

University of Milano-Bicocca

Doctoral School in Statistics and Maths Applied to Finance

PhD in Statistics

XXVII Cycle



**SPECIFICATION AND ESTIMATION OF COMPLEX
STRUCTURAL MODELS WITH COVARIATE EFFECTS**

PIA CLARA PAFUNDI

December 10th, 2014

University of Milano-Bicocca

Doctoral School in Statistics and Maths Applied to Finance

PhD in Statistics

XXVII Cycle



**SPECIFICATION AND ESTIMATION OF COMPLEX
STRUCTURAL MODELS WITH COVARIATE EFFECTS**

Doctoral Dissertation

Pia Clara Pafundi

December 10th, 2014

Acknowledgements

I would first like to sincerely thank my tutor, Prof. Piergiorgio Lovaglio, for guiding me in my research path with wise counsel. In addition, I express my gratitude for conveying to me, at the same time, his professional and human experience.

Special thanks are also addressed to Prof. Giorgio Vittadini, who has always followed and supported me giving me valuable advice to conclude in the best way this difficult path.

Thank you also to the Coordinator of the PhD in Statistics, Prof. Fulvia Mecatti. A particular thank to the colleague and friend Gianmarco Vacca for his valuable support.

To my parents and my brother, whom I have always been able to rely on in times of joy and in the frequent periods of despondency of my PhD, a sincere thanks. I am in debt to them, who supported the most important personal and professional choices of my life and they have never failed to unconditional love, listening and attention, always spurring me to the best.

INDEX

Abstract

INTRODUCTION	9
Chapter 1 - Structural Equation Modelling	13
1.1 What are Structural Equation Models? Covariance Structured Analysis and Component Analysis approaches	14
1.2 The covariance based approach and Partial Least Squares	18
1.3 Covariance Structured Analysis limits	23
1.4 Partial Least Squares limits	24
1.5 Covariance Structured Analysis and Partial Least Squares limitations with external covariates	26
Chapter 2 – Component Analysis and Redundancy Analysis approaches	29
2.1 Redundancy Analysis, Multiblock Redundancy Analysis and external covariates	31
2.2 Extended Redundancy Analysis (ERA) and its limitations	35
2.3 An extension of Extended Redundancy Analysis: the Generalized Redundancy Analysis	44
2.4 Generalized Redundancy Analysis algorithm	46
2.5 Generalized Redundancy Analysis advantages over Extended Redundancy Analysis	50

Chapter 3 – The Simulation Study	53
3.1 The simulation study	54
3.1.1 Study 1: differences in the specification of complex extensions	55
3.1.2 Study 2: Generalized Redundancy Analysis performances in case of correlation between X and T	56
3.1.3 Study 3: Generalized Redundancy Analysis and Non-normal distributions	60
3.2 Results and discussion	63
3.2.1 Study 1 – Specification scenarios	63
3.2.2 Study 2 – Correlation scenarios	67
3.2.3 Study 3 – Distribution scenarios	76
Conclusion	85
References	90
Appendix	98
Appendix 1A – Study 1 (Scenario I)	(I-X)
Appendix 1B – Study 1 (Scenarios II/III)	(XI-XXI)
Appendix 2 – Study 2	(XXII-XXXII)
Appendix 3 – Study 3 (Scenario I)	(XXXIII-XLIV)
Appendix 3– Study 3 (Scenario II)	(XLV-LVIII)

ABSTRACT

Many statistical approaches have tried to specify and fit Structural Equation Models with external covariates. Those methods differ by measurement models (formative or reflective models, latent or composite variables, presence of observed exogenous and or endogenous variables having direct effects on latent/composite variables) and estimation strategies (parametric/algorithmic).

The present work is mainly focused on allowing for external covariates that may affect both endogenous indicators directly and/or indirectly through the composites.

We firstly introduced the basis of structural equation modelling, briefly describing LISREL, PLS and most novel and powerful techniques, among them Generalized Maximum Entropy (GME), and their limitations, explaining if they allow or not for exogenous covariates. We then discussed the introduction of external covariates in the Component Analysis framework, through methods such as Redundancy Analysis, Multiblock Redundancy Analysis (MbRA) and Extended Redundancy Analysis (ERA). We briefly described every single method, pointing out advantages and disadvantages and improvements which justify the use of a new estimation strategy, named Generalized Redundancy Analysis (GRA), on which we lead a complete simulation study with three different scenarios, aimed to evaluate GRA performances in the specification of complex extensions, covariance and distribution scenarios.

KEY WORDS

Structural Equation Modeling, Covariance Structure Analysis, Component Analysis, Redundancy Analysis, External Covariates

ITALIAN ABSTRACT

Molti approcci statistici hanno studiato la specificazione e il fitting di Modelli ad Equazioni Strutturali con covariate esterne. Tali metodi differiscono per modelli di misurazione (modelli formativi o riflessivi, variabili latenti o composite, presenza di variabili osservate esogene e/o endogene con effetti diretti sulle variabili latenti/composite) e strategie di stima (parametriche/algorithmiche).

Il presente lavoro si focalizza in particolare sulla possibilità di inserire covariate esterne che possono influenzare entrambi gli indicatori endogeni direttamente e/o indirettamente, attraverso le composite.

Sono stati introdotti i fondamenti dei Modelli ad Equazioni Strutturali, descrivendo brevemente LISREL, PLS e le metodologie più nuove e innovative, tra cui la Generalized Maximum Entropy (GME), nonché i loro limiti, spiegando se consentono o meno l'inserimento di covariate esterne. È stata quindi discussa l'introduzione di covariate esterne nell'ambito della Component Analysis, attraverso metodi quali la Redundancy Analysis, la Multiblock Redundancy Analysis (MbRA) e la Extended Redundancy Analysis (ERA). Ogni singola metodologia verrà illustrata, esplicitandone vantaggi e svantaggi e miglioramenti giustificanti l'introduzione di una nuova strategia di stima nota come Generalized Redundancy Analysis (GRA), sulla quale è stato condotto uno studio di simulazione completo, con tre diversi scenari, volto a valutare le performance del GRA nella specificazione di estensioni complesse, covariate e differenti scenari di distribuzione.

INTRODUCTION

In real applications, the underlying theory may specify either structural links between observed covariates and LVs or structural links between observed covariates and endogenous observed variables (Moustaki, 2003).

In practice, often the underlying theory may specify the presence of exogenous covariates which may enter the causal model, exogenous covariates that do not strictly belong to the formative blocks of unobservable composites, but may causally affect observed endogenous variables and composites too. Such covariates hereinafter will be called under the name of concomitant indicators and they will be the focus of the present work.

Many statistical approaches have tried to specify and fit Structural Equation Models with external covariates. Those methods differ by measurement models (formative or reflective models, latent or composite variables, presence of observed exogenous and or endogenous variables having direct effects on latent/composite variables) and estimation strategies (parametric/algorithmic).

Such external covariates may affect either observed endogenous variables or composites, without strictly belonging to the formative/reflective blocks of unobservable composites.

For an example, (Figure 1) shows two different path diagrams, Scenario I and Scenario II, respectively, with two exogenous blocks $\mathbf{X}_1 = \mathbf{x}_1|\mathbf{x}_2$, $\mathbf{X}_2 = \mathbf{x}_3|\mathbf{x}_4$ affecting the composites f_1 and f_2 , respectively, while t_1 represents the single concomitant indicator, which affects, in scenario I only an endogenous variable y_1 and, in scenario II, both an endogenous variable y_1 and a composite f_1 .

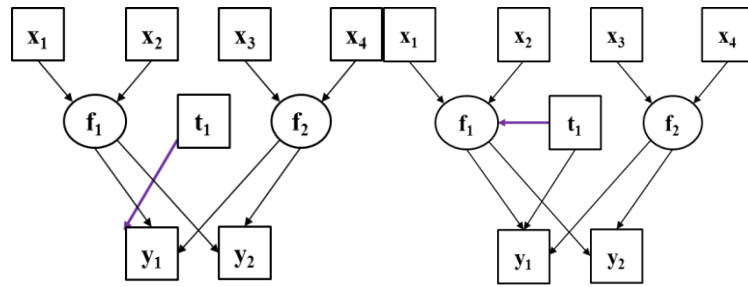


Figure 1. Two path diagrams with concomitant indicators, at increasing complexity. On the left we can observe Scenario I, with a single covariate t_1 affecting an endogenous variable y_1 and, on the right, Scenario II, where a new link has been added, thus the covariate affects both an endogenous variable y_1 and a LC (f_1).

Thus, a comprehensive system would also take into account such “concomitant indicators”.

The present thesis analyses recent methodological proposal for specification and fit of latent variables and latent composites at individual level with external covariates, discussing advantages and disadvantages of the proposed methods. In particular, we will mainly focus on allowing for external covariates that may affect both endogenous indicators directly and/or indirectly through the composites.

In Chapter 1 we will introduce what structural equation models are and briefly present LISREL and PLS and their limitations. Finally, we will explain if these methods allow or not exogenous covariates.

Chapter 2 discusses the implementation of exogenous covariates in the Component Analysis framework, through methodologies such as Redundancy Analysis, Multiblock Redundancy Analysis (MbRA) and Extended Redundancy Analysis (ERA). We will present a brief description of every single method, pointing out advantages and disadvantages and improvements which justify the introduction of a new estimation specification and fit strategy called Generalized Redundancy Analysis (GRA).

In Chapter 3 we will conduct a complete simulation study for GRA with three different scenarios, in order to study the differences in the specification of

complex extensions, covariance and distribution scenarios.

For each study we will describe the model, show and discuss the obtained results. At the end we will give conclusions and future perspective of research in this field.

Chapter 1

-

Structural Equation Modelling

1.1 What are Structural Equation Models? Covariance Structured Analysis and Component Analysis approaches

Structural Equation Models (SEM) represents one of the most common and specific group of methods used to represent hypotheses about the means, variances, and covariances of observed data in terms of a smaller number of 'structural' parameters defined by a hypothesized underlying conceptual or theoretical model. SEM can be defined as "a stochastic model in which every equation represents a causal link, rather than a pure empirical association" (Goldberger, 1972). We distinguish between observed variables and latent constructs (either measurable variables affected by measure errors which make them not observable, or theoretical constructs by themselves neither observable nor measurable (e.g.: socio economic status)).

Historically, SEM derived from the hybrid of two separate statistical traditions¹:

- a) Factor Analysis (FA), in order to develop a model which explains how observed variables are connected and explain the variance-covariance matrix through the simultaneous solution of equations which represent the model, so reducing the dimensionality of our data.
- b) Simultaneous equation modeling (the so called Path Analysis, Wright, 1934), mainly developed in econometrics.

All the hypotheses are examined through constrictions among the groups. Every constriction introduces a restriction in the value that the parameters can assume, thus it increases the parsimony of the model.

¹ So SEM let us investigate simultaneously model on several samples (multiple groups), which consists in a great advantage.

The measurement of the construct can happen only indirectly through observable variables affected by measurement errors (imperfect empirical indicators, measures consisting in observed scores and subjected to errors in the observations).

Historically, the models concerning the relationships among constructs and measures didn't focus on indirect effects (Blalock, 1964; Bollen and Lennox, 1991; Costner, 1969; MacCallum and Browne, 1993). We distinguish two different types of models: reflective (model of the principal factor) and formative (latent composite variable model), (Figure 1.1, Table 1.1).

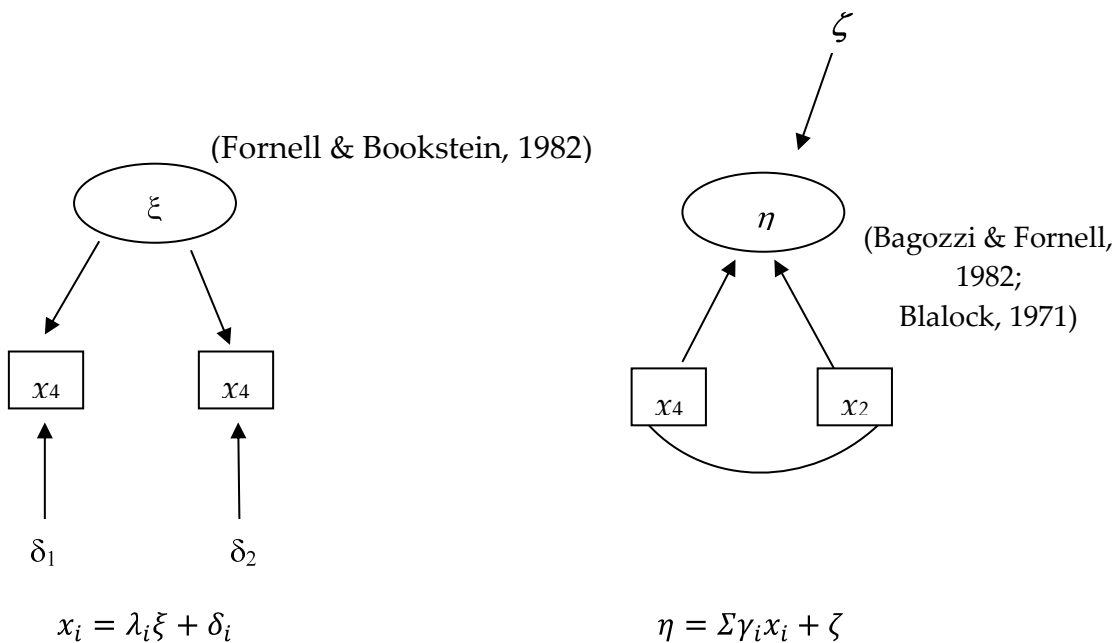


Figure 1.1. On the left the reflective model (Fornell and Bookstein, 1982), on the right the confirmative model (Bagozzi and Fornell, 1982; Blalock, H.M., Jr., 1971).

Latent Measurement Model with “reflective” measures	Aggregated Measurement Models with “formative” measures
The direction of causality is from the construct to the measure.	The direction of causality is from the measure to the construct.
We expect a correlation among the measures (internal reliability – coherence), interchangeable.	No reason to expect a correlation among the measures (the model doesn’t imply the internal coherence) – not interchangeable.
The elimination of an indicator from the measurement model doesn’t affect the significance of the construct.	The elimination of an indicator from the measurement model may affect the significance of the construct.
The measurement error is taken into consideration at a level of single item.	The measurement error is taken into consideration at the level of the construct.
The construct exists at an abstraction level higher than its measures (the construct has a surplus of significance).	The construct exists at the same level of its measures vs the construct has a surplus of significance.

Table 1.1. Reflective (on the left) and formative (on the right) measurement model features.

In the complete form, SEM include four types of variables: (i) latent exogenous variables, exclusively independent (ξ); (ii) latent endogenous variables, dependent (η); (iii) observed variables (item) which measure the exogenous latent variables (x) and (iv) observed variables (item) which measure the endogenous latent variables (y).

The most basic specification in Structural Equation Models is the distinction between exogenous and endogenous variables. To summarize, the presumed causes of exogenous variables are not represented in structural models, instead (a) the causes of exogenous variables are unmeasured (or unknown as far as the model is concerned) and (b) exogenous variables are considered free both to vary and to covary. The latter (covariance, unanalyzed association) is often represented in model diagrams by the symbol with a curved line with two arrowheads that connects every pair of exogenous variables. In contrast, the

presumed measured causes of endogenous variables are explicitly represented in the model, which also implies that endogenous variables are not free to vary or covary.

The common symbolism hereinafter used is presented in Table 1.2:

Variables		
Latent	Endogenous	η
	Exogenous	ξ
Observed	Endogenous	Y
	Exogenous	X
Stochastic errors	of the η	ζ
	of the Y	ε
	of the X	δ
Structural coefficients	η - Y	λ^Y
	ξ - X	λ^X
	η - η	β
	ξ - η	γ
Variances - covariances of exogenous and errors	Between the ξ	ϕ
	Between the errors ζ	ψ
	Between the errors ε	θ^ε
	Between the errors δ	θ^δ

Table 1.2. Common SEM symbolism.

Two different approaches have been proposed for SEM: Covariance Structure Analysis (CSA), where the first and most famous model is represented by the LISREL Model (Jöreskog, 1981) and Partial Least Squares (PLS) (Wold, 1982).

CSA and PLS deal with structural equation models from statistically different perspectives (Velicer and Jackson, 1990): the former entails factor analysis, whereas the latter involves component analysis (CA).

1.2 The covariance based approach and Partial Least squares

As a CSA approach we focused on LISREL approach, firstly introduced by Karl Jöreskog around the early '70. Initially he focused on the estimate of the structural coefficients of the AF with the ML method, but at the end this approach went beyond the first objective, becoming a general procedure for models based on systems of structural equations. LISREL deals simultaneously with: (a) the problem of the measure of LV (the connections among indicators and latent constructs) in order to give us measures of validity and reliability (the capacity to express well and stably those purely theoretical constructs); (b) the problem of causality, instruments and methods to test empirically the existence of nexus hypothetically formulated during the theoretical phase; (c) the need to establish causal connections among the latent variables.

So LISREL model is made up of a structural model which specifies the causal relationships between the latent exogenous and endogenous variables, and a measurement model, where the relationship between the LV and the associated indicators (connection typically only reflective) is specified.

LISREL starts from the variance-covariance matrix among the manifest variables Σ to end with the values of the parameters of a model of structural

equations which describe the causal links among the variables (a causal model implies one and just one variance and covariance matrix!).

LISREL model is made up of three phases, plus one if the model is falsified: (a) Formulation of the theoretical model (we translate a system of structural equations in theory); (b) Estimate of structural parameters (we let interact, iteratively, model and observed data in order to minimize the distances among produced and observed data); (c) Test of the model (we compare the theoretical model with the observed data, covariance matrix “observed” and “expected” by the parameters just estimated); (d) Model Modification (we restart the cycle operating some adjustments based both on theoretical evaluations and falsified model).

The complexity and the necessity of adopting certain assumptions in the CSA framework have encouraged alternative “soft modelling” approaches such as PLS and other methodologies in the framework of component analysis (CA, Meredith and Millsap, 1985; Schonemann and Steiger, 1976).

