi *et al.* [Jedidi et al. 1997a, Jedidi et al. 1997b] (cf. subsection 4.2.1) and the Bayesian Finite Mixture SEM by Zhu & Lee [2001] (cf. subsection 4.2.2). Further, unobserved heterogeneity in PLS Path Modeling (cf. section 3.4.1) framework will be presented. In this framework, several approaches will be described. Namely, the Finite Mixture PLS [Hahn et al. 2002, Ringle et al. 2008] (cf. subsection 4.3.1), the PLS Typological Path Model [Squillacciotti 2005, Trinchera et al. 2006] (cf. subsection 4.3.3), the PATHMOX [Sanchez & Aluja 2006, Sanchez & Aluja 2007] (cf. subsection 4.3.2) and the PLS Path Modeling Clustering [Ringle & Schlittgen 2007] (cf. subsection 4.3.4). To conclude, *response-based* techniques for clustering in GSCA (cf. subsection 3.4.2) will be investigated by the Fuzzy Clusterwise Generalized Structured Component Analysis of Hwang et al. [2007] (cf. subsection 4.4.1).

A new technique to obtain *response-based* clustering in PLS Path Models, the Response Based Unit Segmentation in PLS-PM (REBUS-PLS) [Trinchera 2007, Trinchera, Squillacciotti, Esposito Vinzi & Tenenhaus 2007, Trinchera, Romano & Esposito Vinzi 2007, Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Esposito Vinzi, Amato & Trinchera 2008], will be presented in chapter five.

To conclude, once the groups are identified it is very important to assess the differences (and similarities) among the detected classes of

units. In a Structural Equation framework, this essentially entails comparing the obtained local models to one another and with the global model. It is for this reason that the last section of this chapter will be devoted to presenting the different techniques allowing us to compare local models (cf. section 4.5). Since Structural Equation Models are complex models, comparing local models entails taking into account several aspects. Hence, the different ways to compare model parameters in the several approaches to Structural Equation Models will be first discussed (cf. subsection 4.5.2). Subsequently, latent variable scores comparison (cf. subsection 4.5.3), as well as model quality comparison (cf. subsection 4.5.4), will be examined.

4.2 Unobserved Heterogeneity in LISREL-type models

In SEM-ML the multigroup Structural Equation Modeling of Jöreskog [1971] and Sörbom [1974] is also usually used to handle unobserved heterogeneity. If no well-known moderating variables are available, several clustering techniques, such as K-means, are applied to the manifest variables in order to form *a priori* classes of units, i.e. classes of units built out of the model. Then, multigroup analysis is performed on such detected classes of units.

Jedidi *et al.* in 1997 initially felt the necessity for a *response-based* clustering technique in SEM-ML. The authors proposed to apply the Finite Mixture Model to the Structural Equation Model and presented

the STEMM (STructural Equation finite Mixture Model) [Jedidi et al. 1997a, Jedidi et al. 1997b]. Since then, the Finite Mixture Models have been used also in a PLS Path Modeling context (cf. section 4.3). More recently, Zhu and Lee developed a Bayesian approach to analyze mixtures in Structural Equation Models [Zhu & Lee 2001]. Since then, other works on the same topic have been presented [S.Y.Lee & Song 2002, Lee 2007]. Here, we first present the Jedidi *et al.* approach to Finite Mixture Models in SEM framework (cf. subsection 4.2.1), and then the Bayesian approach (cf. subsection 4.2.2).

4.2.1 Finite Mixtures in SEM-ML

The Structural Equation Finite Mixture Model (STEMM) by Jedidi *et al.* [Jedidi et al. 1997a, Jedidi et al. 1997b] is a model-based clustering technique which allows us to obtain *response-based* unit clustering in a SEM-ML framework. This method simultaneously forms classes and obtains class-specific estimates for the model parameters, i.e. for the measurement and the structural parameters.
Considering the presence of K latent classes, the measurement and the structural models in LISREL-type methods (cf. section 3.2) can be rewritten for each class k as:

$$\begin{aligned} \boldsymbol{x}_i^{(J)}|k &= \boldsymbol{\nu}_k^{(J)} + \boldsymbol{\Lambda}_k^{(J)}\boldsymbol{\xi}_{ik}^{(J)} + \boldsymbol{\epsilon}_{ik}^{(J)} \\ \boldsymbol{x}_i^{(M)}|k &= \boldsymbol{\nu}_k^{(M)} + \boldsymbol{\Lambda}_k^{(M)}\boldsymbol{\xi}_{ik}^{(M)} + \boldsymbol{\epsilon}_{ik}^{(M)} \end{aligned} \tag{4.1}$$

and

$$\left(1 - \boldsymbol{B}_k^{(J)}\right) \boldsymbol{\xi}_{ik}^{(J)} = \boldsymbol{B}_k^{(M)} \boldsymbol{\xi}_{ik}^{(M)} + \boldsymbol{\zeta}_{ik} \tag{4.2}$$

where $\boldsymbol{x}_i^{(M)}|k$ and $\boldsymbol{x}_i^{(J)}|k$ are the vectors of the manifest variables linked respectively to the exogenous and to the endogenous blocks for the i-th unit in the k-th latent class, $\boldsymbol{\xi}_{ik}^{(M)}$ and $\boldsymbol{\xi}_{ik}^{(J)}$ are, respectively, the J by one and the M by one vectors of the endogenous and exogenous latent variables for the i-th unit in the k-th latent class, $\boldsymbol{\Lambda}_k^{(M)}$, $\boldsymbol{\Lambda}_k^{(J)}$ and $\boldsymbol{B}_k$ are the matrices containing the group-specific parameters of the measurement and of the structural models, $\boldsymbol{\epsilon}_{ik}$ and $\boldsymbol{\zeta}_{ik}$ are the vectors of the group-specific errors for unit i associated to the measurement and to the structural models.
Let $\boldsymbol{x}_i|k$ be the joint vector, of dimension $[P \times 1]$ composed of the manifest variables linked to both the exogenous and the endogenous blocks:

$$\boldsymbol{x}_i|k = \begin{bmatrix} \boldsymbol{x}_i^{(J)}|k \\ \boldsymbol{x}_i^{(M)}|k \end{bmatrix} \tag{4.3}$$

under the assumption that all measures are error-free, *i.e.* $E\left(\boldsymbol{\epsilon}_{ik}^{(M)}\right) = 0$, $E\left(\boldsymbol{\epsilon}_{ik}^{(J)}\right) = 0$ and $E\left(\boldsymbol{\zeta}_{ik}\right) = 0$, the conditional mean vectors μ_k of the $\boldsymbol{x}|k$ is:

$$\mu_k = \begin{bmatrix} \boldsymbol{\nu}_k^{(J)} + \boldsymbol{\Lambda}_k^{(J)} \boldsymbol{B}_k^{(M)-1} \boldsymbol{B}_k^{(J)-1} \boldsymbol{\tau}_k^{\xi^{(M)}} \\ \boldsymbol{\nu}_k^{(M)} + \boldsymbol{\Lambda}_k^{(M)} \boldsymbol{\tau}_k^{\xi^{(M)}} \end{bmatrix} \tag{4.4}$$

where $\boldsymbol{\tau}_k^{\xi^{(M)}}$ is the mean vector of the exogenous latent variables in the k-th latent class, *i.e.* $E\left(\boldsymbol{\xi}_k^{(M)^{-1}}\right) = \boldsymbol{\tau}_k^{\xi^{(M)}}$
Moreover, let the covariance matrix of $\boldsymbol{\xi}_k^{(M)}$ be equal to $\boldsymbol{\Phi}_k$, i.e.:

$$\boldsymbol{\Phi}_k = E\left[\left(\boldsymbol{\xi}_k^{(M)} - \boldsymbol{\tau}_k^{\xi^{(M)}}\right)\left(\boldsymbol{\xi}_k^{(M)} - \boldsymbol{\tau}_k^{\xi^{(M)}}\right)^T\right], \tag{4.5}$$

the covariance matrices of the measurement errors be equal to $\boldsymbol{\Theta}_k^{(J)}$ and $\boldsymbol{\Theta}_k^{(M)}$, i.e.:

$$E\left(\boldsymbol{\epsilon}_{ik}^{(J)}\boldsymbol{\epsilon}_{ik}^{(J)^T}\right) = \boldsymbol{\Theta}_k^{(J)} \tag{4.6}$$

and

$$E\left(\boldsymbol{\epsilon}_{ik}^{(M)}\boldsymbol{\epsilon}_{ik}^{(M)^T}\right) = \boldsymbol{\Theta}_k^{(M)} \tag{4.7}$$

with these last two matrices not necessarily diagonal (so, with measurement errors correlated with one another), the covariance matrix of the structural error equal to Ψ_k, i.e.:

$$E\left(\boldsymbol{\zeta}_{ik}{\boldsymbol{\zeta}_{ik}}^T\right) = \Psi_k, \tag{4.8}$$

and under the assumption that the structural errors are uncorrelated with the endogenous latent variables, the conditional covariance matrix of the joint vector $\boldsymbol{x}|k$ is:

$$\widehat{\boldsymbol{\Sigma}}_k = \tag{4.9}$$

$$= \begin{bmatrix} \boldsymbol{\Lambda}_k^{(J)}\left(\boldsymbol{I} - \boldsymbol{B}_k^{(J)^{-1}}\right)\left(\boldsymbol{B}_k^{(M)}\boldsymbol{\Phi}_k\boldsymbol{B}_k^{(M)^T} + \boldsymbol{\Psi}_k\right)\left(1 - \boldsymbol{B}_k^{(J)^{-1}}\right)^T\boldsymbol{\Lambda}_k^{(J)^T} + \boldsymbol{\Theta}_k^{(J)} & \boldsymbol{\Lambda}_k^{(J)}\left(1 - \boldsymbol{B}_k^{(J)^{-1}}\right)\boldsymbol{B}_k^{(M)}\boldsymbol{\Phi}_k\boldsymbol{\Lambda}_k^{(M)^T} \\ \boldsymbol{\Lambda}_k^{(M)}\boldsymbol{\Phi}_k\boldsymbol{B}_k^{(M)}\boldsymbol{B}_k^{(M)^T}\left(\boldsymbol{I} - \boldsymbol{B}_k^{(J)^{-1}}\right)^T\boldsymbol{\Lambda}_k^{(J)^T} & \boldsymbol{\Lambda}_k^{(M)}\boldsymbol{\Phi}_k\boldsymbol{\Lambda}_k^{(M)^T} + \boldsymbol{\Theta}_k^{(M)} \end{bmatrix}$$

according to equation 3.34.
Assuming that the joint vector $\boldsymbol{x}_i|k$ is multivariate normally distributed within each class, with parameters equal to μ_k and $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{x}_i|k \sim f_{ik}\left(\boldsymbol{x}_i|\mu_k, \boldsymbol{\Sigma}_k\right) \tag{4.10}$$

then, the unconditional density function can therefore be represented as a mixture of the conditional, i.e. class specific, density functions:

$$\boldsymbol{x}_i \sim \sum_{k=1}^{K} \pi_k f_{ik}\left(\boldsymbol{x}_i|\mu_k, \boldsymbol{\Sigma}_k\right) \tag{4.11}$$

where π_k's are the mixing proportions or equivalently the size of the clusters, subject to standard constraints as expressed in equations 2.3 and 2.4, i.e. to be non-negative values and to sum up to one across classes.
The Log-likelihood function for the whole sample is then:

$$\log L = \sum_{i=1}^{N} \sum_{k=1}^{K} \pi_k f_{ik}\left(\boldsymbol{x}_i|\mu_k, \boldsymbol{\Sigma}_k\right) \tag{4.12}$$

The estimation of the free parameters can be obtained by maximizing the equation 4.12 under the constraints on the mixing proportions π_k, and under the condition that $|\boldsymbol{\Sigma}_k| > 0$ for all the classes. This last condition is necessary since consistent estimators are not possible when $\boldsymbol{\Sigma}_k$ is not singular [Jedidi et al. 1997b]. Moreover, this condition entails a minimum sample size of $\frac{P(P+1)}{2}$ units within each group.

A modified EM algorithm (cf. subsection 2.4.2) is used to solve the maximization problem.
Once the estimates ($\widehat{\pi}_k, \widehat{\mu}_k$ and $\widehat{\boldsymbol{\Sigma}}_k$) of the parameters are obtained, it is possible to apply Bayes' theorem [Bayes 1763/1958] to estimate the posterior probability of memberships of each unit in each latent class:

$$\rho_{ik} = \frac{\hat{\pi}_k f_{ik}\left(\boldsymbol{x}_i|\widehat{\mu}_{ik}, \widehat{\boldsymbol{\Sigma}}_k\right)}{\sum_{k=1}^{K} \hat{\pi}_k f_{ik}\left(\boldsymbol{x}_i|\hat{\mu}_k, \widehat{\boldsymbol{\Sigma}}_k\right)} \tag{4.13}$$

Basing on the ρ_{ik}, a fuzzy classification of the units is obtained. Moreover, K local models, one for each class are defined according to the parameters estimated through an EM algorithm.

Since an EM algorithm is used to estimate the mixing components, all the drawbacks and the positive aspects of the EM algorithm are still valid (cf. subsection 2.4.2). Namely, even if the EM algorithm always assures convergence, it has a tendency to fall into a local optimum. For this reason, several starting values have to be tested in order to choose the best estimates of the mixing components. Moreover, the problem of the convergence in local optimum seems to increase in importance when the number of parameters to be estimated is high, that is often the case in complex Structural Equation Models.
Another problem affecting the STEMM algorithm is that the number of classes to take into account has to be decided *a priori*. If *a priori* information is not available, STEMM needs to be performed with successive numbers of classes. All the available procedures to select the

number of classes to take into account defined in subsection 2.4.3, are still available in the STEMM framework. Among them the Akaike's Information Criteria (AIC), the Controlled Criterion (CAIC) and the Bayesian Information Criterion (BIC). The model for which the chosen criterion is the smallest is selected.

Also the usual indexes to assess class separation in Mixture Models (cf. subsection 2.4.4) are still available in STEMM. In particular the entropy index (EN) as described in equation 2.22 is the most widely used also in the STEMM context.

To conclude Jedidi *et al.* assess that the STEMM is equivalent to multigroup Structural Equation Modeling [Jöreskog 1971] when the number of groups and the membership values are known *a priori* [Jedidi et al. 1997b].

4.2.2 Bayesian Finite Mixtures in SEM-ML

In 2001, Zhu and Lee proposed a Bayesian analysis to Finite Mixture in the LISREL-type models [Zhu & Lee 2001]. Since then, other works have been presented on this topic, in particular in 2002 Lee and Song developed a Bayesian approach to analyze mixtures in Structural Equation Models with an unknown number of classes (i.e. the components of the mixture) [S.Y.Lee & Song 2002].
More recently Lee [2007] published a complete work on the Bayesian approach to Structural Equation Models.

Defining the measurement and the structural models for each unit in each latent class as usual, i.e:

$$\boldsymbol{x}_i^{(M)}|k = \boldsymbol{\nu}_k^{(M)} + \boldsymbol{\Lambda}_k^{(M)}\boldsymbol{\xi}_{ik}^{(M)} + \boldsymbol{\epsilon}_{ik}^{(M)}, \tag{4.14}$$

$$\boldsymbol{x}_i^{(J)}|k = \boldsymbol{\nu}_k^{(J)} + \boldsymbol{\Lambda}_k^{(J)}\boldsymbol{\xi}_{ik}^{(J)} + \boldsymbol{\epsilon}_{ik}^{(J)} \tag{4.15}$$

and

$$\left(\boldsymbol{I} - \boldsymbol{B}_k^{(J)}\right)\boldsymbol{\xi}_{ik}^{(J)} = \boldsymbol{B}_k^{(M)}\boldsymbol{\xi}_{ik}^{(M)} + \boldsymbol{\zeta}_{ik} \tag{4.16}$$

where $\boldsymbol{x}_i^{(M)}|k$ and $\boldsymbol{x}_i^{(J)}|k$ are the vectors of the manifest variables for the i-th unit in the k-th latent class respectively associated to the exogenous and to the endogenous latent variables, $\boldsymbol{\xi}_{ik}$ is the vector of the generic latent variable for the i-th unit in the k-th latent class (with $\boldsymbol{\xi}_{ik}^{(J)}$ and $\boldsymbol{\xi}_{ik}^{(M)}$ defining respectively, the vector of the endogenous and the exogenous latent variables), $\boldsymbol{\epsilon}_{ik}^{(M)}$ and $\boldsymbol{\epsilon}_{ik}^{(J)}$ are the measurement residuals associated to the exogenous and to the endogenous blocks for the i-th unit in the k-th latent class, and $\boldsymbol{\zeta}_{ik}$ are the the structural residuals for the i-th unit in the k-th latent class. Remembering that $\boldsymbol{\Lambda}_k^{(M)}$ and $\boldsymbol{\Lambda}_k^{(J)}$ are the matrices containing the group-specific parameters of the measurement model, that $\boldsymbol{B}_k^{(M)}$ and $\boldsymbol{B}_k^{(J)}$ are the matrices containing the group-specific parameters of the structural model, that $\boldsymbol{\mu}_k$ is a vector containing the group-specific means for the manifest

variables and that $\boldsymbol{\Phi}_k$, $\boldsymbol{\Theta}_k^{(M)}$, $\boldsymbol{\Theta}_k^{(J)}$ and $\boldsymbol{\Psi}_k$ are the covariance matrices, respectively, of the exogenous latent variables, of the measurement errors and of the structural errors as expressed in equations 4.5, 4.6 and 4.8.

The idea is that for each unit the $\boldsymbol{x}_i$ arises from a mixture of distributions, the unconditional distribution of $\boldsymbol{x}_i$ can be written as:

$$f\left(\boldsymbol{x}_i|\boldsymbol{\Omega}_k\right) = \sum_{k=1}^{K} \pi_k f_{ik}\left(\boldsymbol{x}_{ik}|\boldsymbol{\Omega}_k\right) \tag{4.17}$$

where $\boldsymbol{\Omega}_k$ contains all the unknown parameters of the model, i.e. $\boldsymbol{\Lambda}_k^{(M)}$, $\boldsymbol{\Lambda}_k^{(M)}$, $\boldsymbol{B}_k^{(M)}$, $\boldsymbol{B}_k^{(J)}$, $\boldsymbol{\Phi}_k$, $\boldsymbol{\Theta}_k^{(M)}$, $\boldsymbol{\Theta}_k^{(M)}$ and $\boldsymbol{\Psi}_k$, as well as the main vector $\boldsymbol{\mu}_k$, and the mixing proportion π_k,

$$\boldsymbol{\Omega}_k = \left(\boldsymbol{\mu}_k, \pi_k, \boldsymbol{\Lambda}_k^{(M)}, \boldsymbol{\Lambda}_k^{(J)}\boldsymbol{B}_k^{(M)}, \boldsymbol{B}_k^{(J)}, \boldsymbol{\Phi}_k, \boldsymbol{\Theta}_k^{(M)}, \boldsymbol{\Theta}_k^{(J)}\boldsymbol{\Psi}_k\right).$$

Assuming the membership values correspond to the i-th unit, z_i is a latent allocation variable *i.i.d.* as a multinomial with probabilities π_k:

$$p\left(z_i = k|\boldsymbol{X}\right) = \pi_k \tag{4.18}$$

Standard Bayesian analysis provides an easy evaluation of the posterior distribution $p\left(\boldsymbol{\Omega}|\boldsymbol{X}\right)$. Nevertheless, since SEM Mixture Models are more complex models, the Bayesian estimation of $p\left(\boldsymbol{\Omega}|\boldsymbol{X}\right)$ is here more complicated. As a matter of fact, the standard problem of the

Bayesian analysis needs to take into account not only the manifest variables, but also the latent variable scores and the membership values in the posterior analysis. That is why, in this context the aim is to provide an evaluation of the posterior distribution of $(\boldsymbol{\Omega}, \boldsymbol{\Xi}, \boldsymbol{Z})$ given $\boldsymbol{X}$, where $\boldsymbol{\Xi}$ is the matrix containing both the exogenous and the endogenous latent variable scores, and $\boldsymbol{Z}$ is the matrix containing the membership values. In other words here we are interested in analyzing the $P(\boldsymbol{\Omega}, \boldsymbol{\Xi}, \boldsymbol{Z}|\boldsymbol{X})$.

The Bayesian estimates of $\boldsymbol{\Omega}$ and $\boldsymbol{\Xi}$ are obtained by computing the posterior means of $\boldsymbol{\Omega}$ and $\boldsymbol{\Xi}$ in the posterior distribution of $(\boldsymbol{\Omega}, \boldsymbol{\Xi}, \boldsymbol{Z}|\boldsymbol{X})$. This is done by simulating a sufficiently large sample of observations from this posterior distribution, in order to approximate the Bayesian estimates by the sample means.

Usually, in Bayesian Analysis applied to SEMs, a Gibbs sampler [Geman & Geman 1984] is used to generate the sample of observations from $p(\boldsymbol{\Omega}, \boldsymbol{\Xi}, \boldsymbol{Z}|\boldsymbol{X})$. A detailed discussion on this procedure goes beyond the aim of this work. For more details please refer to [Lee 2007]. Nevertheless, here a brief overview of the used procedure is given.

At the t-th iteration with current values $\boldsymbol{\Omega}^{(t)}$, $\boldsymbol{\Xi}^{(t)}$ and $\boldsymbol{Z}^{(t)}$, the Gibbs sampler procedure can be summarized in three steps:

(a) generate $\left(\boldsymbol{Z}^{(t+1)}, \boldsymbol{\Xi}^{(t+1)}\right)$ from $p\left(\boldsymbol{\Xi}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\Omega}^{(t)}\right)$

(b) generate $\boldsymbol{\Omega}^{(t+1)}$ from $p\left(\boldsymbol{\Omega}|\boldsymbol{X}, \boldsymbol{\Xi}^{(t+1)}, \boldsymbol{Z}^{(t+1)}\right)$

(c) Reorder the label through the permutation sampler to fulfill the identifiability.

Moreover, since $p(\boldsymbol{\Xi}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\Omega}) = p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\Omega})\, p(\boldsymbol{\Xi}|\boldsymbol{X}, \boldsymbol{Z\Omega})$, the step (a) can be decomposed into two sub-steps:

$(\mathbf{a}_1)$ generate $\boldsymbol{Z}^{(t+1)}$ from $p\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\Omega}^{(t)}\right)$

$(\mathbf{a}_2)$ generate $\boldsymbol{\Xi}^{(t+1)}$ from $p\left(\boldsymbol{\Xi}|\boldsymbol{X}, \boldsymbol{Z}^{(t+1)}\boldsymbol{\Omega}^{(t)}\right)$

Under several assumptions, such as that there are no cross-group constraints,

$$\left\{\left(\boldsymbol{\Omega}^{(j)}, \boldsymbol{\Xi}^{(j)}, \boldsymbol{Z}^{(j)}\right), j = 1, \cdots, J\right\} \tag{4.19}$$

are the observations of $(\boldsymbol{\Omega}, \boldsymbol{\Xi}, \boldsymbol{Z})$ generated by the Gibbs sampler from the posterior distribution of $(\boldsymbol{\Omega}, \boldsymbol{\Xi}, \boldsymbol{Z}|\boldsymbol{X})$.
The Bayesian estimates of $\boldsymbol{\Omega}$ and $\boldsymbol{\Xi}$ are then obtained by sample means of the generated observations, i.e.:

$$\widehat{\boldsymbol{\Omega}} = G^{-1} \sum_{g=1}^{G} \boldsymbol{\Omega}^{(g)}, \tag{4.20}$$

and

$$\widehat{\boldsymbol{\Xi}} = G^{-1} \sum_{g=1}^{G} \boldsymbol{\Xi}^{(g)} \tag{4.21}$$

with G is the total number of observations estimated by the Gibbs sampler procedure. These are consistent estimates of the corresponding posterior means. Moreover, it is possible to obtain estimates also for the parameters covariance matrix, as well as for the latent variable matrix. To conclude, it is possible to use simulated observations to compute other statistical inferences (such as deriving confidence intervals) on the latent variable scores and on the model parameters.

Once the estimate of $\boldsymbol{\Omega}$ and $\boldsymbol{\Xi}$ are obtained via the Gibbs sampler according to the latter equations, an approximation of the posterior probability $p\left(z_i = k|\boldsymbol{X}\right)$ can be obtained for each class:

$$p\left(z_i = k|\boldsymbol{X}\right) \approx G^{-1} \sum_{g=1}^{G} I\left(z_i^{(g)} = k\right) \tag{4.22}$$

A Bayesian classification of the units can be reached using the membership values contained in $\boldsymbol{Z}$. As a matter of fact, using a "percentage correctly classified" loss function (see [Richardson & Green 1997, Zhu & Lee 2001]), a Bayesian classification of the i-th unit is:

$$\hat{z}_i = \arg\max_k \left\{p\left(z_i = k|\boldsymbol{X}\right)\right\} \tag{4.23}$$

This technique allows us also to compute a Bayesian classification of a new observation i^* not used to define the classes. Since the inclusion of a new vector of the manifest variables $\boldsymbol{x}_{i^*}$ changes the posterior distribution, for each given class $\bar{k}$ an approximation of the posterior probability associated to the new unit can be obtained as:

$$P\left(z_{i^*} = \bar{k}|\boldsymbol{X}, \boldsymbol{x}_{i^*}\right) \approx G^{-1} \sum_{g=1}^{G} \frac{\pi_{i\bar{k}}^{(g)} f_{i\bar{k}}\left(\boldsymbol{x}_{i^*}|\boldsymbol{\mu}_{\bar{k}}^{(g)}, \boldsymbol{\Omega}_{\bar{k}}^{(g)}\right)}{\sum_{k=1}^{K} \pi_k^{(g)} f_{ik}\left(\boldsymbol{x}_{i^*}|\boldsymbol{\mu}_k^{(g)}, \boldsymbol{\Omega}_k^{(g)}\right)} \tag{4.24}$$

4.3 Unobserved Heterogeneity in PLS-PM

Similarly to classical *covariance-based* methods, also PLS Path Modeling (cf. subsection 3.4.1) assumes homogeneity over the observed set of units: all units are supposed to be well represented by a unique model estimated on all the units.
Nevertheless, in many cases it is reasonable to expect that different classes showing heterogeneous behaviors may exist in the observed set of units. In these cases, treating all units as a single class may lead to biased results both in terms of model parameters and of validation indexes [Jedidi et al. 1997a, Jedidi et al. 1997b].
Recently, several works have been proposed to deal with unobserved heterogeneity in PLS Path Modeling framework [Hahn et al. 2002, Ringle, Wende & Will 2005, Squillacciotti 2005, Trinchera & Esposito Vinzi 2006, Trinchera et al. 2006, Sanchez & Aluja 2006, Trinchera 2007, Sanchez & Aluja 2007, Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Ringle et al. 2008, Squillacciotti 2008, Esposito Vinzi, Amato & Trinchera 2008]. To the author's knowledge, four approaches exist to handle heterogeneity in the PLS-PM: the Finite Mixture PLS, proposed by Hahn et al. [2002] and modified by Ringle et al. [2008], the PLS Typological Path Model presented by Squillacciotti [2005] and modified by Trinchera & Esposito Vinzi [2006] and Trinchera et al. [2006], the PATHMOX by Sanchez & Aluja [2006] and the PLS-PM based Clustering (PLS-PMC) by Ringle & Schlittgen [2007].
In this section all these methods will be discussed in detail. Properties

and drawbacks of each approach will be analyzed and compared. Moreover, a new and original approach to detect homogeneous groups of units in PLS-PM, i.e. the Response Based Unit Segmentation in PLS-PM, will be presented in the next chapter (cf. chapter 5).

4.3.1 The Finite Mixture PLS

As described by Hahn *et al.* [Hahn et al. 2002], FInite MIXture PLS (FIMIX-PLS) is an extension of the Finite Mixture Models in SEM-ML (cf. subsection 4.2.1) to a PLS-PM framework. This technique joins a Finite Mixture procedure (cf. section 2.4) with an EM algorithm (cf. subsection 2.4.2), which specifically concerns the PLS-PM predictions, obtained by means of classical OLS regressions.
FIMIX-PLS is based on the assumption that if separate classes of units exist, the unobserved heterogeneity will be concentrated in the structural model, i.e. in the relationships among latent variables. The measurement model is therefore kept constant among detected classes. As in STEMM, the population is supposed to be the mixture of two or more sub-populations (hereby called classes), each characterized by a different distribution, and mixed in different proportions. The aim is to identify the probability of each unit to belong to each class, as well as to estimate model parameters within each detected class.

The first step of FIMIX-PLS consists in estimating the defined path model on all units through a standard PLS-PM algorithm (cf. subsection 3.4.1), i.e. to estimate the so-called *global model*. Then, the

estimated latent variable scores are used to detect the classes by an EM based procedure. In order to ensure model identification, a normality assumption is required, at least at the endogenous latent variable level.
Moreover, in FIMIX-PLS, heterogeneity is supposed to be concentrated only in the structural model. Therefore, in order to identify the classes and calculate the latent variable scores, the measurement model is kept constant over the iterations. In other words, in all local models, the outer weights are constant and equal to those obtained for the global model.

In more formal terms, FIMIX-PLS assumes that the vector of the J endogenous latent variables for the i-th observation ($\boldsymbol{\xi}_i^{(J)}$) is distributed as a finite mixture of conditional multivariate normal densities $f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\phi}_k\right)$, i.e.:

$$\boldsymbol{\xi}_{ik}^{(J)} \sim f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\phi}_k\right) \tag{4.25}$$

$$\boldsymbol{\xi}_i^{(J)} = \sum_{k=1}^{K} \pi_k f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\Omega}_k\right) \tag{4.26}$$

where, as usual $\boldsymbol{\phi}_k = (\pi_k, \boldsymbol{\Omega}_k)$ is the vector of all unknown parameters in the k-th class, and π_k's are the mixing proportions subject to the usual constraints as expressed in equation 2.3 and 2.4, i.e. to be non-negative values and to sum up to one across classes.

In a Finite Mixture Model applied to PLS Path Modeling the parameters to be estimated are the $(J \times J)$ matrix of the path coefficients linking the endogenous latent variables to each other, $(\boldsymbol{B}_k^{(J)})$, the $(J \times M)$ matrix of the path coefficients linking each endogenous variable to the exogenous ones, $(\boldsymbol{B}_k^{(M)})$, as well as the variances from each regression in the structural model.
The vector $\boldsymbol{\Omega}$ is so composed by the vector of the M exogenous latent variables in the inner models $\boldsymbol{\xi}_i^{(M)}$ and by the parameters to be estimated, i.e. by: $\boldsymbol{B}_k^{(J)}$, $\boldsymbol{B}_k^{(M)}$, and by the diagonal matrix of the exogenous latent variable variance $\boldsymbol{\Psi}_k$.
Therefore, vector $\boldsymbol{\Omega}$ is :

$$\boldsymbol{\Omega} = \left(\boldsymbol{\xi}_i^{(M)}, \boldsymbol{B}_k^{(J)}, \boldsymbol{B}_k^{(J)}, \boldsymbol{\Psi}_k\right) \tag{4.27}$$

Assuming multivariate normal density distribution for the vector of the endogenous latent variables $(\boldsymbol{\xi}_i^{(J)})$, and keeping in mind that the structural model as defined in subsection 3.4.1 can be expressed for the generic k-th class as:

$$\boldsymbol{\xi}_i^{(J)} \widetilde{\boldsymbol{B}}_k^{(J)} + \boldsymbol{\xi}_i^{(M)} \boldsymbol{B}_k^{(M)} = \boldsymbol{\zeta}_i \tag{4.28}$$

where$\widetilde{\boldsymbol{B}}_k^{(J)} = \left(\boldsymbol{I} - \boldsymbol{B}_k^{(J)}\right)$.
The equation 4.25 can be rewritten as:

$$\boldsymbol{\xi}_i^{(J)} \sim \sum_{k=1}^{K} \pi_k \left[\frac{\left|\widetilde{\boldsymbol{B}}_k^{(J)}\right|}{\sqrt[J]{2\pi}\sqrt{|\boldsymbol{\Psi}_k|}} e^{\frac{1}{2}\left(\left(\boldsymbol{\xi}_i^{(J)}\widetilde{\boldsymbol{B}}_k^{(J)} + \boldsymbol{\xi}_i^{(M)}\boldsymbol{B}_k^{(M)}\right)^T \boldsymbol{\Psi}_k^{-1} \left(\boldsymbol{\xi}_i^{(J)}\widetilde{\boldsymbol{B}}_k^{(J)} + \boldsymbol{\xi}_i^{(M)}\boldsymbol{B}_k^{(M)}\right)\right)} \right] \tag{4.29}$$

The likelihood function and the log-likelihood function as expressed in equation 2.5 and 2.6 can be reformulated for the N vectors $\left(\boldsymbol{\xi}_1^{(J)}, \boldsymbol{\xi}_2^{(J)} \cdots \boldsymbol{\xi}_N^{(J)}\right)$ as:

$$L\left(\boldsymbol{\phi}; \boldsymbol{\xi}^{(J)}\right) = \tag{4.30}$$
$$\prod_{i=1}^{N}\left[\sum_{k=1}^{K}\pi_k\left[\frac{\left|\widetilde{\boldsymbol{B}}_k^{(J)}\right|}{\sqrt[J]{2\pi}\sqrt{|\boldsymbol{\Psi}_k|}}e^{\frac{1}{2}\left(\left(\boldsymbol{\xi}_i^{(J)}\widetilde{\boldsymbol{B}}_k^{(J)}+\boldsymbol{\xi}_i^{(M)}\boldsymbol{B}_k^{(M)}\right)^T\boldsymbol{\Psi}_k^{-1}\left(\boldsymbol{\xi}_i^{(J)}\widetilde{\boldsymbol{B}}_k^{(J)}+\boldsymbol{\xi}_i^{(M)}\boldsymbol{B}_k^{(M)}\right)\right)}\right]\right]$$

and

$$\log L\left(\boldsymbol{\phi}; \boldsymbol{\xi}^{(J)}\right) = \tag{4.31}$$
$$\sum_{i=1}^{N}\sum_{k=1}^{K}\log\left(\pi_k\left[\frac{\left|\widetilde{\boldsymbol{B}}_k^{(J)}\right|}{\sqrt[J]{2\pi}\sqrt{|\boldsymbol{\Psi}_k|}}e^{\frac{1}{2}\left(\left(\boldsymbol{\xi}_i^{(J)}\widetilde{\boldsymbol{B}}_k^{(J)}+\boldsymbol{\xi}_i^{(M)}\boldsymbol{B}_k^{(M)}\right)^T\boldsymbol{\Psi}_k^{-1}\left(\boldsymbol{\xi}_i^{(J)}\widetilde{\boldsymbol{B}}_k^{(J)}+\boldsymbol{\xi}_i^{(M)}\boldsymbol{B}_k^{(M)}\right)\right)}\right]\right)$$

In FIMIX-PLS, an EM algorithm (cf. subsection 2.4.2) is used to maximize the likelihood function expressed in equation 4.30. An exhaustive description of this procedure will be given later in this subsection.

Once the class specific parameters of the model, $\boldsymbol{B}_k^{(J)}$, $\boldsymbol{B}_k^{(M)}$, $\boldsymbol{\Psi}_k$, are estimated through the EM algorithm, a fuzzy clustering of the units can be obtained.
The posterior probability of each unit to belong to each detected latent class (ρ_{ik}) is computed by means of the Bayes' theorem [Bayes

1763/1958] as:

$$\rho_{ik} = \frac{\pi_k f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\xi}_i^{(M)}, \widehat{\boldsymbol{B}}_k^{(J)}, \widehat{\boldsymbol{B}}_k^{(M)}, \widehat{\boldsymbol{\Psi}}_k\right)}{\sum_{k=1}^{K} \pi_k f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\xi}_i^{(M)}, \widehat{\boldsymbol{B}}_k^{(M)}, \widehat{\boldsymbol{B}}_k^{(M)}, \widehat{\boldsymbol{\Psi}}_k\right)} \tag{4.32}$$

In FIMIX-PLS the number of classes is not known *a priori* nor is included as a parameter in the estimation process. In order to detect the optimal partition, FIMIX-PLS has to be repeated using each time a different choice for the number of classes K, i.e. $K = 1$, $K = 2$, $K = 3 \ldots$.