Partial Least Squares, firstly introduced by H. Wold (1975) under the name NIPALS (Nonlinear Iterative PArtial Least Squares), focuses on maximizing the variance of the dependent variables explained by the independent ones instead of reproducing the empirical covariance matrix. Like any SEM, a Partial Least Squares model (PLS) consists of a structural model, reflecting the relationships between the latent variables (LVs), and a measurement model, which shows how the LVs and their indicators are related, plus a third component, the weight relationships, used to estimate case-values for the LVs (Chin and Newsted, 1999). So, PLS refers to a set of iterative algorithms based on least squares that implement an extensive spectrum of both explanatory and

exploratory multivariate techniques, which range from regression to path modeling.

PLS belongs to the family of Alternating Least Squares (ALS) algorithms (Henseler and Sarstedt, 2013), which extend Principal Component and Canonical Correlation Analysis (PCA and CCA, respectively). It was designed by Herman Wold (1966, 1974, 1982, 1985a,b, 1989) in order to analyze high dimensional data in a low-structure environment and, during the following decades, it has been variously extended and modified.

The fundamental assumption of all PLS methods is that the observed data are generated by a system, a process driven by a small number of LVs. PLS-Path Models (PLS-PM) are formally defined by two sets of linear equations (Figure 1.2): an **inner model**, focused on the relationships between the LVs, and an **outer model** which, instead, is mainly interested in the relationships between a LV and its manifest indicators (MVs).

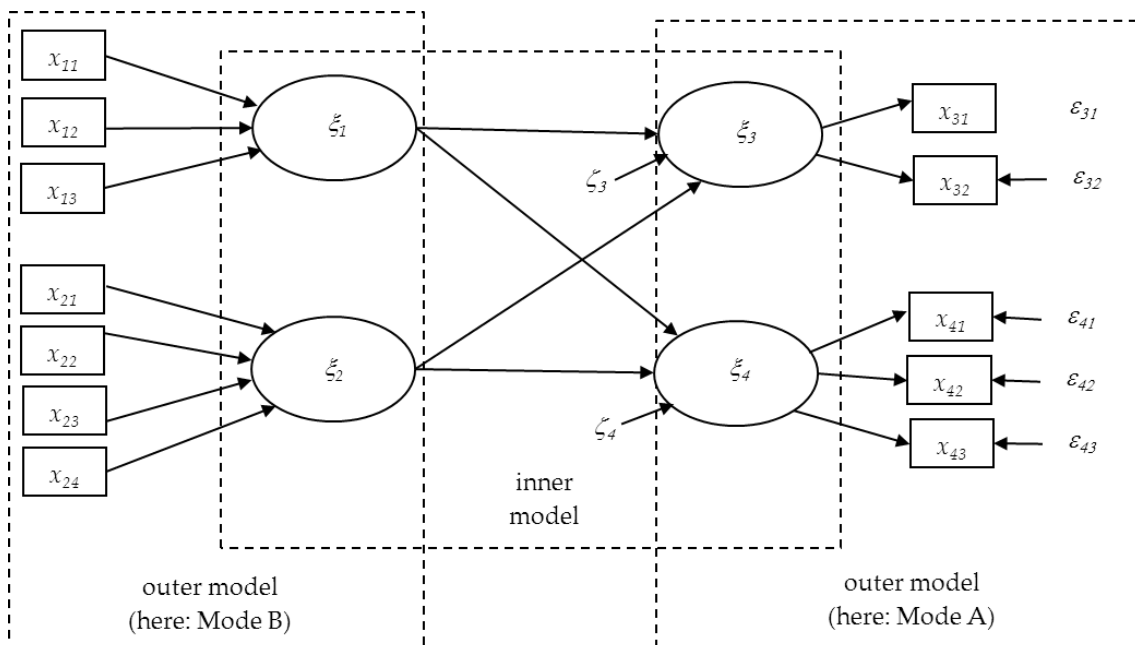


Figure 1.2. A simple PLS path model (Lohmöller, 1989).

Figure 1.2 depicts an example of a PLS-PM. Without a loss of generality, it can be assumed that LVs and MVs are centered so that the location parameters can be split in the two models. The inner model is given by equation (1.1)

$$\Xi = \Xi B + Z \quad (1.1)$$

where Ξ is the vector of latent variables ($\xi_1, \xi_2, \xi_3, \xi_4$), B denotes the matrix of path coefficients, and Z (ζ_3, ζ_4) represents the residuals (basic PLS design assumes a recursive inner model which is subject to predictor specification, so reducing equation (1.1) to following equation (1.2)

$$E(\Xi|\Xi) = \Xi B \quad (1.2)$$

The outer model instead can be either Mode A, which tends to optimize a covariance criterion (Tenenhaus and Tenenhaus, 2011) or Mode B, which instead tends to optimize a correlation criterion (Hanafi, 2007), and the choice between one and the other typically derives from a decision to define an outer model as either reflective or formative (Fornell and Bookstein 1982).

Basic PLS path modeling algorithm (Wold, 1982), which was later extended by Lohmöller (1989), includes the following three stages: (1) the iterative approximation of latent variable scores, (2) the estimation of outer weights, outer loadings, and path coefficients and (3) the estimation of location parameters. Only the first stage is iterative and comprehends four steps, repeated until the change in the outer weights between two iterations drops below a predefined limit.

PLS has been proven to be a very powerful versatile data analytical tool applied in many areas of research. Computational and implementation simplicity of PLS is a strong aspect of this approach. Among many advantages of such an approach we can find: a) the ability to analyze the importance of individual observed variables potentially leading to the deletion of unimportant variables with a significant cost reduction and without a high risk associated with the “blind” variables deletion; b) PLS “involves no assumptions about the population or scale of measurement” (Fornell and Bookstein, 1982) and consequently works without distributional assumptions and with nominal, ordinal, and interval scaled variables.

PLS, on its own, has the advantages that it has a very easy algorithm (fixed point algorithm), which helps in reaching systematic convergence and, as such, PLS does not need any restrictive distributional assumptions. Moreover, it can be used with a few individuals and lots of variables, LV estimates have a practical meaning and it lets Multiblock analysis and it doesn't suffer from improper solutions and indeterminate factor scores.

As we pointed out at the beginning, a useful extension of both the basic CSA models and the PLS models for better prediction of the endogenous latent variables (LVs) is the accommodation of external covariates in the structural equation of the model (Lee and Shi, 2000; Lee and Song, 2003; Moustaki, 2003). Despite a number of benefits in both CSA and PLS methodologies to fit Structural Equation Models, both presents unsolved problems.

1.3 Covariance Structured Analysis limits

Typically Component Structured Analysis (CSA) estimates model parameters by the Maximum Likelihood (ML) method, following the assumption of multivariate normality of the variables. However, such a distributional assumption, is often violated. Another, more serious, problem is linked to improper solutions (e.g., negative variance estimates), highly frequent in practice. Moreover, factor scores or latent variable scores are indeterminate. Hence, an Asymptotically Distribution-Free (ADF) estimator can be used to fit non-normal data. The ADF estimator, however, is accurate only with very large samples and it still does not guarantee the absence of improper solutions and factor score indeterminacy.

Finally, the main limitation of CSA SEM (LISREL approach) is represented by indeterminacy of latent scores, due to the exceeding of the rank of the covariance matrix for latent variables (endogenous, exogenous, errors in equations and measurement errors) on the rank-of-covariance matrix for the MVs (Schönemann and Steiger, 1976; Schönemann, 1971; Vittadini, 1989; Vittadini et al., 2007). Due to this, LVs and errors are not unique even with a precise identification of the model (Schönemann and Steiger, 1976; Vittadini 1989; Hwang and Takane, 2004).

Finally, there is no necessary and sufficient condition available for model identification, reason for which Jöreskog (1981) suggested that “the identification problem be studied on a case-by-case basis, examining the equations and choosing the restrictions, not only in number, but also in position”.

1.4 Partial Least Squares limits

As far as PLS is concerned, its main limitation is related to the fact that we cannot use this technique to empirically explore a system of highly structured relationships modelled by specified matrices (weights and/or loadings) restricting fixed elements (zeroes, unities, or equal constraints), based on the specified links between observed variables and composites (Lovaglio and Vittadini, 2014). Its main intention is to predict in situations of low theoretical information, not emphasizing confirmatory analysis (Mc Donald, 1996; Garthwife, 1994; Tenenhaus et al., 2005), hence, concomitant indicators are not allowed.

Second, PLS algorithm for the estimation of composite scores suffers from logic inconsistency, meaning that, in some situations, the weights that define the composite scores are performed with arbitrariness by means of regressions, which do not respect the role of the observed variables in the measurement models (Vittadini et al., 2007). For example, in presence of an unobservable composite with reflective indicators, composite scores (estimated as a linear combination of reflective indicators) are obtained by treating the observed variables as formative (causes), whereas they are effects of the composite.

Another problem is represented by the so called "consistency at large". Because the case-values for the LVs in PLS are aggregates of MVs that involve measurement error, they must be considered as inconsistent (Fornell and Cha, 1994). Therefore, "the path coefficients estimated through PLS converge on the parameters of the latent-variable model [only] as both the sample size and the number of indicators of each latent variable become infinite" (McDonald, 1996, p. 248). Hence, in all real-life situations, with finite number of cases in the sample and of indicators per LV, PLS tends to underestimate the correlations

between the LVs and overestimate the loadings (i.e., Dijkstra, 1983). Only when both the numbers tend to infinity, then the LV case-values approaches the true values and the problem disappears (Lohmöller, 1989).

In addition to this, PLS does not solve a global optimization problem for parameter estimation, which makes difficult the evaluation of an overall model fit. Furthermore, path modelling softwares are much more confidential than the covariance based ones such as LISREL; PLS algorithm is also more an heuristic one; it does not allow, differently from CSA-SEM, for imposing value or equality constraints on path coefficients and, last, it is not likely that the obtained PLS solutions are optimal in any well-defined sense.

That's why further researches have been made to improve it, overcoming these issues and allowing for the inclusion of external covariates in the model.

1.5 Covariance Structured Analysis and Partial Least Squares limitations with external covariates

In fact, hypothetically, both CSA and PLS could be employed to fit simple SEM-RA models, but some aspects related to the SEM-RA framework and to limitations of the methodologies reduce their use.

CSA allow the specification of observed (exogenous or endogenous) variables in the structural model: under specific constraints (unitary loading and zero measurement error), an observed variable is made to coincide with a fictive latent variable admitting it as single reflective indicator. Moreover, other specific approaches address covariate effect in the CSA framework (Muthen, 2002; Lee and Song, 2003; Moustaki 2003) or factor analysis (Sammel and Ryan, 1996). Also recent innovations in the context of nonlinear SEM, or Bayesian SEM, were developed allowing manifest covariates (see Lee et al., 2007 for a discussion).

We have previously described how in a latent variable model we have a structural model and a measurement model with direct effects. Thus external covariates are allowed to affect indirectly the indicators through the latent variables or directly.

Latent variable models with covariate effects have on their own the fact that they contain large numbers of parameters, thus implying high complexity, in particular when we want to estimate simultaneously the parameters. Hence, it would be time consuming apply CSA approach (Moustaki et al., 2004). An alternative would be to estimate the measurement model first, also computing the factor scores in order to be used as dependent variables for further analysis, and then the structural one.

In addition, the effects of the violation of the bivariate normality on the estimated parameters has not been understood yet.

Besides this, estimating lots of parameters can increase the possibility of both non-convergence and improper solutions (Boomsma and Hoogland, 2001). In this context particularly the occurrence of improper solutions is most likely to interfere with meaningful analysis (Kiers et al., 1996) because the complexity of the structural models often determines an ambiguous application of the principles of causation between latent variables. Alternatively, if a two stage approach is used, firstly estimating the measurement model and then the structural model (Croon and Bolck, 1997), thus implying the use of the factor scores as observed variables regressed on a set of covariates, this can yield high biases of the estimates. Hence, some corrections needs to be applied.

With regard to Partial Least Squares, this approach is not usable for the empirical exploration of a system of relationships underlying a specific model with structured matrices (weights and/or loadings) restricting fixed elements (zeros, unities, or equal constraints), based on the specified links between observed variables and composites (Lovaglio and Vittadini, 2014). Specifically, structural models with concomitant indicators cannot be specified and fit using PLS, mainly because:

1. PLS does not allow for structural relationships between observed variables and composites ($\mathbf{f} \rightarrow \mathbf{X}$ or $\mathbf{X} \rightarrow \mathbf{f}$), but just between composites ($\mathbf{f}_1 \rightarrow \mathbf{f}_2$, e.g.). In fact the scores of the observed variables change during the (inner) steps of the algorithm even with their specification as composites with single indicators;

2. PLS does not allow the presence of concomitant indicators, such as covariates (\mathbf{T}) not strictly belonging to the measurement model, which may affect the observed endogenous variables ($\mathbf{T} \rightarrow \mathbf{Y}$) and the composites ($\mathbf{T} \rightarrow \mathbf{f}$). In such a context the evaluation of both direct and indirect relationships, either strength or direction, between an exogenous and an endogenous variable can be meaningfully affected by the concomitant indicators and this goes under the name of “moderating analysis”(Henseler and Fassott, 2009). Though, in PLS path modelling the observed variables are excluded from the role of moderator, which is played only by the composites and only on the direct relationships between exogenous and endogenous composites.

Thus, since both CSA and PLS are not suitable for the specification and the fitting of SEM-RA models with concomitant indicators, their accommodation in the CA framework will be analyzed. In the next chapter we will discuss the Redundancy Analysis framework, particularly focusing on the Extended Redundancy Analysis and the proposed alternative, called Generalized Redundancy Analysis (GRA).

Chapter 2

-

Component Analysis and Redundancy Analysis approaches

Estimation techniques based on components aim to provide an estimate of the values of the Latent Variables (LVs), so that they are the most correlated among them (according to a path diagram structure) and the most representative of each corresponding block of manifest variables (MVs). The measurement model in this kind of methods can be either reflective or formative and the estimates of the values of the LVs play a major role in the estimation methods component-based. Thus, in such a sense, these techniques appear more as an explorative approach rather than a confirmative one and have to be considered as a generalization of the Principal Component Analysis (PCA) with several data tables causally linked. PCA, in fact, can be viewed as an approximation of a weighted component analysis “with a weighted matrix approximately equal to the inverse of the variance-covariance matrix of the residuals” (Meredith and Millsap, 1985). Thus, in the Component Analysis (CA) framework the weighting problem (McDonald, 1968, 1996) plays a crucial role.

We have previously shown PLS limits in accommodating external covariates. In order to overcome this issue, it has been tried to introduce them in the Redundancy Analysis (RA) framework (Van den Wollenberg, 1977). Thus, in this context, we will briefly present this kind of methodologies, the classical RA, Multiblock Redundancy Analysis (MbRA: Bougeard, Qannari, Rose, 2011) and then Extended Redundancy Analysis (ERA, Hwang and Takane, 2005) to finally focus on a recent proposal, namely Generalized Redundancy Analysis (Lovaglio and Vittadini, 2013).

2.1 Redundancy Analysis, Multiblock Redundancy Analysis and external covariates

RA is the multivariate analog of simple linear regression and it is based on similar principles as Principal Components Analysis (PCA), thus making similar assumptions about the data. It is appropriate when the expected relationship between dependent and independent variables is linear. RA is the simplest type of structural-equation model between two sets of observed variables, in which latent variables are intended as components. Its aim is to extract a series of linear components from a set of exogenous variables in such a way that they are mutually orthogonal, thus the first redundancy component explains the maximum proportion of the total sum of squares (SS) in criterion variables, the second one, orthogonal to the first, accounts for the maximum proportion of unexplained SS after the first component is extracted, and so on (Lambert et al., 1988). Hence, RA is a sort of special type of SEM where: (a) a formative relationship is always assumed between unobserved and observed (exogenous) variables and (b) endogenous variables are always manifest ones. Despite conditions (a) and (b), structural equation models in the RA framework (hereinafter SEM-RA) are quite comprehensive, enabling us to specify and fit various and complex structural models.

With the aim to analyze and model the relationships between two data tables $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p)$ and \mathbf{Y} , it has been displayed in literature that two popular methods, PLS and Redundancy Analysis (RA) are clearly connected. In fact, if RA is a sort of regression of \mathbf{Y} upon linear combinations of \mathbf{X} , Partial least square (PLS) regression (Wold, 1966) may be seen as an alternative method to RA in the case of ill-conditioned problems.

At the basis of both Redundancy Analysis and PLS regression there is the same

criterion to maximize, namely $cov^2(t, u)$ with $\mathbf{t} = \mathbf{X}\mathbf{w}$ and $\mathbf{u} = \mathbf{Y}\mathbf{v}$. What changes are the norm constraints imposed on the components to be determined (Burnham et al., 1996; Bougeard et al., 2008): $\|\mathbf{w}\|=\|\mathbf{v}\|=1$ for PLS regression and $\|\mathbf{t}\|=\|\mathbf{u}\|=1$ for Redundancy Analysis.

In order to extend RA to more than two sets of variables (Velu, 1991; Reinsel and Velu, 1998) an extension of RA to the multiblock setting, alternative to multiblock PLS, has been proposed in the literature. This technique, called Multiblock Redundancy Analysis (MbRA, Bougeard et al., 2008) explores and models the relationships between one block of dependent variables with several blocks of explanatory variables, maximizing the sum of the covariances between the latent constructs obtained from each explanatory block and the latent construct obtained from the dependent block (Pafundi and Vacca, 2014). Thus, a multiblock setting with $(K+1)$ datasets is considered, where:

- \mathbf{Y} is a dataset with Q variables;
- \mathbf{X}_k are K datasets from which \mathbf{Y} has to be predicted. Each \mathbf{X}_k has P_k variables;
- $[\mathbf{X}_1 | \dots | \mathbf{X}_K]$ is the merged dataset \mathbf{X} , with $P = \sum_k P_k$ explanatory variables.
- N is the number of individuals on which all the quantitative variables, column centered, are measured.