Since Mixture Models are not asymptotically distributed as a *chi-square* and consequently the Likelihood Ratio Test (LRT) has not an asymptotically full rank quadratic form, then the LRT statistic is not valid [Aitkin & Rubin 1985, Titterington 1990].

For a detailed discussion on how to choose the appropriate number of classes to be considered in a Mixture Model, please refer to the subsection 2.4.3.

All the available procedures defined in subsection 2.4.3 are still available in a FIMIX-PLS framework. As a matter of fact, both Hahn et al. [2002] and Ringle et al. [2008] suggest using different indexes to choose the number of classes to be considered, such as the Akaike's Information Criteria (AIC), the Controlled Criterion (CAIC) and the Bayesian Information Criterion (BIC). The model with the smallest values for the chosen criteria is selected.

Also the usual indexes to assess class separation in the Mixture Model (cf. subsection 2.4.4) are still available in FIMIX-PLS. In particular, the entropy index (EN) as described in equation 2.22 is the most widely used in FIMIX-PLS.
As usual, the EN value will increase with the improvement of class separation. As a matter of fact, values higher than 0.5 indicate an unambiguous segmentation. Experience has however shown that the choice of the appropriate number of classes is not at all straightforward in empirical applications. When trying FIMIX-PLS with an increasing number of classes, the method may lead to unacceptable results, such as R^2 higher than 1 or negative variances. These problems appear especially when class sizes are too small [Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008].

Another drawback of FIMIX-PLS is that this technique only focuses on the heterogeneity concentrated in the structural model. If units differ with respect to the measurement model, FIMIX-PLS is not able to capture this source of heterogeneity as long as the structural coefficients are similar among the classes. In other words, the outer weights for the local models, i.e. the weights linking the latent variables to the correspondent manifest variables, are the same as in the global model. To overcome this problem, Ringle et al. [2008] propose to perform an *ex-post* analysis in the last step of the procedure. The *ex-post* analysis consists in looking for an external variable able to lead to the same classes as those identified by FIMIX-PLS. Once this variable is

detected, multi-group PLS-PM is performed over these *a priori* segmented data leading to class-specific latent variable scores, as well as different measurement and structural models. Very rarely, however, is possible to find one (or few) external variables leading unambiguously to the same classes as indicated by FIMIX-PLS.

Finally, the main issue concerning FIMIX-PLS is that it requires conditional multivariate normal density assumptions for predicted latent variable scores, at least for the endogenous latent variables. The PLS Path Model, instead, is a distribution-free technique that does not require assumptions on manifest variables: hence the estimated latent variable scores are unlikely to follow a normal distribution except for the case of latent variables corresponding to super-blocks in PLS hierarchical models that may approach a normal distribution even if both the manifest variables and the other latent variable scores are far from being normal [Tenenhaus, Mauger & Guinot 2008]. That is why it will be suitable to apply such a distribution-free clustering technique to obtain unit clustering in the PLS-PM framework.

An EM formulation for the FIMIX-PLS

The EM algorithm is a widely-used procedure to estimate likelihood parameters in Mixture Models. For a detailed discussion on the EM algorithm and of its features please refer to the subsection 2.4.2. Here, an application of the EM algorithm to the PLS-PM framework is described.

FIMIX-PLS firstly estimates both endogenous and exogenous latent variable scores by applying standard PLS-PM to the whole set of units. In a second stage, the estimated scores are used to perform a set of regressions between endogenous and exogenous latent variables according to the path structure.

As well known, the EM algorithm provides a solution to the maximum likelihood estimation in *incomplete-data* frameworks. In the Mixture Models the unobserved data to be replaced are the membership values, π_k's, and the additional information to be added is the expected membership values, z_{ik}'s. Each iteration of the EM algorithm is composed of two steps, the E-step (Expectation Step) in which the expectations of the membership value, z_{ik}, are computed given a provisional estimate of $\boldsymbol{\omega}$, and the M-step in which the expectation of the log-likelihood obtained in E-step is maximized with respect to the parameters.

Once the data is "completed" by means of the z's values, and assuming that the vector $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iK})$ is *i.i.d.* as a multinomial with probabilities π_k, the complete-data log-likelihood function defined in equation 4.31 can be rewritten as:

$$\log L\left(\boldsymbol{\phi}; \boldsymbol{\xi}^{(J)}\right) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \log\left(f_{ik}\left(\boldsymbol{\xi}_i^{(J)} | \boldsymbol{\xi}_i^{(M)}, \widetilde{\boldsymbol{B}}_k^{(J)}, \boldsymbol{B}_k^{(M)}, \boldsymbol{\Psi}_k\right)\right) + \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \log\left(\pi_k\right) \tag{4.33}$$

In the first E-step the expectation of the log-likelihood expressed in equation 4.33 is evaluated assuming such provisional estimates $\widetilde{\boldsymbol{B}}_k^{(J)*}$, $\boldsymbol{B}_k^{(M)*}$, $\boldsymbol{\Psi}_k^*$, and π_k^* for the parameters $\widetilde{\boldsymbol{B}}_k^{(J)}$, $\boldsymbol{B}_k^{(M)}$, $\boldsymbol{\Psi}_k$, and π_k respectively. These estimates can be easily obtained by a random sample of the membership values π_k, or on the basis of prior knowledge or analysis of the classes and/or the coefficients.

The expectation of the log-likelihood function expressed in 4.33 is:

$$\begin{aligned} E\left(\log L\left(\boldsymbol{\phi}^*;\boldsymbol{\xi}^{(J)}\right)\right) &= \sum_{i=1}^{N}\sum_{k=1}^{K} E\left(z_{ik};\boldsymbol{\xi}_i^{(M)},\pi_k^*,\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*|\boldsymbol{\xi}_i^{(J)}\right) \\ &\quad \log\left(f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\xi}_i^{(M)},\boldsymbol{\Pi}_k^*\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*\right)\right) \\ &\quad + \sum_{i=1}^{N}\sum_{k=1}^{K} E\left(z_{ik};\boldsymbol{\xi}^{(M)},\pi_k^*,\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*|\boldsymbol{\xi}^{(J)}\right)\log\left(\pi_k^*\right) \end{aligned} \tag{4.34}$$

where the conditional expectation of z_{ik} with $\boldsymbol{B}_k^{(M)*}$, $\widetilde{\boldsymbol{B}}_k^{(J)*}$, $\boldsymbol{\Psi}_k^*$, and π_k^* fixed, can be calculated as:

$$E\left(z_{ik};\boldsymbol{\xi}^{(M)},\pi_k^*,\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*|\boldsymbol{\xi}^{(J)}\right) = \frac{\pi_k^* f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\xi}_i^{(M)},\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*\right)}{\sum_{k=1}^{K}\pi_k^* f_{ik}\left(\boldsymbol{\xi}_i^{(J)}|\boldsymbol{\xi}_i^{(M)},\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*\right)} \tag{4.35}$$

Having defined $E\left(z_{ik};\boldsymbol{\xi}^{(M)},\pi_k^*,\widetilde{\boldsymbol{B}}_k^{(J)*},\boldsymbol{B}_k^{(M)*},\boldsymbol{\Psi}_k^*|\boldsymbol{\xi}^{(J)}\right)$ as above, the equation 4.33 is maximized in the M-step of the algorithm in order to obtain new provisional estimates of the parameters $\boldsymbol{B}_k^{(J)}$, $\widetilde{\boldsymbol{B}}_k^{(J)}$, $\boldsymbol{\Psi}_k$, and π_k. These new parameter estimates are then used in a subsequent

E-step to obtain update estimates of z_{ik} according to equation 4.35, and so forth.
The two steps are alternated until there is convergence on the increase of the log-likelihood function value.

In the M-step provisional estimates of the parameters $\widetilde{\boldsymbol{B}}_k^{(J)*}$, $\boldsymbol{B}_k^{(M)*}$, $\boldsymbol{\Psi}_k$, and π_k are obtained through a number of independent OLS regressions according to the path model scheme.
In particular, one OLS regression is performed for each endogenous latent variable $\boldsymbol{\xi}_j^{(J)}$. For a detailed formulation of the M-step in FIMIX, please refer to Hahn et al. [2002].

Since an EM algorithm is used in FIMIX-PLS to estimate the mixing components, all the drawbacks and the positive aspects of the EM algorithm are still valid. Namely, the EM algorithm, even if it always assures convergence, has a tendency to fall in a local optimum. For this reason several starting values have to be tested in order to choose the best estimates of the mixing components. Moreover, the problem of the convergence in local optimum seems to increase in importance when the number of parameters to be estimated is large, that is often the case in complex PLS Path Models.

4.3.2 The PATHMOX algorithm

Path Modeling Segmentation Tree algorithm (PATHMOX algorithm) was recently presented by Sanchez & Aluja [2007]. It provides a path

model tree having a decision tree-like structure. Each node in the decision tree-like structure is represented by a local PLS Path Model. The PATHMOX algorithm uses external concomitant variables, such as socio-demographical variables, to split units. Nevertheless, the split order is obtained taking into account the capacity of each concomitant variable to identify local models as different as possible. In other words, the clustering is obtained by means of external information, and it is somehow optimized with respect to the model.
The algorithm starts by estimating the global model at the root node. Then, all the possible two-way splits obtained by the categories of the concomitant variables are investigated. The several obtained local models are compared first of all as regards the structural models via a test for comparing the path coefficients. In addition, the diversity among measurement models is assessed. Once the best segmentation variable is detected, the algorithm provides a new estimation of the local models in each node. The process goes on by looking for new segmentation variables able to provide the best split.
Usually the number of units in a node, as well as the significance level for the best split, are considered as stopping criterion.

In more formal terms, consider all the categories of a concomitant variable and all the possible two-way splits of these categories, then for each of those two-way splits, units are divided into two groups of size n_1 and n_2, respectively.
For each group the structural model is estimated taking into account

the latent variable scores computed at the previous step. The "potential" structural model of the child nodes are compared with the structural model of the parent node. An extension of the test presented by Lebart, Morineau & Piron [1995] for testing the equality of two regression models has been developed for this purpose. In particular, under the null hypothesis all the path coefficients are assumed to be identical between two models, while alternative hypothesis assumes that two models are different as regards at least one of the path coefficients.
Define the structural models for the two models to be tested as:

$$\begin{aligned} \boldsymbol{\Xi}_1 &= \boldsymbol{\Xi}_1 \boldsymbol{B}_1 + \boldsymbol{E}_1 \\ &\text{and} \\ \boldsymbol{\Xi}_2 &= \boldsymbol{\Xi}_2 \boldsymbol{B}_2 + \boldsymbol{E}_2 \end{aligned} \tag{4.36}$$

where $\boldsymbol{\Xi}_k$ is the matrix of dimension N by Q containing all the latent variable scores, $\boldsymbol{\xi}_q$, computed for the k-th latent class, $\boldsymbol{B}_k$ is the matrix of dimension M by Q containing the path coefficients estimated for the k-th latent class, and $\boldsymbol{E}_k$ is the class specific matrix of dimension N by Q containing the residuals of all the regressions in the structural models.
Under the null hypothesis H_0 all the coefficients are to be considered equal, i.e.: $\boldsymbol{B}_1 = \boldsymbol{B}_2 = \boldsymbol{B}$.

In matrix notation H_0 and H_1 can be expressed as:

$$H_0 : \begin{bmatrix} \boldsymbol{\Xi}_1 \\ \boldsymbol{\Xi}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Xi}_1 \\ \boldsymbol{\Xi}_2 \end{bmatrix} [\boldsymbol{B}] + \begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \end{bmatrix} \tag{4.37}$$

$$H_1 : \begin{bmatrix} \boldsymbol{\Xi}_1 \\ \boldsymbol{\Xi}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Xi}_1 & 0 \\ 0 & \boldsymbol{\Xi}_2^{(M)} \end{bmatrix} \begin{bmatrix} \boldsymbol{B_1} \\ \boldsymbol{B_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \end{bmatrix} \tag{4.38}$$

The sum of the squared errors (SSE) is computed under H_0 and under H_1 and a test statistic is computed as:

$$F = \frac{(n^* - 2M)}{M} \frac{SSE_0 - SSE_1}{SSE_1} \tag{4.39}$$

where M is the total number of exogenous latent variables in the model and $n^* = NJ$ with J equal to the number of endogenous latent variables in the model and N equal to the total number of units in the sample, with $N = n_1 + n_2$. This test statistic distributes approximately as an F distribution with M and $(n^* - 2M)$ degrees of freedom.
The partition showing the most significant p-value is considered a candidate for the best split.
The same process is repeated for each concomitant variable selecting the partition with the minimum p-value among all the candidates as the optimal split.

Once the child node is identified, the child model and the parent model are compared as regards the measurement models. A Ryan-Joiner test

[Ryan & Joiner 1976] is used to assess how close to unity the correlation between the latent variable scores in the parent node and the latent variable scores in the child node is. The basic idea underlining this procedure is that, if the measurement models are the same between the two tested models, then, the correlation between the estimated latent variable scores in the parent node and the estimated latent variable scores in the child node have to be as high as possible, i.e. close to the unity.

With respect to the other clustering techniques presented in this chapter, PATHMOX directly uses concomitant variables to classify the units. Even if it provides local models that are different as regards the structural and the measurement models, the unit clustering is not made using the model directly. If no external/concomitant variables other than the manifest variables used in the model are available, this technique is no more applicable. Moreover, the test used to compare structural models requires the normality assumption on the structural model residuals.

4.3.3 The PLS Typological Path Model

The PLS Typological Path Model (PLS-TPM) [Squillacciotti 2005, Trinchera & Esposito Vinzi 2006, Trinchera et al. 2006] algorithm aims at overcoming the drawbacks of FIMIX-PLS, namely the normality assumption on latent variable scores and the assumption that unobserved heterogeneity is focused only in the structural model. This

method is an iterative algorithm firstly proposed by Squillacciotti [2005]. It allows us to estimate at the same time both the memberships of units to classes and the parameters of the local models without making any kind of distributional assumption.

The algorithm iteratively assigns the units to the classes corresponding to the closest local model, according to a measure which stems from the $DModY$ distance used in PLS Regression [Tenenhaus 1998] and the $DModY, N$ index in PLS Typological Regression (PLS-TR) [Esposito Vinzi & Lauro 2003]. In particular, the $DModY$ distance used in PLS Regression measures the distance between the i-th unit and the regression model in the space spanned by the endogenous variables using the model residuals. For more details, please refer to Tenenhaus [1998]. In PLS Typological Regression, instead, Esposito Vinzi & Lauro [2003] following the $DModY$ optic, define a distance measure to cluster units in PLS Regression framework taking into account the predictive purpose of the PLS Regression. The $DModY, N$ index developed for PLS Typological Regression is defined as:

$$DModY, N_k = \sqrt{\frac{\frac{\sum_{j=1}^{J}\left[r_{ijk}^2/Rd(T_k,y_j)\right]}{(J-a_k)}}{\frac{\sum_{i=1}^{N}\sum_{j=1}^{J}\left[r_{ijk}^2/Rd(T_k,y_j)\right]}{(n_k-a_k-1)(J-a_k)}}} \tag{4.40}$$

where J is the number of endogenous variables, a_k is the number of extracted components in the k-th class, $Rd\,(T_k, y_j)$ is the portion of the j-th endogenous variable variance explained by the a_k components within the k-th class, and r_{ijk}^2 is the square of the i-th residual for the

j-th endogenous variable in the PLS Regression model estimated for the k-th class.

An extension of this distance measure is used in PLS Typological Path Model, i.e.:

$$D_{ik} = \sqrt{\frac{\frac{\sum_{p=1}^{P_{j^*}}\left[v_{ipj^*k}^2 / Rd\left(\boldsymbol{\xi}_{j^*k}^{(J)}, \boldsymbol{x}_{pj^*k}\right)\right]}{\left(P_{j^*} - a_k\right)}}{\frac{\sum_{i=1}^{N}\sum_{p=1}^{P_{j^*}}\left[v_{ipj^*k}^2 / Rd\left(\boldsymbol{\xi}_{j^*k}^{(J)}, \boldsymbol{x}_{pj^*k}\right)\right]}{(n_k - a_k - 1)\left(P_{j^*} - a_k\right)}}} \tag{4.41}$$

where j^* is a target endogenous block, a_k is the number of exogenous latent variables linked to the target block in the local model estimated for the k-th latent class, $Rd\left(\boldsymbol{\xi}_{j^*k}^{(J)}, \boldsymbol{x}_{pj^*k}\right)$ is the redundancy index (cf. subsection 3.4.1) computed for the p-th manifest variable linked to the target block j^* in the k-th latent class, and v_{ipj^*k} is the i-th residual of the "redundancy" model in the k-th latent class.

The redundancy residuals v_{ipj^*k} are obtained as the residuals of the regression of each p-th manifest variable linked to the target block j^* over the target endogenous latent variable scores ($\boldsymbol{\xi}_{j^*k}^{(J)}$) estimated for the k-th latent class. A redundancy residual is computed for each unit with respect to each latent class, i.e. at each iteration NK redundancy residuals are computed.

It is important to notice that the chosen measure of unit-model distance in PLS-TPM requires the presence of a well-identified target latent variable among the J endogenous latent variables. Neverthe-

less, in a PLS Path Model the identification of a unique endogenous target latent variable, on which to compute the unit-model distance defined in equation 4.41, is not always possible.

In the original formulation of the PLS-TPM by [Squillacciotti 2005], the algorithm starts by estimating the global model over the whole set of observed units. After having randomly assigned the units to a previously chosen number K of classes, the starting local models are estimated, and the distances between each unit and each local model are computed. Units are then re-assigned to the closest local model on the basis of the unit-model distance measure in 4.41. If this leads to modifications in the classes' composition, updated local models are estimated and new distances computed. The algorithm repeats these steps until stability is reached on the classes' compositions. The final local models are then compared and classes are eventually characterized by means of available external (concomitant) variables that, however, do not play any explorative role in the identification of the classes.
Nevertheless, once again the number of classes is not considered as a parameter to be estimated. As a matter of fact, the main problem with this procedure concerns the choice of the number of classes. As in all latent class detection procedures presented so far, the appropriate number of classes is generally *a priori* unknown, and PLS-TPM has to be repeated with different values of K in order to choose the optimal partition. Differently from FIMIX-PLS, however, neither indicators to

assess class separation nor Information criteria are available in PLS-TPM. In fact, it does not lead to a "fuzzy" clustering but rather to a "hard" one and is not model based. This makes the choice of the number of classes to retain more difficult.

In a more recent version of the algorithm Trinchera & Esposito Vinzi [2006] propose to obtain the number of classes to retain and the initial unit assignment to classes by means of a hierarchical classification over the redundancy residuals (v_{ipj^*k}) computed for the global model. Once the number of classes is identified, units are iteratively assigned to the class corresponding to the closest local model, according to the distance measure defined in equation 4.41. Models concerning the classes are re-estimated at each iteration, leading to a dynamic re-estimation of all elements in the models (path coefficients, latent variable scores, outer weights, etc.). This leads to final local models that are different with respect to both the measurement and the structural models. Stability of results in terms of class' compositions is considered as a stopping criterion.
In the modified PLS-TPM by Trinchera & Esposito Vinzi [2006] the number of classes to take into account is not chosen *a priori* by the users, but it is directly obtained by the algorithm. This allows us to apply PLS-TPM even if prior information on the number of classes is not available.

In both the formulations, the PLS-TPM approach leads to a clustering

of the units according to the specified path model. Nevertheless, the units' assignment to classes is obtained only according to the structural and the measurement model for the target block. As underlined by Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus [2008] "*Obtained local models lead only to a higher predictivity in terms of R^2 value associated to the target latent variable*", but not to better local models in a more general meaning. Moreover, PLS-TPM is applicable only to PLS Path Models including only reflective indicators.

4.3.4 The PLS Path Model based Clustering

Recently, Ringle & Schlittgen [2007] presented a new class of methods for clustering in PLS-PM: the PLS-PMC (PLS Path Model based Clustering) methods. The idea is to use model residuals to improve an initial partition of units according to the model features.
The initial unit partition can be obtained either by using Genetic Algorithms [Cowgill, Harvey & Watson 1999] as the case in PLS Genetic Algorithm Segmentation (PLS-GAS) or by a random assignment of the units as in PLS-SPS.
A common drawback to all these methods is that the number of classes to take into account is not determined by the algorithm but has to be defined by comparing the models. The procedure needs to be repeated taking into account a successive number of latent classes, as in FIMIX-PLS and in the original formulation of the PLS Typological Path Model.
To conclude, further research on these new techniques has to be done

in order to evaluate the capacity of these techniques to detect latent classes.

4.4 Unobserved Heterogeneity in GSCA

Generalized Structured Component Analysis (GSCA) models are estimated under the assumption that data are homogeneous. Nevertheless, it is often more realistic to assume that units come from heterogeneous subgroups. Until now, only one method is available to handle this kind of problem in the GSCA framework: the Fuzzy Clusterwise Generalized Structured Component Analysis (FGSCA) by Hwang et al. [2007]. This method will be discussed in detail in the following subsection.

4.4.1 The Fuzzy GSCA

An extension of the Generalized Structured Component Analysis models was recently presented by Hwang et al. [2007] combining fuzzy clustering with GSCA: the Fuzzy Clusterwise Generalized Structured Component Analysis (FCGSCA). This method allows us to obtain at the same time both a fuzzy clustering of units into overlapping clusters (cf. chapter 2) and GSCA parameters for each detected class. Given an *a priori* chosen number of classes K, for each class the matrix $\boldsymbol{Z}_k$ is a matrix containing the membership values of each unit in the k-th class, under the classical assumptions of fuzzy clustering, i.e. under the constraints that $0 \leq z_{ik} \leq 1$ and $\sum_{k=1}^{K} z_{ik} = 1$. Moreover, let

m be the fuzzifier, i.e. the predetermined fuzzy weight scalar which influences the degree of fuzziness of the solution.
Then, the equation to be minimized in GSCA as expressed in equations 3.84 and 3.85 can be rewritten taking into account the classes as:

$$\vartheta = SS\left(\boldsymbol{U}_k - \boldsymbol{U}_k\boldsymbol{A}_k\right)_{\boldsymbol{Z}_k^m} \tag{4.42}$$

where:

$$\boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{W}_k \end{bmatrix} \boldsymbol{X} \tag{4.43}$$

and

$$\boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda}_k \\ \boldsymbol{0} & \boldsymbol{B}_k \end{bmatrix}, \tag{4.44}$$

are the matrices containing the class-specific model parameters (i.e. the external weights $\boldsymbol{w}_k$, the path coefficients $\boldsymbol{\beta}_k$ and the external loadings $\boldsymbol{\lambda}_k$), and $SS\left(\boldsymbol{X}\right)_{\boldsymbol{Y}} = trace\left(\boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{X}\right)$
Of course in the case of a unique class, *i.e.* if $K = 1$, the equation 4.42 reduces to equation 3.84. In other words, GSCA is a particular case of FCGSCA when the number of classes taken into account is equal to one.

Solving the minimizing problem expressed in equation 4.42 under the

constraints on the z_{ik}'s by using a Lagrangian multiplier (λ) is equivalent to minimizing:

$$L = \sum_{k=1}^{K} \sum_{i=1}^{N} z_{ik}^{m} r_{ik} - \lambda \left(\sum_{k=1}^{K} z_{ik} - 1 \right) \tag{4.45}$$

where r_{ik} is the class-specific residual obtained for the i-th unit, *i.e.* $r_{ik} = SS\left(\boldsymbol{U}_k - \boldsymbol{A}_k \boldsymbol{U}_k\right)$.
So, solving L with respect to z_{ik}:

$$\partial L / \partial z_{ik} = m z_{ik}^{m-1} r_{ik} - \lambda = 0 \tag{4.46}$$

and λ:

$$\partial L / \partial \lambda = \sum_{k=1}^{K} z_{ik} - 1 = 0 \tag{4.47}$$

leads to:

$$\hat{z}_{ik} = \left(\frac{\lambda}{m r_{ik}} \right)^{1/(m-1)} \tag{4.48}$$

and

$$\hat{\lambda} = \left(\left(\sum_{k=1}^{K} 1 / \left(m r_{ik} \right) \right)^{1/(m-1)} \right)^{1-m} \tag{4.49}$$

Inserting the equation 4.49 in the equation 4.48 allows us to obtain an estimate of the membership values, given the class-specific parameters:

$$\hat{z}_{ik^*} = \frac{1}{\sum_{k=1}^{K} \left(\frac{r_{ik^*}}{r_{ik}}\right)^{1} / (m-1)} \tag{4.50}$$

The class-specific estimates of the parameters, as well as the class-specific membership values, are obtained by an optimization procedure composed of two steps.
In the first step the class-specific parameters ($\boldsymbol{W}_k$ and $\boldsymbol{A}_k$) are estimated for fixed membership values. This is equivalent to minimizing:

$$\begin{aligned} \vartheta = & \sum_{k=1}^{K} SS\left((\boldsymbol{Z}_k^m)^{1/2}\left(\boldsymbol{X}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{W}_k\end{bmatrix} - \boldsymbol{X}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{W}_k\end{bmatrix}\boldsymbol{A}_k\right)\right) \\ = & \sum_{k=1}^{K} SS\left(\boldsymbol{X}_k\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{W}_k\end{bmatrix} - \boldsymbol{X}_k\boldsymbol{A}_k\boldsymbol{U}_k\right) \end{aligned} \tag{4.51}$$

where $\boldsymbol{X}_k = \left((\boldsymbol{Z}_k^m)^{1/2}\boldsymbol{X}\right)$. Equations expressed in 4.51 are nothing but the sum of the equation 3.85 of the GSCA across the K classes. Under such provisional membership values, an Alternating Least Squares algorithm is used as in GSCA to update the parameters estimates within each latent class.

Once the class-specific parameters are estimated, the membership values are updated using equation 4.50.

Since FCGCSA is an extension of GSCA, it bears all the drawbacks of the GSCA, such as improper solutions or indeterminacy of latent variable scores. In addition, FCGCSA requires a very large number of units to be applied. As a matter of fact, a minimum sample size of $P(P+1)/2$ in each latent class is needed in order to obtain a positive definite covariance matrix within each class [Wedel & Kamakura 2000]. Moreover, being based on the EM algorithm (cf. subsection 2.4.2) FCGCSA requires the data in each latent class to be normally distributed.

Nevertheless, also the good aspects of the GSCA are still available in the FCGSCA. As a matter of fact, the two quality indexes proposed in GSCA, the FIT and the AFIT (cf. subsection 3.4.2) can be rewritten taking into account the class-specific values in order to allow us to compare models:

$$FIT = 1 - \frac{SS\left(\boldsymbol{U}_k - \boldsymbol{U}_k\boldsymbol{A}_k\right)}{SS\left(\boldsymbol{U}_k\right)} \tag{4.52}$$

and

$$AFIT = 1 - \left(1 - FIT\right)\frac{df_0}{df_1} \tag{4.53}$$

where $df_0 = NP$ is the number of degrees of freedom for the model for which $\boldsymbol{W}_k = 0$ and $\boldsymbol{A}_k = 0$, and $df_1 = NP - fp$ are the degrees for the model to test and fp is the number of free parameters, including the unknown elements in $\boldsymbol{W}_k = 0$ and $\boldsymbol{A}_k = 0$, as well as all the membership values z_{ik}.

As for the GSCA, the AFIT index takes into account the model complexity in evaluating the fit of the model. Simpler models are usually preferred over complex models showing similar explanatory power. Nevertheless, the AFIT index is useful as long as $fp < NP$. Since fp depends on the number of classes considered, AFIT can be used only in the case of $K < P$. As a matter of fact, if $K = P$, the number of membership values becomes equivalent to NP, and so df_1 becomes equal to zero. This drawback is not so relevant in SEM models since the number of manifest variables (P) is usually large enough.

Moreover, FCGSCA provides also several indexes able to assess class separation as in Mixture Models based clustering techniques. Being based on fuzzy clustering, the FCGSCA borrows from classical fuzzy theory a number of cluster validity measures [Bezdek 1981, Roubens 1982]. In particular, according to the Rubens works, the Fuzziness Performance Index (FPI) and the Normalized Classification Entropy (NCE) are the most useful cluster validity measurements in the fuzzy clustering procedure. That is why, Hwang et al. [2007] suggest using these two indexes also in FCGSCA to assess how well the detected classes are separated. These two indexes are defined as:

$$FPI = 1 - \frac{(K \times PC - 1)}{(K - 1)} \tag{4.54}$$

and

$$NCE = \frac{PE}{\log K} \tag{4.55}$$

where:

$$PC = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{ik}^2 \tag{4.56}$$

is the Partition Coefficient of Bezdek [1974] and

$$PE = -\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{ik} \log z_{ik} \tag{4.57}$$

is the Partition Entropy [Bezdek 1974].
Both the indexes in equations 4.54 and 4.55 select the model yielding the smallest value.

Two main weak points affect the FCGSCA. The first one pertains to the number of classes to be considered. Once again the number of classes to taken into account is not a parameter of the model. Just like any other technique discussed in this chapter, if *a priori* information about the number of classes is not available, the method needs to be repeated for successive numbers of classes. Model quality indexes, as well as cluster validity measurements, are then used to select the "best" number of latent classes.
The second main drawback of the FCGSCA, common to any fuzzy

clustering analysis, is related to the choice of the fuzzy weight m. Until now, there has been no theoretical justifiable way to select m. As a matter of fact, m can be any of the values bounded between one and infinity. Nevertheless, values too close to one may lead to a non-overlapping clustering of the units, with membership values close to zero or to one. Instead, high m values lead to an excessively overlapping clustering, with membership values constant across classes and equal to $1/K$. As a consequence of this open-endedness, a m value of two is the most popular choice in fuzzy clustering [Bezdek 1981, Gordon 1999, Hruschka 1986, Steenkamp & Wedel 1993]. Also in FCGSCA $m = 2$ is the standard value for the fuzzy weight.

4.5 Assessment of model diversity

In a clustering framework, once the groups are identified it is very important to assess the differences (and similarities) among the detected classes of units. In our specific framework, this essentially entails comparing the obtained local models to one another and with the global model. Hence, in heterogeneous Structural Equation Models, group comparison can be seen as model comparison.

Since Structural Equation Models are very complex systems, comparing Structural Equation Models estimated on different groups of units is a very difficult task. As a matter of fact, and as stated by Liao [2002], various levels of comparison have to be taken into ac-

count when comparing Structural Equation Models. In particular Liao [2002], identified four different degrees of comparison in LISREL-type models:

- Comparing distribution
- Comparing data structure
- Comparing model structure
- Comparing model parameters

Here we focus our attention only on the last item, i.e. on comparing the model parameters. Moreover local models will be compared also as regards the goodness of fit and the latent variable scores. The model structure, i.e. the relationships involved in the model, are considered constant across the different classes.

In LISREL-type models, standard tests are available to compare two or more groups of units. In PLS-PM, in FIMIX and in GSCA context, instead, only non parametric procedures and resampling methods, such as a bootstrap based technique [Efron 1982], are available. Nevertheless, since the bootstrap procedure is not yet available in the case of two or more samples, and since the detected groups can be considered different samples, bootstrap empirical confidence intervals can not be used to compare the model parameters.
Other non parametric techniques have been developed for this purpose. Among them, the permutation tests allow us to compute test

statistics in a non parametric framework. For reference on permutation test please refer to Edgington [1987].

In this section first an overview of the permutation tests will be given (cf. subsection 4.5.1), then the different ways to compare model parameters in the several approaches to SEM will be discussed (cf. subsection 4.5.2). Furthermore, latent variable scores comparison (cf. subsection 4.5.3), as well as model quality comparison (cf. subsection 4.5.4), will be examined.

4.5.1 Permutation tests

Permutation tests [Edgington 1987] are based on the permutation of units among classes. In particular, let k_1, k_2 be two groups of units and s a statistic that allows us to test the null hypothesis H_0. Then, these tests need to compute the statistic s several times on different samples obtained by unit permutation in order to obtain an empirical distribution of the statistic s under the null hypothesis. H_0 is rejected if the p-value obtained by the empirical distribution is lower than a certain threshold α. In other words, H_0 is rejected if the value of the statistic s computed on the original groups, $s_{original}$, is an extreme value of the empirical distribution of the statistics s computed on the permuted data, $s_{permuted}$. The probability of $s_{original} < s_{permuted}$ is:

$$P\left(s_{original} < s_{permuted}\right) = \frac{1}{G+1}\left(\sum_{g=1}^{G} I\left(s_{original} < s_{permuted_g}\right) + 1\right) \tag{4.58}$$

where

$$I\left(s_{original} < s_{permuted_g}\right) = \begin{cases} 1 \text{ if } s_{original} < s_{permuted_g} \\ 0 \text{ if not} \end{cases} \tag{4.59}$$

and G is the number of random permutations.
The null hypothesis is rejected if the probability value expressed in equation 4.58 is lower than a certain value α.

As a matter of fact, the procedure behind a permutation test can be summarized by the following scheme:

1. The statistic s is computed in each sample, in our case a statistic s is computed for each of the two groups in order to obtain $s_{original}$

2. The units are grouped in a unique sample $k_1 \cup k_2$.

3. A random permutation of the unique sample is performed in order to obtain two groups of k_1^* and k_2^* units having the same size as the original groups.

4. The statistic s is computed for each of the permuted samples, $s_{permuted_g}$.

5. The steps 3 to 4 are repeated G times.

6. An empirical distribution of the $s_{permuted}$ under H_0 is obtained.

7. If the original value of the statistic s, $s_{original}$, is an extreme value of the $s_{permuted}$ distribution, then the null hypothesis is rejected.

This procedure is shown to be a valid technique to check hypothesis [Edgington 1987].

4.5.2 Comparing model parameters

Comparing model parameters is the easiest way to assess if two models are different. In a latent class context this means to define if the detected classes show different behaviors as regards the model parameters. Of course, different procedures are available to compare model parameters according to the estimation techniques used to estimate the Structural Equation Model. That is why in this subsection comparing model coefficients will be discussed for each of the estimation techniques presented in chapter 3.

Comparing coefficients in LISREL-type models

In LISREL-type techniques, it is possible to test if the detected latent classes meet the assumption that they are equal among the groups by examining whether different matrices in the model (which represent sets of path coefficients) are "invariant". In other words, it is possible to test whether the matrices of the coefficients in the model are equal across the groups.
Model parameters could be tested at different degrees, i.e. different assumptions of class equality can be tested. As a matter of fact,

assuming that the model has the same form across the different classes, it is possible to test model equality at the path coefficients level, as well as at external weights level, or at the level of the covariance matrices of the errors in the models. Moreover, they are usually tested in a particular order [Bollen 1989], i.e. from the least restrictive test to the hypothesis imposing the most constraints on the parameters. Usually, supposing the structure of the model be the same across groups, the order of the tests for equality in Structural Equation Model is:

$H_{\mathbf{\Lambda}}$ test for equality in the measurement models, i.e.: $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2$.

$H_{\mathbf{\Lambda B}}$ test for equality in the structural models, given equal measurement models, i.e.: $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2$, $\boldsymbol{B}_1 = \boldsymbol{B}_2$.

$H_{\mathbf{\Lambda B \Theta}}$ test for equality in the covariance matrix of the measurement errors, given equal measurement and structural coefficients,i.e.: $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2$, $\boldsymbol{B}_1 = \boldsymbol{B}_2$, $\mathbf{\Theta}_1 = \mathbf{\Theta}_2$.