The main concept which MbRA is that there are latent variables, linear combinations of the variables derived from each dataset, which sum up each dataset (Figure 2.1).

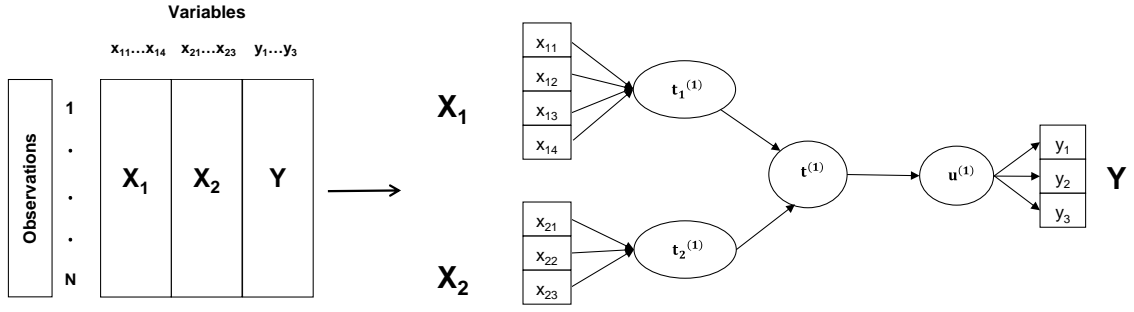


Figure 2.1. Conceptual scheme of the multiblock data structure. Here the relationships between the variable blocks ($\mathbf{X}_1, \dots, \mathbf{X}_K, \mathbf{Y}$) and the associated latent variables ($\mathbf{t}_1, \dots, \mathbf{t}_K, \mathbf{u}$) for the first dimension ($h = 1$) (Bougeard et al., 2011).

More precisely, through MbRA a global component $\mathbf{t}^{(1)}$ is derived, oriented toward the explanation of \mathbf{Y} and summing up partial components ($\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_K^{(1)}$) respectively associated with the blocks ($\mathbf{X}_1, \dots, \mathbf{X}_K$). The components are computed through the following maximization problem (equation (2.1)).

$$cov^2(\mathbf{u}^{(1)}, \mathbf{t}^{(1)}) \text{ with } \mathbf{t}^{(1)} = \sum_{k=1}^K \mathbf{a}_k^{(1)} \mathbf{t}_k^{(1)}, \sum_{k=1}^K \mathbf{a}_k^{(1)2} = 1, \quad (2.1)$$

$$\mathbf{u}^{(1)} = \mathbf{Y}\mathbf{v}^{(1)}, \mathbf{t}_k^{(1)} = \mathbf{X}_k\mathbf{w}_k^{(1)} \text{ and } \|\mathbf{t}_k^{(1)}\| = \|\mathbf{v}^{(1)}\| = 1$$

The partial components ($\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_K^{(1)}$) are given projecting $\mathbf{u}^{(1)}$ normalized onto each subspace spanned by variables in blocks $\mathbf{X}_1, \dots,$ and \mathbf{X}_K respectively, as detailed in equation (2.2).

$$\mathbf{t}_k^{(1)} = \frac{\mathbf{P}_{\mathbf{X}_k} \mathbf{u}^{(1)}}{\|\mathbf{P}_{\mathbf{X}_k} \mathbf{u}^{(1)}\|} \text{ for } k = (1, \dots, K) \quad (2.2)$$

with $\mathbf{P}_{\mathbf{X}_k} = \mathbf{X}_k(\mathbf{X}_k' \mathbf{X}_k)^{-1} \mathbf{X}_k'$ as the projector onto the subspace spanned by the variables from the X_k dataset. The coefficients $a_k^{(1)}$ are instead computed through equation (2.3).

$$a_k^{(1)} = \frac{\text{cov}(\mathbf{u}^{(1)}, \mathbf{t}_k^{(1)})}{\sqrt{\sum \text{cov}^2(\mathbf{u}^{(1)}, \mathbf{t}_j^{(1)})}} = \frac{\|P_{X_k} \mathbf{u}^{(1)}\|}{\sqrt{\sum_l \|P_{X_l} \mathbf{u}^{(1)}\|^2}} \text{ for } k = (1, \dots, K) \quad (2.3)$$

which implies the global component given by equation (2.4).

$$\mathbf{t}^{(1)} = \sum_k a_k^{(1)} \mathbf{t}_k^{(1)} = \frac{\sum_k P_{X_k} \mathbf{u}^{(1)}}{\sqrt{\sum_l \|P_{X_l} \mathbf{u}^{(1)}\|^2}} \quad (2.4)$$

Then, to obtain the other order solutions, we just have to regress the variables in the datasets (X_1, \dots, X_K) onto the first global component $\mathbf{t}^{(1)}$ and consider the residuals of these regressions. The same maximization is then performed by replacing the datasets by their associated residuals. Subsequent components can be found through reiteration of the process.

What makes MbRA slightly different from the other existing $(K+1)$ methods is represented by its focus on the global components rather than on the partial components. In fact, global components allow a much deeper analysis of the object of the study, providing valuable instruments both to investigate and predict the relationships among the datasets. Moreover, global components are each other orthogonal, which makes possible better interpret similarities and difference between the individuals, also thanks to graphs which highlight the relationships among all the variables. The main advantage of the proposed model when compared to the inner model of PLS - Path Modelling is that the regression coefficients are directly derived from the eigensolution. Thus, MbRA may be seen as a competitor technique to PLS-PM in case of variables organized in more than one dataset.

However, the multiblock approach presents some limitations. In particular MbRA is limited either to relationships among three sets of variables or to particular types of models.

Further developments in multiblock analysis should allow new innovation in the statistical processing of complex data. This issue have brought to the proposal of a further extension, which goes under the name Extended Redundancy Analysis (ERA, Hwang and Takane, 2002).

2.2 Extended Redundancy Analysis and its limitations

ERA employs a linear combination of the manifest variables to obtain the latent composites (LCs), subsequently fitted with the endogenous block. The estimates of the parameters are obtained minimizing a global LS criterion, overcoming the limits of CSA (improper solutions, indeterminacy of the factor scores, etc...) and the lack of global optimization of PLS-PM. ERA is multi-block, deals with highly structured parameter matrices whose structure, given prior to the analysis, may contain fixed (zero) elements. In addition, it allows analyzing more complex relationships among variables, unlike MbRA, also giving a global goodness of fit index. In ERA the relationships between the manifest exogenous and the manifest endogenous variables are moderated by the presence of linear composites (LCs), which are estimated as exact linear combinations of formative indicators, and both component weights and component loadings are estimated by consistently minimizing a single criterion. In ERA also a global least squares (LS) criterion to estimate model parameters is employed. Thus, it offers an overall model fit without recourse to the normality assumption.

After these simple illustrations, we can now look more deeply at a general SEM-RA model in presence of concomitant indicators. We can denote with $\mathbf{Y}_{n \times p}$ and $\mathbf{X}_{n \times q}$ the matrices of the observed endogenous variables and observed exogenous variables, respectively, with the columns mean centered and scaled to unit variance. If an observed variable is both exogenous and endogenous, it will be included in both the matrices.

Thus we can write a simple model (see Figure 2.2), at the basis of several more complex structural models, as:

$$\mathbf{Y} = \mathbf{X}\mathbf{W}\mathbf{A}' + \mathbf{E} = \mathbf{F}\mathbf{A}' + \mathbf{E} \quad \text{rank}(\mathbf{W}\mathbf{A}') = D \leq \min(q, p) \quad (2.5)$$

where $\mathbf{W}_{q \times D}$ is the matrix of composite weights, $\mathbf{A}'_{D \times p}$ the matrix of composite loadings, $\mathbf{E}_{n \times p}$ the matrix of residuals, $\mathbf{F}_{n \times D} (= \mathbf{X}\mathbf{W})$ the matrix of composite (or LVs scores). As we want the model to be identified it must hold that $\text{diag}(\mathbf{F}'\mathbf{F}) = \mathbf{I}_D$.

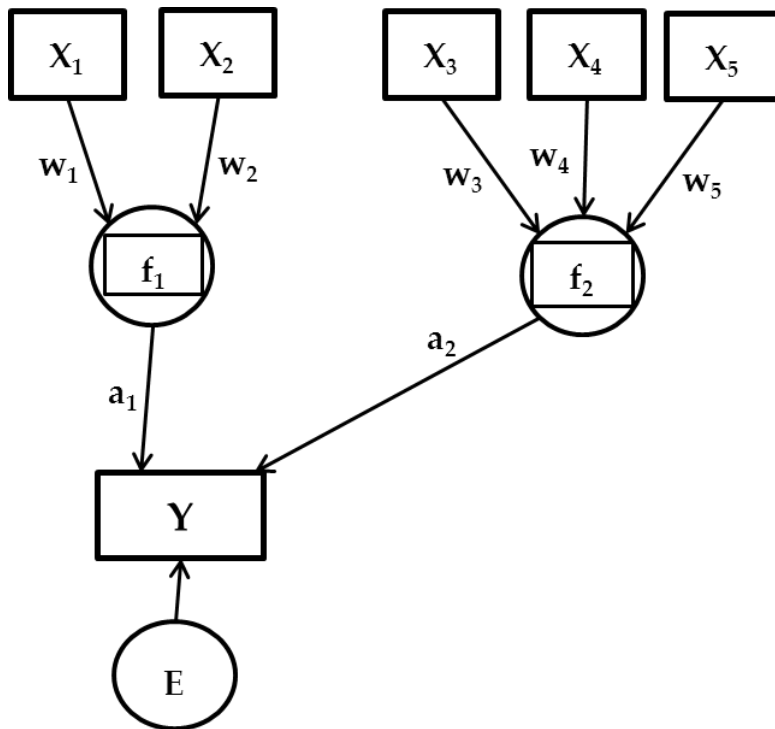


Figure 2.2. A simple structural equation model without concomitant indicators. Squares inside circles indicate vectors of composites, circles the errors, arrows the relationships (weights and loadings as vectors) and rectangles the observed variables (collected in matrices). The arrows which start from a block or arrive at a block of observed variables involve all the observed variables in the block.

ERA allows the specification of more complex models in which a block of \mathbf{X} can behave both as endogenous and exogenous.

Now, if \mathbf{X} is composed of two blocks ($\mathbf{X} = [\mathbf{X}_1 | \mathbf{X}_2]$) of observed variables that generate $\mathbf{F} = [\mathbf{f}_1 | \mathbf{f}_2]$, with one exogenous block (e.g., \mathbf{X}_2) having an impact on \mathbf{f}_1 (Figure 2.3), we can rewrite model (2.5) according to the rule that when an observed variable block is both exogenous and endogenous it is included either in \mathbf{Y} and \mathbf{X} (Hwang and Takane, 2005), as:

$$\tilde{\mathbf{Y}} = [\mathbf{X}_1 | \mathbf{X}_2] \begin{bmatrix} \mathbf{w}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_2 \end{bmatrix} \begin{bmatrix} \mathbf{a}'_1 & \mathbf{a}'_2 \\ \mathbf{0}' & \mathbf{a}'_3 \end{bmatrix} + [\mathbf{e}_2 | \mathbf{e}_1] = \mathbf{XWA}' + \mathbf{E} \quad (2.6)$$

with $\tilde{\mathbf{Y}} = [\mathbf{X}_2|\mathbf{Y}]$ new matrix of dependent variables, \mathbf{w}_1 and \mathbf{w}_2 composite weights for \mathbf{X}_1 and \mathbf{X}_2 ; \mathbf{a}'_1 , \mathbf{a}'_2 and \mathbf{a}'_3 composite loadings for \mathbf{f}_1 on \mathbf{X}_2 and for \mathbf{f}_1 and \mathbf{f}_2 on \mathbf{Y} and $\mathbf{F}_{n \times 2(=D)} = [\mathbf{f}_1|\mathbf{f}_2]$ a matrix of unit variance composite scores.

Notice that if no variables are shared by both \mathbf{Y} and \mathbf{X} , and no constraints other than $\text{rank}(\mathbf{W}\mathbf{A}')$ are imposed on \mathbf{W} and \mathbf{A}' , such a model reduces to the classic RA.

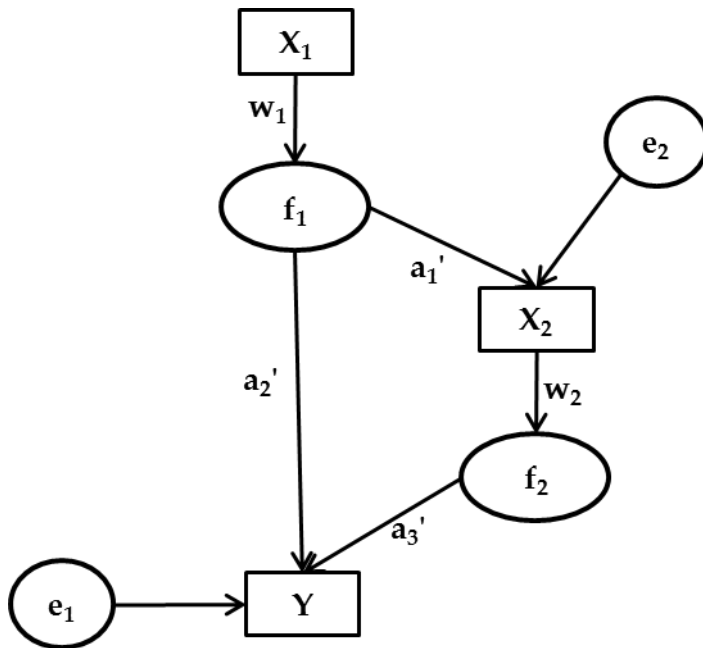


Figure 2.3. A path diagram for ERA model where \mathbf{X} is composed of two blocks ($\mathbf{X} = [\mathbf{X}_1|\mathbf{X}_2]$) of observed variables that generate $\mathbf{F} = [\mathbf{f}_1|\mathbf{f}_2]$, with one exogenous block (e.g., \mathbf{X}_2) having an impact on \mathbf{f}_1 .

The major innovation introduced by ERA is the inclusion of direct effects of observed variables on endogenous variables (Lovaglio and Vittadini, 2013), either taking part in the formation of the latent composite or acting as external covariates. Thus, based on the simple model depicted in figure 2.2, we can distinguish two different scenarios: (Scenario I) there are K concomitant indicators (embedded in matrix \mathbf{T}) in the model with a causal impact (thanks to $\mathbf{A}'_{\mathbf{Y}K \times p}$) on observed endogenous variables and (Scenario II) with a causal impact also onto the latent composites (see Figure 2.4 for both scenarios).

In particular, the D composites (\mathbf{F}) can be expressed as a linear combination of q strictly formative (\mathbf{X}) and K concomitant indicators (\mathbf{T}) as:

$$\mathbf{F} = \mathbf{X}\mathbf{W} + \mathbf{T}\mathbf{W}_T, \quad \text{diag}(\mathbf{F}'\mathbf{F}) = \mathbf{I}_D \quad (2.7)$$

with \mathbf{W}_T a $K \times D$ matrix of weights for \mathbf{T} and $D \leq \min(K + q, p)$. Thus the associated ERA model is equivalent to that of Scenario I.

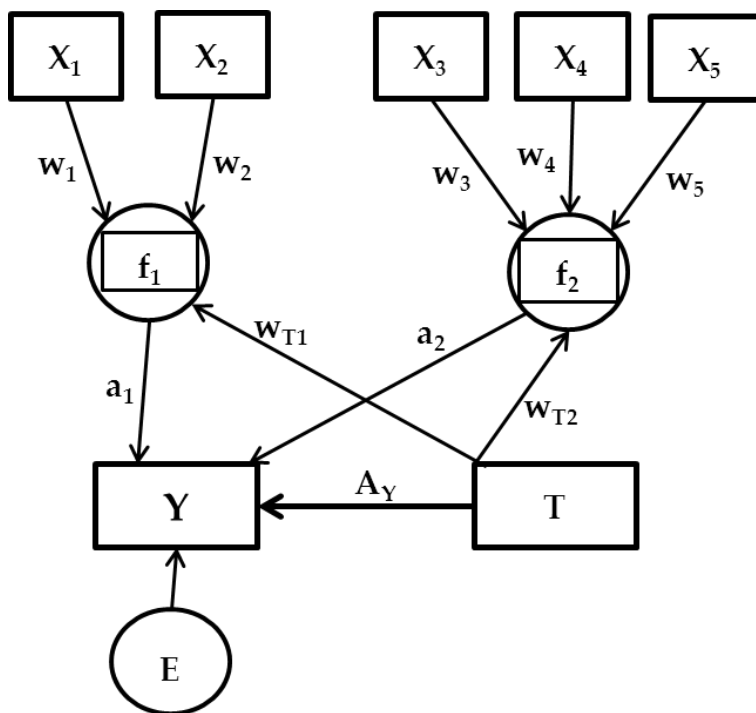


Figure 2.4. Structural Equation Model with concomitant indicators in Scenario II is shown (for Scenario I, $\mathbf{W}_T = [\mathbf{w}_{T1}, \mathbf{w}_{T2}] = \mathbf{0}_{K \times D}$). Squares inside circles indicate vectors of composites, circles the errors, arrows the relationships (weights and loadings as vectors) and rectangles the observed variables (collected in matrices). The arrows which start from a block or arrive at a block of observed variables involve all the observed variables in the block (e.g., \mathbf{w}_2 and \mathbf{w}_{T2} are vectors of weights linearly combining the columns of \mathbf{X}_2 and \mathbf{T} , respectively, to define \mathbf{f}_2) [Lovaglio and Vittadini, 2014].

In Scenario I, ERA model specification of direct effects can be written as:

$$\mathbf{Y} = \mathbf{T}\mathbf{A}'_Y + \mathbf{F}\mathbf{A}' + \mathbf{E}$$

$$\text{rank}(\mathbf{W}\mathbf{A}') = D \leq \min(q, p) \quad \text{diag}(\mathbf{F}'\mathbf{F}) = \mathbf{I}_D \quad (2.8)$$

Moreover, ERA also let us specify in a unique model both the structural equation and the measurement model (depending on the scenario) thanks to the introduction of new matrices of regressors, weights and loadings:

$$\mathbf{Y} = \mathbf{X}^\dagger \mathbf{W}^\dagger \mathbf{A}^\dagger + \mathbf{E} \quad (2.9)$$

with

$$\begin{aligned} \mathbf{X}_{n \times (K+q)}^\dagger &= [\mathbf{T}_{n \times K} | \mathbf{X}_{n \times K}] & \mathbf{A}_{p \times (K+D)}^\dagger &= [\mathbf{A}_{Y \ p \times K} | \mathbf{A}_{p \times D}] \\ \mathbf{W}_{(K+q) \times (K+D)}^\dagger &= \text{diag}[\mathbf{P}_{K \times K} | \mathbf{W}_{q \times D}] \end{aligned}$$

\mathbf{P} is a matrix of fixed elements (zero or unities), and the unity in the k -th column and a non-zero coefficient in the j -th row ($j = 1, \dots, p$) of \mathbf{A}_Y selects a direct link between the k -th concomitant indicator and the j -th endogenous indicator (Lovaglio and Vittadini, 2014).

For Scenario II (Figure 2.4) the ERA model is the same as Scenario I. What changes are the block matrices, which become: $\mathbf{X}^\dagger = [\mathbf{T} | \mathbf{X} | \mathbf{T}]$, $\mathbf{A}^\dagger = [\mathbf{A}_Y | \mathbf{A} | \mathbf{A}]$ and $\mathbf{W}^\dagger = \text{diag}[\mathbf{P} | \mathbf{W} | \mathbf{W}_T]$.

The estimation algorithm, unlike RA, instead of the classical minimization procedure, exploits an iterative method, a simple revision (in case \mathbf{Y} and \mathbf{X} consist of distinct variables in a single population) of the classic alternating least squares (ALS) algorithm (Kiers and ten Berge, 1989). This version of ALS algorithm is still monotonically convergent (De Leeuw et al., 1976) and consists

of two main steps²: (a) **W** is updated keeping **A'** fixed (initialized with arbitrary values) and (b) **A'** is updated for fixed **W**.

In order to assess the goodness of fit of an ERA model, as CSA indices (e.g., \mathbf{X}^2 , GFI, AIC, etc.) cannot be used due to the specific distributional assumptions required, a specific index has been introduced, the ERA fit index:

$$\text{Fit} = 1 - \frac{\text{SS}(\mathbf{Y} - \mathbf{XWA}')}{\text{SS}(\mathbf{Y})} \quad (2.10)$$

which uses the total variance of the endogenous variables minus its unexplained variance. ERA fit index ranges from 0 to 1 and the larger it is, the more variance of the endogenous variables is explained by the exogenous variables.

Finally, other two measures are really important in ERA model:

- (i) Standard Errors (SE) of parameter estimates in ERA, which can be computed through resampling methods, in particular with bootstrap (Efron, 1982; Efron and Tibshirani, 1998), and are used to evaluate the reliability of the parameter estimates;
- (ii) Critical Ratios (CR) like (e.g., parameter estimates/SE), useful to test the significance of the parameter estimates (e.g., $\text{CR} > |2|$ is considered significant at 0.05 level).

Comparing ERA estimation procedure with CSA-SEM and PLS we will observe that CSA typically bases on maximum likelihood estimation (MLE). Through MLE parameters are estimated so that the probability of obtaining the observed

² These steps are alternated until convergence, so, until the decrease in the function value falls below a certain threshold value (10^{-4}).

data (or the sample covariance matrix derived from the data) is maximum, given the hypothesized model and under certain distributional assumptions. In contrast, ERA estimation method aims to minimize “the sum of squares of discrepancies between the observed endogenous variables and their predicted counterparts from the exogenous variables without any explicit distributional assumptions” (Hwang, 2000).

In this sense ERA seems similar to PLS, but it has on its own the advantage of overcoming the lack of global optimization of PLS.

However, though ERA allows for concomitant indicators, even in this case we can find out some methodological drawbacks, with particular regard to the estimation of latent scores, which discourage from its use. These handicaps are mainly imputable to the way of specification of the direct effects. In fact, if we extend ERA formal specification in this case without any alteration either in the formulation of the model or in the estimation algorithm we may obtain whether inefficient solutions or misinterpretation of the coefficients. Hence, we can distinguish mainly three limitation at this methodology.

First of all, the approach ERA utilizes in managing direct effects. ERA estimates loadings and weights by a sequential algorithm (\mathbf{W}^+ for fixed \mathbf{A}^+ and \mathbf{A}^+ for fixed \mathbf{W}^+). In each first step, \mathbf{W}^+ is essentially estimated by reconstruction of their unknown blocks: first using an estimate of \mathbf{W} and then replenishing the known elements of \mathbf{P} . But step 1 is the same as fitting an ERA model between \mathbf{Y} and \mathbf{X} , hence the presence of concomitant indicators in the estimation procedure of the LC scores is ignored (particularly the direct effects on \mathbf{Y}). In addition to this, as in the second step \mathbf{A}^+ is updated for fixed \mathbf{W}^+ , the effects of the model are estimated in two stages (the measurement model is fitted first and then the LC scores are used as independent variables with concomitant indicators to

estimate \mathbf{A}^\dagger). The same limitation happens in Scenario II, where the direct effect of \mathbf{T} on \mathbf{Y} (\mathbf{A}_Y) are ignored.

The other limitations deal with the model specification of ERA, which collects in a single exogenous block concomitant indicators and formative variables for LCs (e.g. $\mathbf{X}^\dagger \mathbf{W}^\dagger$ of Eq. 2.9).

Firstly, in Scenario II the LC scores are estimated as linear combinations of both formative and concomitant indicators. However, the (typical) correlation between \mathbf{T} (concomitant indicators) and \mathbf{X} (strictly formative indicators) and their different effects on the endogenous variables (indirect effect of \mathbf{X} on \mathbf{Y} , mediated by LC, and both direct and indirect of \mathbf{T} on \mathbf{Y} , mediated by \mathbf{F}), the separate contributions of the two blocks in determining the LC scores cannot be distinguished (Pafundi and Vacca, 2014).

As last limitation, a problem of collinearity between \mathbf{T} and $\mathbf{f}_i (= \mathbf{X}_i \mathbf{w}_i + \mathbf{T} \mathbf{w}_{T1})$ in the estimation of the direct effects (\mathbf{a}_i and \mathbf{A}_Y), confirmed by the authors of ERA themselves (Takane and Hwang, 2005). This may verify because of the specification inside the same block of formative variables \mathbf{X}_i (e.g. in $\mathbf{X}^\dagger = [\mathbf{T} | \mathbf{X}_i]$) of (i) exogenous/concomitant variables \mathbf{T} directly affecting both a composite \mathbf{f}_i and (ii) an endogenous variable \mathbf{y} .

Hence, although in principle ERA allows for concomitant indicators, a further extension of ERA methodology is needed, which takes into account all specified effects in a correct way, allows the separation of the different contributions of \mathbf{X} and \mathbf{T} indicators on the composite scores and limits the collinearity problem among the exogenous variables.

2.3 An extension of Extended Redundancy Analysis: the Generalized Redundancy Analysis

Though ERA allows for concomitant indicators, as previously said, we can find out some methodological limitations, in particular regarding the estimation of latent scores, which discourage from its use. These disadvantages are mainly imputable to the way in which direct effects are specified. Thus, a more consistent approach would be to take into account all specified effects in such a way that all the separate contributions of strictly formative and concomitant indicators on the composite scores can be examined and collinearity among exogenous variables is avoided. This further extension in the SEM-RA framework goes under the name of Generalized Redundancy Analysis (GRA: Lovaglio and Vittadini, 2013, 2014; Lovaglio and Boselli, 2013).

For the more general Scenario II (impact of the concomitant indicators on the latent composites), the block of concomitant indicators \mathbf{T} is directly explicated in the model and all the unknown weight and loading matrices (\mathbf{W}_T and \mathbf{A}'_Y) are simultaneously estimated through an iterative method. Thus, Generalized Redundancy Analysis (GRA) neutralizes the effect of the correlations between the exogenous matrix \mathbf{X} and the concomitant indicators matrix \mathbf{T} through an orthogonalization and adds two new steps to the ERA algorithm:

- (i) the separation of the contribution of \mathbf{W} from that of \mathbf{W}_T ³;
- (ii) a final step to account for the direct effect of \mathbf{A}'_Y separately from the other steps.

³ \mathbf{TA}'_Y block is isolated and estimated separately with the purpose of better focusing on the direct effect of \mathbf{A}'_Y .

Also, correlations between matrices \mathbf{X} and \mathbf{T} , and the influences of \mathbf{W} and \mathbf{W}_T matrices on the LCs are accounted for by matrix manipulation (uncorrelation will be attributed by manipulating \mathbf{W} , \mathbf{W}_T , \mathbf{X} and \mathbf{T}).

GRA let us also fit diverse complex relationships among variables, such as highly structured matrices.

In order to separate the contribution of \mathbf{W} from \mathbf{W}_T the classic measurement model⁴ is equivalently specified with the orthogonal regressors' matrices \mathbf{T}° and \mathbf{X} as:

$$\mathbf{F} = \mathbf{X}\mathbf{W}^\circ + \mathbf{T}^\circ\mathbf{W}_T, \quad \text{diag}(\mathbf{F}\mathbf{F}') = \mathbf{I}_D, \quad (2.11)$$

where $\mathbf{T}^\circ = \mathbf{T} - \mathbf{X}\mathbf{X}^+\mathbf{T}$ and $\mathbf{W}^\circ = \mathbf{W} + \mathbf{X}^+\mathbf{T}\mathbf{W}_T$ and \mathbf{X}^+ is the Moore-Penrose generalized inverse of \mathbf{X} ⁵. Thus, putting eq. (2.11) into (2.8) we will obtain:

$$\mathbf{Y} = \mathbf{T}\mathbf{A}'_Y + (\mathbf{X}\mathbf{W}^\circ + \mathbf{T}^\circ\mathbf{W}_T)\mathbf{A}' + \mathbf{E} \quad (2.12)$$

$$= \mathbf{T}\mathbf{A}'_Y + \mathbf{X}\mathbf{W}^\circ\mathbf{A}' + \mathbf{T}^\circ\mathbf{W}_T\mathbf{A}' + \mathbf{E} \quad (2.13)$$

Given this we anyway have to deal with the lack of a closed form solution for the minimization of the loss function even in this case, as \mathbf{W}° , \mathbf{W}_T and \mathbf{A}' are fixed parameters. That's why an ALS algorithm has been used, in four iterative steps, involving alternately the four matrices \mathbf{A}'_Y , \mathbf{W}° , \mathbf{W}_T and \mathbf{A}' ⁶.

More in detail, in the first step \mathbf{W}° is updated independently from \mathbf{W}_T , keeping fixed \mathbf{A}' and \mathbf{A}'_Y . In the second step \mathbf{W}_T is updated, fixed \mathbf{A}' and \mathbf{A}'_Y , with \mathbf{W}° from the first step. In the third step, \mathbf{A} is updated keeping fixed \mathbf{A}' and \mathbf{A}'_Y , \mathbf{W}° and \mathbf{W}_T and, finally, in the last step \mathbf{A}'_Y is updated keeping fixed \mathbf{A} , \mathbf{W}° and \mathbf{W}_T .

⁴ $\mathbf{F} = \mathbf{X}\mathbf{W} + \mathbf{T}\mathbf{W}_T$

⁵ Singular Value Decomposition (SVD).

⁶ Note that every other matrix no involved in the considered step must be kept constant.

2.4 Generalized Redundancy Analysis algorithm

In the first step \mathbf{A}'_Y will be initialized with arbitrary values and a matrix $\mathbf{Y}^* = \mathbf{Y} - \mathbf{T}\mathbf{A}'_Y$ will be defined. Thus, the loss function to be minimized associated to equations (2.12-2.13) will be:

$$f = \text{SSQ}[\mathbf{Y}^*\mathbf{X}\mathbf{W}^\circ\mathbf{A}' - \mathbf{T}^\circ\mathbf{W}_T\mathbf{A}'] \quad (2.14)$$

The separate estimation of regression parameters is possible thanks to orthogonal regressors. In particular, if we assume $\mathbf{X} = \mathbf{Q}\mathbf{R}'$ (QR decomposition of \mathbf{X}), with $\mathbf{Q}_{n \times q}$ orthonormal matrix and $\mathbf{R}'_{q \times n}$ upper-triangular matrix, we will obtain the following loss function to be minimized:

$$\begin{aligned} f &= \text{SSQ}[\mathbf{Y}^* - \mathbf{Q}\mathbf{Q}'\mathbf{Y}^* + \mathbf{Q}\mathbf{Q}'\mathbf{Y}^* - \mathbf{Q}\mathbf{R}'\mathbf{W}^\circ\mathbf{A}' - \mathbf{Q}\mathbf{Q}'\mathbf{T}^\circ\mathbf{W}_T\mathbf{A}'] \\ &= \text{SSQ}[(\mathbf{Y}^* - \mathbf{Q}\mathbf{Q}'\mathbf{Y}^*) + \mathbf{Q}\mathbf{Q}'\mathbf{Y}^* - \mathbf{Q}(\mathbf{R}'\mathbf{W}^\circ\mathbf{A}' - \mathbf{Q}'\mathbf{T}^\circ\mathbf{W}_T\mathbf{A}')] \end{aligned} \quad (2.15)$$

Equation (2.15) can be rewritten as:

$$f = \text{SSQ}[(\mathbf{Y}^* - \mathbf{Q}\mathbf{Q}'\mathbf{Y}^*)] + \text{SSQ}[\mathbf{Q}(\mathbf{Q}'\mathbf{Y}^* - \mathbf{R}'\mathbf{W}^\circ\mathbf{A}')] \quad (2.16)$$

In fact, (i) the orthogonality of \mathbf{X} and \mathbf{T}° implies that the term $\mathbf{Q}'\mathbf{T}^\circ\mathbf{W}_T\mathbf{A}'$ is null and

$$(ii) \text{tr} \left[(\mathbf{Y}^* - \mathbf{Q}\mathbf{Q}'\mathbf{Y}^*)' \mathbf{Q}(\mathbf{Q}'\mathbf{Y}^* - \mathbf{R}'\mathbf{W}^\circ\mathbf{A}') \right] = 0.$$

The first term of eq. (2.16) is independent from unknown weights and loadings, thus the minimization of eq. (2.15) is equivalent to the minimization of:

$$\begin{aligned} f &= \text{SSQ}[\mathbf{Q}(\mathbf{Q}'\mathbf{Y}^* - \mathbf{R}'\mathbf{W}^\circ\mathbf{A}')] = \text{SSQ}[\mathbf{Q}'\mathbf{Y}^* - \mathbf{R}'\mathbf{W}^\circ\mathbf{A}'] = \\ &= \text{SSQ}[\text{vec}(\mathbf{Q}'\mathbf{Y}^*) - (\mathbf{A} \otimes \mathbf{R}')\text{vec}(\mathbf{W}^\circ)] = \text{SSQ}[\text{vec}(\mathbf{Q}'\mathbf{Y}^*) - \mathbf{w}^\circ] \end{aligned} \quad (2.17)$$

with $\mathbf{\Omega} = (\mathbf{A} \otimes \mathbf{R}')$, \otimes the Kronecker product and $\mathbf{w}^\circ = \text{vec}(\mathbf{W}^\circ)$ the supervector made up of all the columns of \mathbf{W}° one below the other. As we want to estimate \mathbf{W}° without destroying its structure, we define a vector \mathbf{w}_*° such that it selects the non-zero elements from \mathbf{w}° and a matrix $\mathbf{\Omega}_*$ devoid of the columns of $\mathbf{\Omega}$ containing the zero elements of \mathbf{w}° . Thus the LS estimate of \mathbf{w}_*° will be:

$$\mathbf{w}_*^\circ = (\mathbf{\Omega}'_* \mathbf{\Omega}_*)^{-1} \mathbf{\Omega}'_* \text{vec}(\mathbf{Q}' \mathbf{Y}^*) \quad (2.18)$$

Then \mathbf{w}° is obtained by reconstruction of the zero elements to their original positions and then \mathbf{W}° .

In the second step, analogously, for \mathbf{W}_T we simply need to modify the previous loss function isolating the term with \mathbf{W}_T and defining $\mathbf{Y}^{**} = \mathbf{Y}^* - \mathbf{X} \mathbf{W}^* \mathbf{A}'$, thus obtaining:

$$f = \text{SSQ}[\mathbf{Y}^* - \mathbf{X} \mathbf{W}^\circ \mathbf{A}' - \mathbf{T}^\circ \mathbf{W}_T \mathbf{A}'] \quad (2.19)$$

$$= \text{SSQ}[\text{vec}(\mathbf{Y}^{**}) - (\mathbf{A} \otimes \mathbf{T}^\circ) \text{vec}(\mathbf{W}_T)] \quad (2.20)$$

More in detail, the LS estimate of \mathbf{w}_{*T} ⁷ can be obtained as

$$\mathbf{w}_{*T} = (\mathbf{\Omega}'_{*T} \mathbf{\Omega}_{*T})^{-1} \mathbf{\Omega}'_{*T} \text{vec}(\mathbf{Y}^{**}) \quad (2.21)$$

with $\mathbf{\Omega}_T = \mathbf{A} \otimes \mathbf{T}^\circ$, $\mathbf{w}_T = \text{vec}(\mathbf{W}_T)$. As in the first step \mathbf{w}_T (and then \mathbf{W}_T) is obtained by reconstruction of the zero elements to their original positions.

Once got the estimates of \mathbf{W}_T and \mathbf{W}° the values of the score matrix \mathbf{F}^8 are then normalized in order to satisfy the condition $\text{diag}(\mathbf{F}' \mathbf{F}) = \mathbf{I}_D$.