$H_{\mathbf{\Lambda B \Theta \Psi}}$ test for the equality in the covariance matrix of the structural errors, given equal measurement loadings and structural coefficients, as well as similar covariance matrix of the measurement errors, i.e.: $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2$, $\boldsymbol{B}_1 = \boldsymbol{B}_2$, $\mathbf{\Theta}_1 = \mathbf{\Theta}_2$, $\mathbf{\Psi}_1 = \mathbf{\Psi}_2$.

$H_{\mathbf{\Lambda B \Theta \Psi \Phi}}$ test for equality in the covariance matrix of the exogenous latent variables, given equal all the other parameters in the models, i.e: $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2$, $\boldsymbol{B}_1 = \boldsymbol{B}_2$, $\mathbf{\Theta}_1 = \mathbf{\Theta}_2$, $\mathbf{\Psi}_1 = \mathbf{\Psi}_2$, $\mathbf{\Phi}_1 = \mathbf{\Phi}_2$.

This last test is the most restrictive hypothesis. Under $H_{\mathbf{\Lambda B \Theta \Psi \Phi}}$ all parameter matrices are constrained to be the same among groups. If

$H_{\boldsymbol{\Lambda B \Theta \Psi \Phi}}$ is accepted, the results are consistent with the assumption that parameters have the same level in the two groups. In other words the model is invariant among the groups.
Of course, this is not a restrictive order. If a particular hypothesis has to be tested, the order in which parameter equalities are tested can be altered. Nevertheless, once the testing hierarchy is established, it is possible to perform the tests and assess which degree of invariance best fits the data.

Whatever the level at which the test is performed, independently from the H_0 under which the test is performed, the same procedure is applied.
As a matter of fact, in all the cases the test is performed on the group-specific covariance matrix ($\boldsymbol{S}_k$). Under each of the above specified hypothesis an implied covariance matrix can be estimated as a function of the model's parameters, i.e. $\boldsymbol{\Sigma}_k\left(\widehat{\boldsymbol{\Omega}}_k\right)$, where $\boldsymbol{\Omega}_k$ is the matrix of model parameters. Let $\widehat{\boldsymbol{\Sigma}}_k = \boldsymbol{\Sigma}_k\left(\widehat{\boldsymbol{\Omega}}_k\right)$, the closer the $\widehat{\boldsymbol{\Sigma}}_k$ is to the $\boldsymbol{S}_k$'s for all groups, the better the model fit. The global fit function is obtained as a weighted combination of the fit for all the groups, with weights equal to relative group sizes, i.e:

$$F = \sum_{k=1}^{K} \frac{n_k}{N} F_k\left(\boldsymbol{S}_k, \widehat{\boldsymbol{\Sigma}}_k\right) \tag{4.60}$$

where F_k is a general fit function defined according to the different estimation modes (for example ML or GLS) in LISREL-type Struc-

tural Equation Models (cf. subsection 3.3.1 and equation 3.11), n_k is the size of the k-th group and N is the total number of units, with $N = n_1 + \cdots n_K$ in the case of K groups.

Under the null hypothesis, the constraints in all the groups are corrected. The $(N-1)\,F$ following a *chi-square* distribution with $\left(\frac{KP(P+1)}{2} - fp\right)$ degrees of freedom, where fp is the number of independent parameters is estimated in all the groups.
Moreover, since the hierarchy of testing hypothesis presented above contains nested models, and since the differences in *chi-squares* for nested models is distributed as a *chi-square* with degrees of freedom equal to the difference in the degrees of freedom for the two models (that in the case of the nested model is equal to one), then it is easy to perform the test while scrolling the hierarchy up and down.
The null hypothesis of equality among the groups, for a given testing level, is rejected if the χ^2 value is higher then a given α value, i.e. if the associated p-value is small.

Comparing coefficients in PLS-PM

In the PLS Path Modeling approach several methods are available to compare the parameters of models having the same structure. Usually only path coefficients are taken into account when comparing several models. Nevertheless, many of the presented procedures can be easily extended to the external weights.
Four approaches have been developed to test if differences in model

parameters across groups is significant. These are:

- the classical (parametric) approach,
- the non-parametric approach,
- the permutation based approach,
- the moderator variable based approach.

Each of these approaches will be discussed in details.

Parametric approach

This approach is based on standard bootstrap techniques [Efron 1982]. For each group, the parameter to be investigated linking the m-th exogenous variable to the j-th endogenous one, β_{mj}, is estimated by performing standard PLS Path Modeling analysis. Then, the standard deviation ($s^2_{\beta_{mj}}$) of each estimated parameter β_{mj_k} is estimated by means of bootstrapping. The following test statistic is calculated:

$$t = \frac{\beta_{mj_1} - \beta_{mj_2}}{\sqrt{\frac{(n_1-1)^2}{n_1+n_2-2} \cdot {s_{\beta_{mj_1}}}^2 + \frac{(n_2-1)^2}{n_1+n_2-2} \cdot s^2_{\beta_{mj_2}}} \cdot \sqrt{\frac{1}{n_1} + \frac{2}{n_2}}} \tag{4.61}$$

Under several distribution assumptions, such as the normality of the residuals, the test statistic defined in equation 4.61 is asymptotically distributed as a *Student* with $(n_1 + n_2 - 2)$ degrees of freedom.
A parametric test can be performed and the null hypothesis on the equality of coefficients tested.

This procedure is relatively easy to be applied, nevertheless it requires a distributional assumption, at least on the residuals. As shown by Henseler & Fassott [2007] this assumption does not always hold. For this reason, the use of this procedure to assess differences among model parameters has to be carefully evaluated.

Non-parametric approach

This approach was recently presented by Henseler & Fassott [2007]. It is a bridge between the parametric and the permutation approaches. The basic idea is to obtain, by means of bootstrapping the empirical cumulative distribution of the parameters of interest. The procedure requires four steps, that are:

1. For each group, estimate the parameter of interest, and fix the null hypothesis.
   For instance, supposing there are two groups, and that the path coefficient linking the m-th exogenous latent variable to the j-th endogenous one is greater in group one than in group two, i.e. $H_0 : \beta_{mj_1} > \beta_{mj_2}$.

2. For each group, build G bootstrap samples and compute the G estimates for the parameter of interest.

3. Build all the possible combinations (G^K) of the bootstrap parameters across groups, in the case of two groups we will have G^2 possible combinations.

4. Count how often, in the G^K combinations, the null hypothesis is rejected. In our case, count how often the path coefficient of group one is smaller than or equal to the one estimated for group two.

The relative frequency of these counts reflects the error probability, i.e. the probability that in the population the path coefficient computed for group one is smaller than or equal to the one computed for group two:

$$P\left(\beta_{mj_1} > \beta_{mj_2}\right) = 1 - \frac{1}{G^2}\sum_{g=1}^{G}\sum_{s=1}^{G} I\left(\beta^g_{mj_1} \leq \beta^s_{mj_2}\right) \tag{4.62}$$

where $\beta^g_{mj_1}$ is the parameter estimated for group one in the g-th bootstrap sample, and I is a boolean function with:

$$I\left(\beta^g_{mj_1} \leq \beta^s_{mj_2}\right) = \begin{cases} 1 \text{ if } \beta^g_{mj_1} \leq \beta^s_{mj_2} \\ 0 \text{ otherwise} \end{cases} \tag{4.63}$$

This method is easily applied by using the available software for PLS-PM analysis together with a spreadsheet software. Moreover, it does not require any distributional assumption. Nevertheless, increasing the number of classes directly increases the number of bootstrap sample combinations to take into account.

Permutation tests approach

Chin in 2003 proposed to apply permutation tests to compare path coefficients of PLS Path Models estimated on different samples [Chin 2003].
In the case of two groups, the null hypothesis to be tested is:

$$H_0\colon\ \beta_{mj_1} = \beta_{mj_2} \tag{4.64}$$

where, as usual, β_{mj_1} and β_{mj_2} are the path coefficients linking the m-th exogenous latent variable to the j-th endogenous latent variable in group one and in group two. As described in subsection 4.5.1, a s statistic needs to be identified. Chin [2003] in his work does not specify the used statistic. As a matter of fact, since the aim is to test if the path coefficients are different across groups, both the $s = \left|\beta_{mj_1} - \beta_{mj_2}\right|$ and the $s = \left(\beta_{mj_1} - \beta_{mj_2}\right)^2$ can be considered in a permutation procedure as expressed in 4.5.1.
Then a permutation test procedure as expressed in subsection 4.5.1 is performed.

Using moderating variables to assess differences among model parameters

Moderating variables are categorical or metric variables influencing the relationship, in terms of strength and/or direction, between an endogenous and an exogenous variable (fig. 4.1) [Baron & Kenny 1986]. Following this idea, group effects are nothing else than a moderating effect of a categorical moderating variable expressing group member-

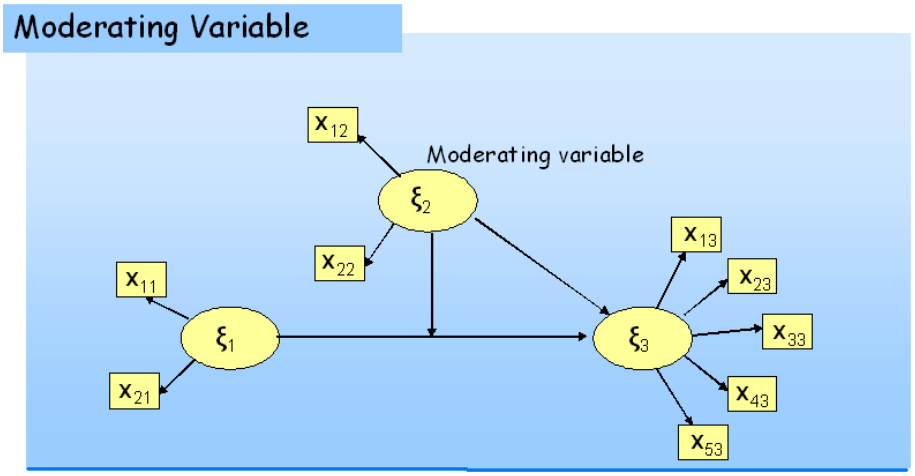

Figure 4.1: *Moderating Variable in a simple SEM*

ship. Several solutions have been proposed to consider moderating variable in regression-like techniques. Nevertheless, usually a moderator effect is modeled by taking into account product terms considering the effects of the moderating variables.

In other words, moderating variables have been integrated in Structural Equation Models by adding a so-called interaction term as an additional latent variable in the model [Kenny & Judd 1984]. In a simple model, with only one exogenous variable and one endogenous variable, the interaction term is obtained as the product of the manifest variables linked to the exogenous latent variable and the moderating variable (fig. 4.2). In such a model, it does not matter which

variable is moderating and which one is the exogenous one. Moreover, problems arise in the interpretation of the product term.

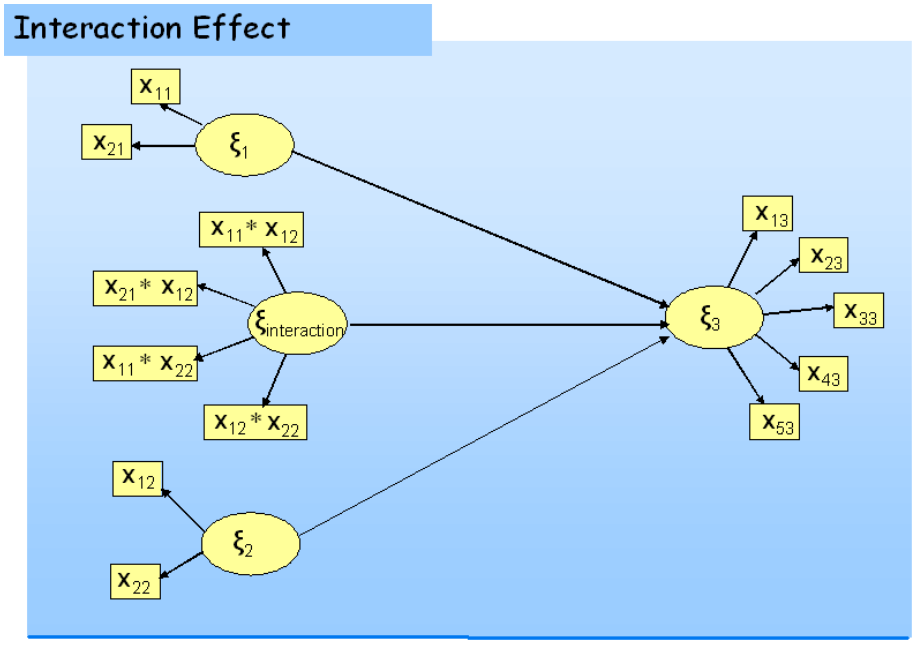

Figure 4.2: *Creating interaction term in a simple SEM by product*

A first attempt to take into account moderating variables in PLS-PM by including interaction effects was made by Chin, Marcolin & Newsted [2003]. Since then, other proposals exist for modeling moderating effects in PLS-PM framework, as the one by Tenenhaus et al. [2008] and the one by Hensler & Fassott [2008].

As Hensler & Fassott [2008] suggest, in the case that the exogenous variable or the moderating variable is formative, the pairwise multiplication of the manifest variables is not feasible. In this case they propose to use a two step procedure to include product terms. In the first step they suggest performing PLS-PM by considering both the exogenous variable and the moderating variable as independent latent variables in the model. Once latent variable scores are estimated, the product term is computed as the elementwise product of the exogenous latent variable scores and the moderating latent variable scores. A multiple linear regression between the endogenous latent variable scores and the exogenous, the moderating, and the product term latent variable scores is then performed. The interaction effect is estimated. A scheme of procedure proposed by Hensler & Fassott [2008] is shown in figure 4.3.
Chin et al. [2003] suggest to assessing moderating by comparing the R^2 value, i.e. the proportion of the variance explained by the model, computed for the model without moderating effects with the R^2 value obtaining for the model taking into account interaction effects. The effect size, f^2, is computed as:

$$f^2 = \frac{R^2_{\text{model with moderating}} - R^2_{\text{model without moderating}}}{1 - R^2_{\text{model without moderating}}} \tag{4.65}$$

Moderating effects with an effect size f^2 of 0.02 are regarded as weak, an effect size between 0.15 and 0.35 as moderated and an effect size higher than 0.35 as strong [Chin et al. 2003]. Nevertheless, the au-

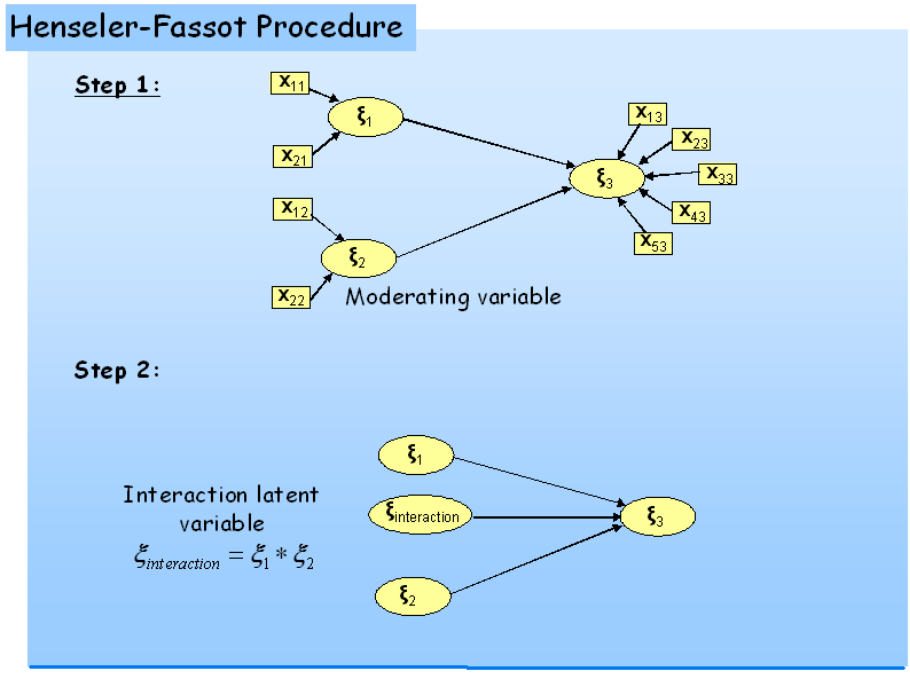

Figure 4.3: *Henseler and Fassot procedure to model interaction effect in a simple SEM with formative manifest variables*

thors stress that a lower effect size does not necessarily mean that the considered moderating effect is negligible.

The significance of the coefficient linked to the interaction effect can be tested also by means of bootstrap-based techniques [Hensler & Fassott 2008].

It is important to notice that, usually, when comparing model coefficients, information about model fit is not taken into account. Future research in this framework might be able to test differences across model parameters taking into account also the fit of the models.
Moreover, even if all these approaches have been developed in PLS-PM framework, they may be easily extended to Structural Equation Models estimated by ML or by GSCA.

Comparing coefficients in GSCA

Fuzzy Clusterwise Generalized Structured Analysis uses bootstrap techniques [Efron 1982] to compute standard errors of parameter estimates. Critical Ratios, obtained by dividing the parameter estimates by their standard errors, can be used to test the parameters significance without distributional hypothesis.
Nevertheless, no specific technique is available to compare parameters across groups. Techniques developed in PLS-PM framework could be easily extended to GSCA context.

4.5.3 Comparing latent variable scores

Especially in PLS-PM framework, latent variable scores assume a key role. As a matter of fact, PLS-PM directly provides latent variable scores, while in SEM-ML latent variable scores can only be obtained

indirectly. In any case, generic latent variable scores are obtained as:

$$\boldsymbol{\xi}_q = \frac{\sum_{p=1}^{P_q} w_{pq}\boldsymbol{x}_{pq}}{\sum_{p=1}^{P_q} w_{pq}} \tag{4.66}$$

Usually these values are scaled between 0 and 100 by means of the following equation:

$$\boldsymbol{\xi}_q^* = 100 \times \frac{\boldsymbol{\xi}_q - x_{min}}{x_{max} - x_{min}} \tag{4.67}$$

Two different hypotheses can be tested on the scaled values. Both tests are obtained by means of a non-parametric approach: the permutation tests [Edgington 1987] specified in subsection 4.5.1.

First of all, it is possible to look for differences across groups in latent variable means. Let $\widehat{\mu}_{\boldsymbol{\xi_{q1}^*}}$ be the mean value for the q-th latent variable in the first group, and $\widehat{\mu}_{\boldsymbol{\xi_{q2}^*}}$ the mean value for the same variable in the group two, then it is possible to compare the mean values under the null hypothesis that the two values are equal, i.e.:

$$H_0\colon\ \mu_{\boldsymbol{\xi}_{q1}^*} = \mu_{\boldsymbol{\xi}_{q2}^*} \tag{4.68}$$

where the latent variable mean values are estimated by means of the expectation of each latent variable in each class, i.e. $\mu_{\boldsymbol{\xi}_{q1}^*} = E\left(\boldsymbol{\xi}_{q1}^*\right)$ and $\mu_{\boldsymbol{\xi}_{q2}^*} = E\left(\boldsymbol{\xi}_{q2}^*\right)$, and a permutation test is used to perform the test.

As discussed in subsection 4.5.1, in a permutation test we have to

identify a statistic s allowing us to test the null hypothesis. Under the null hypothesis expressed in equation 4.68, the statistic s to be used is:

$$s = \left| E\left(\boldsymbol{\xi}^*_{q1}\right) - E\left(\boldsymbol{\xi}^*_{q2}\right) \right| \tag{4.69}$$

If the null hypothesis is rejected, differences between mean values allow us to assess that each group has a specific mean value for the q-th latent variable. Of course, for each latent variable a similar test has to be performed.

The same procedure can be applied to test differences across the groups as regards the variance values of the latent variables in the model. In this case the null hypothesis to be tested will be:

$$H_0\colon\ \sigma^2_{\boldsymbol{\xi}^*_{q1}} = \sigma^2_{\boldsymbol{\xi}^*_{q2}} \tag{4.70}$$

Under this null hypothesis, the statistic s to be tested becomes:

$$s = \left| S_{\boldsymbol{\xi}^*_{q1}} - S_{\boldsymbol{\xi}^*_{q2}} \right| \tag{4.71}$$

where S is the sample variance. The permutation tests performed on this statistic allow us to compare the group dispersion as regards the q-th latent variable scores.

4.5.4 Comparing model quality

Comparing the quality of Structural Equation Models requires applying different procedures as regards the estimation method used to estimate it. Moreover, since SEM are complex models, defining a quality index in a SEM framework requires taking into account several aspects. In this section an overview of the different methods available to compare model quality in the different Structural Equation frameworks will be provided. First of all comparing model quality in a LISREL-type framework will be presented. Later, the cases of both GSCA and PLS-PM will be considered. A new way to assess if the local models perform better than the global model will be introduced in chapter 5 (cf. section 5.3).

Comparing models in SEM-ML framework

As expressed in subsection 3.3.1 LISREL-type Structural Equation Models aim to reproduce the sample covariance matrix. In this framework the quality of a model is strictly related with its ability to fit the data. The various indexes proposed in subsection 3.3.1 can be easily compared across groups using standard parametric tests based on the likelihood ratio test.

Comparing models in PLS-PM framework

In PLS-PM framework, three different quality indexes are available as expressed in subsection 3.4.1. Jakobowicz [2007] in his doctoral thesis

proposed an extension of the Chin [2003] approach, presented in subsection 4.5.2, in order to compare models estimated on different groups of units. The basic idea is to apply permutation tests (cf. subsection 4.5.1) on the PLS-PM quality indexes under the null hypothesis that the quality of the model is the same across groups. In the case of two groups:

$$H_0\text{: } GF_1 = GF_2 \tag{4.72}$$

where GF_k is a quality index for the k-th group, with $k = 1, 2$.
Under this null hypothesis, the statistic to be used in a permutation test is expressed by:

$$s = |GF_1 - GF_2| \tag{4.73}$$

A permutation procedure as expressed in subsection 4.5.1 is applied, and the null hypothesis is rejected if the obtained p-value is lower than a certain value α, usually $\alpha = 0.05$.
The Jakobowicz [2007] procedure can be applied to all the quality indexes usually available in PLS-PM framework. Therefore, the GF can be one of the three indexes presented in subsection 3.4.1, i.e. the average communality index, the average redundancy index and the GoF index. The use of the different quality indexes as GF allows us to test differences in model quality with respect to the structural model (by using the redundancy index), to the measurement model (by using communality index) and to the whole model (by using the GoF index).

In latent class detection, the Jakobowicz [2007] procedure can be used to test differences between the performance of the global model and the performance of the local models in order to establish whether taking into account a group structure in the data would improve model quality.

Comparing models in GSCA framework

Fuzzy Clusterwise Generalized Structured Component Analysis furnishes an overall fit measure as discussed in subsection 3.4.2: the FIT index. However this index does not take into account model complexity. For this reason Hwang et al. [2007] propose to use a modified version of the FIT, i.e. the adjusted FIT expressed by equation 4.74.

$$AFIT = 1 - (1 - FIT)\,\frac{df_0}{df_1} \tag{4.74}$$

where $df_0 = NP$ are the degrees of freedom of the null model and $df_1 = NP - fp$ are the degrees of freedom of the model being tested, with fp equal to the number of free parameters including the unknown elements in the matrices of external weights ($\boldsymbol{W}$) and in the matrix of model coefficients ($\boldsymbol{A}$), as well as the membership values (z_{ik}).
A simpler model showing similar explanatory power is preferred to a more complex model. Moreover, a model showing a higher value of the $AFIT$ index has to be preferred over competing models. Nevertheless, $AFIT$ is valid until $fp < NP$, so until the number of classes consid-

ered is smaller than the number of manifest variables in the model. As a matter of fact, the number of free parameters becomes equal to NP in the case of $K = P$. This is not a serious drawback since SEM models usually take into account a larger number of manifest variables.

In this chapter a detailed overview of the available techniques to handle unobserve heterogeneity in Structural Equation Models has been provided. The different approaches have been discussed and the ways to assess model diversity shown. To the author's knowledge, no unique and complete discussion of the available techniques to detect latent classes in Structural Equation Models has been presented until now. The author wishes to make up for this gap with this work.
Moreover, in the next chapter (cf. chapter 5) a new method to obtain *response-based* clustering in PLS Path Modeling will be presented: the REBUS-PLS (REsponse Based Unit Segmentation in PLS path models) algorithm [Trinchera 2007, Trinchera, Squillacciotti, Esposito Vinzi & Tenenhaus 2007, Trinchera, Romano & Esposito Vinzi 2007, Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Esposito Vinzi, Amato & Trinchera 2008].

Chapter 5

The REBUS-PLS algorithm

5.1 Introduction

A new method for unobserved heterogeneity detection in PLS Path Modeling is proposed in this chapter: the REBUS-PLS (REsponse Based Unit Segmentation in PLS-PM) [Trinchera 2007, Trinchera, Squillacciotti, Esposito Vinzi & Tenenhaus 2007, Trinchera, Romano & Esposito Vinzi 2007, Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Esposito Vinzi, Amato & Trinchera 2008].
REBUS-PLS is an iterative algorithm, which allows us to estimate at the same time both the unit memberships to latent classes and the class specific parameters of the local models without making any kind of distributional assumption either on the manifest variables or on the latent variables. The core of the algorithm is a so-called *closeness measure* between units and models based on residuals and directly

developed by the author. The idea behind the definition of this new measure is that if latent classes exist, units belonging to the same latent class will have similar local models, i.e. similar performance as regards the global model.
Moreover, if a unit is assigned to the correct latent class, its performance in the local model computed for that specific class will be better than the performance obtained by the same unit considered as supplementary in all the other local models.

Unlike FIMIX-PLS (cf. subsection 4.3.1) and coherent with PLS Path Modeling features (cf. subsection 3.4.1), REBUS-PLS does not require distributional hypotheses. Moreover, REBUS-PLS may lead to local models that are different both in terms of structural and measurement models. Furthermore, differently from PLS Typological Path Model, REBUS-PLS involves a new "distance", taking into account all the endogenous latent variables and the measurement models of all blocks. Thus it removes the requirement of a well-identified target endogenous latent variable (cf. subsection 4.3.3). To conclude, unlike the PATHMOX, REBUS-PLS does not require external/concomitant variables to cluster the units.
Furthermore, the number of classes to take into account is directly defined by the algorithm. So, REBUS-PLS can be applied even if no *a priori* information on the number of latent classes to consider is available.

In this chapter the REBUS-PLS algorithm will be explained in detail (cf. section 5.2).
Moreover, the *Group Quality Index*, i.e. a new index to evaluate the obtained unit partition, will be presented in section 5.3.

5.2 The REsponse BAsed Unit Segmentation algorithm

Despite following the procedure defined in PLS Typological Path Modeling, the REBUS-PLS algorithm is based on a different logic. PLS Typological Path Modeling searches for classes optimizing the local model predictivity relative only to a target block (latent and manifest variables), hence leading to high values of R^2 associated with the target latent variable [Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008]. This is reflected in the choice of the "distance": units are assigned to the class corresponding to the local model minimizing the redundancy residuals (see equation 4.41).
In REBUS-PLS, instead, the "distance" of a unit from a model is defined by taking into account the model performance for both the structural and the measurement model. Since the "distance" measure chosen in PLS-TPM (see equation 4.41) is a sum of squared residuals, it would be better defined as a measure of *closeness* of units to the model, than to a "distance" measure. This is the reason why in REBUS-PLS, and in the rest of this work, a measure to assess distance between unit and model based on residuals is referred to as a *closeness*

measure (CM) rather than as a distance.

In order to obtain local models that fit better than the global model, the chosen *closeness measure* is defined according to the structure of the Goodness of Fit (GoF) index (cf. subsection 3.4.1), the only available measure of global fit for a PLS Path Model. The GoF index, as already presented in subsection 3.4.1, is defined as:

$$GOF = \sqrt{\frac{\sum_{q=1}^{Q}\sum_{p=1}^{P_q} Cor^2\left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q\right)}{\sum_{q=1}^{Q} P_q} \times \frac{\sum_{j=1}^{J} R^2\left(\boldsymbol{\xi}_j, \{\boldsymbol{\xi}_q^*\text{'s explaining } \boldsymbol{\xi}_j\}\right)}{J}} \tag{5.1}$$

The left product term in 5.1, the average communality index (see equation 3.68), can be considered as an index measuring the quality of the measurement models: it is obtained as the mean of the squared correlations linking each manifest variable ($\boldsymbol{x}_{pq}$) to the correspondent latent variable ($\boldsymbol{\xi}_q$) over all the Q blocks.
The term on the right side, the average R^2, is instead an index measuring the quality of the structural model (see equation 3.73).
Since the GoF is obtained as the geometric mean of the average communality and the average R^2 value, a model with a high GoF value shows a better performance on both the structural and measurement models.

Following the GoF structure, as expressed in equation 5.1, a new *closeness measure* between unit and model has been defined. This

index is based on the residuals of the communality model (i.e., the regressions of the manifest variables over their respective latent variables) and of the structural model (the regressions of the endogenous latent variables over their respective explanatory latent variables).
In more formal terms, the *closeness measure* (CM) of the i-th unit to the k-th local model, i.e. to the latent model corresponding to the k-th latent class, is defined as:

$$CM_{ik} = \sqrt{\frac{\sum_{q=1}^{Q}\sum_{p=1}^{P_q}\left[e_{ipqk}^2/Com\left(\boldsymbol{x}_{pq},\boldsymbol{\xi}_{qk}\right)\right]}{\sum_{i=1}^{N}\sum_{q=1}^{Q}\sum_{p=1}^{P_q}\left[e_{ipqk}^2/Com\left(\boldsymbol{x}_{pq},\boldsymbol{\xi}_{qk}\right)\right]} \times \frac{\sum_{j=1}^{J}\left[f_{ijk}^2/R^2\left(\boldsymbol{\xi}_j,\{\boldsymbol{\xi}_{q^*}\text{'s explaining }\boldsymbol{\xi}_j\}\right)\right]}{\sum_{i=1}^{N}\sum_{j=1}^{J}\left[f_{ijk}^2/R^2\left(\boldsymbol{\xi}_j,\{\boldsymbol{\xi}_{q^*}\text{'s explaining }\boldsymbol{\xi}_j\}\right)\right]}} \tag{5.2}$$

where:
$Com\left(\boldsymbol{x}_{pq},\boldsymbol{\xi}_{qk}\right)$ is the communality index computed following equation 3.67 for the p-th manifest variable of the q-th block in the k-th latent class;
e_{ipqk} is the measurement model residual for the i-th unit in the k-th latent class, corresponding to the p-th manifest variable in the q-th block, i.e. the communality residuals;
f_{ijk} is the structural model residual for the i-th unit in the k-th latent class, corresponding to the j-th endogenous block;
N is the total number of units;
m_k is the number of extracted components. Since all blocks are supposed to be reflective, this figure will always be equal to 1.

As for the GoF index, the left-side term of the product in equation 5.2 refers to the measurement models for all the Q blocks in the model (whereas in PLS Typological Path Model the measurement model is taken into account only for the target block), while the right-side term refers to the structural model. It is important to notice that both the measurement and the structural residuals are computed for each unit with respect to each local model regardless of the memberships of the units to the specific latent class.
The idea behind this is that if latent classes exist, units belonging to the same latent class will have similar local models, i.e. similar performance as regards the global model. Moreover, if a unit is assigned to the correct latent class, for example to class two out of five classes, its performance in the local model computed for the second class will be better than the performance obtained by the same unit considered as supplementary in all the other local models. That is why residuals of each unit from each local model are computed. In computing the residual from the k-th latent model, we expect that units belonging to the k-th latent class show smaller residuals than units belonging to the other $(K-1)$ latent classes.

Two kinds of residuals are used to evaluate the closeness between a unit and a model: the measurement (or communality) residuals and the structural residuals. The firsts are taken into account in order to evaluate unit performance as regards the measurement model, while the seconds check for homogeneity in the structural model.

In a reflective measurement scheme a communality residual is computed for each manifest variable in the model.
In more formal terms, the measurement residual of the i-th unit from the k-th local model, i.e. the local model computed for the k-th latent class, is obtained as:

$$e_{ipqk} = x_{ipq} - \hat{x}_{ipqk} \tag{5.3}$$

where x_{ipq} is the observed value of the p-th manifest variable in the q-th block for the i-th unit and $\hat{x}_{ipqk}$ is the estimated value of x_{ipq} in the k-th latent class.
Hence, for each unit i, $\hat{x}_{ipqk}$ is obtained by regression of $\boldsymbol{x}_{pq}$ on the q-th latent variable computed for the k-th latent class, i.e. $\boldsymbol{\xi}_{qk}$.
Thus as:

$$\hat{x}_{ipqk} = \lambda_{pqk}\xi_{iqk} \tag{5.4}$$

with λ_{pqk} representing the class-specific loading associated with the p-th manifest variable of the q-th block in the k-th latent class, and ξ_{iqk} being the score of the q-th latent variable for the i-th unit.
This last value is obtained by using the external weights estimated for the k-th latent class according to:

$$\xi_{iqk} = \sum_{p=1}^{P_q} w_{pqk}x_{ipq} \tag{5.5}$$

where w_{pqk} is the external weight linking the p-th manifest variable of the q-th block to the corresponding latent variable $\boldsymbol{\xi_{qk}}$ in the k-th local model.
The generic external weight w_{pqk} is obtained by performing a PLS Path Model on units belonging to the k-th latent class, i.e. it is class-specific. In other words, the communality residuals are the residuals of the simple regressions of each manifest variable on the corresponding latent variable computed according to the class-specific parameters.

The structural residuals f_{ijk}, instead, are computed in order to evaluate unit performance in the structural model.
They are obtained for each unit i as the difference between each endogenous latent variable score (i.e. the latent variable value estimated using external weights obtained by PLS Path Model performed for the k-th latent class, as defined in equation 5.5) and the inner estimation of the latent variable obtained by the path diagram relations (ϑ_{ijk}).
Therefore, the generic structural residual f_{ijk} is computed as:

$$f_{ijk} = \xi_{ijk} - \vartheta_{ijk} \tag{5.6}$$

where ξ_{ijk} is obtained according to 5.5 and ϑ_{ijk} is:

$$\vartheta_{ijk} = \sum_{q^*=1}^{Q^*\text{'s on } j} \beta_{q^*jk}\xi_{ijk} \tag{5.7}$$

where β_{q^*jk} is the path coefficient linking the q^*-th exogenous latent variable to the j-th endogenous latent variable computed in the k-th local model. In other words, the structural residuals are the residuals of the multiple regression of the endogenous latent variables on their exogenous latent variables. Consequently, a structural residual is computed for each endogenous latent variable in the model.

The choice of the *closeness measure* in equation 5.2 as a criterion for assigning units to classes has two major advantages.
Firstly, unobserved heterogeneity can now be detected in both the measurement and the structural models. If two models show identical structural coefficients, but differ with respect to one or more outer weights in the exogenous blocks, REBUS-PLS is able to identify this source of heterogeneity, which might be of major importance in practical applications (cf. chapter 6).
Moreover, since the *closeness measure* is defined according to the structure of the Goodness of Fit (GoF) index, the identified local models will show a higher value for both the GoF and the R^2 indexes (cf. chapter 6).