⁷ \mathbf{w}_{*T} and $\mathbf{\Omega}_{*T}$ represent the non-zero counterparts of \mathbf{w}_T and $\mathbf{\Omega}_T$.

⁸ Weights to obtain \mathbf{F} values are taken projecting on the principal directions of the exogenous MVs the matrix of the endogenous indicators, after controlling for the direct effect of \mathbf{T} and \mathbf{X} , respectively, onto \mathbf{Y} .

As a result of the fact that \mathbf{XW}° are strictly formative indicators, while $\mathbf{XW}^\circ + \mathbf{T}^\circ\mathbf{W}_T = \mathbf{F}$ represents the overall composite scores in the specified model (Lovaglio and Vittadini, 2014), we compute \mathbf{F} as a combination of these two terms.

In the third step \mathbf{A}' is estimated, keeping fixed \mathbf{W}, \mathbf{W}_T and \mathbf{A}'_Y . Once computed the score matrix \mathbf{F} , the loss function (2.14) can be written as

$$f = \text{SSQ}[\mathbf{Y}^* - \mathbf{FA}'] = \text{SSQ}[\text{vec}(\mathbf{Y}^*) - (\mathbf{I} \otimes \mathbf{F})\text{vec}(\mathbf{A}')] \quad (2.22)$$

As in the previous step we now define $\mathbf{a} = \text{vec}(\mathbf{A}')$, $\mathbf{\Gamma} = \mathbf{I} \otimes \mathbf{F}$ (\mathbf{a}_* and $\mathbf{\Lambda}_*$ are their non-zero counterparts) and we compute the LS estimate of \mathbf{a}_* as:

$$\mathbf{a}_* = (\mathbf{\Gamma}'_*\mathbf{\Gamma}_*)^{-1}\mathbf{\Gamma}'_*\text{vec}(\mathbf{Y}^*) \quad (2.23)$$

Then, from \mathbf{a}_* we easily reconstruct \mathbf{a} and \mathbf{A}' .

The last step involves the estimation of \mathbf{A}'_Y for fixed \mathbf{W}, \mathbf{W}_T and \mathbf{A}' . Here we employ a multivariate regression with fixed elements to update \mathbf{A}'_Y , regressing \mathbf{T} onto $(\mathbf{Y} - \mathbf{FA}')$ scores⁹.

All these steps are then alternated until the algorithm reaches the convergence. Notice that in Scenario I (the concomitant indicators affect the observed endogenous variables), where we do not have \mathbf{W}_T , step 2 disappears.

Looking more deeply at the method we can show that the estimate of the original weights \mathbf{W} can be easily computed reversing the equation of the estimated matrix \mathbf{W}° ($\mathbf{W} = \mathbf{W}^\circ - \mathbf{X}^+\mathbf{TW}_T$). Such a model offers noticeable advantages in the latent scores interpretation:

⁹ \mathbf{F} and \mathbf{A} have been estimated in the previous steps.

1. The score matrix \mathbf{F} is obtained coalescing (a) \mathbf{XW}° , a combination of strictly formative indicators, independently of the contribution of concomitant indicators to the LS and (b) $\mathbf{T}^\circ\mathbf{W}_T$, an orthogonal (to \mathbf{X}) combination of concomitant indicators (Vacca, A.A. 2012-13).
2. Weights have meaningful geometric interpretations. In fact, the first set of weights \mathbf{W}° is the projection of endogenous observed indicators with scores adjusted by the direct effect of \mathbf{T} on \mathbf{Y} ($\mathbf{Q}'\mathbf{Y}^*$) on the space spanned by the principal directions of the QR decomposition of \mathbf{X} ($\text{vec}(\mathbf{\Omega}_*)$). Instead, the second set of weights \mathbf{W}_T is the projection of the endogenous observed indicators adjusted by either the direct effect of \mathbf{T} on \mathbf{Y} and the indirect effect of \mathbf{XW}° on \mathbf{Y} , through \mathbf{F} ($\mathbf{Y}^{**} = \mathbf{Y}^* - \mathbf{XW}^\circ\mathbf{A}'$) onto the column space spanned by \mathbf{T}° .

In conclusion, GRA solves a global optimization problem. GRA allows for the evaluation of the total fit of a hypothesized model, measured, as in ERA, by the total variance of all the observed endogenous variables explained by the exogenous variables (notice that if no variables are shared both by \mathbf{Y} and \mathbf{X} , $\tilde{\mathbf{Y}}$ reduces to \mathbf{Y} , see figure 2.3). If we have concomitant indicators, since they belong to the exogenous variables as long as formative indicators, the ERA fit index Ψ turns out to be

$$\Psi = \text{SS}[\tilde{\mathbf{Y}} - (\mathbf{TA}'_Y + \mathbf{FA}')] / \text{SS}[\tilde{\mathbf{Y}}] \quad (2.24)$$

where $\tilde{\mathbf{Y}}$ collects all the endogenous variables¹⁰. The larger the fit value, the more variance of the endogenous variables is explained by the exogenous variables.

¹⁰ Notice that the index still ranges from 0 to 1.

2.5 Generalized Redundancy Analysis advantages over Extended Redundancy Analysis

As shown, the limitations of ERA in accommodating concomitant indicators justify a better strategy (GRA), both in the model specification and estimation phase.

Specifically, ERA and GRA embody two different methods of specifying and fit the same SEM-RA model. Explicitly, for what regards model specification, the structural model does not change in the two methods. What changes is the measurement model. For what regards estimation instead, ERA estimates parameters through a loss function which embeds two block super-matrices, whereas GRA through a loss function where the different contributions of concomitant indicators and strictly formative indicators are disjointed, and thus embeds four matrices: \mathbf{W}° , \mathbf{W}_T (component weights) and \mathbf{A}_Y , \mathbf{A} (component loadings), with the clear consequences on the ALS estimation algorithm, which changes from the two steps of ERA to the four steps of GRA.

Thanks to these devices in the specification and estimation strategy of GRA, ERA drawbacks can be solved.

First of all, in GRA the estimation of the loadings \mathbf{A}_Y , associated to the direct effects, truly reflects the impact of an exogenous variables on endogenous variables, net of its effect through the component it contributes to “form” (fourth step of the algorithm). Hence the meaning of \mathbf{A}_Y and of the LC scores in the structural model is respected. In fact, these LC scores have been obtained in the previous step net of the contributions of concomitant indicators to the composite scores (second step of the algorithm). Notice that ERA, instead, estimates the sub-matrix corresponding to the direct effects using latent

components, whose scores have been obtained (in a previous step) ignoring the presence of these direct effects by an ERA model between \mathbf{X} and \mathbf{Y} .

Secondly, the orthogonality of both strictly formative and concomitant indicators in GRA model (Eq. (2.6)) GRA permits the investigation of the separate contributions of both blocks on the composite scores, contrary to ERA where LC scores are estimated as linear combinations of (typically correlated) formative and concomitant indicators.

This orthogonality, moreover, solves the problem of collinearity between concomitant indicators and LCs in the estimation of the direct effects.

In conclusion, the proposed method is simple and versatile enough to fit various complex relationships among variables, including direct effects of observed variables and/or concomitant variables.

However, also GRA presents some limitations, which require further extensions.

A first essential extension of the actual version of GRA regards the specification of measurement models for endogenous variables ($\mathbf{f} \rightarrow \mathbf{Y}$), not allowed in the current version.

The second issue instead is related to direct structural links between an exogenous and an endogenous Latent Composite (e.g. $f_1 \rightarrow f_2$). In fact, such an association at now needs the mediation of formative indicators (e.g., \mathbf{X}_2) of an endogenous LC (e.g., $f_1 \rightarrow \mathbf{X}_2 \rightarrow f_2$).

Both these aspects strongly depend on the two conditions on which the SEM-RA reposes. Moreover, RA mainly aims to obtain linear composites of observed exogenous variables, thus these extensions cannot be introduced unless more general models outside the redundancy analysis framework are proposed

(e.g., Generalized Structured Component Analysis (GSCA, Hwang and Takane, 2004)).

Furthermore, GRA can just accommodate formative schemes for the LCs, even capturing reflective relationships among a LC and the formative indicators of another LC. However, such an issue is more an apparent limitation than a big deal as the only formative scheme is represented by the measurement model consistently with CA (Millsap and Meredith, 1988), where any unobservable composite is defined as a linear combination of its observed indicators and where this kind of limitation is often present.

Chapter 3

-

The Simulation Study

3.1 The simulation study

In this section, to reinforce our methodological findings, the GRA model in three different scenarios has been evaluated through a complete simulation study, focusing on how biases and accuracy of the estimates behave in presence of different concomitant links:

- a. Study 1: differences in the specification of complex extensions
- b. Study 2: differences in case of correlation
- c. Study 3: different distributional scenarios (non-normality).

The results of each study will be presented and discussed, foregone by a detailed description of the model specification. To notice that in order to obtain significant results, all the samples have been extracted from a random variables, and not from a finite population, through a Monte Carlo analysis (Robert and Casella, 2004), thus we do not have a fixed N.

The evaluation of each simulation study has been based on the analysis of two indices representing the accuracy of the estimates:

- 1) The congruence coefficient $\phi = \theta^* \theta / [(\theta^* \theta^*)(\theta \theta)]^{1/2}$ (Lorenzo-Seva and ten Berge, 2006) between the estimates θ^* and the true parameters θ (vectors) and
- 2) the relative bias (in absolute value), $\gamma = 100(|\theta - \theta^*|/\theta)$ of each parameter estimate.

Estimates have been obtained through the Monte Carlo analysis conducted, for each sample size.

3.1.1 Study 1: differences in the specification of complex extensions

The first study is pointed toward the evaluation of GRA performances in three different path diagrams at increasing complexity (Figure 3.1) through simulation, focusing on the behaviour of both biases and accuracy of the estimates in presence of external covariates, in a clear step-by-step increase in complexity of the GRA model specification:

- (i) a single covariate affecting a single endogenous variable (Scenario I);
- (ii) a single concomitant indicator affecting also LCs (Scenario II) and
- (iii) a concomitant indicator affecting both the endogenous variables (Scenario III).

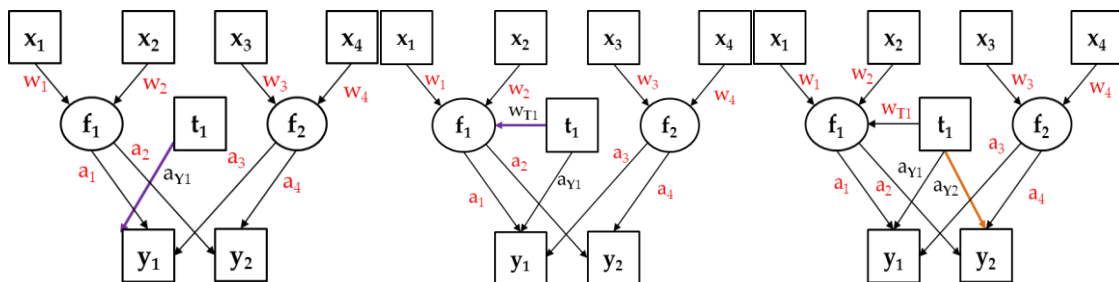


Figure 3.1. Three GRA path specifications, with increasing complexity. Starting from the left we find Scenario I, then Scenario II and, on the right, Scenario III. (Pafundi and Vacca, 2014).

We started from a basic GRA model specification, with the covariate t_1 just affecting an endogenous variable, to then move towards the more complex scenarios, in which also a latent variable is involved, which has been our main interest. In particular the third scenario shows the influence of the concomitant indicator not only on one endogenous variable and one LC, but on the other endogenous variable too.

For each scenario, the distributions underlying \mathbf{X} , t_1 and \mathbf{E} simulated data, with variance scaled to unit and \mathbf{X} and t_1 independent, are $\mathbf{X} \sim N_4(0, \Sigma_x)$, $t_1 \sim N(0, 1)$,

and $E \sim N_2(0, \Sigma_E)$. The covariance matrices and the arbitrary parameters for \mathbf{W} and \mathbf{A}' have been defined as follows (as in Hwang and Takane, 2002):

$$\Sigma_{\mathbf{X}\mathbf{T}} = \begin{bmatrix} 1 & 0.6 & 0.05 & 0.05 & 0 \\ 0.6 & 1 & 0.05 & 0.05 & 0 \\ 0.05 & 0.05 & 1 & 0.6 & 0 \\ 0.05 & 0.05 & 0.6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.6 & 0 \\ 0.6 & 0 \\ 0 & 0.6 \\ 0 & 0.6 \end{bmatrix}, \mathbf{A}' = \begin{bmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}, \Sigma_E = \begin{bmatrix} 1 & \\ 0.5 & 1 \end{bmatrix}$$

For Scenario I, with only one external link (t_1 affects just y_1), \mathbf{W}_T is null and $\mathbf{A}_Y' = [0.3 \ 0]$; for Scenario II, with a concomitant link (t_1 affects both y_1 and f_1), $\mathbf{W}_T = [0.4 \ 0]$ and $\mathbf{A}_Y' = [0.3 \ 0]$ and, finally, for Scenario III, with the second external link (t_1 affects both y_1 and y_2 , and f_1): $\mathbf{W}_T = [0.4 \ 0]$ and $\mathbf{A}_Y' = [0.3 \ 0.3]$.

Once extracted the samples through a Monte Carlo simulation, we have computed the endogenous matrix \mathbf{Y} based on the above specifications (see Appendices 1A, 1B).

3.1.2 Study 2: Generalized Redundancy Analysis performances in case of correlation between X and T

In order to further estimate the robustness of the Generalized Redundancy Analysis (GRA) procedure in the estimation of structural equation models, we then tested GRA for multicollinearity.

Thus, the second step of our analysis, which is based on the previous "Scenario II" (Figure 3.2), on GRA performances has been to consider the presence of correlations between both the concomitant ($\mathbf{x}_1, \mathbf{x}_2$) and non-concomitant ($\mathbf{x}_3, \mathbf{x}_4$) blocks and t_1 .

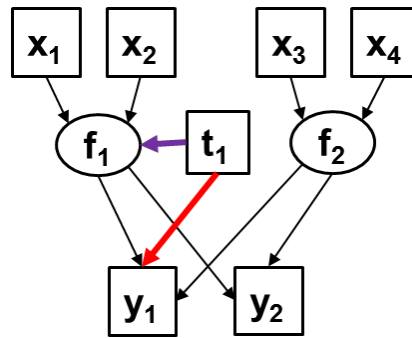


Figure 3.2. GRA path specification, with a single concomitant indicator (t_1) affecting also LCs (Scenario II).

In order to do this we modified the script (see Appendix 2) by the introduction of matrix $[X|T]$, which presents, in the last column and last row of the variances and covariances matrix, the correlations between each block of X (concomitant and non-concomitant) and t_1 , in six different combinations of values, for testing GRA behavior in presence of either low or high correlation when compared to the uncorrelated case. We fixed the correlations between the x of the same block at 0.3 and the correlations between blocks at 0.1. The different cases analyzed differ for the correlation between both the concomitant (x_1 and x_2) and non-concomitant (x_3 and x_4) X blocks and t_1 . Thus we can sum up all the cases as follows:

		Scenario II			
Case		σ_{x1t1}	σ_{x2t1}	σ_{x3t1}	σ_{x4t1}
i)	Uncorr	0	0	0	0
ii)	Low Corr – Total	0.3	0.3	0.3	0.3
iii)	Low Corr – Concomitant, t_1	0.3	0.3	0	0
iv)	Low Corr - Non Concomitant, t_1	0	0	0.3	0.3
v)	High Corr - Concomitant, t_1	0.7	0.7	0	0
vi)	High Corr – Non Concomitant, t_1	0	0	0.7	0.7

Looking more in detail at the whole variances and covariances matrix Σ_{xT} , the six cases can be rewritten as:

(i) Absence of correlation (all covariances between both the concomitant block $(\mathbf{x}_1, \mathbf{x}_2)$ and the non-concomitant block $(\mathbf{x}_3, \mathbf{x}_4)$ and \mathbf{t}_1 are equal to zero);

(ii) Correlation between both the concomitant block $(\mathbf{x}_1, \mathbf{x}_2)$ and non-concomitant block $(\mathbf{x}_3, \mathbf{x}_4)$ and \mathbf{t}_1 fixed at 0.3

$$\left(\Sigma_{\text{XT}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0.3 \\ 0.3 & 1 & 0.1 & 0.1 & 0.3 \\ 0.1 & 0.1 & 1 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 1 \end{bmatrix} \right);$$

(iii) Low correlation between the concomitant block $(\mathbf{x}_1, \mathbf{x}_2)$ and \mathbf{t}_1 (null covariance between the non-concomitant block $(\mathbf{x}_3, \mathbf{x}_4)$ and \mathbf{t}_1 , covariance

between $(\mathbf{x}_1, \mathbf{x}_2)$ and \mathbf{t}_1 equal to 0.3) $\left(\Sigma_{\text{XT}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0.3 \\ 0.3 & 1 & 0.1 & 0.1 & 0.3 \\ 0.1 & 0.1 & 1 & 0.3 & 0 \\ 0.1 & 0.1 & 0.3 & 1 & 0 \\ 0.3 & 0.3 & 0 & 0 & 1 \end{bmatrix} \right);$

(iv) Low correlation between the non-concomitant block $(\mathbf{x}_3, \mathbf{x}_4)$ and \mathbf{t}_1 (null covariance between the concomitant block $(\mathbf{x}_1, \mathbf{x}_2)$ and \mathbf{t}_1 , covariance

between $(\mathbf{x}_3, \mathbf{x}_4)$ and \mathbf{t}_1 equal to 0.3) $\left(\Sigma_{\text{XT}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0 \\ 0.3 & 1 & 0.1 & 0.1 & 0 \\ 0.1 & 0.1 & 1 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.3 & 1 & 0.3 \\ 0 & 0 & 0.3 & 0.3 & 1 \end{bmatrix} \right);$

(v) High correlation between the concomitant block $(\mathbf{x}_1, \mathbf{x}_2)$ and \mathbf{t}_1 (equal to case (c), changing the value 0.3 with 0.7)

$$\left(\Sigma_{\text{XT}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0.7 \\ 0.3 & 1 & 0.1 & 0.1 & 0.7 \\ 0.1 & 0.1 & 1 & 0.3 & 0 \\ 0.1 & 0.1 & 0.3 & 1 & 0 \\ 0.7 & 0.7 & 0 & 0 & 1 \end{bmatrix} \right);$$

(vi) High correlation between non-concomitant block ($\mathbf{x}_3, \mathbf{x}_4$) and \mathbf{t}_1 (equal to case (d), changing the value 0.3 with 0.7)

$$\left(\Sigma_{\mathbf{X}\mathbf{T}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0 \\ 0.3 & 1 & 0.1 & 0.1 & 0 \\ 0.1 & 0.1 & 1 & 0.3 & 0.7 \\ 0.1 & 0.1 & 0.3 & 1 & 0.7 \\ 0 & 0 & 0.7 & 0.7 & 1 \end{bmatrix} \right).$$

The distributions underlying $[\mathbf{X}|\mathbf{T}]$ and \mathbf{E} simulated data are $[\mathbf{X}|\mathbf{T}] \sim N_5(0, \Sigma_{\mathbf{X}\mathbf{T}})$, and $\mathbf{E} \sim N_2(0, \Sigma_{\mathbf{E}})$. $[\mathbf{X}|\mathbf{T}]$ and \mathbf{E} covariance matrices and the arbitrary parameters for \mathbf{W} and \mathbf{A}' have been defined as follows (e.g., Hwang and Takane, 2002) for the uncorrelation situation (case (i)):

$$\Sigma_{\mathbf{X}\mathbf{T}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0 \\ 0.3 & 1 & 0.1 & 0.1 & 0 \\ 0.1 & 0.1 & 1 & 0.3 & 0 \\ 0.1 & 0.1 & 0.3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \Sigma_{\mathbf{E}} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}; \mathbf{W} = \begin{bmatrix} 0.6 & 0 \\ 0.6 & 0 \\ 0 & 0.6 \\ 0 & 0.6 \end{bmatrix}; \mathbf{A}' = \begin{bmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{bmatrix};$$

where, in the matrix $\Sigma_{\mathbf{X}\mathbf{T}}$, the covariances between \mathbf{X} and \mathbf{T} (last column and row) are put to 0 in case of uncorrelation. In the other cases, as depicted above, 0 values have been substituted with the other chosen covariances.