Nevertheless, the CM expressed by equation 5.2 is only the core of an iterative algorithm allowing us to obtain a *response-based* clustering of the units.
As a matter of fact, REBUS-PLS is an iterative algorithm that starting from the global model allows us to detect local models performing

better than the global model (cf. figure 5.1). The steps of the REBUS-

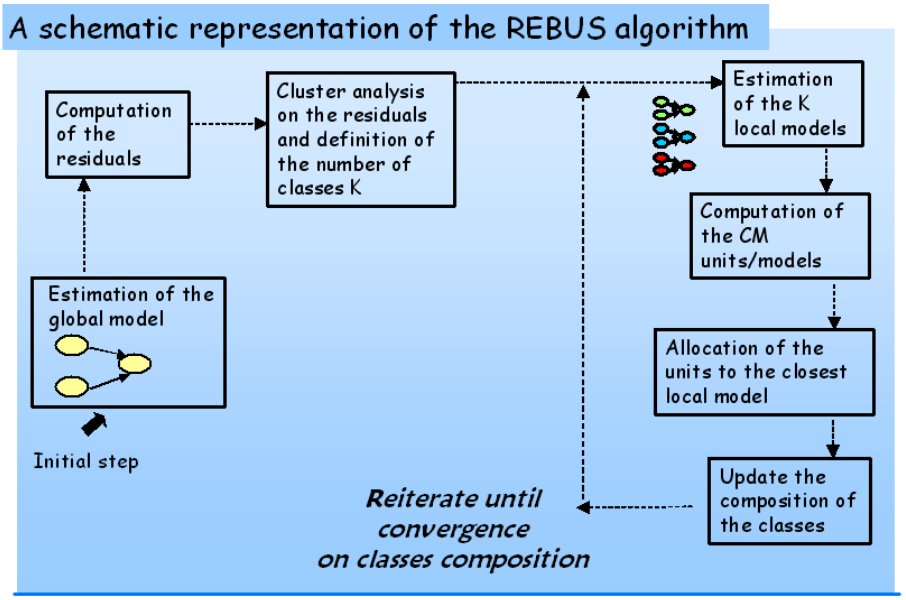

Figure 5.1: *A schematic representation of the REBUS-PLS algorithm*

PLS algorithm overlap the ones of the PLS Typological Path Modeling [Trinchera & Esposito Vinzi 2006]. However, as already said, the two methods are different as regards the way in which they look for latent classes among units.

The first step of the REBUS-PLS algorithm involves computing the global model on all the observed units, by performing a simple PLS

Path Modeling analysis. In the second step, the communality and the structural residuals of each unit from the global model are obtained according to equations 5.3 and 5.6.
The number of classes (K) to be taken into account during the successive iterations and the initial composition of the classes are obtained by performing a hierarchical cluster analysis on the computed residuals (both from the measurement and the structural models).
Once the number of classes to consider and the initial composition of the classes are obtained, a PLS Path Modeling analysis is performed on each formed class and K provisional local models are estimated.
The group-specific parameters computed at the previous step are so used to compute the communality and the structural residuals of each unit from each local model according to equations 5.3 and 5.6. Then the CM of each unit from each local model is obtained according to equation 5.2.
Each unit is, therefore, assigned to the closest local model, i.e. to the model from which shows the smaller CM value. Once the composition of the classes is updated, K new local models are estimated.
The algorithm goes on until the threshold of a stopping rule is achieved.

A schematic representation of the REBUS-PLS algorithm is shown in figure 5.1 and in the algorithm 2.

As in PLS-TPM, stability on class composition from one iteration to the other is considered as a stopping rule. The author suggests using

Algorithm 2 REBUS-PLS algorithm

Input: $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_q, \ldots, \boldsymbol{X}_Q]$ standardized MV's;
Output: $\boldsymbol{\beta}_{jk}$, $\boldsymbol{w}_{qk}$, $\boldsymbol{\xi}_{qk}$, $\boldsymbol{Z}$;

1: Estimate a global PLS Path Model
2: **for all** $i = 1, \ldots, N$ **do**
3: Compute the communality and structural residuals as:
4: $f_{ijk} = \xi_{ijk} - \vartheta_{ijk}$ and $e_{ipqk} = x_{ipq} - \hat{x}_{ipqk}$
5: **end for**
6: Perform a hierarchical cluster analysis on the residuals computed at step 2
7: Choose the number of classes (K) to take into account according to the dendograme obtained at step 3 and assignment of units to the class according to the cluster analysis results.
8: **for all** $k = 1, \ldots, K$ **do**
9: Estimate the local PLS Path Model
10: **end for**
11: **for all** $i = 1, \ldots, N$ **do**
12: **for all** $k = 1, \ldots, K$ **do**
13: Compute the communality and structural residuals as:
14: $f_{ij} = \xi_{ij} - \vartheta_{ij}$ and $e_{ipq} = x_{ipq} - \hat{x}_{ipq}$
15: **end for**
16: Compute the CM measure for each unit from each local model
17: according to equation 5.2
18: **end for**
19: Assign each unit to the closest local model
20: **Steps 4 to 7 are repeated until there is convergence** on class composition
21: **Once the convergence is achieved**:

(i) Estimate the final local models

(ii) Compute the Group Quality Index according to equation 5.12

the threshold of less than 0.05% of units changing class from one iteration to the other as stopping rule. As a matter of fact, REBUS-PLS usually assures convergence in a small number of iterations (i.e. less than 15).
It is also possible not to define a threshold as a stopping rule and run the algorithm until the same groups are formed in successive iterations. In fact, if no stopping rule is imposed once the "best" model is obtained in the REBUS-PLS viewpoint, i.e. once each unit is correctly assigned to the closeness local model, the algorithm provides the same partition of the units at successive iterations.
If the sample size is large, it is possible to have such boundary units that change classes time after time at successive iterations. This leads us to obtain a series of partitions (i.e. of local model estimates) that repeat themselves in successive iterations.
In order to avoid the "boundary" unit problem the author suggests always defining a stopping rule.

Once the stability on class composition is reached, the final local models are computed. The class-specific parameters are then compared in order to explain differences among detected latent classes.
Moreover the quality of the obtained partition can be evaluated through a new index (i.e. the *Group Quality Index* (GQI)) expressly developed (cf. section 5.3). A permutation test procedure (cf. subsection 4.5.1) applied on the GQI, can be used to validate the detected latent classes.
Furthermore, if external concomitant variables are available, an *ex-*

post analysis on the detected classes can be performed. This allows us to use such concomitant variables to characterize the detected latent classes.

Until now, REBUS-PLS has only been able to be applied in models showing the reflective measurement model. As a matter of fact, the measurement residuals, as stated in equation 5.3, are the residuals of the simple regression between each manifest variable in a block and the corresponding latent variable. Therefore, they are defined only for reflective indicators. Developments of the REBUS-PLS algorithm to take into account also formative indicators are on going.

5.3 A new index to assess group separation

A new index to assess if local models perform better than the global model will be introduced in this section. Of course, if local models perform better than the global model, this directly entails assessing the quality of the detected partition. As a matter of fact, if the detected local models perform better than the global model, and better than a random partition of the units, this can be considered an index of the quality of the detected partition.

The *Group Quality Index* (GQI) presented here, is a reformulation of the *Goodness of Fit* (GoF) index in a multi-group optic, and as the

CM used in REBUS-PLS algorithm it is based on residuals.

As is well known the R^2 index in a simple regression is an indicator of how well the model fits the data. According to this, the R^2 value is commonly expressed as the proportion of the dependent variable variability explained by the regression model.
In other words, the smaller the variability of the residual values around the regression line relative to the overall variability is, the better the prediction obtained by the model is. Once again, the residuals play a central role in stating the quality of a model.
Following this idea the simple R^2 index can be expressed as:

$$R^2 = 1 - \frac{Dev\,(E)}{Dev\,(Y)} \tag{5.8}$$

where $Dev\,(E)$ is the deviance of the errors in the model, and $Dev\,(Y)$ is the deviance of the dependent variable.

Remembering the *Goodness of Fit* index as presented in subsection 3.4.1, it is easy to notice that both the terms on the left-side of the index and on the right-side are R^2 value:

$$GoF = \sqrt{\frac{\sum_{q=1}^{Q}\sum_{p=1}^{P_q} Cor^2\left(\boldsymbol{x}_{pq}, \boldsymbol{\xi}_q\right)}{\sum_{q=1}^{Q} P_q} \times \frac{\sum_{j=1}^{J} R^2\left(\boldsymbol{\xi}_j, \{\boldsymbol{\xi}_q^*\text{'s explaining}\boldsymbol{\xi}_j\}\right)}{J}} \tag{5.9}$$

Thus, it is possible to rewrite the GoF index according to the R^2 formulation expressed in equation 5.8, i.e. based on the residuals:

$$GoF = \sqrt{\frac{\sum_{q=1}^{Q}\sum_{p=1}^{P_q}\left(1 - \frac{\sum_{i=1}^{N} e_{ipq}^2}{\sum_{i=1}^{N}\left(x_{ipq} - \overline{x}_{pq}\right)^2}\right)}{\sum_{q=1}^{Q} P_q} \times \frac{\sum_{j=1}^{J}\left(1 - \frac{\sum_{i=1}^{N} f_{ij}^2}{\sum_{i=1}^{N}\left(\xi_{ij} - \overline{\xi}_j\right)^2}\right)}{J}} \tag{5.10}$$

Remembering that the total number of manifest variables in the model is P, with $P = \sum_{q=1}^{Q} P_q$, the equation 5.10 can be rewritten as:

$$GoF = \sqrt{\frac{1}{P}\sum_{q=1}^{Q}\sum_{p=1}^{P_q}\left(1 - \frac{\sum_{i=1}^{N} e_{ipq}^2}{\sum_{i=1}^{N}\left(x_{ipq} - \overline{x}_{pq}\right)^2}\right) \times \frac{1}{J}\sum_{j=1}^{J}\left(1 - \frac{\sum_{i=1}^{N} f_{ij}^2}{\sum_{i=1}^{N}\left(\xi_{ij} - \overline{\xi}_j\right)^2}\right)} \tag{5.11}$$

This reformulation of the GoF allows us to assess model quality directly as regards the measurement and the structural residuals, i.e. directly in the REBUS-PLS optic.

If more than one class is taken into account, i.e. if the N units are split into K classes each one of size n_k, the GoF index as expressed in equation 5.11 can be reformulated leading to the *Group Quality Index*. Therefore, in the case of K classes the *Group Quality Index* can be

expressed as:

$$GQI = \sqrt{\sum_{k=1}^{K} \frac{n_k}{N} \left[\frac{1}{P} \sum_{q=1}^{Q} \sum_{p=1}^{P_q} \left(1 - \frac{\sum_{i=1}^{n_k} e_{ipqk}^2}{\sum_{i=1}^{n_k} (x_{ipq} - \overline{x}_{pqk})^2} \right) \right] \times \sum_{k=1}^{K} \frac{n_k}{N} \left[\frac{1}{J} \sum_{j=1}^{J} \left(1 - \frac{\sum_{i=1}^{n_k} f_{ijk}^2}{\sum_{i=1}^{n_k} \left(\xi_{ijk} - \overline{\xi}_{jk} \right)^2} \right) \right]} \tag{5.12}$$

This index is equal to the GoF in the case of a unique class, i.e. when $K = 1$ and $n_1 = N$.
In other words, the *Group Quality Index* computed for the whole sample as a unique class is equal to the GoF index computed for the global model.

If local models performing better than the global model are detected the GQI index will be higher than the GoF value computed for the global model. As a matter of fact, local models performing better than the global model mean working with residuals that are smaller than the one computed for the global model. And this directly entails obtaining higher GQI index than the one obtained for the global model. Of course, the GQI can be considered as an average of the class specific GoF index. Nevertheless, expressing the *Group Quality Index* as in equation 5.12, allows us to directly compare the same index among different partitions of the units (and with the aggregate solution of the global model too).

Simulation study (cf. subsection 6.1.3) has suggested that an im-

provement of the GQI index from the global model to the detected local model higher than the 25% can be considered as a satisfactory threshold to prefer the detected unit partition to the aggregate data solution.
The improvement of the GQI index as regards the global model can be computed as:

$$GQI_{improvement} = \frac{GQI_K - GQI_1}{GQI_1} \tag{5.13}$$

where GQI_1 is the *Group Quality Index* computed for the aggregate data, i.e. the GoF value computed for the global model, and GQI_K is the *Group Quality Index* computed for the detected partition of the units in K latent classes.

To conclude, to assess the quality of the detected partition it is possible to perform a permutation test procedure involving T random replications of the unit partition (keeping constant the group proportions as detected by REBUS-PLS).
In this way an empirical distribution of the GQI index will be obtained. The GQI obtained for the REBUS-PLS based partition will be compared to the empirical distribution, in order to assess if the REBUS-PLS based partition performs better than random assignment of the units, and better than the global model.
As a matter of fact, simulation studies, as well as real data case (cf. chapter 6), have shown that in the case of unobserved heterogeneity, apart from the outlier solutions, the GQI index computed for the

aggregate level is the minimum value obtained for the empirical distribution of the GQI.

REBUS-PLS features, as well as GQI proprieties will be investigated in the following chapter through a simulation study involving 600 experimental data-sets. Furthermore, REBUS-PLS will be applied on an empirical data-set coming from a marketing study.

Chapter 6

Simulation study and application to real data

6.1 Simulation study

6.1.1 Design of the Numerical Example and Data Simulation

A key area for identifying and forming segments in social sciences is related to the specific behaviors of certain groups of observed units. Although the naming of latent variables is a trivial matter for numerical examples using simulated data, this study focuses on the area of customer satisfaction, as well as segmentation of markets, and consumers. In this section, the REBUS-PLS algorithm will be tested in

order to investigate its capability of detecting unobserved heterogeneity. These kind of examples allow us to better illustrate the features of the REBUS-PLS algorithm that could be easily transferable to other research disciplines in social sciences.

Customer satisfaction has achieved a fundamental and well documented results in business research. Forming groups of consumers that are homogeneous in terms of the benefits they seek or their response to marketing programs (e.g. product offering, price discounts) is therefore a key element for marketers to establish and improve their targeted marketing strategies [Wedel & Kamakura 2000].
Here, a simple marketing type model will be used to assess the REBUS-PLS capability. The postulated model overlaps the one used by Jedidi et al. [1997a] and by Esposito Vinzi, Ringle, Squillacciotti & Trinchera [2007] for their numerical examples. It is composed of one latent endogenous variable, *Customer Satisfaction*, and two latent exogenous variables, *Price Fairness* and *Quality* (cf. figure 6.1).
Each latent exogenous variable (*Price Fairness* and *Quality*) has five manifest variables (reflective mode), and the latent endogenous variable (*Customer Satisfaction*) is measured by three indicators (reflective mode). However, it is not relevant for this study to include an additional level of complexity by exemplifying path model details regarding the manifest variables and the theoretical reasoning for choosing reflective instead of formative measurement models. Since REBUS-PLS has been established for path models with reflective blocks, our

analysis will be limited to that kind of measurement model.

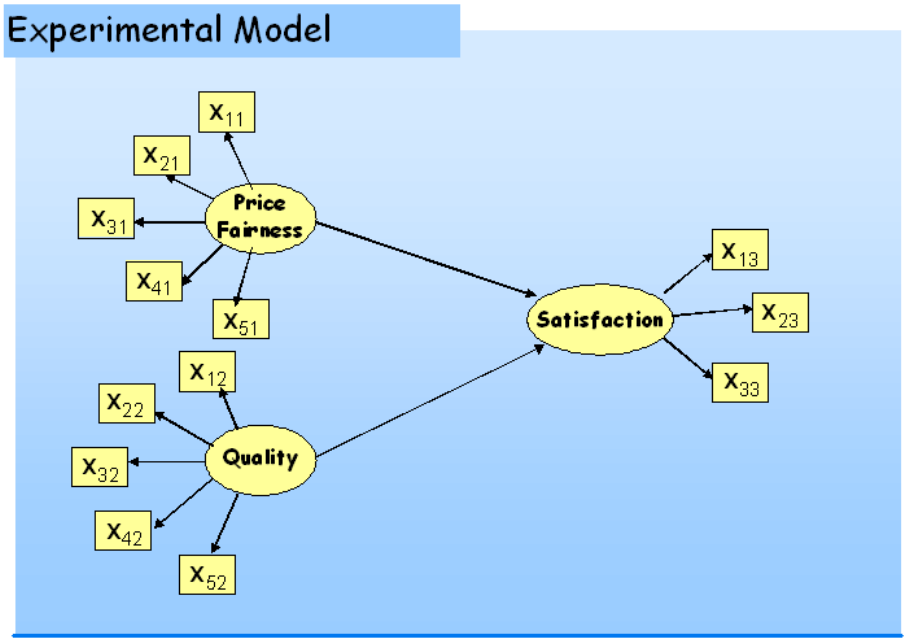

Figure 6.1: *Experimental model*

This study intentionally uses a clear cut example of a marketing related path model for data simulation purposes. The data generation procedure is based on the LISREL-type approach. In other words, once the model parameters are established, the data are generated according to the implied covariance matrix, using a specific SAS-IML

macro developed by the author.
Two latent classes showing different local models are supposed to exist. Each one is composed of 200 units. Thus, the data on the aggregate level for each one of the numerical examples includes 400 units.

This simulation study aims at testing the REBUS-PLS capability in handling unobserved heterogeneity in three different situations, i.e. when the unobserved heterogeneity is focused only on the structural models (simulation scheme 1), when it concerns only the measurement models (simulation scheme 2), and when units are heterogeneous as regards both the structural and measurement models (simulation scheme 3).
For each of the postulated simulation schemes, 100 sets of simulated data will be used.

In total, the analysis involves 300 marketing related numerical examples on different sets of simulated data. Each set includes computation of the PLS Path Modeling results for (a) the aggregate data level (global model), (b) each class of simulated data (group models) and (c) the local model solutions for REBUS-PLS obtained by using a SAS-IML macro developed by the author (cf. A.2).
In all the cases the results of the global model exhibit the requirement for addressing heterogeneity of model estimates. A comparison of the outcomes in the local model estimates facilitates an assessment of the REBUS-PLS algorithm. The class-specific parameters, as well as the

quality indexes as the R^2 and the GoF of the local models, are benchmarks for REBUS-PLS. The rate of correctly assigned cases will also be used as a performance indicator.
To conclude, the *Group Quality Index* as defined in section 5.3, will be used to assess the performance of the local models compared with the global model.

In this section, the results concerning the REBUS-PLS capability of detecting unobserved heterogeneity focused on structural models (simulation scheme 1) will be presented first (cf. subsection 6.1.2). Then, the REBUS-PLS performance in handling unobserved heterogeneity concerning the measurement models (simulation scheme 2) will be investigated (cf. subsection 6.1.3). To conclude (cf. subsection 6.1.4), REBUS-PLS will be tested in local model detection when both the measurement and structural models are different among classes (simulation scheme 3).

6.1.2 Unobserved heterogeneity focused on the structural model

Unobserved heterogeneity focused only on the structural models directly means working with local models that are different only as regards the path coefficient intensities. In a simple model, as the one postulated above, heterogeneity in the structural model implies detecting price sensitive consumers, or those requiring price fairness, and consumers who have the strongest preference for another partic-

ular product attribute, e.g. quality.
Thus the experimental sets of data consist of two latent classes with the following characteristics:

(a) Class 1 - **price fairness seeking customers** - characterized by a strong relationship between *Price Fairness* and *Customer Satisfaction* and a weak relationship between *Quality* and *Customer Satisfaction*;

(b) Class 2 - **quality oriented customers** - characterized by a strong relationship between *Quality* and *Customer Satisfaction* and a weak relationship between *Price Fairness* and *Customer Satisfaction*.

Data simulation for the group of price fairness seeking consumers involves a strong relationship of 0.9 between *Price Fairness* and *Customer Satisfaction* and a weak relationship of 0.1 between *Quality* and *Customer Satisfaction* in the structural model (Class 1). Another group of data reflects the characteristics of the quality oriented consumers (Class 2), with a path coefficient close to 0.9 between *Quality* and *Customer Satisfaction*, and a weak relation (close to 0.1) between *Price Fairness* and *Customer Satisfaction*.
Each of the two groups is composed of 200 units. Therefore, at the aggregate level each experimental data-set is composed of 400 units.

100 data-sets keeping the postulated features (two groups of 200 units each one, the first characterized by a strong relationship between *Price*

Fairness and *Customer Satisfaction*, and the second by a strong relationship between *Quality* and *Customer Satisfaction*) have been simulated.
The REBUS-PLS algorithm has been applied to each of the 100 aggregate data-sets. A summary of the results obtained at aggregate level, as well as at detected local model level, are shown in the tables 6.3 and 6.5, as well as in the figures 6.2 and 6.4.

In all the cases, REBUS-PLS detect two classes of units overlapping the simulated groups. As a matter of fact, looking at the distribution of the structural model coefficients (cf. figure 6.2) it is possible to notice that the path coefficient estimates obtained for the first detected class are always coherent with the simulated one (close to 0.90 for the latent variable *Price*, and close to 0.10 for the latent variable *Quality*).

It is the same for the second detected class. In this case, the path coefficient estimates linked to the latent variable *Price* are close to 0.10, while the same for the latent variable *Quality* are close to 0.90.

The PLS Path Modeling results on the aggregate data level are significantly different compared with the segment specific computations for each *a priori* simulated group of data. In these numerical examples, estimates for the overall sets of data are close to the weighted average of group specific coefficients. As a consequence, the PLS Path Modeling results are ambiguous when heterogeneity is not accounted for.

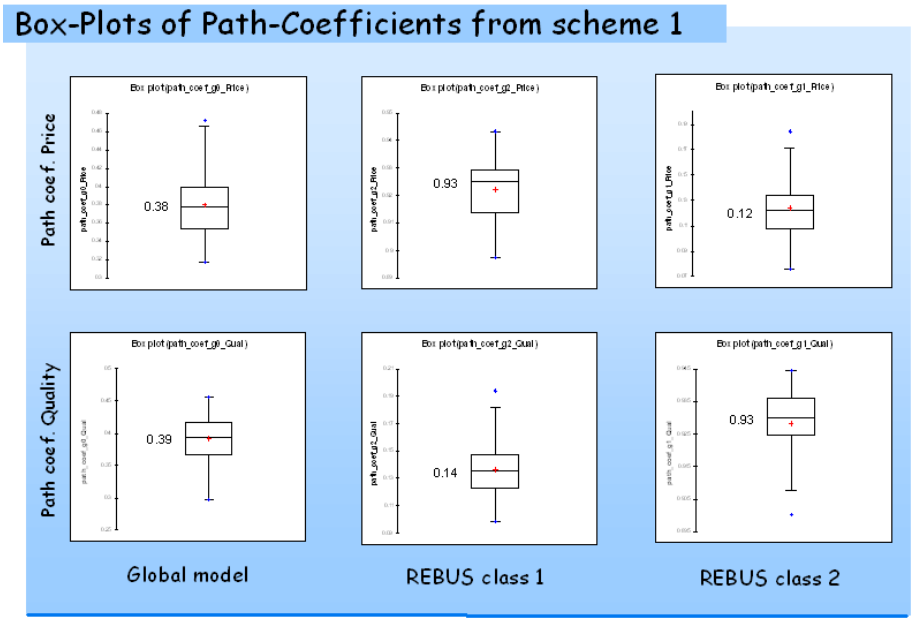

Figure 6.2: *Box-Plots for path coefficient estimates for simulation scheme* 1

As a matter of fact, in a simulation scheme as the one here adopted (characterized by two structural relationships of about 0.9 and 0.1 for one class and vice versa for the other class), to perform a PLS Path Modeling analysis without taking into account heterogeneity in the data turns out to have a value on the aggregate data level between 0.30 and 0.46 for both the relationships (cf. table 6.3).

Statistics	Global Model		Class 1			Class2		
	coeff_PRICE	coeff_QUALITY	n_g1	coeff_PRICE	coeff_QUALITY	n_g2	coeff_PRICE	coeff_QUALITY
No. of simulated data-sets	100	100	100	100	100	100	100	100
Simulated Values	n. a.	n. a.	200	0.900	0.100	200	0.100	0.900
Minimum	**0.317**	**0.297**	186	**0.898**	**0.099**	185	**0.076**	0.900
Maximum	**0.472**	**0.456**	215	**0.943**	**0.194**	214	**0.185**	0.945
1st Quartile	0.354	0.367	199	0.914	0.123	194	0.108	0.925
Median	0.378	0.394	202	0.925	0.136	198	0.122	0.930
3rd Quartile	0.400	0.416	206	0.929	0.147	201	0.134	0.936
Mean	0.379	0.392	202	0.922	0.137	198	0.124	0.928
Variance (n-1)	0.001	0.001	37	0.000	0.000	37	0.001	0.000
Standard deviation (n-1)	0.035	0.035	6	0.010	0.020	6	0.023	0.010
Variation coefficient	0.092	0.088	0	0.011	0.146	0	0.186	0.011
Lower bound on mean (95%)	0.372	0.385	201	0.920	0.133	197	0.119	0.926
Upper bound on mean (95%)	0.386	0.399	203	0.924	0.141	199	0.128	0.930

Figure 6.3: *Descriptive Statistics for path coefficient estimates and detected class size obtained from the* 100 *data-sets simulated according to simulation scheme* 1

As regards the quality indexes, i.e. the R^2 value associated with the endogenous latent variable *Satisfaction*, and the GoF value computed for each of the 100 data-sets, their values are always definitely higher at local model level than at aggregate level (cf. figure 6.4 and table 6.5).

In particular, the R^2 values at the aggregate level are significantly lower than the ones computed for the REBUS-PLS based clusters. This is logically due to the simulation scheme that only involved the structural model.

The GoF values computed for the global models, instead, even if lower than the ones obtained for the local models, are still "higher" at the

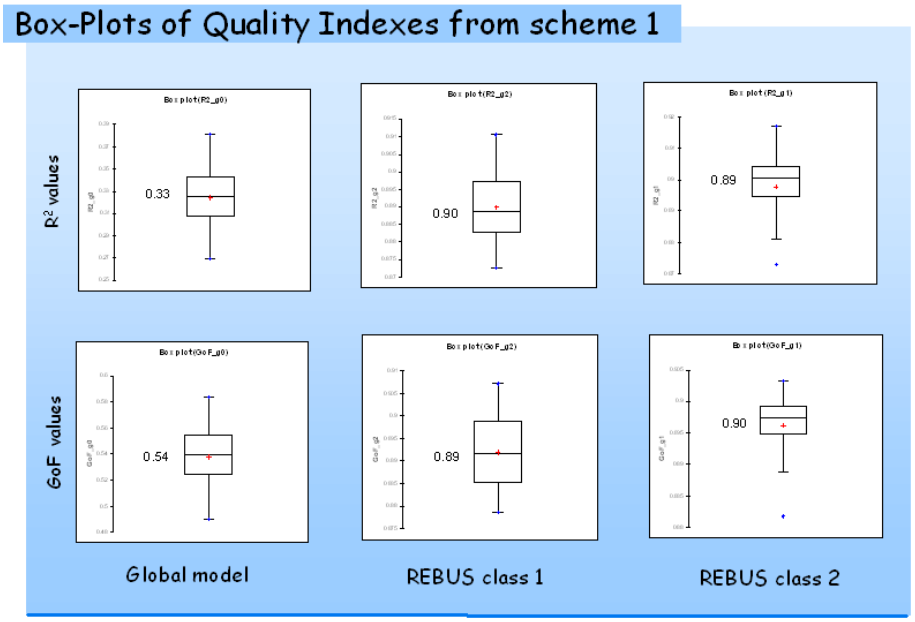

Figure 6.4: *Box-Plots for* R^2 *and GoF values computed for simulation scheme* 1

aggregate level. This is due to the *GoF* nature. As a matter of fact, the *GoF* index is a quality index that takes into account both the model performance in the structural and in the measurement models. In the postulated simulation scheme, the measurement model are similar in the two simulated groups. This means, that even at aggregate level they are well estimated. In other words, the communality indexes are still higher at the aggregate level (and close to the ones obtained

at the local model level).
This leads to *GoF* values that at the aggregate level are only slightly affected by unobserved heterogeneity.

Statistics	Global Model		Class 1		Class2		GQI	% well classified	Improvement of GQI
	R^2	GoF	R^2	GoF	R^2	GoF			
No. of simulated data-sets	100	100	100	100	100	100	100	100	100
Minimum	**0.269**	**0.490**	**0.873**	**0.879**	**0.873**	**0.882**	**0.885**	**0.910**	**0.533**
Maximum	**0.381**	**0.584**	**0.911**	**0.907**	**0.917**	**0.903**	**0.904**	**0.968**	**0.811**
1st Quartile	0.307	0.524	0.883	0.885	0.895	0.895	0.890	0.930	0.615
Median	0.326	0.540	0.889	0.892	0.900	0.897	0.894	0.940	0.657
3rd Quartile	0.343	0.554	0.897	0.899	0.904	0.899	0.898	0.953	0.707
Mean	0.324	0.538	0.890	0.892	0.898	0.896	0.894	0.940	0.664
Variance (n-1)	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004
Standard deviation (n-1)	0.026	0.021	0.009	0.008	0.010	0.005	0.005	0.014	0.066
Variation coefficient	0.080	0.040	0.010	0.009	0.011	0.005	0.005	0.014	0.098
Lower bound on mean (95%)	0.319	0.534	0.888	0.890	0.896	0.895	0.893	0.937	0.651
Upper bound on mean (95%)	0.329	0.542	0.892	0.893	0.900	0.897	0.895	0.943	0.677

Figure 6.5: *Descriptive Statistics for the* R^2 *values, the GoF values, the GQI values, the well-classified rate and the improvement of the GQI obtained from the* 100 *data-sets simulated according to simulation scheme* 1

Moreover, to assess the REBUS-PLS capability to detect the simulated group of data, the well-classified rate can be used. REBUS-PLS is always able to correctly assign units to the corresponding simulated group, with a well-classified rate never lower than 91% (cf. table 6.5). To conclude, the *Group Quality Index*, as presented in section 5.3, is always higher than 0.885, i.e. about double the one obtained at the

aggregate level (i.e. the GoF index computed for the global models). This means, that REBUS-PLS clustering leads to an improvement of the model quality (in terms of the GoF value, cf. equation 5.13) always higher than 53% (cf. table 6.5).
The Box-Plots summarizing the distribution of the GQI and of the well-classified rate in the 100 simulated data-sets are provided in figure 6.6

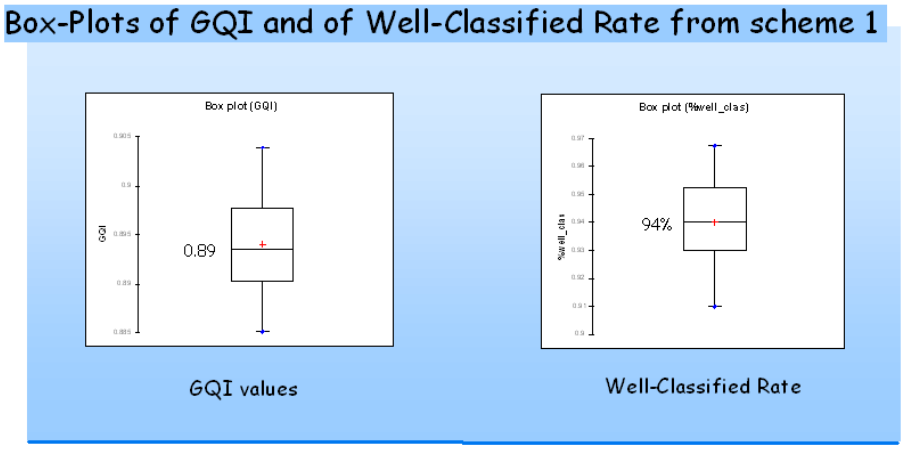

Figure 6.6: *Box-Plots for GQI and well-classified rate computed for simulation scheme* 1

We can conclude stating that if heterogeneity in the structural model is not identified by the researcher, this can lead to incorrect estimates

of the path coefficients and to models performing worse than homogeneous models.
In our example, if heterogeneity in the structural model is not taken into account the effects of *Price Fairness* and *Quality* on *Satisfaction* seem to be equally important. As a consequence of these, PLS Path Modeling results, marketers may focus on the areas of *Price Fairness* and *Quality* at the same time for all consumers. Uncovering heterogeneity in the structural model relationships and forming distinctive classes of price fairness seeking and quality oriented customers allows marketers to develop better targeted and more effective business strategies.

In the next subsection the problems linked to the presence of heterogeneous measurement models will be investigated.

6.1.3 Unobserved heterogeneity focused on the measurement model

The second simulation scheme involves working with local models that are different at measurement model level. In other words, the simulated local models are different as regards the outer weight intensities and the correlations between each manifest variable and the corresponding latent variable.
The path coefficients, instead, are supposed to be the same among the two groups. In other words, the two exogenous latent variables (*Price Fairness* and *Quality*) are supposed to have the same impact on the

endogenous latent variable *Satisfaction*.
In order to build such different measurement models the author decided to identify, for each group of units, a manifest variable showing a slight correlation with the corresponding manifest variable. This means that each group is characterized by a unique and well defined outer weight smaller than all the others in the model.
Therefore, the experimental sets of data consist of two latent classes with the following characteristics:

(a) Class 1 - characterized by a **weak correlation** between the **3rd manifest variable of the *Price Fairness* block** and the corresponding latent variable;

(b) Class 2 - characterized by a **weak correlation** between the **3rd manifest variable of the *Quality* block** and the corresponding latent variable.

In particular, data simulation for the first group involves a relationship of 0.7 between *Price Fairness* and *Customer Satisfaction* and between *Quality* and *Customer Satisfaction* in the structural model, and an external normalized weight close to 0.1 for the third manifest variable of the *Price Fairness* block (Class 1).
The second group of data, instead, shows a path coefficient close to 0.7 between *Quality* and *Customer Satisfaction*, and between *Price Fairness* and *Customer Satisfaction*, and an external normalized weight close to 0.1 for the third manifest variable of the *Quality* block.
As usual, each of the two groups is composed of 200 units, and 100

data-sets keeping the postulated features have been simulated.

The REBUS-PLS algorithm has been applied to each of the 100 aggregate data-sets. A summary of the results obtained at aggregate level, as well as at detected local model level, is shown in the tables 6.7 and 6.8, as well as in the figures, 6.9, 6.10 and 6.11.

Statistic	Global Model		Class 1			Class2		
	weight 3VM PRICE	weight 3VM QUALITY	n_g1	weight 3VM PRICE	weight 3VM QUALITY	n_g2	weight 3VM PRICE	weight 3VM QUALITY
No. of simulated data-sets	100	100	100	100	100	100	100	100
Simulated Values	n. a.	n. a.	200	~0.1	>0.2	200	>0.2	~0.1
Minimum	**0.132**	**0.114**	187	0.052	-0.498	190	**-0.167**	**0.059**
Maximum	**0.190**	**0.183**	210	0.185	0.420	213	**0.358**	**0.185**
1st Quartile	0.150	0.149	197	0.069	0.204	198	0.194	0.085
Median	0.161	0.159	200	0.086	0.208	200	0.200	0.098
3rd Quartile	0.169	0.167	202	0.106	0.212	203	0.209	0.107
Mean	0.160	0.157	200	0.094	0.192	200	0.199	0.100
Variance (n-1)	0.000	0.000	14	0.001	0.010	14	0.002	0.001
Standard deviation (n-1)	0.013	0.015	4	0.034	0.101	4	0.044	0.027
Variation coefficient	0.080	0.094	0	0.356	0.526	0	0.219	0.267
Lower bound on mean (95%)	0.157	0.154	199	0.087	0.172	200	0.190	0.095
Upper bound on mean (95%)	0.162	0.160	200	0.101	0.212	201	0.208	0.105

Figure 6.7: *Descriptive Statistics for normalized outer weight estimates and detected class size obtained from the* 100 *simulated data-sets simulated according to simulation scheme* 2

In table 6.8 it is possible to notice that the improvement of GQI, obtained according to equation 5.13, reaches a minimum value of 0.088. In other words, there are some data-sets for which the obtained unit partition does not improve the model quality (in terms of GoF value).

Statistic	GoF Global	GoF Class 1	GoF Class 2	GQI	% well classified	Improvement of GQI
No. of simulated data-sets	100	100	100	100	100	100
Minimum	0.559	0.531	0.574	0.689	0.543	**0.088**
Maximum	0.678	0.861	0.864	0.851	0.993	**0.472**
1st Quartile	0.597	0.820	0.824	0.823	0.970	0.305
Median	0.612	0.832	0.833	0.833	0.978	0.344
3rd Quartile	0.633	0.838	0.841	0.839	0.983	0.387
Mean	**0.614**	**0.815**	**0.828**	0.822	0.940	0.339
Variance (n-1)	0.001	0.004	0.001	0.001	0.012	0.005
Standard deviation (n-1)	0.026	0.062	0.034	0.035	0.109	0.069
Variation coefficient	0.042	0.076	0.040	0.042	0.116	0.202
Lower bound on mean (95%)	0.609	0.802	0.821	0.815	0.918	0.326
Upper bound on mean (95%)	0.620	0.827	0.834	0.829	0.961	0.353

Figure 6.8: *Descriptive Statistics for the GoF values, the GQI values, the well-classified rate and the improvement of the GQI obtained from the* 100 *data-sets simulated according to simulation scheme* 2

In particular, for 13 data sets out of 100, the obtained partition shows an improvement of GQI smaller than 25% (cf. table 6.12). This is due to the fact that the unobserved heterogeneity is focused only on the measurement model. Since REBUS-PLS is based on a measure that takes into account both the structural and measurement model it is not always able to handle local models that differ only as regards the measurement model.

It is for this reason that in 13 data sets out of the 100 simulated according to the simulation scheme 2, REBUS-PLS does not detect the

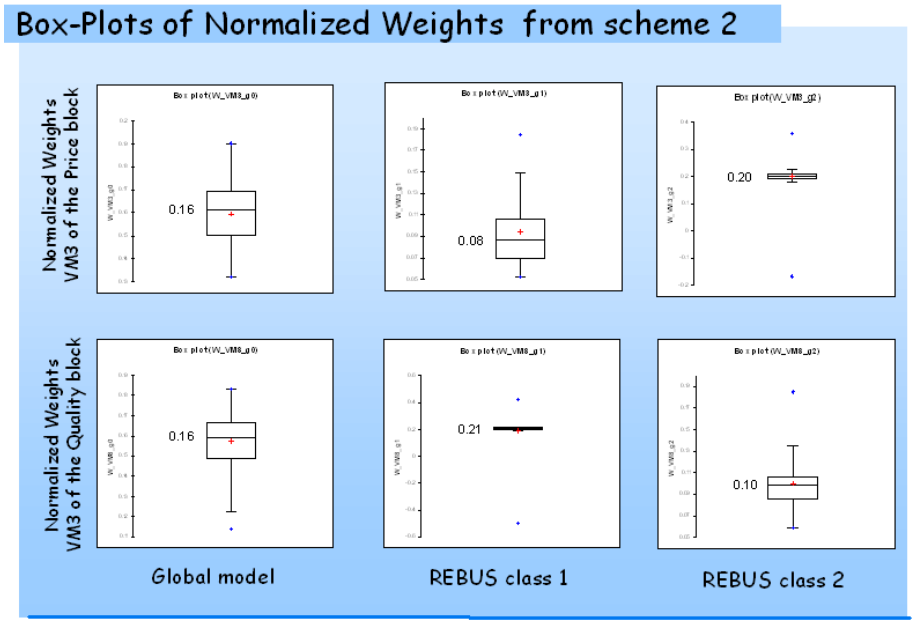

Figure 6.9: *Box-Plots for normalized weight estimates for simulation scheme* 2

"true" (simulated) local model, showing an average well-classified rate close to 70% (cf. table 6.12), and no difference in class-specific outer weights (cf. table 6.13).

For the remaining 87 data sets, instead, the REBUS-PLS results are extremely positive. As a matter of fact, for these cases the average well-classified rate is close to 98% and never lower than 87% (cf. table

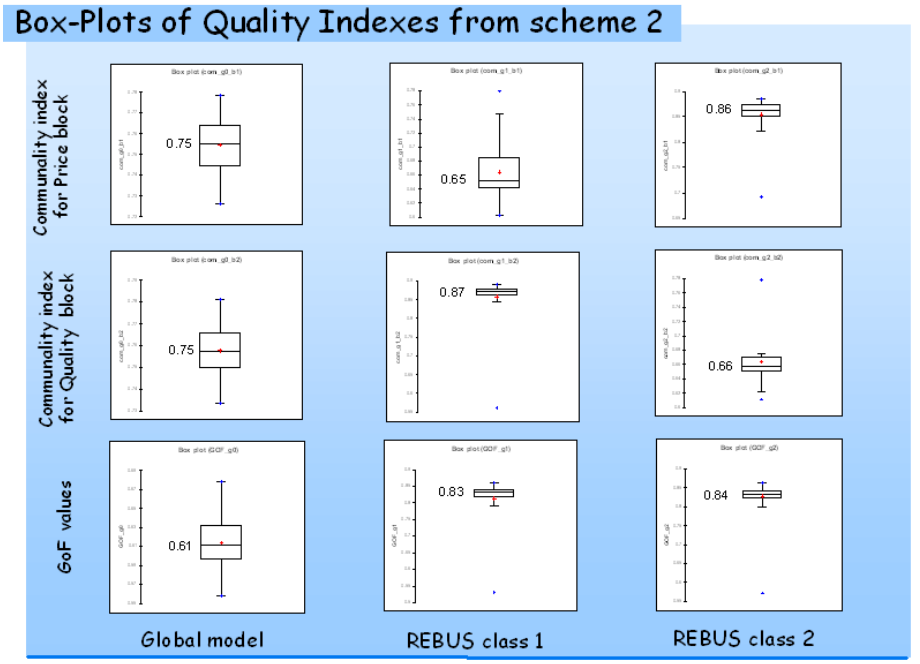

Figure 6.10: *Box-Plots for R^2 and GoF values computed for simulation scheme* 2

6.14).

The same encouraging results are obtained for the GQI values that have a mean value of 0.83 with an improvement of the GoF value never smaller than $27,4\%$ (cf. table 6.14).

Not taking into account heterogeneity leads us to neglect both the

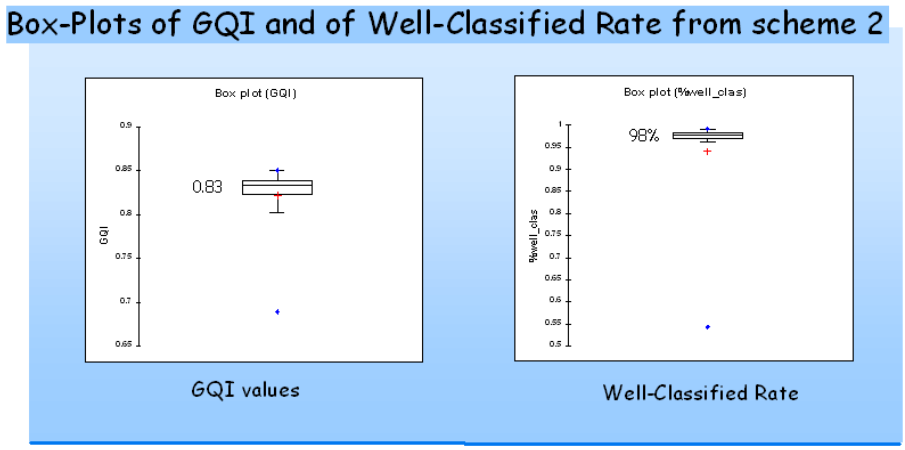

Figure 6.11: *Box-Plots for GQI and well-classified rate computed for simulation scheme* 2

3rd manifest variable of the *Price Fairness* block and the 3rd manifest variable of the *Quality* block (cf. table 6.15). As a matter of fact, at the aggregate level both the normalized weights of the 3rd manifest variable associated to the *Price Fairness* block and the 3rd manifest variable of the *Quality* block show average values close to 0.16.
The group specific estimates, instead, exactly overlap the simulated value for the normalized outer weights of both the manifest variables. As a matter of fact, for the selected 87 simulated data-sets, the normalized outer weight associated to the 3rd manifest variable of the *Price Fairness* block for the first detected class is bounded between

Statistic	GoF Global	GoF Class 1	GoF Class 2	GQI	% well classified	Improvement of GQI
No. of simulated data-sets	13	13	13	13	13	13
Minimum	0.560	**0.531**	**0.574**	0.689	**0.543**	**0.088**
Maximum	0.678	0.857	0.844	0.850	0.983	0.256
1st Quartile	0.606	0.582	0.808	0.708	0.578	0.188
Median	0.620	0.718	0.820	0.760	0.610	0.225
3rd Quartile	0.643	0.807	0.827	0.780	0.750	0.245
Mean	**0.622**	**0.703**	**0.791**	**0.751**	0.702	0.208
Variance (n-1)	0.001	0.015	0.006	0.003	0.026	0.002
Standard deviation (n-1)	0.034	0.122	0.075	0.053	0.162	0.048
Variation coefficient	0.053	0.167	0.091	0.068	0.221	0.221
Lower bound on mean (95%)	0.601	0.629	0.746	0.719	0.604	0.180
Upper bound on mean (95%)	0.642	0.777	0.836	0.783	0.800	0.237

Figure 6.12: *Descriptive Statistics for the GoF values, the GQI values, the rate of well-classified and the improvement of the GQI obtained for the "worst"* 13 *data-sets out of the* 100 *simulated according to simulation scheme* 2

0.052 and 0.165, with an average value of 0.085, i.e. close to the simulated value of 0.100. While, the 3rd manifest variable of the *Quality* block shows an average normalized weight close to 0.212, i.e. close to the ones associated with the other manifest variables of the block, usually close to 0.200 (cf. table 6.15).

Similar results are obtained for the second detected class. Also in this case, the obtained results overlap the simulated group specific values. In fact, in the *Quality* block the 3rd manifest variable is the weakliest correlated with the corresponding latent variable, with a normalized

Statistic	Global Model		Class 1			Class2		
	weight 3VM PRICE	weight 3VM QUALITY	n_g1	weight 3VM PRICE	weight 3VM QUALITY	n_g2	weight 3VM PRICE	weight 3VM QUALITY
No. of simulated data-sets	13	13	13	13	13	13	13	13
Simulated Values	n. a.	n. a.	200	˜0.1	»0.2	200	»0.2	˜0.1
Minimum	0.157	0.134	187	**0.088**	**-0.498**	190	**-0.167**	**0.059**
Maximum	0.188	0.183	210	**0.185**	**0.420**	213	**0.358**	**0.185**
1st Quartile	0.164	0.152	193	0.162	-0.049	196	0.161	0.115
Median	0.167	0.158	200	0.170	0.160	200	0.173	0.161
3rd Quartile	0.171	0.163	204	0.176	0.209	207	0.193	0.169
Mean	0.168	0.157	199	0.157	0.059	201	0.163	0.144
Variance (n-1)	0.000	0.000	53	0.001	0.061	53	0.013	0.001
Standard deviation (n-1)	0.008	0.014	7	0.033	0.247	7	0.113	0.037
Variation coefficient	0.045	0.086	0	0.200	4.018	0	0.664	0.245
Lower bound on mean (95%)	0.164	0.149	195	0.137	-0.090	197	0.095	0.122
Upper bound on mean (95%)	0.173	0.166	203	0.177	0.208	205	0.231	0.166

Figure 6.13: *Descriptive Statistics for normalized outer weight estimates and detected class size obtained for the "worst"* 13 *data-sets out of the* 100 *simulated according to simulation scheme* 2

outer weight value bounded between 0.059 and 0.135. While, in the *Price Fairness* block all the manifest variables show the same level of correlation with the latent variable, with an average normalized outer weight equal to 0.204 (cf. table 6.15). Moreover, looking at figure 6.9, it is possible to notice that the distributions of the normalized outer weights are very different in the two groups.

Statistic	GoF Global	GoF Class 1	GoF Class 2	GQI	% well classified	Improvement of GQI
No. of simulated data-sets	87	87	87	87	87	87
Minimum	0.559	0.792	0.765	0.780	**0.868**	**0.274**
Maximum	0.664	0.861	0.864	0.851	0.993	0.472
1st Quartile	0.597	0.823	0.826	0.830	0.973	0.320
Median	0.612	0.833	0.835	0.834	0.978	0.363
3rd Quartile	0.633	0.838	0.844	0.840	0.983	0.391
Mean	**0.613**	**0.831**	**0.833**	**0.832**	0.975	0.359
Variance (n-1)	0.001	0.000	0.000	0.000	0.000	0.002
Standard deviation (n-1)	0.025	0.015	0.017	0.012	0.018	0.047
Variation coefficient	0.040	0.018	0.020	0.014	0.018	0.130
Lower bound on mean (95%)	0.608	0.828	0.830	0.830	0.972	0.349
Upper bound on mean (95%)	0.619	0.835	0.837	0.835	0.979	0.369

Figure 6.14: *Descriptive Statistics for the GoF values, the GQI values, the well-classified rate and the improvement of the GQI obtained for the "best"* 87 *data-sets out of the* 100 *simulated according to simulation scheme* 2

6.1.4 Unobserved heterogeneity involves both the measurement and the structural models

This last simulation scheme involves working with local models that are different at both the measurement and the structural model levels. Here, the simulation scheme 1 (cf. subsection 6.1.2) and the simulation scheme 2 (cf. subsection 6.1.3) are mixed in order to obtain groups of units with specific values for both the path coefficients and the measurement model parameters.

Statistic	Global Model		Class 1			Class2		
	weight 3VM PRICE	weight 3VM QUALITY	n_g1	weight 3VM PRICE	weight 3VM QUALITY	n_g2	weight 3VM PRICE	weight 3VM QUALITY
No. of simulated data-sets	87	87	87	87	87	87	87	87
Simulated Values	n. a.	n. a.	200	~0.1	>0.2	200	>0.2	~0.1
Minimum	**0.132**	**0.114**	194	**0.052**	**0.194**	193	**0.186**	**0.059**
Maximum	**0.190**	**0.182**	207	**0.165**	**0.304**	206	**0.283**	**0.135**
1st Quartile	0.148	0.149	198	0.068	0.205	198	0.196	0.080
Median	0.159	0.160	200	0.079	0.208	200	0.201	0.097
3rd Quartile	0.169	0.167	202	0.103	0.212	203	0.209	0.104
Mean	0.158	0.157	200	0.085	0.212	200	0.204	0.094
Variance (n-1)	0.000	0.000	9	0.000	0.000	9	0.000	0.000
Standard deviation (n-1)	0.013	0.015	3	0.022	0.016	3	0.014	0.018
Variation coefficient	0.081	0.095	0	0.255	0.075	0	0.070	0.189
Lower bound on mean (95%)	0.156	0.154	199	0.080	0.208	200	0.201	0.090
Upper bound on mean (95%)	0.161	0.160	200	0.089	0.215	201	0.207	0.097

Figure 6.15: *Descriptive Statistics for normalized outer weight estimates and detected class size obtained for the "best"* 87 *data-sets out of the* 100 *simulated according to simulation scheme* 2

Thus, the experimental sets of data consist of two latent classes with the following characteristics:

(a) Class 1 - **price fairness seeking customers** - characterized by a strong relationship between *Price Fairness* and *Customer Satisfaction* and a weak relationship between *Quality* and *Customer Satisfaction*, as well as by a weak correlation between the 3rd manifest variable of the *Price Fairness* block and the corresponding latent variable;

(b) Class 2 - **quality oriented customers** - characterized by a strong relationship between *Quality* and *Customer Satisfaction*

and a weak relationship between *Price Fairness* and *Customer Satisfaction*, as well as by a weak correlation between the 3rd manifest variable of the *Quality* block and the corresponding latent variable.

In particular, data simulation for the group of price fairness seeking consumers involves a strong relationship of 0.9 between *Price Fairness* and *Customer Satisfaction* and a weak relationship of 0.1 between *Quality* and *Customer Satisfaction* in the structural model, and an external normalized weight close to 0.1 for the third manifest variable of the *Price Fairness* block (Class 1).
Another group of data reflects the characteristics of the quality oriented consumers (Class 2), with a path coefficient close to 0.9 between *Quality* and *Customer Satisfaction*, a weak relation (close to 0.1) between *Price Fairness* and *Customer Satisfaction*, and an external normalized weight close to 0.1 for the third manifest variable of the *Quality* block.
As usual, each of the two groups is composed of 200 units. And, 100 data-sets keeping the postulated features have been simulated.

The REBUS-PLS algorithm has been applied to each of the 100 aggregate data-sets. A summary of the results obtained at aggregate level, as well as at detected local model level, is shown in the tables 6.16, 6.18 and 6.22, as well as in the figures 6.17, 6.19 and 6.21.

In all the 100 data-sets, REBUS-PLS detects two classes of units over-

lapping the simulated groups.

As regards the structural model, exactly as in the first simulation schemes, the path coefficient estimates obtained for the first detected class are always coherent with the simulated one (close to 0.90 for the latent variable *Price Fairness*, and close to 0.10 for the latent variable *Quality*) (cf. table 6.16) Moreover, looking at figure 6.17, it is possible

Statistics	Global Model		Class 1			Class2		
	coeff_PRICE	coeff_QUALITY	n_g1	coeff_PRICE	coeff_QUALITY	n_g2	coeff_PRICE	coeff_QUALITY
No. of simulated data-sets	100	100	100	100	100	100	100	100
Simulated Values	n. a.	n. a.	200	0.900	0.100	200	0.100	0.900
Minimum	**0.280**	**0.277**	**186**	**0.868**	**0.046**	**190**	**0.060**	**0.880**
Maximum	**0.432**	**0.451**	**210**	**0.922**	**0.175**	**214**	**0.159**	**0.928**
1st Quartile	0.324	0.337	194	0.897	0.106	200	0.093	0.898
Median	0.346	0.367	197	0.906	0.139	203	0.108	0.905
3rd Quartile	0.367	0.396	200	0.910	0.153	206	0.136	0.911
Mean	0.346	0.366	197	0.903	0.129	203	0.112	0.905
Variance (n-1)	0.001	0.002	20	0.000	0.001	20	0.001	0.000
Standard deviation (n-1)	0.034	0.040	4	0.011	0.031	4	0.026	0.010
Variation coefficient	0.097	0.108	0	0.012	0.237	0	0.229	0.011
Lower bound on mean (95%)	0.340	0.358	196	0.901	0.123	202	0.107	0.903
Upper bound on mean (95%)	0.353	0.374	198	0.905	0.135	204	0.118	0.907

Figure 6.16: *Descriptive Statistics for path coefficient estimates and detected class size obtained from the* 100 *data-sets simulated according to simulation scheme* 3

to notice that the distributions of the structural model coefficients are very different in the two groups.

Once again, the results for the aggregate model are significantly different with respect to the class-specific ones. Not taking into account heterogeneity leads to path coefficient estimates that are similar for

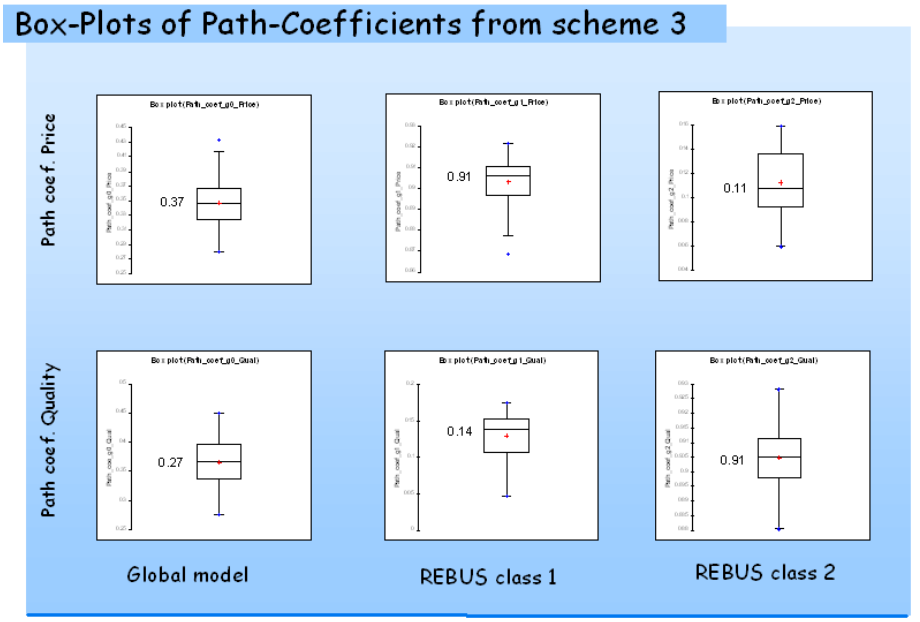

Figure 6.17: *Box-Plots for path coefficient estimates for simulation scheme* 3

both the relationships in the structural model. In other words, at the aggregate data level it is not possible to assess which is the most important driver for *Satisfaction*.

The same happens in measurement model estimates. Nevertheless, as regards the measurement model parameters, not taking into account heterogeneity leads one to neglect both the 3rd manifest variable of

the *Price Fairness* block and the 3rd manifest variable of the *Quality* block (cf. table 6.18). As a matter of fact, at the aggregate level both the normalized weights of the 3rd manifest variable associated to the *Price Fairness* block and the 3rd manifest variable of the *Quality* block show average values close to 0.1

Statistic	Global Model		Class 1			Class2		
	weight 3VM PRICE	weight 3VM QUALITY	n_g1	weight 3VM PRICE	weight 3VM QUALITY	n_g2	weight 3VM PRICE	weight 3VM QUALITY
No. of simulated data-sets	100	100	100	100	100	100	100	100
Simulated Values	n. a.	n. a.	200	~0.1	>0.2	200	>0.2	~0.1
Minimum	**0.046**	**0.063**	186	**0.092**	**0.123**	190	**0.149**	**0.082**
Maximum	**0.138**	**0.154**	210	**0.127**	**0.395**	214	**0.366**	**0.130**
1st Quartile	0.084	0.082	194	0.103	0.208	200	0.187	0.099
Median	0.100	0.096	197	0.107	0.224	203	0.220	0.104
3rd Quartile	0.112	0.114	200	0.115	0.255	206	0.253	0.111
Mean	0.098	0.097	197	0.108	0.234	203	0.225	0.106
Variance (n-1)	0.000	0.000	20	0.000	0.002	20	0.002	0.000
Standard deviation (n-1)	0.020	0.020	4	0.008	0.044	4	0.049	0.011
Variation coefficient	0.200	0.204	0	0.077	0.189	0	0.217	0.104
Lower bound on mean (95%)	0.094	0.094	196	0.107	0.225	202	0.215	0.104
Upper bound on mean (95%)	0.102	0.101	198	0.110	0.242	204	0.235	0.108

Figure 6.18: *Descriptive Statistics for normalized outer weight estimates and detected class size obtained from the* 100 *data-sets simulated according to simulation scheme* 3

The group-specific estimates, instead, exactly overlap the simulated values for the normalized outer weights of both the manifest variables. As a matter of fact, out of all the 100 simulated data-sets, the normalized outer weight associated to the 3rd manifest variable of the *Price Fairness* block for the first detected class is bounded between 0.092 and 0.127, i.e. close to the simulated value of 0.100. While, the

3rd manifest variable of the *Quality* block shows an average normalized weight close to 0.23, i.e. close to the ones associated to the other manifest variables of the block.
As regards the second detected class, the obtained results overlap the simulated group specific values. In fact, in the *Quality* block the 3rd manifest variable is the weakliest correlated with the corresponding latent variable, with a value bounded between 0.082 and 0.130. While, the manifest variables of the *Price Fairness* block all show the same level of correlation with the latent variable.
All this information can be easily checked also referring to figure 6.19.

Differently from the first simulation scheme, where heterogeneity only involved the structural model, here there is a difference in model performance arising in both the measurement and the structural models when comparing the global model to the local ones.
This leads to quality indexes, i.e. the R^2 value associated to the endogenous latent variable *Satisfaction* and the GoF value computed for each of the 100 data-sets, that are always definitely higher at local model level than at aggregate level (cf. figure 6.21 and table 6.22).
As a matter of fact, the R^2 value at the aggregate level is at the most equal to 0.338, while for both the detected classes of units the same is never less than 0.785 (cf. table 6.22). Therefore, the detected local models show R^2 values that are more than double the ones obtained for the aggregate level. Is it the same for the GoF values, even if the difference between the GoF value obtained from global models and

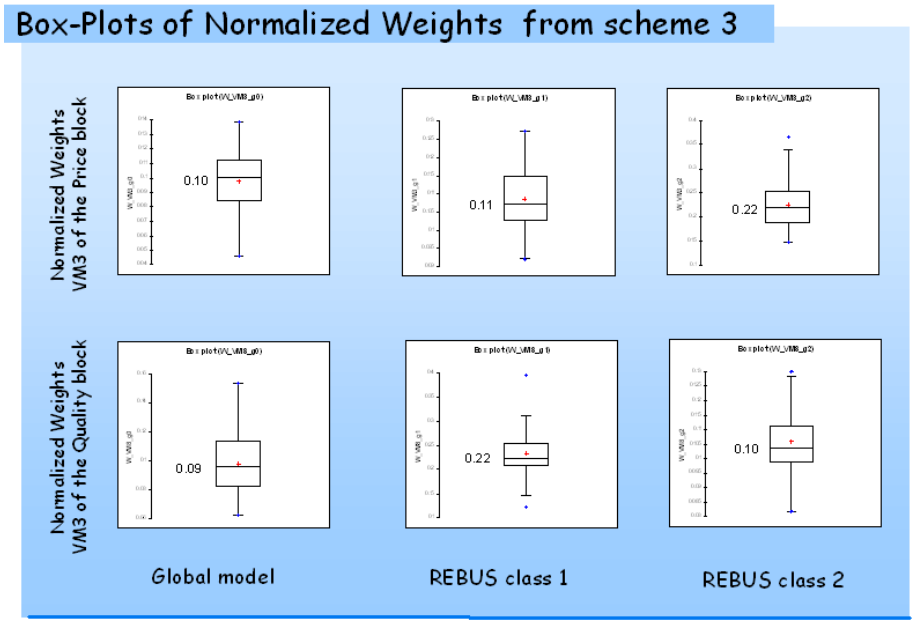

Figure 6.19: *Box-Plots for normalized weight estimates for simulation scheme* 3

the same obtained at local model level is not so strong as the one obtained for the R^2 value. This is due to the *GoF* features and to the fact that the measurement model quality indexes (i.e. the communality indexes) are only slightly affected by heterogeneity in the measurement model. As a matter of fact, the distribution of the average communality values are not different in the global model and the two detected local models (cf. figure 6.21).

Statistics	Global Model		Class 1		Class2		GQI	% well classified	Improvement of GQI
	R^2	GoF	R^2	GoF	R^2	GoF			
No. of simulated data-sets	100	100	100	100	100	100	100	100	100
Minimum	**0.174**	**0.374**	**0.791**	**0.805**	**0.785**	**0.787**	**0.808**	**0.908**	**0.584**
Maximum	**0.338**	**0.526**	**0.878**	**0.850**	**0.884**	**0.851**	**0.843**	**0.958**	**1.187**
1st Quartile	0.242	0.445	0.833	0.823	0.835	0.822	0.823	0.923	0.700
Median	0.273	0.470	0.846	0.831	0.845	0.832	0.831	0.930	0.755
3rd Quartile	0.294	0.489	0.855	0.836	0.856	0.839	0.835	0.936	0.852
Mean	0.269	0.467	0.842	0.830	0.844	0.829	0.830	0.929	0.785
Variance (n-1)	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.015
Standard deviation (n-1)	0.036	0.032	0.017	0.010	0.019	0.013	0.008	0.011	0.121
Variation coefficient	0.134	0.069	0.020	0.011	0.022	0.016	0.009	0.012	0.154
Lower bound on mean (95%)	0.262	0.461	0.839	0.828	0.840	0.827	0.828	0.927	0.761
Upper bound on mean (95%)	0.276	0.473	0.846	0.832	0.848	0.832	0.831	0.931	0.809

Figure 6.20: *Descriptive Statistics for the* R^2 *values, the GoF values, the GQI values, the well-classified rate and the improvement of the GQI obtained from the* 100 *data-sets simulated according to simulation scheme* 3

Moreover, also in this last example the well-classified rate can be used to assess the REBUS-PLS capability to detect the simulated group of data. Once again, REBUS-PLS shows its ability to correctly assign units to the corresponding simulated group, with a well-classified rate never lower than 90.8% (cf. table 6.22).

To conclude, the *Group Quality Index*, as presented in section 5.3, is always higher than 0.808, with an improvement in the model quality (in terms of the *GoF* value) always higher than 58.4% (cf. table 6.22).

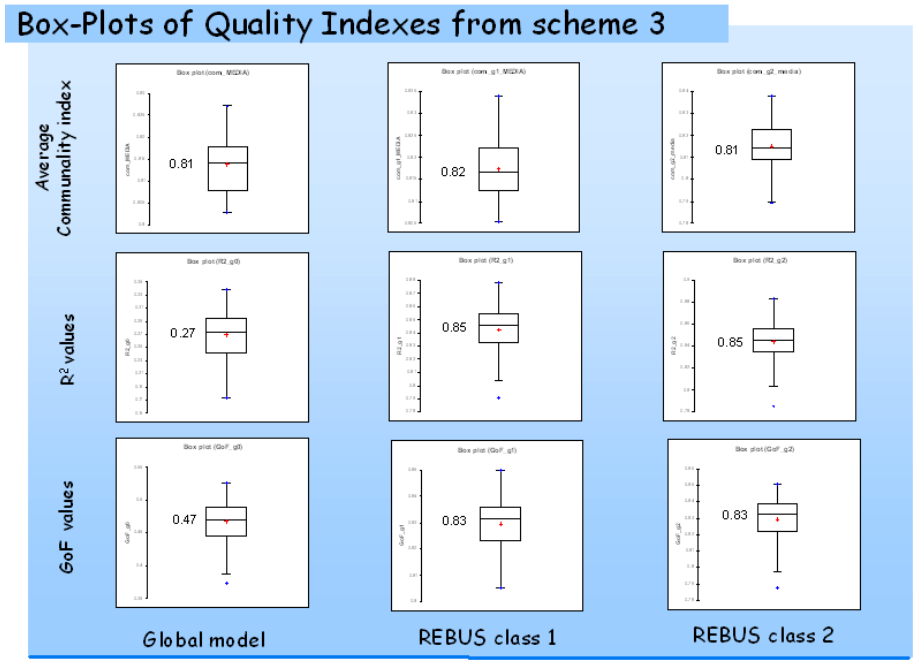

Figure 6.21: *Box-Plots for the Average Communality values, the R^2 values and the GoF values computed for simulation scheme* 3

The Box-Plot for the *GQI* and for the well-classified rate summarizing the distribution of these two values in the 100 simulated data-sets for simulation scheme 3 are provided in figure 6.23

Statistics	Global Model		Class 1		Class2		GQI	% well classified	Improvement of GQI
	R^2	GoF	R^2	GoF	R^2	GoF			
No. of simulated data-sets	100	100	100	100	100	100	100	100	100
Minimum	**0.174**	**0.374**	**0.791**	**0.805**	**0.785**	**0.787**	**0.808**	**0.908**	**0.584**
Maximum	**0.338**	**0.526**	**0.878**	**0.850**	**0.884**	**0.851**	**0.843**	**0.958**	**1.187**
1st Quartile	0.242	0.445	0.833	0.823	0.835	0.822	0.823	0.923	0.700
Median	0.273	0.470	0.846	0.831	0.845	0.832	0.831	0.930	0.755
3rd Quartile	0.294	0.489	0.855	0.836	0.856	0.839	0.835	0.936	0.852
Mean	0.269	0.467	0.842	0.830	0.844	0.829	0.830	0.929	0.785
Variance (n-1)	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.015
Standard deviation (n-1)	0.036	0.032	0.017	0.010	0.019	0.013	0.008	0.011	0.121
Variation coefficient	0.134	0.069	0.020	0.011	0.022	0.016	0.009	0.012	0.154
Lower bound on mean (95%)	0.262	0.461	0.839	0.828	0.840	0.827	0.828	0.927	0.761
Upper bound on mean (95%)	0.276	0.473	0.846	0.832	0.848	0.832	0.831	0.931	0.809

Figure 6.22: *Descriptive Statistics for the R^2 values, the GoF values, the GQI values, the well-classified rate and the improvement of the GQI obtained from the* 100 *data-sets simulated according to simulation scheme* 3

6.1.5 Conclusion

On the basis of this simulation study is possible to state that the REBUS-PLS algorithm is able to detect unobserved heterogeneity not only when it affects the whole model (i.e. both the measurement and structural models), but also when it focuses only on the structural model level or on the measurement model level.
As a matter of fact, out of all the 300 simulated data-sets, REBUS-PLS never achieved a well-classified rate lower than 86%.
Its ability to detect homogeneous group of units, however, is stronger

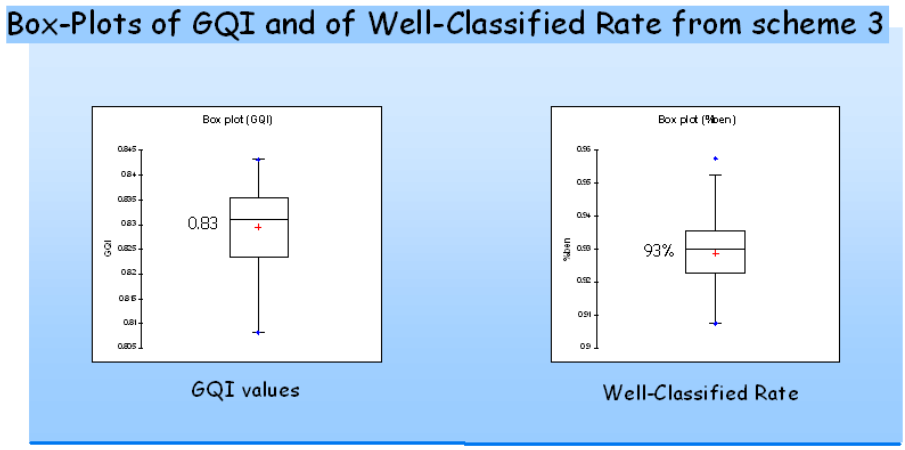

Figure 6.23: *Box-Plots for GQI and well-classified rate computed for simulation scheme* 3

if the unobserved heterogeneity is focused on the structural model or on the whole model, than if it simply affects the measurement model. In the author's opinion, this is not a great problem in real application. As a matter of fact, to assume that heterogeneity only affects the measurement model is an unrealistic assumption. At well as, to assume that the same only affects the structural model as in the case of FIMIX-PLS.

As far as future developments are concerned it would be interesting to evaluate REBUS-PLS capabilities when dealing with alterations

of segment sizes as well as in data distribution. Moreover, it would be interesting to perform a simulation study in order to evaluate the REBUS-PLS behavior under the so-called "null hypothesis" of homogeneity in the sample. The author has already conducted studies in this sense and the results obtained so far are positive. Nevertheless, a more organic and systematic study is necessary.

6.2 Real data example

The REBUS-PLS algorithm has already been tested also on some empirical data [Esposito Vinzi, Trinchera, Squillacciotti & Tenenhaus 2008, Trinchera et al. 2006, Esposito Vinzi, Amato & Trinchera 2008]. Here the author decides to present a simple and clear example to show the REBUS-PLS ability to handle unobserved heterogeneity on empirical data. Moreover, the author decides to use a dataset that has already been used in the literature [Ringle et al. 2008], in order to indirectly compare the REBUS-PLS results with the ones obtained using methods other than REBUS-PLS.

Due to this, the author decides to use empirical data coming from the Gruner&Jahr's 'Brigitte Communication Analysis performed in 2002 that specifically concerns the Benetton fashion brand.
Gruner&Jahr is one of the leading publishers of printed magazines in Germany. Since 1984, they have conducted each year a Communication Analysis Survey. In the survey, over 5.000 women answer numer-

ous questions on brands in different product categories and questions regarding their personality. The women represent a cross section of the German female population. As Ringle et al. [2008] suggest Benetton's aggressive and provocative advertising in the 1990s resulted in a lingering customer heterogeneity that is more distinctive and easier to identify compared with other fashion brands in the Communication Analysis Survey (e.g. Esprit or S.Oliver). For this reason, as well as for comparing the REBUS-PLS results with the Ringle et al. [2008] ones, the author chooses to use the answers to questions on the Benetton fashion brand.