Each case of correlation has been evaluated with the same two indices used in our first simulation: (a) the congruence coefficient ϕ between the estimates θ^* and (b) the real parameters θ and the relative bias γ (in absolute value) of each estimate.

3.1.3 Study 3: Generalized Redundancy Analysis and Non-normal distributions

In our previous simulation studies we always assumed that our data were sampled from a normal population and this usually happens in analyzing data. On the other hand, it may happen, in the empirical research, that the data we need to analyze either present outliers (which determine, at least, an increase in the kurtosis), etc. In other words, it may often happen that the variables we want to analyze present different degrees of skewness and/or kurtosis, which implies that they are non-normally distributed.

In order to verify if GRA methodology behaves well also in case of non-normal distributions, we decided to investigate its behavior in case of non-normality.

We in particular focused on the non-normality of the errors, testing both Scenario I and Scenario II (see Figure 3.1) for three different degrees of kurtosis and skewness, compared to the normal case. The three analyzed cases can be summed up as follows:

Case	Scenario I		Scenario II	
	Skewness	Kurtosis	Skewness	Kurtosis
Normal	0	3	0	3
a	1	0.5	1	0.5
b	1.5	2.5	1.5	2.5
c	2	5.5	2	5.5

In order to perform such a study we needed to simulate the non-normal distribution on which to test GRA algorithm. What is needed in general is a methodology able not only to generate non-normality, also fulfilling a set of conditions:

1. In the method a set of parameter has to be known a priori;
2. The researcher can change distribution with the minimum effort;
3. The generated distribution have to simulate realistically the empirical distributions;
4. Capability to generate a wide spectrum of different distributions;
5. The method must be as more efficient as possible.

Thus, in this specific case, in order to avoid problems of over/underestimation, we generated non-normal distributions (extracted as Uniform in this specific case) through the so called “power method” (Fleishman, 1978)¹¹, implementing it in our script (Appendix 3).

With regard instead to \mathbf{X} and \mathbf{T} simulated data distributions, they are assumed to be normal, with variance scaled to unit, in particular we will have: $\mathbf{X} \sim N_4(0, \Sigma_x)$, $\mathbf{T} \sim N(0, \sigma^2_T)$. Looking specifically to the error matrix \mathbf{E} , the generated data are normal, $\mathbf{E} \sim N_2(0, \Sigma_E)$, then we have applied Fleishman algorithm on them in order to create the three different cases depicted above.

¹¹ Fleishman power method is based on a polynomial transformation, which would be of the form:

$$\mathbf{Y} = a + b\mathbf{X} + c\mathbf{X}^2 + d\mathbf{X}^3 \text{ (or, equivalently } \mathbf{Y} = d\mathbf{X} + c\mathbf{X} + b\mathbf{X} + a),$$

where \mathbf{X} is a random variate normally distributed ($\mathbf{X} \sim N(0,1)$), \mathbf{Y} distributed depending on the constants for different skewness and kurtosis (e.g., as it is convenient for the mean to be null, a condition necessary and sufficient for this is that $a = -c$). This transformation, though it needs a lot of operations, is extremely efficiently performed by the computer. Notice that given a certain skewness value, not all values of kurtosis will provide a solution, but just the points above a certain values. This range of values can be described by the parabola of equation:

$$\text{skew}^2 < 0.0629576 * \text{kurtosis} + 0.0717247$$

which represent a limit of this power method.

In our study, in order to employ this method and have proper solutions, the kurtosis values chosen had to fulfill the following condition:

$$\text{kurt} > -1.13168 + 1.58837 * \text{skew}^2.$$

The relative covariance matrices and the arbitrary parameters for \mathbf{W} and \mathbf{A}' have been defined as follows:

$$\Sigma_{\text{XT}} = \begin{bmatrix} 1 & 0.3 & 0.1 & 0.1 & 0 \\ 0.3 & 1 & 0.1 & 0.1 & 0 \\ 0.1 & 0.1 & 1 & 0.3 & 0 \\ 0.1 & 0.1 & 0.3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.6 & 0 \\ 0.6 & 0 \\ 0 & 0.6 \\ 0 & 0.6 \end{bmatrix}, \mathbf{A}' = \begin{bmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}, \Sigma_{\text{E}} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$$

Each case, even in this study, has been evaluated through the same two indices used in the other studies: (i) the congruence coefficient ϕ between the estimates θ^* and (ii) the real parameters θ and the relative bias γ (in absolute value) of each estimate.

3.2 Results and discussion

3.2.1 Study 1 – Specification scenarios

Looking at the increasing complexity models, in all the three scenarios tested convergence of the estimates has been reached (Table 3.1). Results show a general increase in $\text{Mean}(\phi)$ along with an increase in the sample size, indicating a satisfactory recovery of the population parameters, with $n > 200$, for Scenarios I and II ($\phi > 0.9$). The distribution of ϕ through the replications appears to be increasingly pointed towards high values, with decreasing standard deviations. Also Scenarios III recovers efficiently the true values of the coefficients, with $\phi = 0.902$ already for $n = 200$ and standard deviations coherent with those of Scenario I and Scenario II ($\text{StdDev}(\phi) = 0.023$ at $n = 600$).

Scenario	I	II	III	I	II	III
N	Mean(ϕ)			StdDev(ϕ)		
50	0,696	0,667	0,629	0,270	0,255	0,253
100	0,851	0,799	0,735	0,152	0,174	0,204
200	0,936	0,901	0,902	0,070	0,103	0,129
400	0,969	0,952	0,940	0,029	0,043	0,064
600	0,979	0,969	0,969	0,018	0,024	0,023

Table 3.1. Descriptive statistics of ϕ for the GRA model for $n = \{50; 100; 200; 400; 600\}$ and model complexity. For scenario III convergence criterion has been lowered from 10^{-4} to 10^{-3} .

The reasons at the basis of such a behavior are highlighted analyzing the biases of the estimates (Figure 3.3, Table 3.2), generally satisfactory for all Scenarios I, II and III, although the presence of the concomitant indicator affects some estimates peculiarly:

- a. With t_1 , **W** biases tend to decrease with the sample size, having w_1 and w_2 affected by its presence ($\rightarrow y_1$ in I and also $\rightarrow f_1$ in II), for which $\gamma(w_1)$ and $\gamma(w_2)$, initially increasing for $n = 50 \rightarrow 250$, then they decrease and stabilize at $\sim 2\%$ in model I and, from $n > 250$, at $\sim 5-10\%$ in scenario II;
- b. **A** biases do not seem to be affected by the absence of the concomitant indicators, (a_3 and a_4 in I and II are not affected by the absence of indicators related to f_2 , in fact $\gamma(a_3)$ and $\gamma(a_4)$ are $\sim 2\%$ for scenario I and $\sim 5\%$ for scenario II, regardless of n).
- c. Scenario III is more unstable for lower values of n , with biases mainly on loadings related to f_1 : having t_1 and f_1 sharing the same endogenous variables, should normally lower **A** biases, but the causal link between t_1 and y_2 is only external, influencing them similarly to Scenario I.
- d. Moreover, in Scenario III, the additional external link affects w_{T1} bias, which is quite fluctuating with the sample size ($\gamma(w_{T1}) \sim 60\%$ for $n = 350$). $\gamma(w_{T1})$ reaches values around 20% just for large n ($\gamma(w_{T1}) < 20\%$ for $n = 550$).

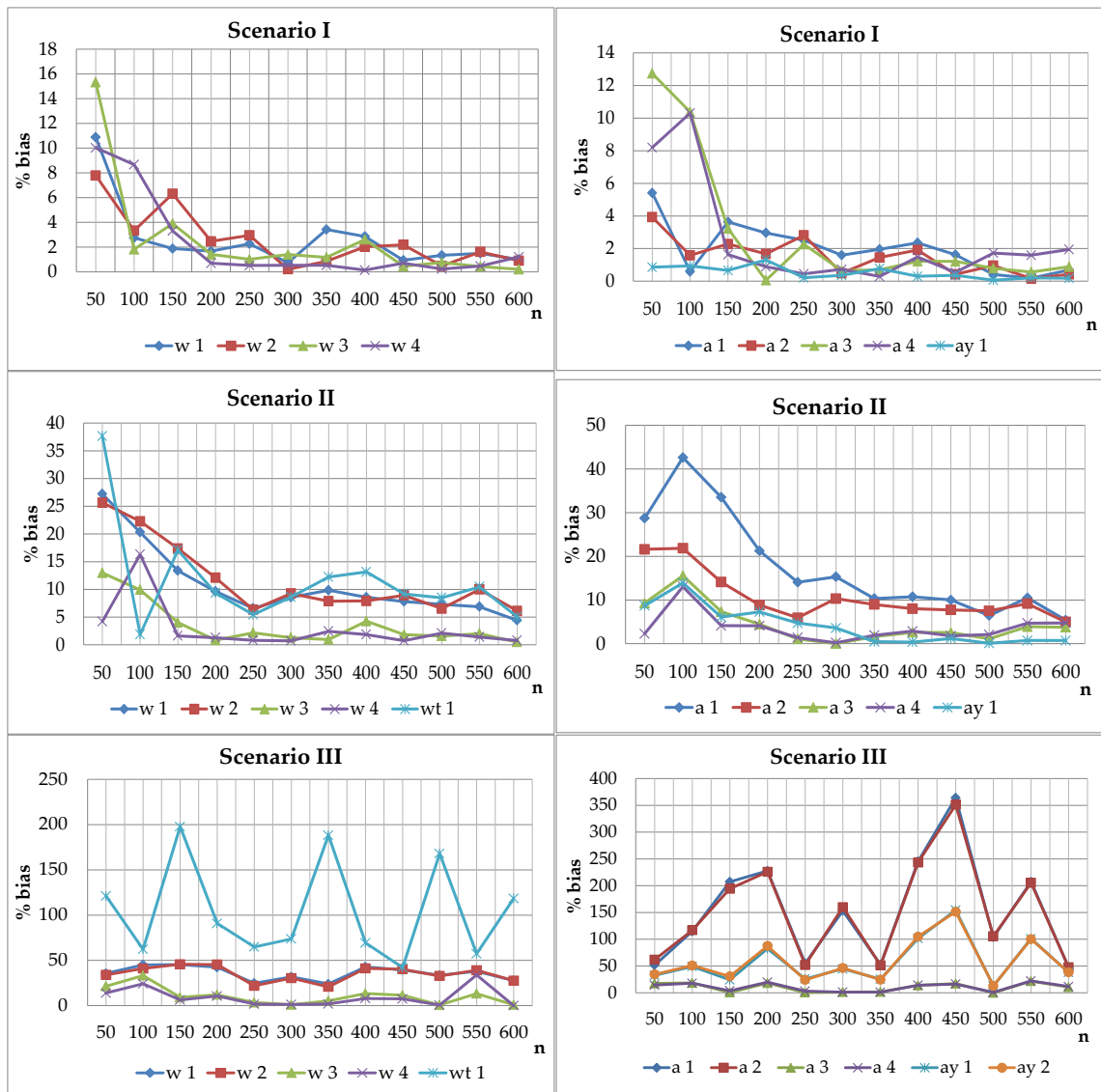


Figure 3.3. The trend of the biases at increasing sample size is shown. Starting from the top we find scenario I, scenario II and scenario III.

Specification and Estimation of Complex Structural Models with Covariate Effects

Scenario			I			II			III		
Par.	Pop.	n	Est.	S.E.	%Bias	Est.	S.E.	Bias%	Est.	S.E.	Bias%
w ₁	0.6	50	0.54	0.445	10.9	0.44	0.38	27.2	0,40	0,375	33,2
		100	0.58	0.326	2.8	0.48	0.289	20.3	0,38	0,252	37,5
		200	0.61	0.242	1.7	0.54	0.235	9.6	0,53	0,232	11,4
		400	0.58	0.16	2.9	0.55	0.158	8.6	0,52	0,164	13,3
		600	0.61	0.132	1	0.57	0.13	4.4	0,55	0,125	8,0
w ₂	0.6	50	0.55	0.622	7.8	0.45	0.534	25.6	0,42	0,498	30,8
		100	0.58	0.377	3.3	0.47	0.334	22.3	0,39	0,331	34,9
		200	0.59	0.239	2.5	0.53	0.22	12.1	0,51	0,233	14,3
		400	0.59	0.151	2.0	0.55	0.148	7.9	0,53	0,168	11,6
		600	0.60	0.131	0.9	0.56	0.13	6.1	0,55	0,137	9,0
a ₁	0.3	50	0.19	0.149	5.4	0.26	0.332	28.7	0,29	0,366	44,2
		100	0.20	0.107	0.6	0.29	0.239	42.6	0,36	0,317	82,1
		200	0.19	0.073	3.0	0.24	0.225	21.3	0,24	0,201	22,0
		400	0.21	0.05	2.4	0.22	0.082	10.8	0,24	0,085	18,7
		600	0.20	0.042	0.7	0.21	0.046	5.4	0,22	0,050	9,5
a ₂	0.3	50	0.19	0.16	3.9	0.24	0.159	21.6	0,30	0,347	51,6
		100	0.20	0.104	1.6	0.24	0.103	21.9	0,37	0,286	86,6
		200	0.20	0.077	1.7	0.22	0.073	8.9	0,25	0,180	22,4
		400	0.20	0.051	1.9	0.22	0.05	8.0	0,24	0,086	18,0
		600	0.20	0.042	0.4	0.21	0.041	5.0	0,22	0,049	9,7
a ₃	0.3	50	0.18	0.155	12.7	0.18	0.17	9.3	0,17	0,172	17,2
		100	0.18	0.111	10.4	0.17	0.116	15.6	0,17	0,120	17,5
		200	0.20	0.074	0.1	0.19	0.079	4.5	0,19	0,079	4,2
		400	0.20	0.052	1.2	0.21	0.052	2.6	0,21	0,053	3,3
		600	0.20	0.04	0.9	0.21	0.04	3.8	0,21	0,040	5,0
a ₄	0.3	50	0.18	0.156	8.2	0.20	0.156	2.3	0,17	0,171	13,2
		100	0.18	0.109	10.3	0.17	0.109	13.0	0,17	0,115	16,9
		200	0.20	0.073	0.9	0.19	0.073	4.1	0,19	0,080	5,7
		400	0.20	0.05	1.5	0.21	0.05	2.9	0,21	0,050	3,7
		600	0.20	0.041	2.0	0.21	0.041	4.8	0,21	0,041	6,2
w _{T1}	0.4	50	-	-	-	0.19	0.532	37.6	-0,05	0,633	115,9
		100	-	-	-	0.29	0.369	1.8	0,46	0,414	52,7
		200	-	-	-	0.27	0.325	9.4	0,33	0,314	10,3
		400	-	-	-	0.26	0.237	13.1	0,34	0,294	12,0
		600	-	-	-	0.29	0.195	5.0	0,40	0,179	31,5

Table 3.2. Estimates, S. Errors and biases for relevant parameters, by scenario and sample size.

3.2.2 Study 2 – Correlation Scenarios

If we then explore the analysis on the multicollinearity, both cases of low (ii, iii, iv) and high (v, vi) correlation show convergence of the estimates (Table 3.3). The results obtained in the three cases of low correlation (ii, iii, iv) and in the two of high correlation (v, vi), respectively, are depicted in the tables below. The results show a general increase in $\text{Mean}(\phi)$ along with an increase in the sample size both in cases of uncorrelation (i), low and high correlations (ii \rightarrow vi), indicating a satisfactory recovery of the population parameters. In fact, a congruence value of $\phi > 0.9$ has been reached for $n > 200$ in all the low correlations scenarios (ii, iii, iv), while in case of high correlation (v, vi) a similar value ($\phi > 0.905$) for $n > 400$ in the case {0.7} and only from $n=600$ ($\phi > 0.898$) for the case {0.7 0}. As highlighted by the table below, the distribution of ϕ through the replications appears to be increasingly pointed towards high values, with decreasing standard deviations.

Case	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(i)	(ii)	(iii)	(iv)	(v)	(vi)
N	Mean(ϕ)						StdDev(ϕ)					
50	0,647	0,615	0,612	0,639	0,581	0,473	0,264	0,285	0,280	0,273	0,267	0,307
100	0,801	0,778	0,730	0,803	0,729	0,633	0,161	0,181	0,260	0,167	0,196	0,231
200	0,912	0,901	0,895	0,913	0,845	0,757	0,078	0,088	0,099	0,079	0,139	0,216
400	0,955	0,948	0,947	0,955	0,905	0,867	0,042	0,044	0,045	0,042	0,093	0,141
600	0,969	0,964	0,963	0,969	0,935	0,898	0,023	0,025	0,026	0,024	0,066	0,120

Table 3.3. Descriptive statistics of ϕ for the GRA model for $n = \{50; 100; 200; 400; 600\}$ in the different cases of correlation: (i) Uncorrelation, (ii) [correlation of both the concomitant ($\mathbf{x}_1, \mathbf{x}_2$) and non-concomitant ($\mathbf{x}_3, \mathbf{x}_4$) blocks with \mathbf{t}_1 is set equal to 0.3], (iii) [low correlation between the concomitant block ($\mathbf{x}_1, \mathbf{x}_2$) and \mathbf{t}_1 set to 0.3, null for the non-concomitant block ($\mathbf{x}_3, \mathbf{x}_4$)], (iv) [low correlation set to 0,3 between the non-concomitant block ($\mathbf{x}_3, \mathbf{x}_4$) and \mathbf{t}_1 , null for the concomitant block], (v) [high correlation between the concomitant block ($\mathbf{x}_1, \mathbf{x}_2$) and \mathbf{t}_1 set to 0.7, null for the non-concomitant block] and, finally, (vi) [high correlation between the non-concomitant block ($\mathbf{x}_3, \mathbf{x}_4$) and \mathbf{t}_1 set to 0.7 and null for the concomitant block].

Deepening the reasons why this happens, we report a detailed analysis of the obtained biases of the estimates (Figures 3.4-3.5-3.6; Tables 3.4-3.5), which reveals a generally good trend in the low correlation cases (ii, iii, iv), although less consistent at higher correlations (v, vi). This behavior is in particular due to the presence of the concomitant indicator, which affects some estimates peculiarly.

In case of uncorrelation (case (i)), loadings biases are characterized by different behaviors. $\gamma(a_3)$ and $\gamma(a_4)$ values are both next to zero with every sample size as the LC f_2 is not affected by the concomitant indicator. $\gamma(a_{y1})$ presents quite the same behavior of $\gamma(a_3)$ and $\gamma(a_4)$ (direct effect of t_1 on y_1), instead $\gamma(a_1)$ and $\gamma(a_2)$ tend to have low values just from higher n ($n > 250$).

In this case, in fact, the relationship is indirectly influenced by the affection of f_1 by t_1 . For what regard the weight matrix \mathbf{W} instead, biases present a quite fluctuating trend. While $\gamma(w_3)$ and $\gamma(w_4)$ are small at all sample sizes, as the concomitant indicator is not involved in this relationship, while $\gamma(w_1)$ and $\gamma(w_2)$ tend to stabilize around low values (under 10) just from $n > 250$, which can be due indirectly to the influence of t_1 on the LC f_1 . Finally, $\gamma(w_{T1})$ is the most fluctuating parameter as we have the direct effect of the concomitant indicator t_1 on f_1 (see Fig. 3.2).

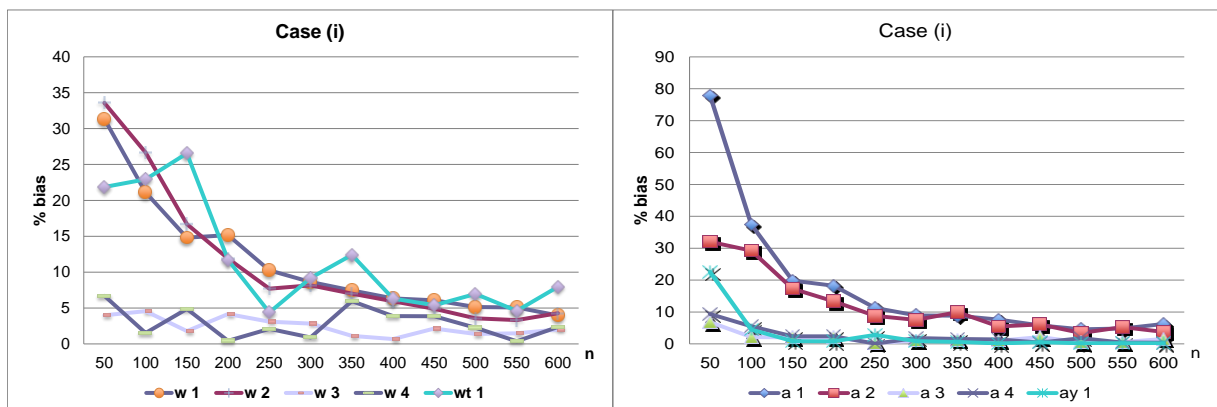


Figure 3.4. Bias trend for weights and loadings in case of uncorrelation (case (i) – the covariances between both the concomitant and non-concomitant \mathbf{X} blocks and t_1 are null).

We then analyzed the case of low correlations (Figure 3.5, Table 3.4), starting from setting covariances between both the concomitant and non-concomitant blocks and t_1 all equal to 0.3 (case (ii)).

In this case $\gamma(a_3)$, $\gamma(a_4)$ and $\gamma(a_{Y1})$ assume constantly small values next to the zero at increasing n . Also $\gamma(a_1)$ and $\gamma(a_2)$ tend to be small, but just from $n > 250$.

For what regard \mathbf{W} matrix behavior instead, we notice that the trend is quite fluctuating. Just $\gamma(w_3)$ and $\gamma(w_4)$ are quite constant both at small and large sample sizes, while $\gamma(w_1)$, $\gamma(w_2)$ and $\gamma(w_{T1})$ reflect the influence, both direct or indirect, of t_1 on f_1 .

Second low correlation test has been on (iii), with low correlation $\{0.3 \ 0\}$ between the concomitant block (x_1, x_2) and t_1 . Here we can observe that $\gamma(a_3)$, $\gamma(a_4)$ and $\gamma(a_{Y1})$ tend to be constant next to zero at every sample size, while $\gamma(a_1)$ and $\gamma(a_2)$ are more fluctuating, with stable values around 20 from $n > 200$. With regard to weights biases instead, $\gamma(w_3)$ and $\gamma(w_4)$ tend to present low values at every sample size, while $\gamma(w_1)$, $\gamma(w_2)$ and $\gamma(w_{T1})$ tend to be high at small sample sizes, becoming stable for values around 15 from $n > 250$.

We can notice that if exogenous and concomitant variables are correlated and each of them participate in the formation of the latent variable, the correspondent parameters would be more bias affected. With larger n the bias in any case tends to "switch off", it decreases, even if more slowly.

Let's now move to the opposite case (iv), with low correlation case $\{0 \ 0.3\}$ between the non-concomitant block (x_3, x_4) and t_1 . Here we can observe that \mathbf{A} biases are all small at increasing sample size; the only high value we register for smaller n is $\gamma(a_1)$. \mathbf{W} parameters instead show very small biases for large n in general. Just $\gamma(w_1)$, $\gamma(w_2)$ and $\gamma(w_{T1})$ are high for $n = 50$.

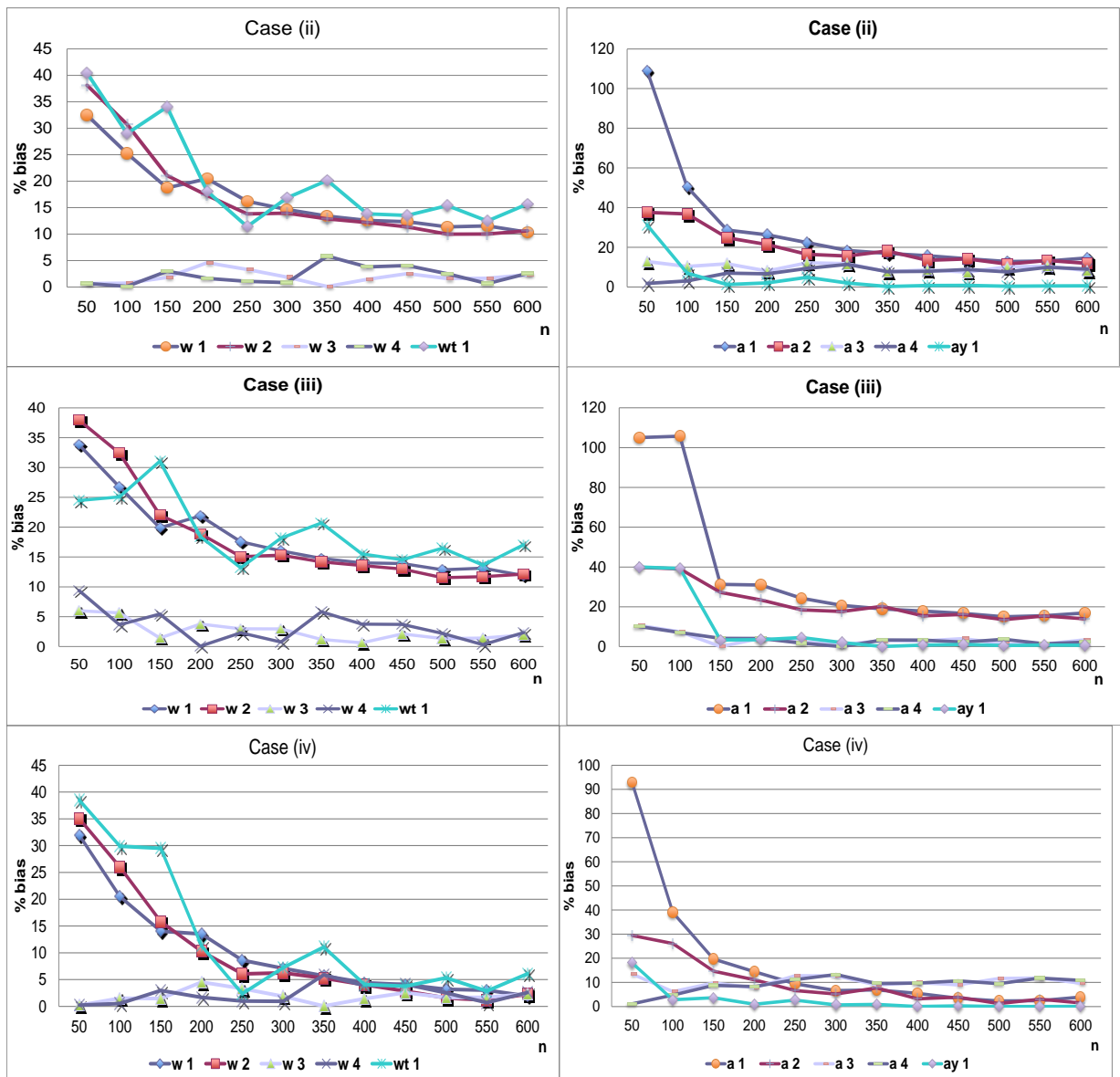


Figure 3.5. The trend of the biases of both loading matrix A and weight matrix W is shown, for all the low correlation cases (ii, iii, iv). From the top we find: case (ii) - correlation of both the concomitant (x_1, x_2) and non-concomitant (x_3, x_4) blocks with t_1 is set equal to 0.3; case (iii) - correlation between the concomitant block (x_1, x_2) and t_1 set to 0.3 and null for the non-concomitant block (x_3, x_4) and, finally, case (iv) - low correlation set to 0.3 between the non-concomitant block (x_3, x_4) and t_1 and null for the concomitant block.

At this point we tested for higher correlation cases (Table 3.5, Figure 3.6). Firstly case (v), with high correlation {0.7 0} between the concomitant block ($\mathbf{x}_1, \mathbf{x}_2$) and \mathbf{t}_1 . Here $\gamma(a_2)$, $\gamma(a_3)$, $\gamma(a_4)$ and $\gamma(a_{Y1})$ tend to be constant every sample size we have to deal with, just $\gamma(a_1)$ presents high values for small sample sizes. Matrix **W** parameters instead show linearly constant biases, higher for $\gamma(w_4)$, $\gamma(w_3)$, lower for $\gamma(w_2)$ and $\gamma(w_1)$. To notice that bias is higher when there is a direct effect of the concomitant indicator on the LC f_1 ($\gamma(w_{T1})$).

Finally, case (vi), with high correlation {0 0.7} between the non-concomitant block ($\mathbf{x}_3, \mathbf{x}_4$) and \mathbf{t}_1 . Here we can notice $\gamma(a_{Y1})$, $\gamma(a_3)$ and $\gamma(a_4)$ tend to fluctuate for small n and then they assume smaller values at larger sample size. $\gamma(a_1)$ and $\gamma(a_2)$ instead are smaller for small n and larger for large sample size. Looking at **W** parameters instead, we have that $\gamma(w_3)$ and $\gamma(w_4)$ are small for both small and large sample size, while $\gamma(w_1)$, $\gamma(w_2)$ and $\gamma(w_{T1})$ still reflect the influence of the concomitant indicator on the LC f_1 . Such a behavior can be due to the issue of the collinearity among the present variables.

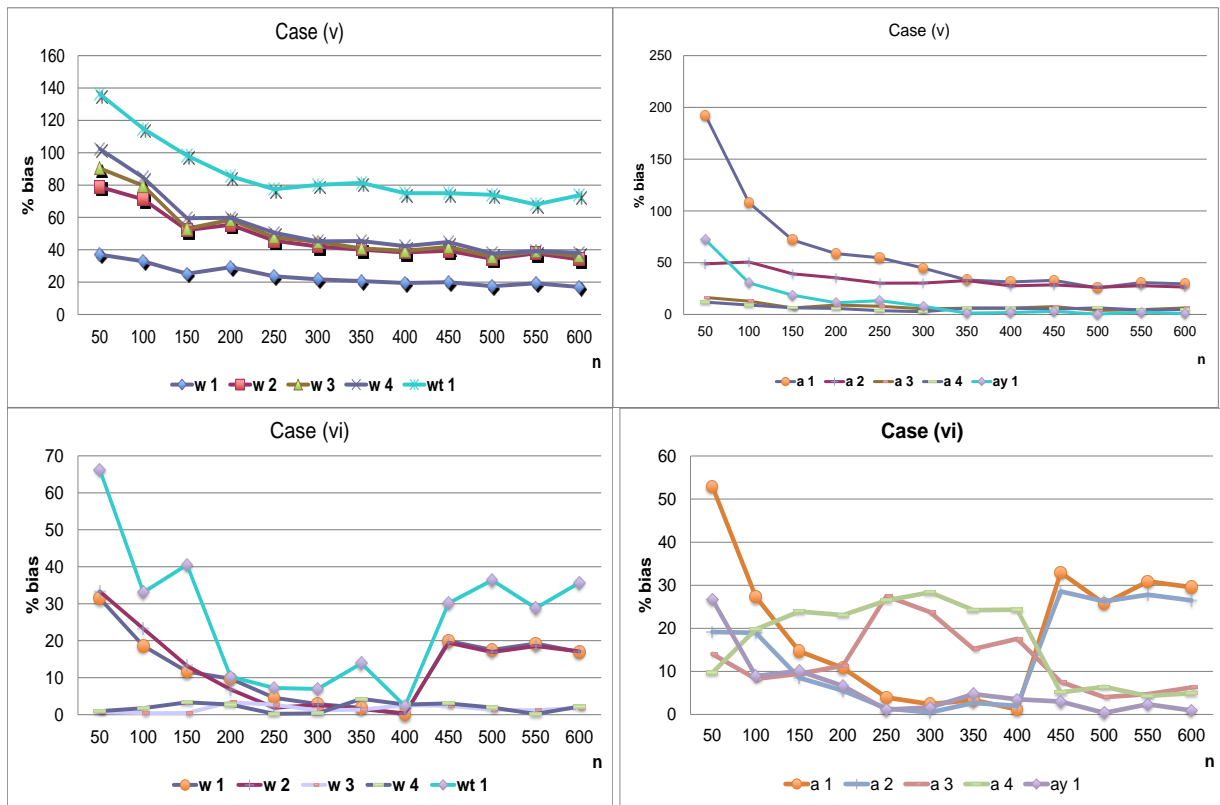


Figure 3.6. The trend of the biases of both loading matrix A and weight matrix W is shown, for the high correlation cases (v, vi) between either the concomitant (x_1, x_2) or the non-concomitant (x_3, x_4) block and t_1 . From the top we find: case (v) - high correlation between the concomitant block (x_1, x_2) and t_1 set to 0.7, null for the non-concomitant block and case (vi) - high correlation between the non-concomitant block (x_3, x_4) and t_1 set to 0.7 and null for the concomitant block.

Eventually we can observe that high degrees of correlation determine not so good performances of GRA algorithm. That's why further work in this sense is needed: an attempt to lower biases in this sense would be by adding further exogenous variables to the latent composites.