The scope of this work neither includes a presentation of a theoretically hypothesized Path Model scheme, nor a discussion on whether the measurement models of latent variables should be operationalized as formative or reflective [Diamantopoulos & Winkelhofer 2001, Rossiter. 2002]. Moreover, an extensive presentation of the survey data went beyond the aim of this paragraph.
Our goal is simply to show the applicability of REBUS-PLS to empirical data for a reduced cause-effect relationship model on branding [Yoo, Donthu & Lee 2000] that principally guides all kinds of Structural Equation Models analysis in marketing employing this clustering technique.

The Benetton dataset, as used by Ringle et al. [2008], is composed of 10 manifest variables observed on 444 German women. Each manifest

variable is a question in the Gruner&Jahr's 'Brigitte Communication Analysis of 2002. The women had to answer each question using a four-point scale from "low" to "high".
The PLS Path Model scheme for Benetton's brand preference, as used by Ringle et al. [2008], consists of one latent endogenous *Brand Preference* variable, and two latent exogenous variables, *Image* and *Character*. All latent variables are linked to the corresponding latent variable via a reflective measurement model. Figure 6.24 illustrates the path model with the latent variables and the particular manifest variables from Gruner&Jahr's 'Brigitte Communication Analysis 2002' employed. A list of the used manifest variables with the corresponding meanings is shown in table 6.25.

A SAS-IML macro developed by the author (cf. appendix A.2) has been used to perform a simple PLS Path Modeling analysis on the whole sample. In other words, the global model estimates have been obtained by using the PLS-PM SAS-IML macro imposing the number of classes equal to zero, i.e. $ncla = 0$. As is obvious, the global model estimates are consistent with the ones obtained by Ringle et al. [2008] in their study. A simple overview of the global model results is proposed in figure 6.26. According to the global model results *Image* seems to be the most important driver for *Brand Preference*, with a path coefficient equal to 0.423. The influence of the latent exogenous *Character* variable is considerably weaker (path coefficient of 0.177). Nevertheless, the R^2 value associated with the endogenous latent vari-

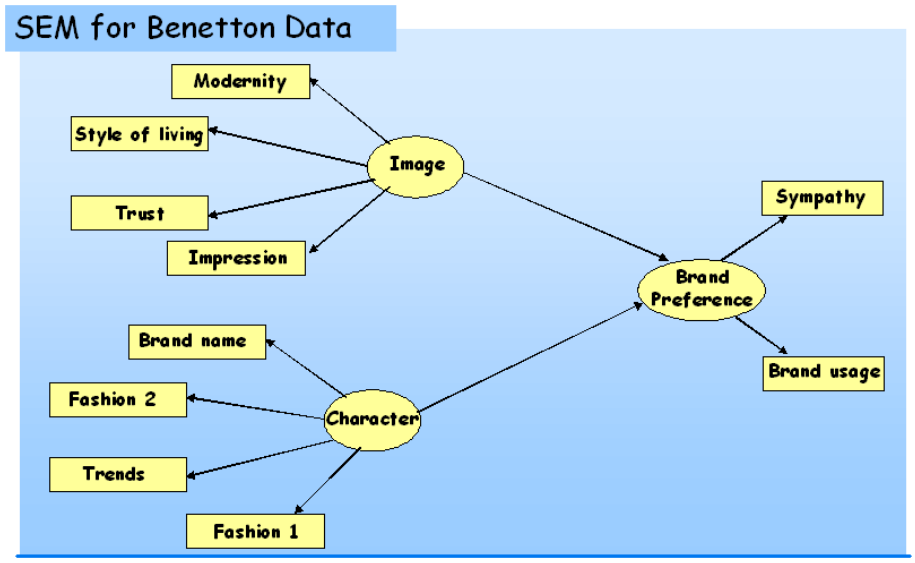

Figure 6.24: *Path diagram for Benetton data*

able *Brand Preference* is quite low, being equal to 0.239. Ringle et al. [2008] consider this value as a moderate level for a PLS Path Model. In the author's opinion a R^2 value of 0.239 has to be considered as unsatisfactory, and could be used as a first sign of heterogeneity in the data.

Looking at the measurement models, all the relationships in the reflective measurement model have high factor loadings (the smallest loading has a value of 0.795, cf. table 6.27). In figure 6.26 the normalized outer weights are shown. Differences in the manifest variables

LV name	MV name	Concept
Image	Modernity	It is modern and up to date
	Style of living	Represents a great style of life
	Trust	This brand can be trusted
	Impression	I have a clear impression of this brand
Character	Brand name	A brand name is very important to me
	Fashion2	I often talk about fashion
	Trends	I am interested in the latest trends
	Fashion1	Fashion is a way to express who I am
Brand Preference	Sympathy	Sympathy
	Brand Usage	Brand Usage

Figure 6.25: *Manifest Variable meanings and block definition for Benetton Data*

impact arise in the *Brand Preference* block. As a matter of fact, the outer weights of the exogenous block are quite similar to each other, while in the endogenous block the latent variable is more correlated with the manifest variable *Sympathy* than with the *Brand Usage*. To conclude, the global model on Benetton data shows a GoF value equal to 0.422 (cf. table 6.28). The quite low value of the GoF index also suggests that we have to look for more homogeneous segments among the units.

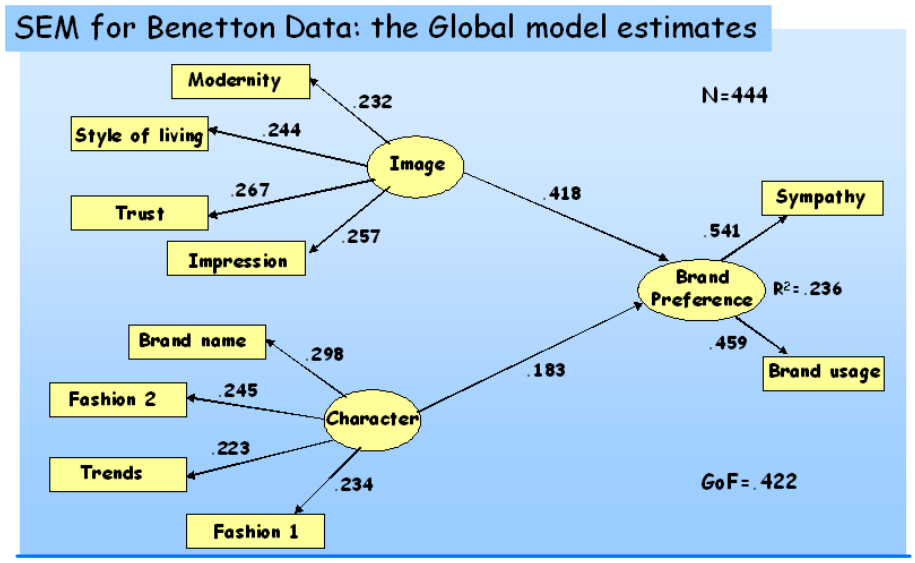

Figure 6.26: *Global model results from Benetton data obtained by using a SAS-IML macro*

A more complete outline of the global model results is provided in table 6.27 and in table 6.28. In these tables all the PLS Path Modeling results, such as the communality indexes and the redundancy indexes, as well as model parameter estimates, and the corresponding interval of confidence obtained by bootstrap, are shown.
These tables contain also the class-specific results in order to make it easier to compare the segments.

		GLOBAL	CLASS 1	CLASS 2	CLASS 3
Number of observations		444	105	141	198
Normalized outer weights of Image	Modernity	0.232	0.223	0.222	**0.188**
	Style of living	0.244	0.287	0.269	0.303
	Trust	0.267	0.245	0.265	0.293
	Impression	0.257	0.245	0.244	0.217
Normalized outer weights of Character	Brand name	0.298	0.295	0.239	0.242
	Fashion2	0.245	0.225	0.302	0.259
	Trends	0.223	0.225	0.284	0.298
	Fashion1	0.234	0.255	**0.176**	0.202
Normalized outer weights of Brand Preference	Sympathy	0.541	0.426	**0.610**	**0.630**
	Brand Usage	0.459	**0.574**	0.390	0.370
Standardized loadings of Image	Modernity	0.795	0.829	0.818	**0.666**
	Style of living	0.834	0.834	0.845	0.886
	Trust	0.891	0.899	0.881	0.870
	Impression	0.865	0.860	0.851	0.831
Standardized loadings of Character	Brand name	0.855	0.842	0.851	0.812
	Fashion2	0.892	0.841	0.929	0.904
	Trends	0.861	0.850	0.903	0.887
	Fashion1	0.794	**0.810**	0.776	**0.767**
Standardized loadings of Brand Preference	Sympathy	0.954	**0.760**	**0.850**	**0.942**
	Brand Usage	0.922	**0.909**	**0.478**	**0.616**
Communality index in the Image block	Modernity	0.632	0.687	0.670	**0.444**
	Style of living	0.696	0.696	0.713	**0.784**
	Trust	0.794	**0.808**	0.776	**0.757**
	Impression	0.748	0.740	0.724	**0.690**
Communality index in the Character block	Brand name	0.731	0.710	0.724	0.659
	Fashion2	0.796	**0.707**	**0.863**	0.817
	Trends	0.741	0.722	**0.816**	0.787
	Fashion1	0.631	**0.656**	0.602	**0.588**
Communality index in the Brand Prefernce block	Sympathy	0.909	**0.578**	**0.722**	**0.887**
	Brand Usage	0.850	0.827	**0.228**	0.379

Figure 6.27: *Measurement model results for the global model and the local models obtained by REBUS-PLS*

		GLOBAL	CLASS 1	CLASS 2	CLASS 3
Number of observations		444	105	141	198
Path coefficients on Brand Preference	Image	0.418 [0.303; 0.497]	0.406 [0.228; 0.542]	**0.692** [0.579; 0.787]	0.460 [0.300; 0.605]
	Character	0.183 [0.104; 0.207]	0.289 [0.075; 0.448]	0.334 [0.229; 0.437]	0.162 **[-0.024; 0.305]**
Redundancy index on Brand Preference	Image	0.215	0.168	0.483	0.227
	Character	0.201	0.240	0.153	0.097
R^2 value on Brand Preference		0.236 [0.164; 0.307]	0.308 [0.164; 0.460]	**0.669** [0.485; 0.750]	0.27 [0.151; 0.423]
R^2 contribute	Image	0.80	0.64	0.77	0.86
	Character	0.20	**0.36**	0,23	**0.14**
GoF value		0.422 [0.346; 0.481]	0.455 [0.325; 0.567]	**0.676** [0.565; 0.727]	0.417 [0.309; 0.555]

Figure 6.28: *Structural model results for the global model and the local models obtained by REBUS-PLS*

Performing REBUS-PLS on Benetton data allows us to detect three different classes of units showing homogeneous behavior. As a matter of fact, the cluster analysis performed on the residuals from the global model (cf. figure 6.29) suggests that we to look for two or three latent classes. Both the partitions have been investigated.

Nevertheless, the three class solution has been preferred to the two class solution. As a matter of fact, the three class partition shows a *Group Quality Index* higher than the two class one. Moreover, the GQI index computed for the two class solution ($GQI = 0.454$) is close to the GoF value computed for the global model (i.e. the GQI index

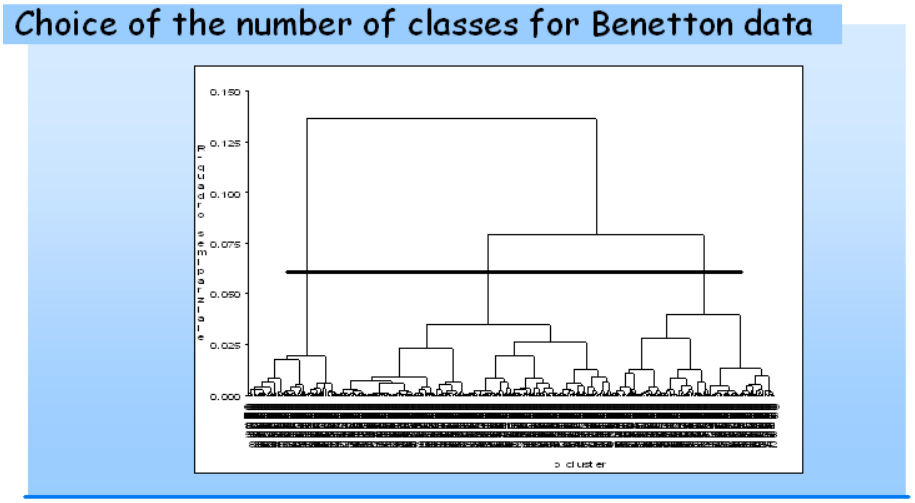

Figure 6.29: *Dendograme obtained by performing a cluster analysis on the residuals from the global model (Step 3 of the REBUS-PLS algorithm)*

in the case of only one class) ($GoF = 0.422$). The 25% improvement forseen to consider the obtained unit partition better than the unique class solution (cf. section 6.1.1) was not achieved.
Here only the results for the three class partition will be presented. Results concerning the two class solutions can be found in the appendix A.1.

As already said, thanks to the REBUS-PLS algorithm the 444 units

have been clustered in three classes that are more homogeneous as regards the model parameters.

The first class is composed of 105 units, i.e around 24% of the whole sample. This class is characterized by a path coefficient linking the latent variable *Character* to the endogenous latent variable higher than the one obtained for the global model. Moreover, differences in unit behaviors arise also as regards the correlations amongs manifest and latent variables in the *Brand Preference* block. Figure 6.30 shows the PLS Path Model parameter estimates obtained for the first class. Differently from the global model, in the *Brand Preference* block the manifest variable *Brand Usage* shows higher outer weight than *Sympathy*. It is the same for the manifest variable *Fashion1* that shows a lower correlation with the corresponding latent variable than in the global model and in the first class model.
The quality index values for the local model computed for the first groups are close to the ones obtained for the global model, with a R^2 associated to the endogenous block equal to 0.308, and a GoF value equal to 0.455.

The second class, instead, shows quality index values definitely higher than the global model, with a GoF value of 0.676 and a R^2 value for the latent variable *Brand Usage* equal to 0.669 (cf. table 6.28). This class is composed of around 32% of the whole sample, and it is characterized by a higher path coefficient associated to the relationship between the

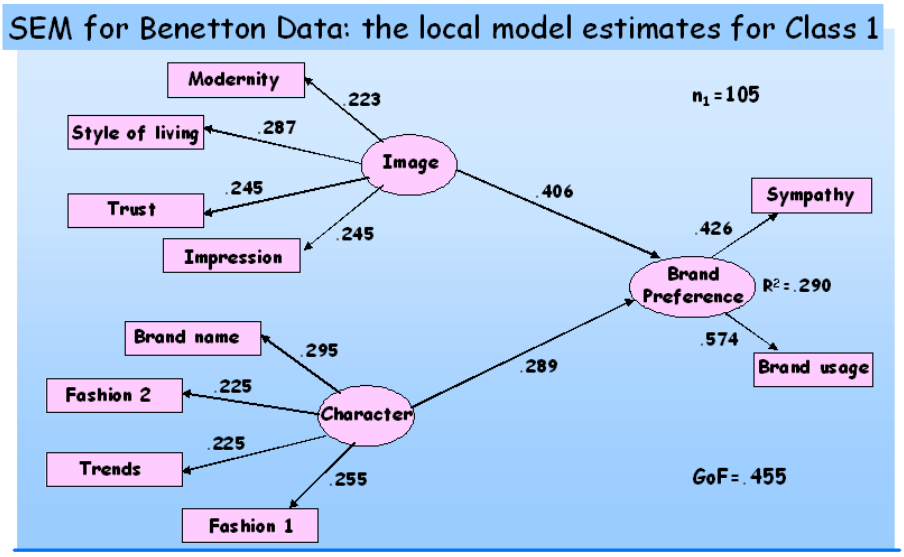

Figure 6.30: *Local model results for the first group detected by performing REBUS-PLS algorithm on Benetton data*

Image and the *Brand Preference*. Looking at the measurement model (cf. table 6.27), differences arise in the *Brand Preference* block and in the *Character* block. As a matter of fact, the communality index (i.e. the square of the correlation) between the manifest variable *Brand Usage* and the corresponding latent variable *Brand Preference* is really lower than the one obtained for the global model and for the first group local model. It is the contrary for the manifest variable *Sympathy* that here shows a higher normalized weight value. As regards

the *Character* block, the manifest variable *Fashion2* shows a higher correlation with the corresponding latent variable *Character* than in the global model or in the local model for group 1.
A synthesis of the results obtained for the second class is provided in figure 6.31.

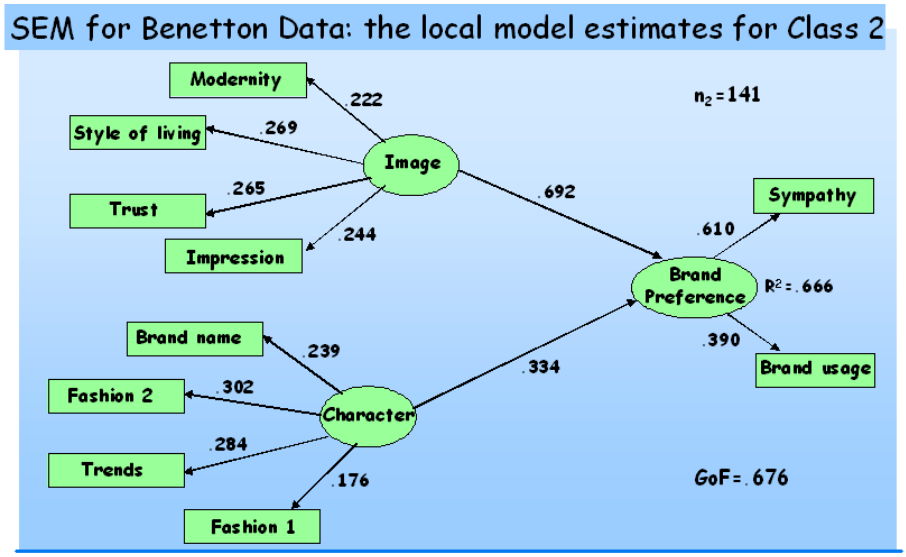

Figure 6.31: *Local model results for the second group detected by performing the REBUS-PLS algorithm on Benetton data*

To conclude, the results for the third detected class are presented in figure 6.32. This class is composed of 198 units., i.e. more than 44%

of the whole sample. It is characterized by a very weak relationship between the latent variable *Character* and the endogenous latent variable *Satisfaction*. Moreover, the bootstrap interval shows that this relation is not significant (cf. table 6.28). Differences arise also as regards the measurement block, notably in the *Image* block. As a matter of fact, in this class the manifest variable *Modernity* shows a very low correlation compared with the other model results. While, the manifest variable *Style of life* seems to be slightly more correlated with the latent variable *Image* than in the other models.
Nevertheless, the quality index values computed for the third local model are only slightly different from the global model one ($R^2 = 0.27$ and $GoF = 0.417$).

The three class solution shows a *Group Quality Index* equal to 0.531. In order to validate the REBUS-PLS based partition, an empirical distribution of the GQI values is computed. Following the permutation test approach (cf. subsection 4.5.1) the whole sample has been randomly divided 300 times into three classes of the same size as the ones detected by REBUS-PLS. The GQI has been computed for each of the random partitions of the units.
The empirical distribution of the GQI values for a three class partition of the units is therefore obtained (cf. figure .38). The GQI value obtained from the REBUS-PLS partition of the units is definitely an extreme value of the distribution. Moreover, analyzing the Box-Plot obtained for the empirical distribution of the GQI values (cf. figure

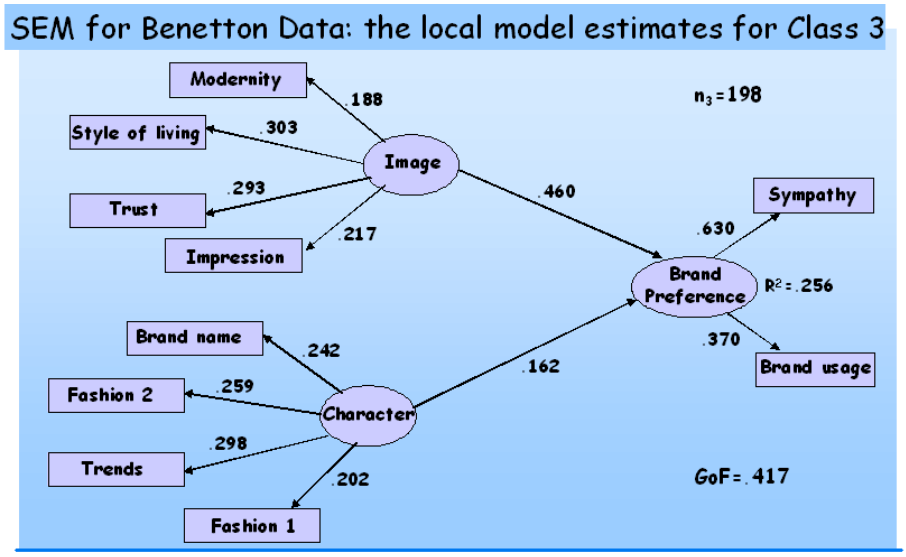

Figure 6.32: *Local model results for the third group detected by performing the REBUS-PLS algorithm on Benetton data*

6.34), it is possible to notice that the GQI computed for the global model (i.e. the GoF value computed for the global model) is the smaller value obtained for the GQI, except for outliers.
This means that a partition of units in latent classes always surpassed the performance of the global model. In other words, the global model has to be definitely considered as affected by heterogeneity. Moreover, the GQI value obtained for the REBUS-PLS based partition is the higher obtained value. This allow us to assess that the REBUS-PLS

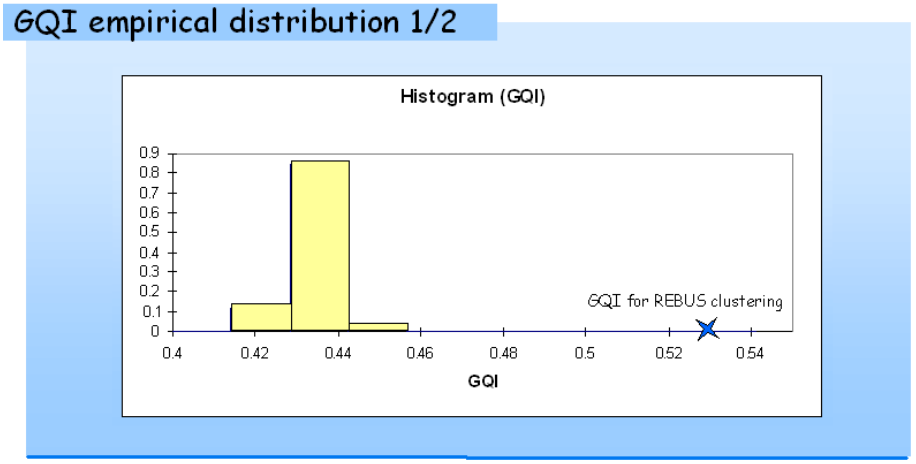

Figure 6.33: *Empirical distribution of the GQI computed on* 300 *random partition of the sample in three classes*

based clustering of the units is better than a random assignment of the units, and is definitely better than the global model solution.

Comparing REBUS-PLS results with the Ringle et al. [2008] ones

Ringle et al. [2008] applied FIMIX-PLS to Benetton data.
As already said (cf. subsection 4.3.1), the FIMIX-PLS look for heterogeneity only in the structural model. The measurement model parameters remain constant among the local models. In other words, the detected classes are different only as regards the path coefficient intensities.

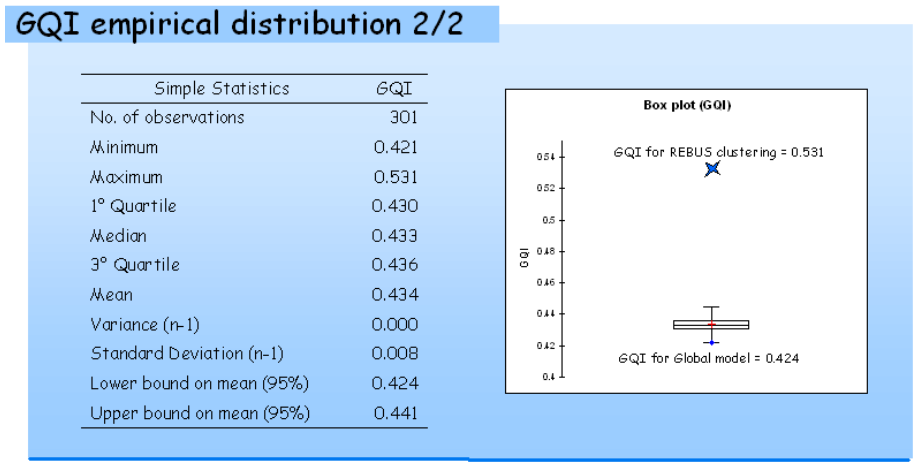

Figure 6.34: *Box-Plots obtained for the empirical distribution of the GQI values*

Moreover, FIMIX-PLS only provides a *fuzzy* clustering of the units. As a matter of fact, all the units are supposed to belong to all the detected latent classes, with a particular degree of membership.
To conclude, if there is no information about the number of classes to take into account, FIMIX-PLS needs to be performed to successive numbers of latent classes in order to identify the better partition.
According to the FIMIX-PLS features Ringle et al. [2008] identify only two segments. The first one (80.9% of the whole sample) overlaps the global model results in terms of path coefficient estimates. Nevertheless, the R^2 value associated to the endogenous latent variable *Satis-*

faction is equal to 0.108. This is a very small value, compared with the already small global model one ($R^2 = 0.236$).
The second detected segment (19.1% of the whole sample), instead, overlaps the results obtained by REBUS-PLS for the second class. As a matter of fact, also in this case the exogenous latent variable *Image* seems be the most important driver for *Brand Preference*, showing an R^2 value close to the unit.

In order to obtain local models that are different also as regards the measurement model, Ringle et al. [2008] applied a two step strategy. In the first step they simple apply FIMIX-PLS. Successively they used such external/concomitant variables to look for groups overlapping the FIMIX-based ones.
Nevertheless, also in this two step procedure the obtained results are not better than the ones provided by the REBUS-PLS based partition. As a matter of fact, the R^2 value and the GoF value of the first local model are smaller than the global model ones. In other words, the local model for the biggest segment (80% of the whole sample) performs worse than the global model, and worst of all the REBUS-PLS based local models.

6.2.1 Conclusion

The detection of unobserved heterogeneity in Structural Equation Models, especially in the marketing field, is a very important task. As a matter of fact, our simulation study (cf. section 6.1.1), as well as

other authors [Jedidi et al. 1997a, Jedidi et al. 1997b] underline that treating a sample as homogeneous when it is not, may lead to model parameter estimates that are biased.
The REBUS-PLS algorithm turned out to be a powerful tool to detect unobserved heterogeneity in both experimental and empirical data.

In particular, as regards the real data application, REBUS-PLS was able to detect three latent classes of units, showing different behavior. Moreover, all the obtained local models perform better than the global model, with R^2 values and GoF values higher than the global model ones.
To conclude, a permutation test performed on the *Group Quality Index* has proved that the REBUS-PLS based partition is the best one according to the prediction capability of the model.

Appendix

A.1 The REBUS-PLS results for the two class solution on Benetton data

Here the results obtained by performing REBUS-PLS algorithm on Benetton data, in the case of two latent class, are presented.

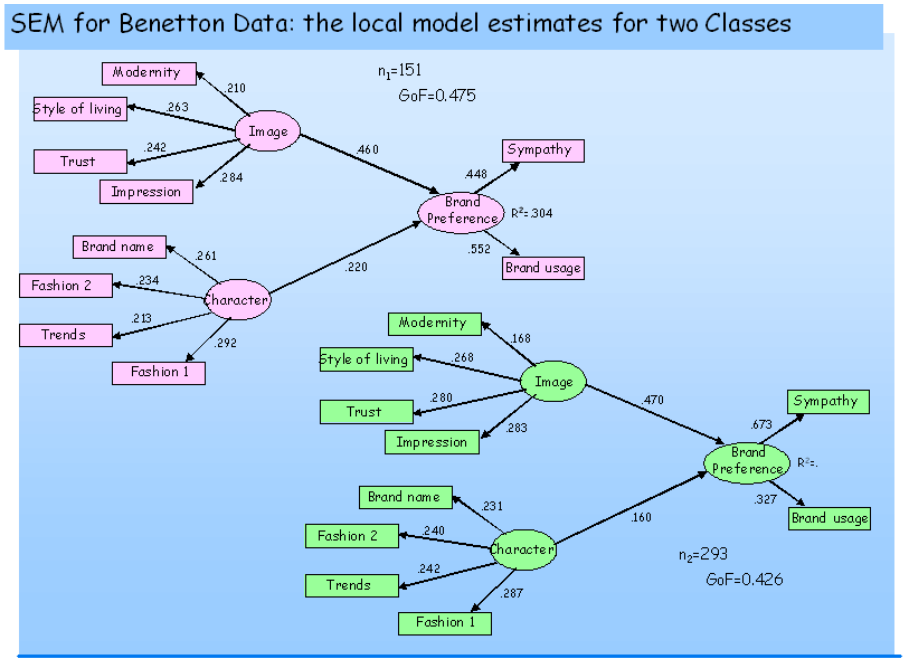

Figure .35: *Local model results for the two groups detected by performing REBUS-PLS algorithm on Benetton data*

		GLOBAL	CLASS 1	CLASS 2
Number of observations		444	151	293
Normalized outer weights of Image	Modernity	0.232	0.210	**0.168**
	Style of living	0.244	0.263	0.268
	Trust	0.267	0.242	0.280
	Impression	0.257	0.284	0.283
Normalized outer weights of Character	Brand name	0.298	0.261	0.231
	Fashion2	0.245	0.234	0.240
	Trends	0.223	0.213	0.242
	Fashion1	**0.234**	0.292	0.287
Normalized outer weights of Brand Preference	Sympathy	0.541	0.448	**0.673**
	Brand Usage	0.459	**0.552**	0.327
Standardized loadings of Image	Modernity	0.795	**0.842**	**0.695**
	Style of living	0.834	0.837	0.855
	Trust	0.891	0.901	0.867
	Impression	0.865	0.892	0.858
Standardized loadings of Character	Brand name	0.855	0.827	0.818
	Fashion2	0.892	0.857	0.901
	Trends	0.861	0.866	0.864
	Fashion1	**0.794**	0.831	0.823
Standardized loadings of Brand Preference	Sympathy	0.954	**0.827**	0.966
	Brand Usage	0.922	**0.922**	**0.562**
Communality index in the Image block	Modernity	0.632	0.709	**0.482**
	Style of living	0.696	0.701	0.731
	Trust	0.794	**0.812**	0.751
	Impression	0.748	0.795	0.737
Communality index in the Character block	Brand name	0.731	**0.684**	0.668
	Fashion2	0.796	**0.735**	**0.813**
	Trends	0.741	0.750	0.746
	Fashion1	0.631	0.691	0.677
Communality index in the Brand Prefernce block	Sympathy	0.909	**0.684**	0.932
	Brand Usage	0.850	0.850	**0.316**

Figure .36: *Measurement model results for the global model and the local models obtained by REBUS-PLS for the two class solution*

		GLOBAL	CLASS 1	CLASS 2
Number of observations		444	151	293
Path coefficients on Brand Preference	Image	0.418	0.460	0.47
		[0.303; 0.497]	[0.306; 0.590]	[0.378; 0.581]
	Character	0.183	0.220	0.160
		[0.104; 0.207]	[0.118; 0.361]	[0.052; 0.291]
Redundancy index on Brand Preference	Image	0.215	0.208	0.247
	Character	0.201	0.259	**0.084**
$R^{\wedge 2}$ value on Brand Preference		0.236	0.304	0.288
		[0.164; 0.307]	[0.196; 0.472]	[0.190; 0.380]
$R^{\wedge 2}$ contribute	Image	0.80	**0.77**	**0.87**
	Character	0.20	**0.23**	**0.13**
GoF value		0.422	0.475	0.426
		[0.346; 0.481]	[0.373; 0.597]	[0.352; 0.508]

Figure .37: *Structural model results for the global model and the local models obtained by REBUS-PLS for the two class solution*

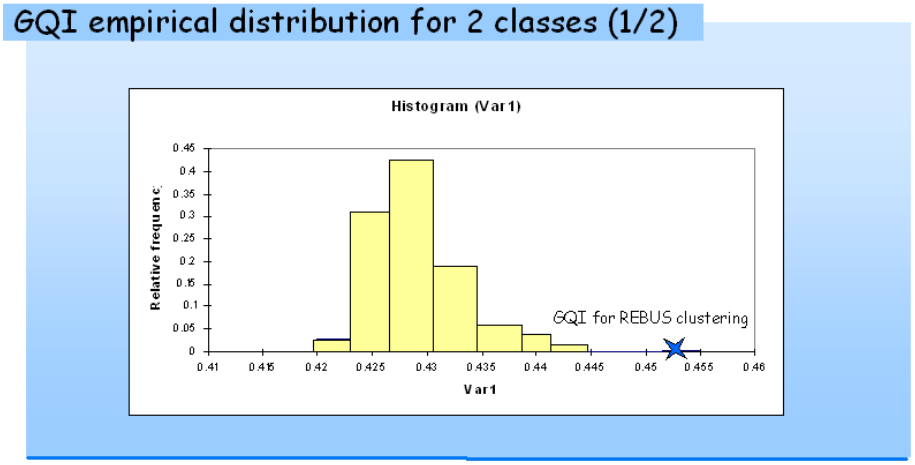

Figure .38: *Empirical distribution of the GQI computed on* 300 *random partitions of the sample in two classes*

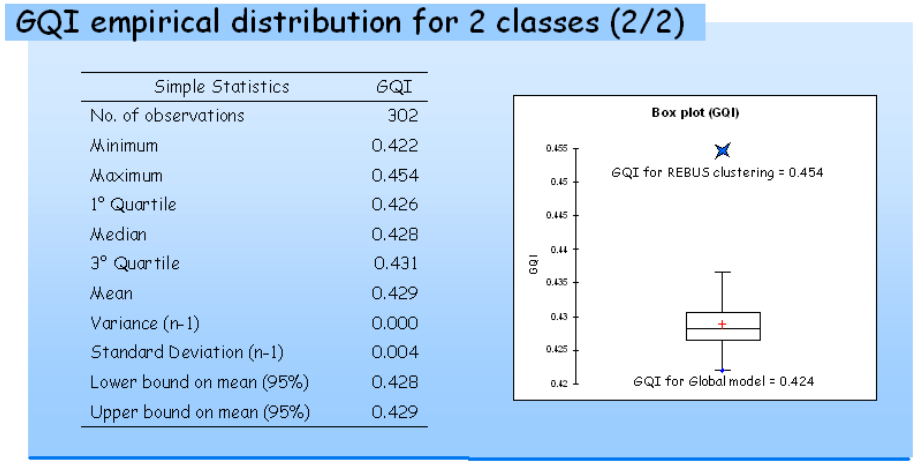

Figure .39: *Box and Whisker Plot obtained for the empirical distribution of the GQI values for two class solution*

A.2 The SAS-IML code for the REBUS-PLS algorithm

```
/*************************************************************************/
/*                                                                       */
/*       This procedure allow to perform the last version  of            */
/*                        REBUS-PLS algorithm.                           */
/*                                                                       */
/* Author : Laura Trinchera - DMS - Università di Napoli FEDERICO II     */
/*                          November 30 2007                             */
/*                                                                       */
/*    The REBUS-PLS is an iterative algorithm that allows to perform     */
/*         response based clustering in a PLS-PM framework.              */
/*                                                                       */
/* 1 At the initial step a global PLS-PM is performed on all units       */
/*                    (Macro PLS_PM with nclas=0)                        */
/*                                                                       */
/* 2 The residuals of each unit from the global model are computed       */
/*                  (Macro dis_res with DIST = 'NO')                     */
/*                                                                       */
/* 3 A cluster analysis is then performed on the residuals computed      */
/*   at step 2                                                           */
/*                       (Macro cluster_for_1_g )                        */
/*                                                                       */
/* 4 Once the number of segments defined, looking at the dendogram       */
/*   obtained in step 4, the composition of classes is obtained          */
/*                         (Macro update_class)                          */
/*                                                                       */
/* 5 The local models (one for each segment) are estimated               */
/*                          (Macro PLS_PM)                               */
```

```
/*                                                                  */
/* 6 A measure of the distance of each unit from each model is then */
/*   computed                                                       */
/*                 (Macro dis_res with DIST = 'YES')                */
/*                                                                  */
/* 7 All units are reassigned to the class corresponding to the     */
/*   closest local model                                            */
/*                       (Macro alloc_units)                        */
/*                                                                  */
/* 8 Local models are estimated for each new class                  */
/*                         (Macro PLS_PM)                           */
/*                                                                  */
/*     THE ALGORITHM IS REITERATED UNTIL CONVERGENCE IS OBTAINED     */
/*  N.B. convergence is obtained when group membership are stable   */
/* from one iteration to the other and the macro alloc_units print  */
/*                            "STOP"                                */
/*                                                                  */
/*  ONCE THE CONVERGENCE RICHED the final local models, as well as  */
/*  the GQI index, are computed by the macro "PLS_PM" with GQI=YES  */
/******************************************************************/

/******************************************************************/
/*                           NOTATION                               */
/*                     (valid for all macros)                       */
/*  - Matrix X containing all the MVs values for all the units      */
/*    (raw data)MUST BE CALLED C1_0                                 */
/*  - the index g&k indicates the number of classes, e.g.:          */
/*     -  COMM_g&0_M is the average communality of the              */
/*        GLOBAL MODEL                                              */
/*     -  COMM_g&1_M is the average communality of the MODEL        */
/*        estimated for group one, and so on...                     */
```

```
/*  - the index b&q refers to the blocks                                */
/*********************************************************************/


/*********************************************************************/
/*                          Macro PLS_PM                               */
/*                                                                     */
/*      This macro allows to estimate the same PLS-PM diagram          */
/*     either on different classes or on a single (global) model       */
/*                                                                     */
/* The macro parameters:                                               */
/*                                                                     */
/* LIBNAME = the library where the results will be stocked and where */
/*           the data to be analyzed are stocked                       */
/* TABLE = name of the table containing the data to analyze            */
/* ID = The unit's identifier. It must be Text type: usually id_1,     */
/*      id_2, id_3, and so on                                          */
/* NCLAS = the number of classes for which the PLS-PM is performed     */
/* NBLOC = the number of PLS_PM blocks (each block is formed  by       */
/*         a LV and the corresponding MVs)                             */
/* NVLendo= the number of endogenous blocks in the model               */
/* GQI = Global Quality Index. GQI = 'YES' allowing the macro to       */
/*       compute the GQI, while GQI = 'NO' does not including the      */
/*       computation of the GQI in the macro.                          */
/*       GQI have to be computed only once the convergence assured     */
/*       to assess unit clustering. Therefor, GQI have to be kipped    */
/*       equal to 'NO' until the last running of this macro!           */
/*                                                                     */
/* N.B. Some steps of this macro depend on the specification of the    */
/*      inner model and of the outer model. Hence, some steps are      */
/*      to be MANUALLY modified by the user according to the model     */
```

```
/*      specification BEFORE running the macro.                    */
/*****************************************************************/

%macro PLS_PM(libname=,table=,id=,nclas=,nbloc=,nVLendo=,GQI=);
data &libname..Cl_0;
  set &libname..&table;
run;
proc iml;
  %do k=0 %to &nclas;
  use &libname..Cl_&k;
  read all into X_g&k [rowname=&id colname=colX&k];
    N_g&k=nrow(X_g&k);
    print N_g&k;

  /*************************************************************/
  /*      Definition of the outer model by allocating each MV,    */
  /*             i.e. each X column, to its block.                */
  /*                                                              */
  /* The blocks are indicated by successive numbers               */
  /*************************************************************/

  /*************************************************************/
  /*  This step is to be MANUALLY completed by the user BEFORE    */
  /*                        running the macro                     */
  /*             (assignment of each MV to its block)             */
  /*************************************************************/

  /* replace into brackets the names of the observed variables for*/
  /* each block and define the name of the latent variable        */

    use &libname..Cl_&k var{IM1 IM2 IM3 IM4};
```

```
    read all into X_image_g&k;
  use &libname..Cl_&k var{Brand1 Brand2 Brand3 Brand4};
    read all into X_brand_g&k;
  use &libname..Cl_&k var{SAT1 SAT2};
    read all into X_sat_g&k;
/*****************************************************************/
/* ALWAYS KEEP THE SAME ORDER FOR THE EXOGENOUS LATENT VARIABLES */
/*****************************************************************/

  %do q=1 %to &nbloc;
    %if &q=1 %then  %do; X_g&k._b&q=X_image_g&k;
    %end;
    %if &q=2 %then  %do; X_g&k._b&q=X_brand_g&k;
    %end;
    %if &q=3 %then  %do; X_g&k._b&q=X_sat_g&k;
    %end;
  %end;

  %do q=1 %to &nbloc;
    mean_VM_g&k._b&q=(X_g&k._b&q[+,])/N_g&k;
    print mean_VM_g&k._b&q;
    quad_VM_g&k._b&q=X_g&k._b&q##2;
    mean_quad_VM_g&k._b&q=(quad_VM_g&k._b&q[+,])/N_g&k;
    print mean_quad_VM_g&k._b&q;
    var_VM_g&k._b&q=mean_quad_VM_g&k._b&q-mean_VM_g&k._b&q##2;
    sqm_VM_g&k._b&q=sqrt(var_VM_g&k._b&q);
    print sqm_VM_g&k._b&q;
    XS_g&k._b&q=standard(X_g&k._b&q);
    correzione=N_g&k/(N_g&k-1);
    correzione=sqrt(correzione);
    XS_g&k._b&q=XS_g&k._b&q*correzione;
```

```
    P_g&k._b&q=ncol(XS_g&k._b&q);
 %end;

/*************************************************************/
/*   P_g&k it is the total number of MVs in the outer model  */
/*       It is obtained as sum of the MVs of each bloc       */
/*************************************************************/

/*************************************************************/
/*  This step is to be MANUALLY completed by the user BEFORE */
/*                      running the macro                    */
/*               (number of blocks to be added)              */
/*************************************************************/

/* if new blocks have been added, add new addend */

 P_g&k=P_g&k._b1+P_g&k._b2+P_g&k._b3;


/*---------weights vectors initialization module----------*/

 start iniz_w (P);
   w=j(1,P,0);
   do i=1 to P;
    if i=1 then w[,i]=1;
      else w[,i]=0;
   end;
   w=w';
   return (w);
 finish iniz_w;
```

```
   %do q=1 %to &nbloc;
   w_g&k._b&q=iniz_w(P_g&k._b&q);
   %end;

  /*************************************************************/
  /*                    The PLS-PM Algorithm                   */
  /*             Maximum number of iterations: 50              */
  /*************************************************************/

   z_g&k._b3=XS_g&k._b3*w_g&k._b3;
   y_old_g&k=z_g&k._b3*100;
   do it=1 to 50 until (converg<0.0000001);
      y_old_g&k=z_g&k._b3;

/* ----- outer estimation of LVs ["csi"] ----- */
      %do q=1 %to &nbloc;
          y_g&k._b&q=XS_g&k._b&q*w_g&k._b&q;
         y_g&k._b&q=standard(y_g&k._b&q);
         y_g&k._b&q=y_g&k._b&q*correzione;
      %end;

     /* ----- estimation of the inner weights ["e"] ------ */

/*************************************************************/
/* This step is to be MANUALLY completed by the user BEFORE  */
/*                     running the macro                     */
/*(weights are obtained according to the path diagram scheme)*/
/*************************************************************/

/* ----- centroid scheme ----- */
/*each weight "e" is obtained as the sign of the correlations*/
```

```
/*---------- between the LVs linked by a causal path --------*/
      e_g&k._b1_b3=(y_g&k._b1‘*y_g&k._b3)/N_g&k;
         if e_g&k._b1_b3>0 then e_g&k._b1_b3=1;
           else if e_g&k._b1_b3<0 then e_g&k._b1_b3=-1;
      e_g&k._b2_b3=(y_g&k._b2‘*y_g&k._b3)/N_g&k;
         if e_g&k._b2_b3>0 then e_g&k._b2_b3=1;
           else if e_g&k._b2_b3<0 then e_g&k._b2_b3=-1;

    /* ----- inner estimation of the LVs ["Z"]----- */

/************************************************************/
/*  This step is to be MANUALLY completed by the user BEFORE */
/*                      running the macro                    */
/*  (inner estimates depend on the path diagram scheme)      */
/************************************************************/

/*check carefully all links between LVS for inner estimation */

      z_g&k._b1=e_g&k._b1_b3*y_g&k._b3;
      z_g&k._b2=e_g&k._b2_b3*y_g&k._b3;
      z_g&k._b3=e_g&k._b1_b3*y_g&k._b1+e_g&k._b2_b3*y_g&k._b2;

      %do q=1 %to &nbloc;
         z_g&k._b&q=standard(z_g&k._b&q);
         z_g&k._b&q=z_g&k._b&q*correzione;
      %end;

    /* estimation of the outer weights ["w"]*/

     /* reflective way */
      %do q=1 %to &nbloc;
```

```
         w_g&k._b&q=inv(z_g&k._b&q'*z_g&k._b&q)*(z_g&k._b&q'*XS_g&k._b&q);
         w_g&k._b&q=w_g&k._b&q';
      %end;

     /* formative way */
     /*%do q=1 %to &nbloc;
     /*  w_g&k._b&q=inv(XS_g&k._b&q'*XS_g&k._b&q)*(XS_g&k._b&q'*z_g&k._b&q);
     /*%end;*/

      converg=(ssq(y_old_g&k-z_g&k._b3))/(ssq(y_old_g&k));
      print converg;
   end;

/* ----the outer weights are non normed ---- */
%do q=1 %to &nbloc;
print w_g&k._b&q;
%end;

  /*****************************************************/
  /*  computation of the LVs using the outer weights w   */
  /*****************************************************/

  %do q=1 %to &nbloc;
     VL_g&k._b&q=XS_g&k._b&q*w_g&k._b&q;
     sqm_VL_g&k._b&q=(VL_g&k._b&q'*VL_g&k._b&q)/(N_g&k);
     sqm_VL_g&k._b&q=sqrt(sqm_VL_g&k._b&q);
     w_tilde_g&k._b&q=w_g&k._b&q/sqm_VL_g&k._b&q;
     print w_tilde_g&k._b&q;
     abs_w_tilde=abs(w_tilde_g&k._b&q);
     somma_w_tilde=abs_w_tilde[+,];
     w_tilde_normal_g&k._b&q=w_tilde_g&k._b&q/somma_w_tilde;
```

```
   print w_tilde_normal_g&k._b&q;
   VLS_g&k._b&q=XS_g&k._b&q*w_tilde_g&k._b&q;
   print VLS_g&k._b&q;
%end;

  /* ----- the LVs are standardized  ----- */

/*************************************************************/
/*          computation of the correlation between           */
/*             each LV and the corresponding MVs             */
/*************************************************************/

%do q=1 %to &nbloc;
   corr_VL_g&k._b&q=(XS_g&k._b&q`*VLS_g&k._b&q)/N_g&k;
   print corr_VL_g&k._b&q;
%end;

/*************************************************************/
/* computation of the Path coefficients and of the R2 value  */
/*                 of the endogenous LVs                     */
/*                                                           */
/* VL_exo_g&k_onXX is the vector containing the exogenous LVs*/
/*              linked to the endogenous LV XX               */
/*************************************************************/

/*************************************************************/
/* This step is to be MANUALLY completed by the user BEFORE  */
/*                     running the macro                     */
/*     (depending on which latent variables are linked       */
/*                      by a causal path)                    */
/*************************************************************/
```

```
  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  VL_exo_g&k._on3=(VLS_g&k._b1||VLS_g&k._b2);

/* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
/*  taking into account only the endogenous blocks  */

%do q=3 %to &nbloc;
      path_coef_g&k._b&q=inv(VL_exo_g&k._on&q'*VL_exo_g&k._on&q)
                          *(VL_exo_g&k._on&q'*VLS_g&k._b&q);
      print path_coef_g&k._b&q;

      VLS_g&k._b&q.att=VL_exo_g&k._on&q*path_coef_g&k._b&q;
      RES_g&k._b&q=VLS_g&k._b&q-VLS_g&k._b&q.att;

      R2_g&k._b&q=1-(RES_g&k._b&q'*RES_g&k._b&q)/
                  (VLS_g&k._b&q'*VLS_g&k._b&q);
      print R2_g&k._b&q;
  %end;

  /* ----- computation of the Communality and Redundancy indexes  ------*/
  %do q=1  %to &nbloc;
      COMM_g&k._b&q._vm=(corr_VL_g&k._b&q)##2;
      print COMM_g&k._b&q._vm;
  %end;
  %do q=1  %to &nbloc;
      COMM_g&k._b&q=sum((corr_VL_g&k._b&q)##2)/P_g&k._b&q;
      print COMM_g&k._b&q;
  %end;
```

```
/*************************************************************/
/*                     AVERAGE COMMUNALITY                   */
/*  This step is to be MANUALLY completed by the user BEFORE */
/*                      running the macro                    */
/*  (the average communality is obtained taking into account */
/*   all the communality indexes, i.e. one per block)        */
/*************************************************************/

 /* if new blocks have been added, add new latent variable */
 /*             in the sum at the numerator               */
 COMM_g&k._M=(P_g&k._b1*comm_g&k._b1+P_g&k._b2*comm_g&k._b2
             +P_g&k._b3*comm_g&k._b3)/P_g&k;
 print COMM_g&k._M;

 /*************************************************************/
 /*                          REDUNDANCY                       */
 /*  This step is to be MANUALLY completed by the user BEFORE */
 /*                      running the macro                    */
 /*      (number of endogenous latent variables in the model) */
 /*************************************************************/

 /*the redundancy indexes are calculated only for the endogenous LVs*/
 /*        IT MUST BE OBTAINED according to the path diagram        */
 %do q=3 %to &nbloc;
     RED_g&k._b&q=COMM_g&k._b&q*R2_g&k._b&q;
     print RED_g&k._b&q;

    RED_g&k._X&q=(CORR_VL_g&k._b&q)##2*R2_g&k._b&q;
    print RED_g&k._X&q;
 %end;
```

```
 /*-------- computation of the GOF index----------*/

  /*************************************************************/
  /*  This step is to be MANUALLY completed by the user BEFORE  */
  /*                    running the macro                      */
  /*    (number of endogenous latent variables in the model)    */
  /*************************************************************/

R2_g&k._M= (R2_g&k._b3);
print R2_g&k._M;

GOF_g&k= sqrt(COMM_g&k._M*R2_g&k._M);
print GOF_g&k;

/*------- creation of output SAS tables------------*/

%do q=1 %to &nbloc;
    varname1={"scores_b&q"};
    create &libname..scoresVLS_g&k._b&q from VLS_g&k._b&q
                [rowname= &id colname=varname1];
       append from VLS_g&k._b&q [rowname= &id];
    close &libname..scoresVLS_g&k._b&q;

    varname1_bis={"scores_orig_b&q"};
    create &libname..scoresVL_orig_g&k._b&q from VL_g&k._b&q
                [rowname= &id colname=varname1_bis];
       append from VL_g&k._b&q [rowname= &id];
    close &libname..scoresVL_orig_g&k._b&q;

    create &libname..CORR_VL_g&k._b&q from CORR_VL_g&k._b&q ;
```

```
      append from CORR_VL_g&k._b&q ;
   close &libname..CORR_VL_g&k._b&q;

   create &libname..w_tilde_g&k._b&q from w_tilde_g&k._b&q;
      append from w_tilde_g&k._b&q;
   close &libname..w_tilde_g&k._b&q;

   create &libname..w_tilde_normal_g&k._b&q from w_tilde_normal_g&k._b&q;
      append from w_tilde_normal_g&k._b&q;
   close &libname..w_tilde_normal_g&k._b&q;

   create &libname..COMM_g&k._b&q  from COMM_g&k._b&q;
      append from COMM_g&k._b&q;
   close &libname..COMM_g&k._b&q;

   create &libname..COMM_g&k._b&q._vm from COMM_g&k._b&q._vm;
      append from COMM_g&k._b&q._vm;
   close &libname..COMM_g&k._b&q._vm;

   create &libname..mean_VM_g&k._b&q from mean_VM_g&k._b&q;
      append from mean_VM_g&k._b&q;
   close &libname..mean_VM_g&k._b&q;

   create &libname..sqm_VM_g&k._b&q from sqm_VM_g&k._b&q;
      append from sqm_VM_g&k._b&q;
   close &libname..sqm_VM_g&k._b&q;
%end;

create correzione from correzione;
   append from correzione;
close correzione;
```

```
  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  %do q=3 %to &nbloc;
      create &libname..path_coef_g&k._b&q from path_coef_g&k._b&q;
         append from path_coef_g&k._b&q;
      close &libname..path_coef_g&k._b&q;

      create &libname..RED_g&k._b&q  from RED_g&k._b&q;
         append from RED_g&k._b&q;
      close &libname..RED_g&k._b&q;

      create &libname..R2_g&k._b&q from R2_g&k._b&q;
         append from R2_g&k._b&q;
      close &libname..R2_g&k._b&q;
  %end;

  %do q=1 %to &nbloc;
      create &libname..VM_oss_b&q._g&k from X_g&k._b&q [rowname= &id];
         append from X_g&k._b&q [rowname= &id];
      close &libname..VM_oss_b&q._g&k;
  %end;
%end;

  /*****************************************************************/
  /*           computation of the  Group Quality Index             */
  /*****************************************************************/
  %if &GQI='YES'%then %do;
    %do k=0 %to &nclas;
      %do q=1 %to &nbloc;
         proc sort data=&libname..VM_oss_b&q._g&k;
            by &id;
```

```
      run;

      proc sort data=&libname..scoresVLS_g&k._b&q;
         by &id;
      run;
  %end;
%end;
proc iml;
%do k=0 %to &nclas;
  %do q=1 %to &nbloc;
  /*************************************************************/
  /*   computation of the first term of Group Quality Index  */
  /*         Q inner residuals have to be computed,          */
  /*            one for each block in the model              */
  /*************************************************************/

    use &libname..VM_oss_b&q._g&k;
        read all into VM_oss_b&q._g&k [rowname=&id];
    use &libname..mean_VM_g&k._b&q;
        read all into mean_VM_g&k._b&q;
    use &libname..scoresVLS_g&k._b&q;
        read all into scoresVLS_g&k._b&q;
    use &libname..sqm_VM_g&k._b&q;
        read all into sqm_VM_g&k._b&q;
    use &libname..CORR_VL_g&k._b&q;
        read all into c_b&q._g&k;

    N_g&k=nrow(VM_oss_b&q._g&k);
    P_g&k._b&q=ncol(VM_oss_b&q._g&k);
    mean_matrix_VM_g&k._b&q=repeat(mean_VM_g&k._b&q,N_g&k,1);
    sqm_matrix_g&k._b&q=repeat(sqm_VM_g&k._b&q,N_g&k,1);
```

```
      XS_g&k._b&q=(VM_oss_b&q._g&k-mean_matrix_VM_g&k._b&q)#
                  (sqm_matrix_g&k._b&q##-1);
      E_VM_b&q._g&k=scoresVLS_g&k._b&q*c_b&q._g&k';

      ex_res_VM_b&q._g&k=XS_g&k._b&q-E_VM_b&q._g&k;
      dif_VM_b&q._VM_g&k._means=XS_g&k._b&q;
      dif_VM_b&q._VM_g&k._means=dif_VM_b&q._VM_g&k._means##2;
   %end;
   /* if the model is composed by more than 3 blocks, */
   /*              add new addends                    */
   P=P_g&k._b1+P_g&k._b2+P_g&k._b3;
%end;
%do k=1 %to &nclas;
   %do q=1 %to &nbloc;
      num_primo_termine_1step_b&q._g&k=ex_res_VM_b&q._g&k##2;
      num_primo_termine_2step_b&q._g&k=
                  num_primo_termine_1step_b&q._g&k[+,];
      den_primo_termine_1step_b&q._g&k=dif_VM_b&q._VM_g&k._means;
      den_primo_termine_2step_b&q._g&k=
                  den_primo_termine_1step_b&q._g&k[+,];
      primo_termine_b&q._g&k=num_primo_termine_2step_b&q._g&k/
                           den_primo_termine_2step_b&q._g&k;
   %end;
   primo_termine_g&k._1step=
    primo_termine_b1_g&k %do q=2 %to &nbloc;||primo_termine_b&q._g&k%end;;
   primo_termine_g&k._2step=primo_termine_g&k._1step[,+]/P;
   primo_termine_g&k._3step=(N_g&k/N_g0)#primo_termine_g&k._2step;
%end;
K=&nclas;
%if &nclas=0 %then %do; K=1; %end;
  primo_termine_1step=
```

```
 primo_termine_g1_3step%do k=2 %to&nclas;||primo_termine_g&k._3step%end;;
primo_termine_2step=primo_termine_1step[,+];
primo_termine=1-primo_termine_2step;
print primo_termine;

/*****************************************************************/
/*     computation of the second term of Group Quality Index     */
/*              J inner residuals have to be computed            */
/*            one for each endogenous block in the model         */
/*****************************************************************/

%do k=1 %to &nclas;
  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  VL_exo_g&k._on3=(scoresVLS_g&k._b1||scoresVLS_g&k._b2);

  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
 %do q=3 %to &nbloc;
    use &libname..path_coef_g&k._b&q;
       read all into path_coef_g&k._b&q;

    mean_scoresVLS_g&k._b&q=scoresVLS_g&k._b&q[+,]/N_g&k;
    mean_matrix_scoresVLS_g&k._b&q=
            repeat(mean_scoresVLS_g&k._b&q,N_g&k,1);
    VL_g&k._b&q.att=VL_exo_g&k._on3*path_coef_g&k._b&q;
    in_RES_VL_g&k._b&q=scoresVLS_g&k._b&q-VL_g&k._b&q.att;
    dif_VL_b&q._VL_g&k._means=
            scoresVLS_g&k._b&q-mean_matrix_scoresVLS_g&k._b&q;
    dif_VL_b&q._VL_g&k._means=dif_VL_b&q._VL_g&k._means##2;
 %end;
%end;
J=&nVLendo;
```

```
%do k=1 %to &nclas;
   /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
   /*   since it is computed only for endogenous LV   */
   %do j=3 %to &nbloc;
     num_secondo_termine_1step_b&j._g&k=in_RES_VL_g&k._b&j##2;
     num_secondo_termine_2step_b&j._g&k=
                       num_secondo_termine_1step_b&j._g&k[+,];
     den_secondo_termine_1step_b&j._g&k=dif_VL_b&j._VL_g&k._means;
     den_secondo_termine_2step_b&j._g&k=
                       den_secondo_termine_1step_b&j._g&k[+,];
     secondo_termine_b&j._g&k=num_secondo_termine_2step_b&j._g&k/
                            den_secondo_termine_2step_b&j._g&k;
   %end;
   /* ADD NEW TERMS BY MEANS OF || ONE FOR EACH ENDOGENOUS LV*/
   secondo_termine_g&k._1step=secondo_termine_b3_g&k;;
   secondo_termine_g&k._2step=secondo_termine_g&k._1step[,+]/J;
   secondo_termine_g&k._3step=(N_g&k/N_g0)#secondo_termine_g&k._2step;
%end;
secondo_termine_1step=
secondo_termine_g1_3step %do k=2 %to &nclas;
 || secondo_termine_g&k._3step
  %end;;
secondo_termine_2step=secondo_termine_1step[,+];
secondo_termine=1-secondo_termine_2step;
print secondo_termine;

GQI=sqrt(primo_termine*secondo_termine);
print GQI;
create &libname..results_GQI from GQI;
   append from GQI;
close &libname..results_GQI;
```

```
 quit;
%end;
quit;
%mend PLS_PM;



/*********************************************************************/
/*                          Macro res_dist                           */
/*                                                                   */
/*    this macro allows to compute the residuals and the distances   */
/*            between each unit and each local model                 */
/*                                                                   */
/* Macro parameters :                                                */
/*  LIBNAME = library where the results are stocked                  */
/*  ID = The unit's identifier. It must be Text type: usually id_1,  */
/*       id_2, id_3, and so on.                                      */
/*  NCLAS= number of classes for which the PLS-PM has been estimated*/
/*  NBLOC = the number of PLS_PM blocks (each block is formed        */
/*          by a LV and the corresponding MVs)                       */
/*  DIST = 'YES' if distance have to be computed,                    */
/*         'NO' if only residual have to be computed                 */
/*  N.B. Some steps of this macro depend on the specification of     */
/*       the inner model and of the outer model.                     */
/*       Hence,some steps are to be MANUALLY modified by the user    */
/*       according to the model specification BEFORE running         */
/*       the macro.                                                  */
/*********************************************************************/

%macro res_dist(libname=,id=,nclas=,nbloc=,dist=);
```

```
/* Computation of LVs scores for all the units */
/*      regardless the class memberships       */
proc iml;
%do k=0 %to &nclas;
  %do q=1 %to &nbloc;
    use &libname..w_tilde_g&k._b&q;
       read all into w_tilde_g&k._b&q;
    use &libname..VM_oss_b&q._g0;
       read all into X_g0_b&q [rowname=&id];
    use &libname..mean_VM_g&k._b&q;
       read all into mean_VM_g&k._b&q;
    use &libname..sqm_VM_g&k._b&q;
       read all into sqm_VM_g&k._b&q;

    N_g&k=nrow(X_g0_b&q);
    mean_matrix_g&k._b&q=repeat(mean_VM_g&k._b&q,N_g&k,1);
    sqm_matrix_g&k._b&q=repeat(sqm_VM_g&k._b&q,N_g&k,1);
    XS_all_g&k._b&q=(X_g0_b&q-mean_matrix_g&k._b&q)
                    #(sqm_matrix_g&k._b&q##-1);

    VL_all_g&k._b&q=XS_all_g&k._b&q*w_tilde_g&k._b&q;
 %end;
%end;

/* computation of the predicted values of the MV linked to all LVs */
%do k=0 %to &nclas;
  %do q=1 %to &nbloc;
    %do m=0 %to &nclas;
       use &libname..CORR_VL_g&k._b&q;
         read all into c_b&q._g&m;
    %end;
```

```
    E_VM_b&q._with_coef_of_g&k=VL_all_g&k._b&q*c_b&q._g&k‘;
  %end;
%end;

/* computation of the predicted values of the endog LVs  */
%do k=0 %to &nclas;
  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  VL_all_exo_g&k._on3=(VL_all_g&k._b1||VL_all_g&k._b2);

  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  %do q=3 %to &nbloc;
    use &libname..path_coef_g&k._b&q;
       read all into path_coef_g&k._b&q;

    VL_g&k._b&q.att=VL_all_exo_g&k._on&q*path_coef_g&k._b&q;
  %end;
%end;
/* ---------- creazione output sas -------- */
%do k=0 %to &nclas;
  %do q=1 %to &nbloc;
     create XS_all_g&k._b&q from XS_all_g&k._b&q [rowname=&id];
        append from XS_all_g&k._b&q [rowname=&id];
     close XS_all_g&k._b&q;

     create VL_all_g&k._b&q from VL_all_g&k._b&q [rowname=&id];
        append from VL_all_g&k._b&q [rowname=&id];
     close VL_all_g&k._b&q;

     create work.E_VM_b&q._with_coef_of_g&k from E_VM_b&q._with_coef_of_g&k
                    [rowname=&id];
        append from E_VM_b&q._with_coef_of_g&K [rowname=&id];
```

```
      close work.E_VM_b&q._with_coef_of_g&k;
   %end;
   /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  %do q=3 %to &nbloc;
     create VL_g&k._b&q.att from VL_g&k._b&q.att [rowname=&id];
        append from VL_g&k._b&q.att [rowname=&id];
     close VL_g&k._b&q.att;
  %end;
%end;

/*************************************************************/
/*    computation of the residuals and of the distances      */
/*************************************************************/

/* --------  residuals "outer" on VM --------- */
%do k= 0 %to &nclas;
  %do q=1 %to &nbloc;
     proc sort data=XS_all_g&k._b&q;
       by &id;
     run;

     proc sort data=work.E_VM_b&q._with_coef_of_g&k;
       by &id;
     run;

     proc iml;
     use work.E_VM_b&q._with_coef_of_g&k;
        read all into E_VM_b&q._with_coef_of_g&k [rowname=&id];

     use XS_all_g&k._b&q;
        read all into VM_oss_b&q._g&k [rowname=&id];
```

```
    ex_res_VM_b&q._from_g&k=VM_oss_b&q._g&k-E_VM_b&q._with_coef_of_g&k;

    varname3={%do c=1 %to 10;
               "ex_res&c._VM_b&q._from_g&k"
              %end;};

    create &libname..ex_res_VM_b&q._from_g&k from ex_res_VM_b&q._from_g&k
               [rowname=&id colname=varname3];
       append from ex_res_VM_b&q._from_g&k [rowname=&id];
    close &libname..ex_res_VM_b&q._from_g&k;
    quit;
  %end;
%end;

/* --------  residuals "inner" on VM --------- */

%do k=0 %to &nclas;
  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  %do q=3 %to &nbloc;
    proc sort data=VL_g&k._b&q.att;
      by &id;
    run;

    proc sort data=VL_all_g&k._b&q;
      by &id;
    run;

    proc iml;
      use VL_all_g&k._b&q;
        read all into VL_all_g&k._b&q [rowname=&id];
```

```
      use VL_g&k._b&q.att;
        read all into VL_g&k._b&q.att [rowname=&id];

      in_RES_g&k._b&q=VL_all_g&k._b&q-VL_g&k._b&q.att;

      varname4={%do c=1 %to 10;
                  "In_res&c._g&k._b&q"
                %end;};

      create &libname..in_Res_g&k._b&q from in_Res_g&k._b&q
               [rowname= &id colname=varname4];
          append from in_Res_g&k._b&q [rowname= &id];
      close &libname..in_Res_g&k._b&q;
    quit;
  %end;
%end;

/* creo la tavola RES con tutti i residui*/
%do k=0 %to &nclas;
  /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
  %do q=3 %to &nbloc;
     proc sort data=&libname..In_res_g&k._b&q;
       by &id;
     run;
%end;

data all_in_res_from_g&k;
   /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
   merge %do q=3 %to &nbloc;
           &libname..in_res_g&k._b&q
         %end;;
```

```
    by &id;
 run;

 %do q=1 %to &nbloc;
    proc sort data=&libname..Ex_res_VM_b&q._from_g&k;
      by &id;
    run;
 %end;

 data all_ex_res_from_g&k;
   merge %do q=1 %to &nbloc;
            &libname..Ex_res_VM_b&q._from_g&k
         %end;;
   by &id;
 run;

 data &libname..res_g&k;
   merge all_in_res_from_g&k
         all_ex_res_from_g&k;
   by &id;
 run;
%end;

%if &dist='YES'%then %do;
   /* --------  Distance --------- */
   %do k=1 %to &nclas;
     proc iml;
     %do q=1 %to &nbloc;
        use &libname..ex_res_VM_b&q._from_g&k;
           read all into ex_res_VM_b&q._from_g&k [rowname=&id];
     %end;
```

```
 /* IT MUST BE DEFINED ACCORDING TO THE PATH DIAGRAM */
 %do q=3 %to &nbloc;
    use &libname..in_Res_g&k._b&q ;
       read all into in_Res_g&k._b&q [rowname=&id];

    use &libname..R2_g&k._b&q;
       read all into R2_g&k._b&q;
%end;

use &libname..cl_&k;
  read all into cl_&k;

%do q=1 %to &nbloc;
   use &libname..Comm_g&k._b&q._vm;
     read all into Comm_g&k._b&q._vm;
%end;

/* ------ NEW distances --------*/
t_g&k=1;
  /* t=1 since the number of extracted components it is */
  /*     always equal to 1...only one LV for block!     */
N=nrow(ex_res_VM_b&q._from_g1);
print N;
secondo_termine_num_g&k=(in_Res_g&k._b3##2/R2_g&k._b3);
  /* the numerator computation have to be changed according*/
  /*    to the number of endogenous blocks in the model    */
secondo_termine_num_g&k=secondo_termine_num_g&k[,+];
  /* new vector containing the "super-residual" for each unit*/
secondo_termine_den_g&k=secondo_termine_num_g&k[+,]/(N-t_g&k-1);
secondo_termine_g&k=secondo_termine_num_g&k/secondo_termine_den_g&k;
```

```
    primo_termine_num_g&k=((ex_res_VM_b1_from_g&k##2*COMM_g&k._b1_vm##-1)
                    ||(ex_res_VM_b2_from_g&k##2*COMM_g&k._b2_vm##-1)
                    ||(ex_res_VM_b3_from_g&k##2*COMM_g&k._b3_vm##-1));
      /* the numerator computation have to be changed according to */
      /*            the number of blocks in the model              */
    primo_termine_num_g&k=primo_termine_num_g&k[,+];
      /* new vector containing the "super-residual" for each unit*/
    primo_termine_den_g&k=primo_termine_num_g&k[+,]/(N-t_g&k-1);
    primo_termine_g&k=primo_termine_num_g&k/primo_termine_den_g&k;
    D_from_g&k=sqrt(primo_termine_g&k#secondo_termine_g&k);
    print D_from_g&k;

    /*************************************************************/
    /*       Creation of the output SAS tables                   */
    /*************************************************************/
    varname5={"D_from_g&k"};
    /*defining a table for distances..*/
    create &libname..dis_from_g&k from D_from_g&k
         [rowname= &id colname=varname5];
       append from D_from_g&k [rowname= &id];
    close &libname..dis_from_g&k;
    quit;
   %end;
  %end;
%mend res_dist;

/*****************************************************************/
/*                        Macro alloc_units                      */
/*                                                               */
/*       Assignment of the units to the closest local model      */
/*                                                               */
```

```
/* Macro parameters :                                                  */
/*  LIBNAME = where the results are stocked                           */
/*  ID = The unit's identifier. It must be Text type: usually id_1    */
/*       id_2, id_3, and so on.                                       */
/*  NCLAS = number of class for which PLS-PM has been estimated       */
/*********************************************************************/

%macro alloc_units(libname=,id=,nclas=);

 %do k=1 %to &nclas;
    proc sort data=&libname..dis_from_g&k;
      by &id;
    run;
 %end;

 proc sort data=&libname..merge;
   by &id;
 run;

 data a;
   set &libname..merge (keep= &id cluster);
 run;

 %do k=1 %to &nclas;
     data a;
        merge a
           &libname..dis_from_g&k;
        by &id;
     run;
 %end;
```

```
data &libname..dis;
  set a;
run;

data &libname..dis;
  set &libname..dis;
    %do k=1 %to &nclas;
       if D_from_g&k<=D_from_g1
         %do h=2 %to &nclas; and D_from_g&k<=D_from_g&h %end;
       then cl_new=&k;
    %end;
run;

proc sort data=&libname..cl_0;
   by &id;
run;

data &libname..merge;
   merge &libname..dis  (keep = &id cluster cl_new)
         &libname..cl_0;
   by &id;
run;

data cluster_old;
   set &libname..merge (keep = &id cluster);
run;

data cluster_new;
   set &libname..merge (keep = &id cl_new);
run;
```

```
data cambio;
   set &libname..merge (keep = &id cl_new);
     rename cl_new=cambio_classe;
run;

data dif_classe;
   merge cluster_old
         cluster_new
         cambio;
   by &id;
run;

data &libname..cambio_classe;
  set dif_classe;
     if cluster^=cl_new
        then cambio_classe=1;
     else cambio_classe=0;
run;

/*-----------  output SAS tables -------------*/
/* defining the cl_&g matrix with the unit belong to the  */
/* 1st class (cl_1),to the second class (cl_2), and so on */
%do k=1 %to &nclas;
   data &libname..cl_&k;
      set &libname..merge (drop=cluster);
        where cl_new=&k;
   run;
   data &libname..cl_&k (rename=(cl_new=cluster));
      set &libname..cl_&k;
   run;
%end;
```

```
data &libname..merge ( drop=cluster rename=(cl_new=cluster));
   set &libname..merge;
run;
/*************************************************************/
/*   Tables cl_g contain all data for units belonging to     */
/*            group g (k=1,...,K)                            */
/*************************************************************/


/******* detecting unit changing class membership *******/
data &libname..Unita_camb;
   set &libname..cambio_classe (keep = &id cambio_classe);
     where cambio_classe=1;
run;

proc print data=&libname..Unita_camb;
  var &id;
run;

proc iml;
  %do k=1 %to &nclas;
     use &libname..cl_&k;
        read all into cl_&k;
     n_g&k=nrow(cl_&k);
     print n_g&k;
  %end;
  use &libname..cambio_classe;
     read all into cambio_classe;
  N_i_cambian_class=cambio_classe[+,3];
  print N_i_cambian_class;
  N=nrow(cambio_classe);
```

```
   tasso_cambio=N_i_cambian_class/N;
   print tasso_cambio;
   if tasso_cambio<0.005 then print "STOP";
     else if tasso_cambio>0.995 then print "STOP";
       else print "GO_ON";
 quit;
%mend alloc_units;


/*************************************************************/
/*                  Macro cluster_for_1_g                    */
/*                                                           */
/*  this macro allow to run a hierarchical cluster analysis on  */
/*   the residuals obtained from the global model running the   */
/*                       dis_res macro                       */
/*                                                           */
/* Macro parameters :                                        */
/*  LIBNAME = library where the results are stocked          */
/*                                                           */
/*************************************************************/
%macro cluster_for_1_g(libname=);

  proc cluster data=&libname..res_g0
    method=ward
     outtree=&libname..tree_res;
  run;

  proc tree data=&libname..tree_res;
  run;

%mend cluster_for_1_g;
```

```
/*****************************************************************/
/*                     Macro update_class                        */
/*                                                               */
/*     this macro allow to build one table for each new class as */
/*         obtained from the cluster analysis performed on the   */
/*                     from the local models                     */
/*                                                               */
/*  The "new" number of classes must be chosen by looking at the */
/*results of the clustering performed in the macro "cluster_for_1_g"*/
/*                                                               */
/* Macro parameters :                                            */
/*  LIBNAME = library where the results are stocked              */
/*  ID = The unit's identifier. It must be Text type: usually id_1, */
/*       id_2, and so on...                                      */
/*  NCLAS_old = number of classes for which the PLS-PM was       */
/*              estimated in the previous step                   */
/*  NCLAS_new = number of classes for which we will estimate     */
/*              the LOCAL models                                 */
/*                                                               */
/*****************************************************************/
%macro update_class(libname=,id=,nclas_old=,nclas_new=);

  %if &nclas_old=0 %then %do;
     proc fastclus data=&libname..res_g0
        maxclusters=&nclas_new out=&libname..clus_&nclas_new;
     run;
  %end;

  %if &nclas_old>0 %then %do;
     proc fastclus data= &libname..res_g0
```

```
        maxclusters=&nclas_new out=&libname..clus_&nclas_new ;
     run;
  %end;

  proc sort data=&libname..clus_&nclas_new  ;
     by &id;
  run;

  proc sort data=&libname..cl_0;
     by &id;
  run;

  data &libname..merge;
     merge &libname..clus_&nclas_new  (keep = &id cluster)
           &libname..cl_0;
     by &id;
  run;

  /* defining the cl_&g matrix with the unit belong to the    */
  /* 1st class (cl_1), to the second class (cl_2), and so on..*/
  %do k=1 %to &nclas_new;
     data &libname..cl_&k;
        set &libname..merge;
          where cluster=&k;
     run;
  %end;

  proc sort data=&libname..merge;
     by cluster;
  run;
%mend update_class;
```

These macros have to be run according to the REBUS-PLS algorithm steps, i.e. following the BENETTON example:

```
libname Benetton'D:....\Reale\Benetton\Benetton_SAS';
%PLS_PM(libname=Benetton,table=Benetton_for_SEM,id=id,nclas=0,nbloc=3,
        nVLendo=1,GQI='No');
%res_dist(libname=Benetton,id=id,nclas=0,nbloc=3,dist='NO');
%cluster_for_1_g(libname=Benetton);

/**************** in the case of two latent classes ******************/
%update_class(libname=Benetton,id=id,nclas_old=0,nclas_new=2);
    /* iterations to be repeated until convergence!: model with 2 classes*/
    %PLS_PM(libname=Benetton,id=id,nclas=2,nbloc=3,nVLendo=1,GQI='No');
    %res_dist(libname=Benetton,id=id,nclas=2,nbloc=3,dist='YES');
    %alloc_units(libname=Benetton,id=id,nclas=2);

/**************** in the case of three latent classes *****************/
%update_class(libname=Benetton,id=id,nclas_old=0,nclas_new=3);
    /*iterations to be repeated until convergence!: model with 3 classes*/
    %PLS_PM(libname=Benetton,id=id,nclas=3,nbloc=3,nVLendo=1,GQI='NO');
    %res_dist(libname=Benetton,id=id,nclas=3,nbloc=3,dist='YES');
    %alloc_units(libname=Benetton,id=id,nclas=3);


/**************** Once the convergence is assured *********************/
%PLS_PM(libname=Benetton,id=id,nclas=3,nbloc=3,nVLendo=1,GQI='YES');
```

Bibliography

Aitkin, M., Anderson, D. & Hinde, J. [1981], 'Statistical modeling of data on teaching styles', *Journal of the Royal Statistical Society* **A144**, 419–461.

Aitkin, M. & Rubin, D. [1985], 'Estimation and hypothesis testing in finite mixture distributions', *Journal of the Royal Statistical Society* **B47**, 67–75.

Al-Nasser, A. [2003], 'Customer satisfaction measurement models: Generalized maximum entropy approach', *Pakistan Journal of Statistics* **19**, 213–226.

Alwin, D. F. & Hauser, R. M. [1975], 'The decomposition of effects in Path', *American Sociological Review* **40**, 36–47.

Amato, S., Esposito Vinzi, V. & Tenenhaus, M. [2005], A global goodness-of-fit index for PLS structural equation modeling, Technical report, HEC School of Managment, France.

Anderson, J. & Gerbing, D. [1984], 'The effects of sampling error on convergence, improprer solution and goodness of fit indeces for maximum likelihood confermatory factor analysis', *Psycometrika* **49**, 155–173.

Balakrrishnan, P., Cooper, M., Jacob, V. & Lewis, P. [1995], 'A study of the classification capabilities of neural networks using unsupervised learning: a comparaison with k-means clustering', *Psychometrika* **59**, 509–524.

Baron, R. & Kenny, D. [1986], 'The moderator-mediator variable distinction in social psychological research: conceptual, strategic, and statistical consideration', *Journal of Personality and Social Psycohology* **51**, 1173–1182.

Bayes, T. [1763/1958], 'Studies in the history of probability and statistics: IX. Thomas Bayes's essay towards solving a problem in the doctrine of chances', *Biometrika* **45**(3/4), 293–315.

Bensmail, H., Celeux, G., Raftery, A. & Robert, C. [1997], 'Inference in model based clustering', *Statistics and Computing* **7**, 1–10.

Bentler, P. [1990], 'Comparative fit indexes in structural models', *Psychological Bulletin* **107**, 238–246.

Bentler, P. & Bonett, D. [1980], 'Significance tests and goodness of fit in the analysis of covariance structure', *Psychological Bulletin* **88**, 588–606.

Bezdek, J. [1974], 'Numerical taxonomy with fuzzy sets', *Journal of Mathematical Biology* **1**, 57–71.

Bezdek, J. [1981], *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York.

Bezdek, J., Coray, C., Gunderson, R. & Watson, J. [1981*a*], 'Detection and characterization of cluster substructure. I. linear structure: Fuzzy c-lines', *SIAM Journal on Applied Mathematics* **40**, 339–357.

Bezdek, J., Coray, C., Gunderson, R. & Watson, J. [1981*b*], 'Detection and characterization of cluster substructure. II. fuzzy c-varieties and convex', *SIAM Journal on Applied Mathematics* **40**, 358–372.

Bishop, Y., Fienberg, S. & Holland, P. [1975], *Discrete Multivariate Analysis: Theory and Practice*, The MIT Press.

Böhning, B., Dietz, E., Schaub, R., Schlattmann, P. & Lindsay, B. [1994], 'The distribution of the likelihood ratio for mixture of density from the one-parameter exponential family', *Annals of the Institute of Statistics and Mathematics* **46**, 373–388.

Bollen, K. A. [1989], *Structural equations with latent variables*, Wiley, New York.

Bottenberg, R. & Christal, R. [1968], 'Grouping criteria - a method which retains maximum predictive efficiency', *Journal of Experimental Education* **36**, 28–34.

Bozdogan, H. [1987], 'Model selection and Akaike's Information Criterion (AIC): the general theory and its analytical extension', *Psychometrika* **52**, 345–370.

Bozdogan, H. [1993], Mixture model cluster analysis using model selection criteria and a new informational measure of complexity, *in* H. Bozdogan, ed., 'Multivariate Statistical Modeling', Vol. 2, Kluwer Accademic Publishers, Drodrrecht, The Netherlands.

Breiman, L., Friedmabn, J., Olshen, R. & Stone, C. [1984], *Classification and Regression Trees*, Wadsworth, Monterey, CA, USA.

Brown, M. [1984], 'Asymtotic distribution free methods in analysis of covariance structures', *British Journal of Mathematical and Statistical Psychology* **37**, 62–83.

Browne, M. & Cudeck, R. [1993], Alternative ways of assessing model fit, *in* K. Bollen & J. Long, eds, 'Testing structural equation models', Sage, Newbury Park, CA, USA.

Buck, S. [1960], 'A method of estimation of missing values in multivariate data suitable for use whit an eletronic computer', *Journal of the Royal Statistical Society* **22**, 302–306.

Carrol, J. & Arabie, P. [1983], 'INDCLUS: an individual differences generalization of the ADCLUS model and the MAPCLUS algorithm', *Psychometrika* **48**, 157–169.

Celeux, G. & Soromenho, G. [1996], 'An entropy based criterion for assessing the number of cluster in a mixtur model', *Journal of Classification* **13**, 195–212.

Chin, W. [2003], A permutation procedure for multi-group comparaison of PLS models, *in* M. Vilares, M. Tenenhaus, P. Coelho, V. Esposito Vinzi & A. Morineau, eds, 'PLS and related methods - Proceedings of the International Symposium PLS'03', DECISIA, pp. 33–43.

Chin, W., Marcolin, B. & Newsted, P. [2003], 'A partial least squares latent variable modeling approach for measuring interaction effects: results from a monte carlo simulation study and an electronical-mail emotion/adoption study.', *Information Systems Research* **14**, 189–217.

Chin, W. W. [1998], The partial least squares approach for structural equation modeling, *in* G. A. Marcoulides, ed., 'Modern Methods for Business Research', Lawrence Erlbaum Associates, London, pp. 295–236.

Christal, R. [1968], 'Jan: a tecnique for analyzing group judgment', *The Journal of Experimental Education* **36**, 24–27.

Ciavolino, E., Al Nasser, A. & D'Ambra, A. [2006], 'The generalized maximum entropy estimation method for the structural equation models', Presented to The 30th Annual Conference of the German Classification Society (GFKL 2006) - Berlin.

Cowgill, M., Harvey, R. & Watson, L. [1999], 'A genetic algorithm approach to cluster analysis', *Computers and Mathematics with Applications* **39**, 99–108.

de Leeuw, J., Young, F. & Takane, Y. [1976], 'Additive structure in qualitative data: an alternating least squares method with optimal scaling features', *Psychometrika* **41**, 471–503.

De Sarbo, W. [1982], 'GENNCLUS: New models for general nonhierarchical clustering analysis', *Psychometrika* **47**, 449–476.

De Sarbo, W., Carroll, J. & Clark, L. [1984], 'Synthesized clustering: a method for amalgamating alternative clustering bases with differential weighting of variables', *Psychometrika* **49**, 57–78.

De Sarbo, W. & Mahajan, V. [1984], 'Constrained classification: the use of a priori information in cluster analysis', *Psychometrika* **49**, 57–78.

De Sarbo, W., Oliver, R. & Rangaswamy, A. [1989], 'A simulated annealing methodology for clusterwise linear regression', *Psychometrika* **54**, 707–736.

De Soete, G. & De Sarbo, W. [1991], 'A latent class probit model for analyzing pck any/N data', *Journal of Classification* **8**, 45–63.

Dempster, A., Laird, N. & Rubin, D. [1977], 'Maximum likelihood from incomplete data via the EM-algorithm', *Journal of the Royal Statistical Society* **B39**, 1–38.

Dennis, J. & Schnabel, R. [1983], *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Pretince Hall, Englewood Cliffs, New Jersey.

Diamantopoulos, A. & Winkelhofer, H. [2001], 'Index construction with formative indicators: an alternative to scale development', *Journal of Marketing Research* **38**, 269–277.

Djkstra, T. [1983], 'Some comments on maximum likelihood and partial least squares methods', *Journal of Econometrics* **22**, 67–90.

Dunn, J. [1974], 'A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters', *Journal of Cybernetics* **3**, 32–57.

Edgington, E. [1987], *Randomization test*, Marcel Dekker Inc.

Efron, B. [1982], *The jackknife, the bootstrap and other resampling plans.*, SIAM, USA: Philadelphia.

Efron, B. & Tibshirani, R. J. [1993], *An Introduction to the Bootstrap*, Chapman&Hall, New York.

Esposito Vinzi, V., Amato, S. & Trinchera, L. [2008], PLS path modeling: Recent developments and open issues for model assessment and improvement, *in* V. Esposito Vinzi, W. Chin, J. Henseler & H. Wang, eds, 'Handbook of Partial Least Squares - Concepts, Methods and Applications', Springer, Berlin, Heidelberg, New York.

Esposito Vinzi, V. & Lauro, C. [2003], PLS regression and classification, *in* 'Proceedings of the PLS'03 International Symposium', DECISIA, France, pp. 45–56.

Esposito Vinzi, V., Ringle, C., Squillacciotti, S. & Trinchera, L. [2007], Capturing and treating unobserved heterogeneity by response based segmentation in PLS path modeling . a comparison of alternative methods by computational experiments, Working paper, ESSEC Business School.

Esposito Vinzi, V., Trinchera, L., Squillacciotti, S. & Tenenhaus, M. [2008], 'REBUS-PLS: A response - based procedure for detecting unit segments in PLS path modeling', *Applied Stochastic Models in Business and Industry (ASMBI)* . (Accepted for publication - to appear in 2008).

Everit, B. [1992], *Cluster Analysis*, Edward Arnold.

Everit, B. & Hand, D. [1981], *Finite Mixture Distribution*, Chapman and Hall, London.

Fisher, W. [1958], 'On grouping for maximum homogeneity', *Journal of American Statistics Association* **53**, 789–798.

Forgy, E. [1965], 'Cluster analysis of multivariate data: Efficiency vs. interpretability of classification', *Biometrics* **21**, 768–769.

Fornell, C. & Bookstein, F. L. [1982], 'Two structural equation models: LISREL and PLS appliead to consumer exit-voice theory', *Journal of Marketing Research* **XIX**, 440–452.

Frank, R. & Green, P. [1968], 'Numerical taxonomy in marketing analysis: a review article', *Journal of Marketing Research* **5**, 83–98.

Geman, S. & Geman, D. [1984], 'Stocastic relaxation, Gibbs distribution, and the bayesian restration of images', *IEEE Transactions on Pattern Analysis and Machine Learning* **6**, 721–741.

Golan, A., Judge, G. & Miller, D. [1996], *Maximum Entropy Econometrics: Robust Estimation*, John Wiley & Sons, New York.

Gordon, A. [1999], *Classification*, 2nd edition edn, Chapman & Hall.

Green, P. [1977], 'A new approach to market segmentation', *Business Horizons* **20**, 61–73.

Green, P. & De Sarbo, W. [1979], 'Componential segmentation in the analysis of consumer trade-offs', *Journal of Marketing* **43**, 83–91.

Hahn, C., Johnson, M., Herrmann, A. & Huber, F. [2002], 'Capturing customer heterogeneity using a finite mixture PLS approach', *Schmalenbach Business Review* **54**, 243–269.

Haughton, D. & Oulabi, S. [1993], 'Direct marketing modeling with CART and CHAID', *Journal of Direct Marketing* **7**, 16–26.

Healy, M. & Westmacott, M. [1956], 'Missing values in experiments analyzed on automatic computers', *Applied Statistics* **5**, 203–206.

Henseler, J. & Fassott, G. [2007], A mew ad simple approach to multi-group analysis in PLS path modeling, *in* 'ESSEC-HEC Research Workshop Series on PLS Developments', ESSEC Business School, Cergy-Pontoise.

Hensler, J. & Fassott, G. [2008], Testing moderating effects in PLS path models: An illustration of available procedure, *in* V. Esposito Vinzi, W. Chin, J. Henseler & H. Wang, eds, 'Handbook of Partial Least Squares - Concepts, Methods and Applications', Springer, Berlin, Heidelberg, New York.

Hoyle, R. [1995], *Structural equation modeling: concepts, issues and applications*, SAGE Publications.

Hruschka, H. [1986], 'Market definition and segmentation using fuzzy clustering methods', *International Journal of Research in Marketing* **3**, 117–134.

Hu, L. & Bentler, P. [1995], *Structural Equation Modelling: concepts, issues, and application*, R.H. hoyle edn, C.A. SAGE, chapter Evaluating model fit, pp. 76–99.

Hwang, H., De Sarbo, W. & Takane, Y. [2007], 'Fuzzy clusterwise generalized structured component analysis', *Psychometrika* **72**, 181–198.

Hwang, H. & Takane, Y. [2004], 'Generalized structured component analysis', *Psychometrika* **69**, 81–99.

Jakobowicz, E. [2007], Contributions aux modèles d'équations structurelles à variables latentes, PhD thesis, CNAM, Paris, France.

Jaynes, E. [1957*a*], 'Information theory and statistical mechanics I', *Physics Review* **106**, 620–630.

Jaynes, E. [1957*b*], 'Information theory and statistical mechanics II', *Physics Review* **108**, 171–190.

Jedidi, K., Jagpal, H. S. & S. De Sarbo, W. [1997b], 'Finite-mixture structural equation models for response-based segmentation and unobserved heterogeneity', *Marketing Science* **16**(1), 39–59.

Jedidi, K., Jagpal, S. & De Sarbo, W. [1997a], 'STEMM: A general finite mixture structural equation model', *Journal of Classification* **14**, 23–50.

Jöreskog, K. [1970], 'A general method for analysis of covariance structure', *Biometrika* **57**, 239–251.

Jöreskog, K. [1971], 'Simultaneous factor analysis in several populations', *Psychometrika* **57**, 409–426.

Jöreskog, K. & Sörbom, D. [1979], *Advances in Factor Analysis and Structural Equation Models*, Abt Books.

Jöreskog, K. & Sörbom, D. [1996], *LISREL 8: Structural Equation Modeling with the SIMPLIS command Language*, Scientific Software International, Hove and London edn.

Jöreskog, K. & Wold, H. [1982], The ML and PLS techniques for modeling with latent variables: historical and comparative aspects, *in* K. Jöreskog & H. Wold,

eds, 'Systems Under Indirect Observation', Vol. Part I, North-Holland, Amsterdam, pp. 263–270.

Kamakura, W. [1988], 'A least squares procedure for benefit segmentation with conjoint experiments', *Journal of Marketing Research* **25**, 157–167.

Kaplan, D. [2000], *Structural Equation Modeling: Foundations and Extensions*, Sage Publications Inc., Thousands Oaks, California.

Kass, G. [1980], 'An exploratory tecnique for investigating large quantitaties of categorical data', *Applied Statistics* **29**, 119–127.

Kenny, D. & Judd, C. [1984], 'Estimating the nonlinear and interactive effects of latent variables', *Psychological Bulletin* **96**, 201–210.

Lachlan, G. M. [1987], 'On bootstrapping the likelihood ratio test statistics for the number of components in a normal mixture', *The Journal of the Royal Statistical Society* **C36**, 318–324.

Lebart, L., Morineau, A. & Piron, M. [1995], *Statistique exploratorie multidimensionelle*, Dunod, Paris.

Lee, S.-Y. [2007], *Structural Equation Modelling: A Bayesian Approach*, Wiley.

Liao, T. [2002], *Statistical Group Comparison*, Willey.

Liu, C. & Rubin, D. [1994], 'The ECME algorithm: a simple extension of EM and EMC with faster convergence', *Biometrika* **81**, 633–648.

Lohmöller, J. [1987], LVPLS program manual, version 1.8, Technical report, Zentralarchiv für Empirische Sozialforschung, Universität Zu Köln, Köln.

Lohmöller, J. [1989], *Latent variable path modeling with partial least squares*, Physica-Verlag, Heildelberg.

Louis, T. [1982], 'Finding the observed information matrix when using the EM algorithm', *Journal of the Royal Statistical Society* **44**, 226–233.

Lutz, J. [1977], 'The multivariate analogue of JAN', *Educational and Psycological Measurement* **37**, 37–45.

Lyttkens, E., Areskoug, B. & Wold, H. [1975], The convergence of NIPALS estimation procedures for six path models with one or two latent variables, Technical report, University of Göteborg.

MacLachlan, D. & Johansson, J. [1981], 'Market segmentation with multivariate aid', *Journal of Marketing* **45**, 74–84.

MacQueen, J. B. [1967], Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability', University of California Press, Berkeley, pp. 281–297.

Manton, K., Woodnury, M. & Tolley, H. [1994], *Statistical Applications using Fuzzy Sets*, John Wiley & Sons, USA, New York.

Marsh, H., Balla, J. & McDonald, R. [1988], 'Goodness of fit indexes in confirmatory factor analysis: the effect of sample size', *Psychological Bulletin* **103**, 391–411.

McDonald, R. [1996], 'Path analysis with composite variables', *Multivariate Behavioral Research* **31**, 239–270.

McDonald, R. & Marsh, H. [247–255], 'Choosing a multivariate model: noncentrality and goodness of fit', *Psychological Bulletin* **107**.

McHugh, R. [1956], 'Efficient estimation and local identification in latent class analysis', *Psychometrika* **21**, 331–347.

McHugh, R. [1958], 'Note on efficient estimation and local identification in latent class analysis', *Psychometrika* **23**, 273–274.

McKendrick, A. [1926], 'Applications of mathematics to medical problems', *Proceedings of the Edinburgh Mathematical Society* **44**, 98–130.

McLachlan, G. [1992], *Discriminant Analysis and Statistical Pettern Recognition*, John Wiley & Sons.

McLachlan, G. J. & Basford, K. E. [1988], *Mixture Models - Inferences and Aplli-cations to Clustering*, Marcel Dekker, Inc., New York and Basel.

McLachlan, G. J. & Krishnan, T. [1997], *The EM Algorithm and Extentions*, John Wiley & Sons, Inc.

McLachlan, G. J. & Peel, D. [2000], *Finite Mixture Models*, John Wiley & Sons, Inc., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto.

McLachlan, G. & Peel, D. [1996], An algorithm for unsupervised learning via normal mixture models, *in* D. Dowe, K. Korb & J. Oliver, eds, 'ISIS: Information, Statistics and Induction in Science', World Scientific, Singapore, pp. 354–363.

McLachlan, G. & Peel, D. [1999], Modelling nonlinearity by mixtures of factor analysers via extension of the EM algorithm, Technical report, Center for Statistics, University of Queensland.

Meilijson, I. [1989], 'A fast improvement to the EM algorithm on its own terms', *Journal of the Royal Statistical Society* **51**, 127–138.

Meng, X. & Rubin, D. [1993], 'Maximum likelihood estimation via the ECM algorithm: a general framework', *Biometrika* **80**, 267–278.

Meng, X. & van Dyk, D. [1995], The EM algorithm - an old folk song sung to a fast new tune, Technical report n.408, Departement of Statistics, University of Chicago, USA: Chicago.

Newcomb, S. [1886], 'A generalized theory of the combination of observation so as to obtain best result', *American Journal of Mathematics* **8**, 343–366.

Ogawa, K. [1987], 'An approach to simultaneous estimation and segmentation in conjoint analysis', *Marketing Science (1986-1998)* **6**, 66–81.

Pearson, K. [1894], 'Contributions to the mathematical theory of evolution', *Philosophical Trans.* **A 185**, 71–110.

Piccolo, D. [1998], *Statistica*, Il Mulino.

Punj, G. & Stewart, D. [1983], 'Cluster analysis in marketing research: review and suggestion for application', *Journal of Marketing Research* **20**, 134–148.

Quandt, R. E. & Ramsey, J. B. [1978], 'Estimating mixtures of normal distributions and switching regressions', *Journal of American Statistical Society* **73**, 730–738.

Richardson, S. & Green, P. [1997], 'On bayesian analysis of mixture with an unknown number of components (with discussion)', *Journal of the Royal Statistical Society , Series B* **59**, 731–792.

Ringle, C. & Schlittgen, R. [2007], A genetic algorithm segmentation approach for uncovering and separating groups of data in PLS path modeling, *in* 'PLS'07: 5th International Symposium on PLS and Related Methods', Oslo, Norway, pp. 75–78.

Ringle, C., Wende, S. & Will, A. [2005], Customer segmentation with FIMIX-PLS, *in* T. Aluja, J. Casanovas, V. Esposito Vinzi, A. Morineau & M. Tenenhaus, eds, 'Proceedings of PLS-05 International Symposium', SPAD Test&go, Paris, pp. 507–514.

Ringle, C., Wende, S. & Will, A. [2008], Finite mixture partial least squares analysis: Methodology and numerical examples, *in* V. Esposito Vinzi, W. Chin, J. Henseler & H. Wang, eds, 'Handbook of Partial Least Squares - Concepts, Methods and Applications', Springer, Berlin, Heidelberg, New York.

Rossiter., J. R. [2002], 'The C-OAR-SE procedure for scale development in marketing', *International Journal of Research in Marketing* **19**, 305–335.

Roubens, M. [1982], 'Fuzzy clustering algorithm and their cluster validity', *European Journal of Operationa Research* **10**, 294–301.

Ryan, T. & Joiner, B. [1976], Normal probability plots and test for normality, Technical report, Statistic Departement, The Pennsylvania State University, USA.

Sanchez, G. & Aluja, T. [2006], PATHMOX: a PLS-PM segmentation algorithm, *in* V. Esposito Vinzi, C. Lauro, A. Braverman, H. Kiers & M. G.Schmiek, eds, 'Proceedings of KNEMO 2006', number ISBN 88-89744-00-6, Tilapia, Anacapri, p. 69.

Sanchez, G. & Aluja, T. [2007], A simulation study of PATHMOX (PLS Path Modeling segmentation tree) sensitivity, *in* '5th International Symposium - Causality explored by indirect observation', Oslo, Norway.

Scales, L. [1985], *Introduction to Numerical Optimization*, Macmillan Publishers, London.

Schwarz, G. [1978], 'Estimating the dimensions of a model', *Annals of Statistics* **6**, 461–464.

Shannon, C. [1948], 'A mathematical teory of communications', *Bell System Technical Journal* **27**, 379–423.

Shepard, R. & Arabie, P. [1978], 'Additive clustering: rapresentation of similarities as combinations of discrete overlapping proprieties', *Psycological Review* **86**, 87–123.

Skilling, J. [1989], The axioms of maximum entropy, *in* J. Skilling, ed., 'Maximum Entropy and Bayesian Methods in Science and Engineering', Kluwer Academic, Dordrecht, pp. 173–187.

Sörbom, D. [1974], 'A general method for studying differences in factor means and factor structures between groups', *British Journal of Mathematical and Statistical Psychology* **27**, 229–239.

Späth, H. [1979], 'Clusterwise linear regression', *Computing* **22**, 367–373.

Späth, H. [1981], 'Clusterwise linear regression', *Computing* **26**, 275.

Späth, H. [1982], 'A fast algorithm for clusterwise linear regression', *Computing* **29**, 175–181.

Squillacciotti, S. [2005], Prediction oriented classification in PLS path modelling, *in* T. Aluja, J. Casanovas, V. Esposito Vinzi, A. Morineau & M. Tenenhaus, eds, 'Proceedings of PLS-05 International Symposium', SPAD Test&go, Paris, pp. 499–506.

Squillacciotti, S. [2008], Prediction oriented classification in PLS path modelling, *in* V. Esposito Vinzi, W. Chin, J. Henseler & H. Wang, eds, 'Handbook of Partial Least Squares - Concepts, Methods and Applications', Springer, Berlin, Heidelberg, New York.

Steenkamp, J.-B. E. & Wedel, M. [1993], 'Fuzzy clusterwise regression in benefit segmentation: Application and investigation into its validity', *Journal of Business Research* **26**, 237–249.

Steiger, J. & Lind, J. [1980], Statistically based tests for the number of common factors, *in* 'Paper presented at the Psycomentric Society Annual Meeeting', Iowa City, IA, USA.

S.Y.Lee & Song, X. [2002], 'Bayesian selection on the number of factors in a factor analysis model', *Behaviormetrika* **27**, 23–39.

Tenenhaus, M. [1998], *La Régression PLS: théorie et pratique*, Technip, Paris.

Tenenhaus, M., Amato, S. & Esposito Vinzi, V. [2004], A global goodness-of-fit index for PLS structural equation modelling, *in* 'Proceedings of the XLII SIS Scientific Meeting', Vol. Contributed Papers, CLEUP, Padova, pp. 739–742.

Tenenhaus, M. & Esposito Vinzi, V. [2005], 'PLS regression, PLS path modeling and generalized procrustean analysis: a combined approach for PLS regression, PLS path modeling and generalized multiblock analysis', *Journal of Chemometrics* **19**, 145–153.

Tenenhaus, M., Esposito Vinzi, V., Chatelin, Y. & Lauro, C. [2005], 'PLS path modeling', *Computational Statistics and Data Analysis* **48**, 159–205.

Tenenhaus, M., Mauger, E. & Guinot, C. [2008], Use of ULS-SEM and PLS-SEM to measure interaction effect in a regression model relating two blocks of

binary variables, *in* V. Esposito Vinzi, W. Chin, J. Henseler & H. Wang, eds, 'Handbook of Partial Least Squares - Concepts, Methods and Applications', Springer, Berlin, Heidelberg, New York.

Thurstone, L. L. [1931], *The theory of multiple factors*, Edwards Brothers, Ann Arbor, MI.

Titterington, D. [1990], 'Some recent research in the analysis of mixture distribution', *Statistics* **4**, 619–641.

Titterington, D., Smith, A. & Makov, U. [1985], *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, USA, New York.

Trasher, R. [1991], 'CART: A recent advance in tree-structured list segmentation methodology', *Journal of Direct Marketing* **5**, 35–47.

Trinchera, L. [2007], REBUS-PLS: Response-based units segmentation in PLS, *in* 'ESSEC-HEC Research Workshop Series on PLS Developments', ESSEC Business School, Cergy-Pontoise.

Trinchera, L. & Esposito Vinzi, V. [2006], Capturing unobserved heterogeneity in PLS path modeling, *in* 'In Proceedings of IFCS 2006 Conference', Ljubljana, Sloveny.

Trinchera, L., Romano, R. & Esposito Vinzi, V. [2007], Latent segments detection in PLS path modeling: a tool to capture unobserved heterogeneity in customers' preferences, *in* 'In proceedings of 3rd German French Austrian Conference on Quantitative Marketing', ESSEC Bussines School, Paris, France.

Trinchera, L., Squillacciotti, S. & Esposito Vinzi, V. [2006], PLS typological path modeling : a model-based approach to classification, *in* V. Esposito Vinzi, C. Lauro, A. Braverman, H. Kiers & M. G.Schmiek, eds, 'Proceedings of KNEMO 2006', number ISBN 88-89744-00-6, Tilapia, Anacapri, p. 87.

Trinchera, L., Squillacciotti, S., Esposito Vinzi, V. & Tenenhaus, M. [2007], PLS path modeling in presence of a group structure: REBUS-PLS, a new response-

based, *in* 'In proceedings of PLS'07: 5th International Symposium on PLS and Related Methods', Oslo, Norway.

Tucker, L. & Lewis, C. [1973], 'The reliability coefficient for maximum likelihood factor analysis', *Psychometrika* **38**, 1–10.

Tukey, J. W. [1964], Causation, regression and path analysis, *in* 'Statistics and Mathematics in Biology', Hafner Publishing Company, New York.

Vermunt, J. & Magidson, J. [2002], Latent class cluster analysis, *in* J. Hagenaars & A. M. Cutcheon, eds, 'Applied latent class models', Cambridge University Press, pp. 89–106.

Wedel, M. & De Sarbo, W. [1994], A review of recent development in latent class regression models, *in* R. P. Bagozzi, ed., 'Methods of Marketing Research', pp. 352–388.

Wedel, M. & Kamakura, W. A. [2000], *Market Segmentation - Conceptual and Methodological Foundations*, II edn, Kluwer Accademic Publishers, Boston Dordrecht London.

Wedel, M. & Kistemaker, C. [1989], 'Consumer benefit segmentation using clusterwise linear regression', *International Journal of Research in Marketing* **6**, 45–49.

Wedel, M. & Steenkamp, J. [1989], 'Fuzzy clusterwise regression approach to benefit segmentation', *International Journal of Research in Marketing* **6**, 45–49.

Wedel, M. & Steenkamp, J. [1991], 'A clusterwise regression method for simultaneous fuzzy market segmentation', *Journal of Marketing Research* **28**, 385–396.

Wildt, A. & Mc Cann, J. [1980], 'A regression model for marketing segmentation studies', *Journal of Marketing Research* **17**, 335–340.

Wilkie, W. & Cohen, J. [1977], An overview of marketing segmentation: Behavioural concepts and research approaches, Technical report, Marketing Science Institute Working Paper.

Wind, Y. [1978], 'Issues and advances in segmentation research', *Journal of Marketing Research* **15**, 317–337.

Wold, H. [1966], Estimation of principal component and related models by iterative least squares, *in* P. R. Krishnaiah, ed., 'Multivariate Analysis', Academic Press, New York, pp. 391–420.

Wold, H. [1975], Modelling in complex situations with soft infromation, *in* 'Third World Congress of Econometric Society', Toronto, Canada.

Wold, H. [1979], Model construction and evaluation when theoretical knowledge is scarce: An example of the use of partial least squares, Technical report, Cahier 79.06 du Département d' Économétrie, Faculté des sciences économiques et sociales, Université de Genève, Genève.

Wold, H. [1982], Soft modeling: the basic design and some extensions, *in* K. G. Jöreskog & H. Wold, eds, 'Systems under Indirect Observation', Vol. Part II, North-Holland, Amsterdam, pp. 1–54.

Wold, H. [1985], Partial Least Squares, *in* S. Kotz & N. L. Johnson, eds, 'Encyclopedia of Statistical Sciences', Vol. 6, Wiley, New York, pp. 581–591.

Yoo, B., Donthu, N. & Lee, S. [2000], 'An examination of selected marketing mix elements and brand equity', *Academy of Marketing Science Journal* **28**, 195–211.

Zadeh, L. [1965], 'Fuzzy sets', *Information and Control* **8**, 338–353.

Zhu, H. & Lee, S. [2001], 'A bayesian analysis of finite mixtures in the LISREL model', *Psychometrika* **66**, 133–152.