3.2.3 Study 3 – Distribution Scenarios

The last scenario we have to look into regards the case of non-normality data. In all cases of variation of skewness and kurtosis values, for both Scenario I and Scenario II we have reached the convergence of the estimates (Tables 3.6-3.7). Results show that a change in skewness and kurtosis parameters does not affect the performance of the model. In fact, as shown in the tables below, the non-normal models behave as the normal ones, with a general increase in Mean(ϕ) along with an increase in the sample size. To reinforce this affirmation we also notice how Mean(ϕ) values are about in the same way in all cases, both normal and non-normal in both Scenarios I and II, with a satisfactory recovery of the population parameters, from $n > 200$, for Scenarios I and II ($\phi > 0.9$). The distribution of ϕ through the replications appears to be increasingly pointed towards high values, with decreasing standard deviations.

Scenario	I				I			
	Skewness/Kurtosis				Skewness/Kurtosis			
Case	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
N	Mean(ϕ)				StdDev(ϕ)			
50	0,696	0,706	0,710	0,707	0,270	0,262	0,260	0,267
100	0,851	0,855	0,855	0,858	0,152	0,148	0,147	0,142
200	0,936	0,926	0,926	0,927	0,070	0,068	0,071	0,069
400	0,969	0,964	0,964	0,964	0,029	0,033	0,033	0,033
600	0,979	0,977	0,977	0,977	0,018	0,020	0,020	0,020

Table 3.6. Descriptive statistics of ϕ for GRA Scenario I for $n = \{50; 100; 200; 400; 600\}$ in case of non-normal distribution, with increasing skewness and kurtosis values: (a) Normal distribution – Skewness-Kurtosis {0, 3}; (b) Skewness-Kurtosis{1, 0.5}; (c) Skewness-Kurtosis {1.5, 2.5}; (d) Skewness-Kurtosis{2, 5.5}.

Scenario	II				II			
	Skewness/Kurtosis				Skewness/Kurtosis			
Case	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
N	Mean(ϕ)				StdDev(ϕ)			
50	0,667	0,672	0,672	0,670	0,255	0,245	0,245	0,247
100	0,799	0,788	0,788	0,789	0,174	0,202	0,202	0,203
200	0,901	0,883	0,883	0,885	0,103	0,122	0,121	0,123
400	0,952	0,947	0,947	0,945	0,043	0,053	0,053	0,065
600	0,969	0,966	0,966	0,967	0,024	0,026	0,026	0,026

Table 3.7. Descriptive statistics of ϕ for GRA Scenario II for n= {50; 100; 200; 400; 600} in case of non-normal distribution, with increasing skewness and kurtosis values: (a) Normal distribution – Skewness-Kurtosis {0, 3}; (b) Skewness-Kurtosis{1, 0.5}; (c) Skewness-Kurtosis {1.5, 2.5}; (d) Skewness-Kurtosis{2, 5.5}.

The reasons why this happens are highlighted analyzing the biases of the estimates (Figures 3.7-3.8, Tables 3.8-3.9); generally satisfactory both in Scenario I and Scenario II, where the presence of the concomitant indicator affects some estimates peculiarly.

We in particular focused on the comparison of the obtained results between the normal scenario and the non-normal one.

What we observed (Figure 3.7, Table 3.8) is that, in Scenario I, with a covariate affecting just the endogenous variable y_1 :

- a. Non normality, thus a change in the values of skewness and kurtosis does not really affect **W** biases, which maintain low values (always below 16%) as skewness and kurtosis increase, having even lower values than the normal case biases for smaller n. Biases tend to converge at a common value for all the non-normality degrees. In particular, all **W** biases tend to have values below 6% for n > 300;

b. Without concomitant indicators, **A** biases are a little smaller than with the concomitant indicators ($\gamma(a_3)$ and $\gamma(a_4)$ are always $\sim 4\%$, regardless of n), which have the same values as a_3 and a_4 just for $n > 300$ ($\gamma(a_1)$ and $\gamma(a_2) > 3\%$ for $n = 300$). Finally, $\gamma(a_{y1})$ has values next to the zero, regardless of the skewness and of the sample size, indicating that the presence of the covariate affecting y_1 does not influence a_{y1} bias.

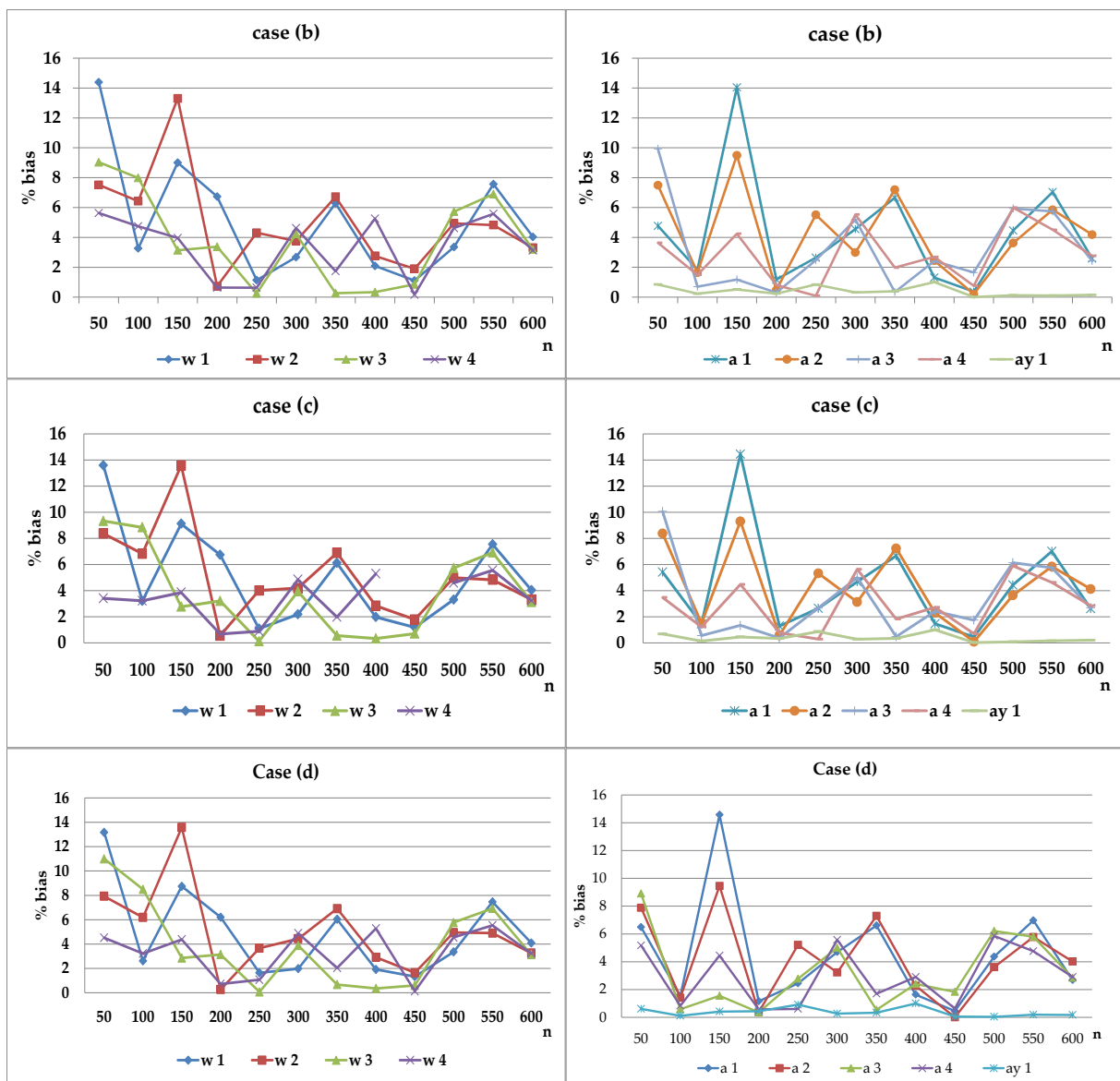


Figure 3.7. The trend of the biases, for a GRA Scenario I (a single concomitant indicator affecting a single endogenous variable), of both weight matrix **W** and loading matrix **A** is shown. From the top we find: case (b) - low skewed distributions (Skewness-Kurtosis[1, 0.5]); case (c) - medium skewed distributions (Skewness-Kurtosis {1.5, 2.5}) and, finally, case (d) - high skewed distributions (Skewness-Kurtosis {2, 5.5}).

With the addition of a concomitant indicator also affecting the LC f_1 (Scenario II), we still observe generally satisfactory performance of GRA model in the different non-normal scenarios, with a less fluctuating behaviour. The reasons why this happens are highlighted analyzing the biases of the estimates (Figure 3.8, Table 3.9):

- a. The presence of the concomitant indicator does not really affect **W** biases, which assume constant low values for $n > 200$. For small sample sizes instead, t_1 presence seems affecting **W** biases, in particular w_1 , w_2 and w_{T1} are affected by its presence, for which $\gamma(w_{T1}) \sim 50\%$ and $\gamma(w_1)$ and $\gamma(w_2) \sim 30\%$ for $n = 50$ in all non-normal scenarios;
- b. Similarly, **A** biases assume constant low values, below 20%, regardless of the sample size and the skewness/kurtosis values, thus indicating that the presence or not of the concomitant indicators does not really affect the loadings biases. The only parameter which seems to be affected by the presence of the concomitant indicator is a_1 for small sample size ($\gamma(a_1) \sim 140$ for $n = 50$).

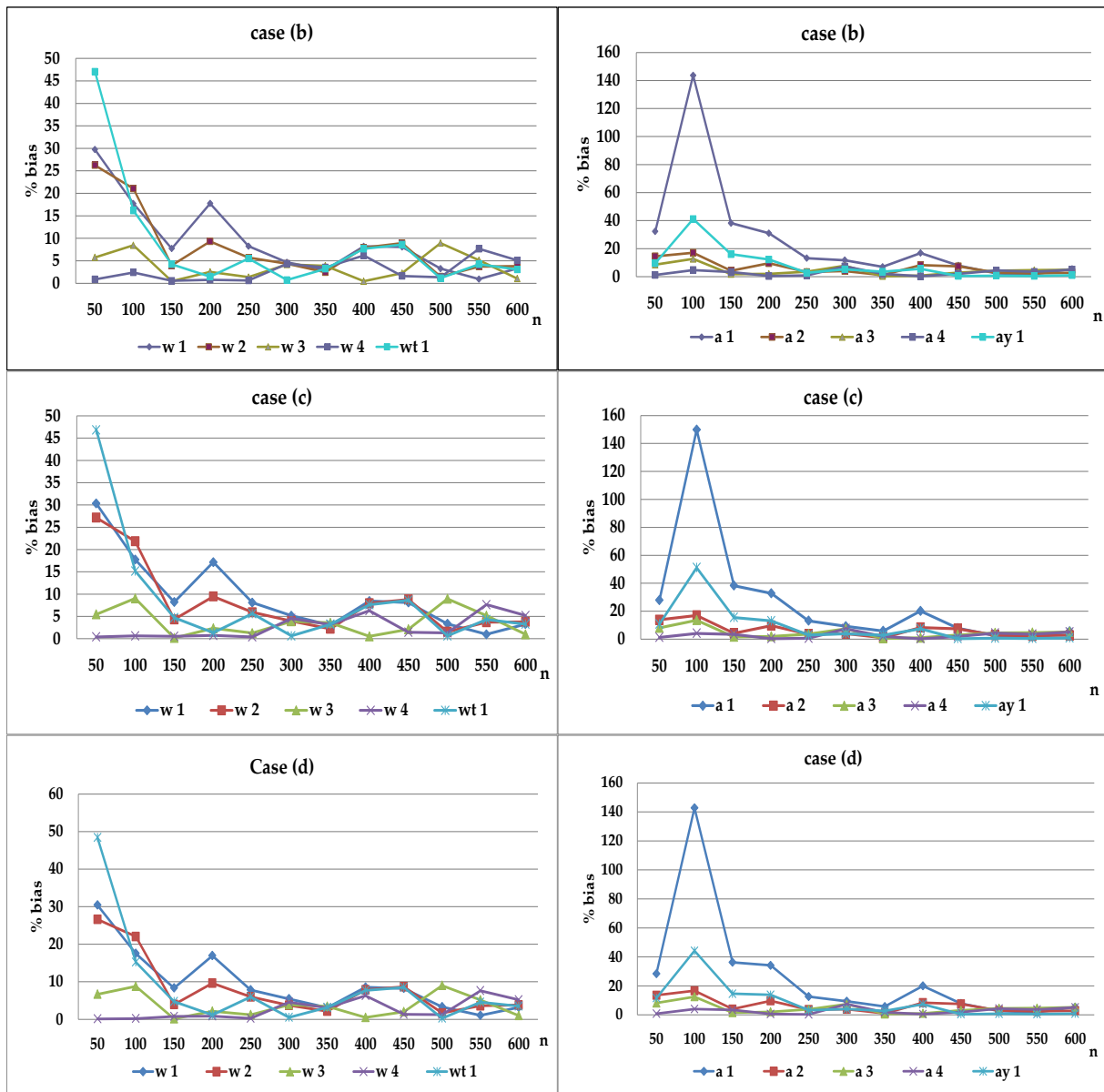


Figure 3.8. The trend of the biases, for a GRA Scenario II (a single concomitant indicator affecting a single endogenous variable and one LC), of both weight matrix W and loading matrix A is shown. From the top we find: case (b) - low skewed distributions (Skewness-Kurtosis {1, 0.5}), case (c) - medium skewed distributions (Skewness-Kurtosis {1.5, 2.5}) and, finally, case (d) - the high skewed distributions (Skewness-Kurtosis {2, 5.5}).

In particular, comparing the normal and non-normal scenarios, we can notice from the tables above (Tables 3.8-3.9), that several estimates with skewed errors provide lower biases than those with the normal scenario.

Focusing on Scenario I results (Table 3.8) $\gamma(w_1)$ is lower than $\gamma(w_1)$ in case (a) (Normal distribution), regardless of the sample size. A similar behavior is followed by $\gamma(w_2)$ and $\gamma(w_3)$ for $n = 200,400$, while $\gamma(w_4)$ does not seem to be positively affected by the degree of skewness. A biases instead are all positively affected by the non-normality (cases (b), (c), (d)), independently of the sample size.

In Scenario II, we have a similar behavior of the biases of both **W** and **A** matrices (Table 3.9). In particular the most interesting situations regard, for **W**, $\gamma(w_2)$, $\gamma(w_3)$ and $\gamma(w_{T1})$ and, for **A**, $\gamma(a1)$ $\gamma(a2)$ and $\gamma(a4)$, for which a high sample size shows lower biases in the skewed distributions (cases (b), (c), (d)). Thus the presence of the concomitant indicator also affecting the LC f_1 does not seem to determine such a behavior.

A possible reason at the basis of this could be a lower variability of the error, besides a higher concentration.

To notice that we have in this specific study chosen increasing and positive skewness values, but not much high values because a high skewness would have led us to take the lowest tails, which would have affected the significance of the results.

High skewness values, according to the Fleishman method, would have brought to extremely high kurtosis (e.g. Skewness = 5 \rightarrow Kurtosis \sim 39). Such a scenario would have influenced the performance of GRA, in the sense that missing values would have been generated, hindering the execution of GRA algorithm.

Thus the conducted study shows us how a change in the distribution of the manifest variables from being symmetric to an increasingly skewed distribution does not seem to affect the biases of the estimates: increasing the skewness maintains low biases (see Tables 3.8-3.9) in both Scenario I and Scenario II. Hence, for realistic situations, the GRA method seems to be reasonably robust also to deviations from symmetry.

CONCLUSION

A useful extension of both the basic CSA models and the PLS models for better prediction of the endogenous latent variables (LVs) is the accommodation of external covariates in the structural equation of the model (Lee and Shi, 2000; Lee and Song, 2003; Moustaki, 2003).

However, as both CSA and PLS are not suitable for the specification and the fitting of SEM-RA models with concomitant indicators, it has been tried to introduce them in the Redundancy Analysis (RA) framework (Van den Wollenberg, 1977).

Thus, after a brief introduction to the estimation techniques based on components, we discussed external covariate accommodation in the Redundancy Analysis framework, looking at the most known methods as classical RA, Multiblock Redundancy Analysis (MbRA: Bougeard, Qannari, Rose, 2011) and Extended Redundancy Analysis (ERA, Hwang and Takane, 2005) to finally focus on a recent proposal, namely Generalized Redundancy Analysis (GRA). In fact, GRA allows us to investigate the separate contributions of both blocks on the composite scores and also direct effects of observed variables and/or concomitant variables.

The major aim here has been to show how GRA behaves with external covariates, analyzing changes in biases and accuracy of the estimates in presence of different concomitant links. Hence I decided to perform a complete simulation study to assess the robustness of this estimation method and, in particular: (i) how GRA behave in presence of complex extensions?; (ii) how GRA behaves in case of multicollinearity?; (iii) how GRA behaves with non-normal distributed data?

For each study I have previously provided a detailed description of the model specification and then shown and discussed all the results.

In the first study, pointed toward the evaluation of GRA performances in three different path diagrams at increasing complexity, I started from a basic GRA model specification, with the covariate t_1 just affecting an endogenous variable, to then move towards the more complex scenarios, in which also a latent variable is involved, which has been our main interest. In particular the third scenario shows the influence of the concomitant indicator not only on one endogenous variable and one LC, but on the other endogenous variable too.

In order to further estimate the robustness of the Generalized Redundancy Analysis (GRA) procedure in the estimation of structural equation models, I have then tested GRA for the presence of correlations between \mathbf{X} and \mathbf{T} matrices, based on a scenario with both the LC and an endogenous variable affected by the external covariate.

In the first two studies however, it has been always assumed the data sampled from a normal population and this usually happens in analyzing data. On the other hand, it may happen, in the empirical research, to have to deal with data non-normally distributed. Hence the interest in analyzing GRA behaviour also in case of non-normal distributions, mainly focusing on the heteroskedasticity of the errors, testing both a basic scenario with just an endogenous variable affected by the external covariate and a scenario with both a LC and an endogenous variable affected, and this for three different degrees of kurtosis and skewness, comparing it the normal case.

All the conducted studies have confirmed a good behavior of GRA methodology in accommodating external covariates. In particular, in the second study it emerges that if exogenous and concomitant variables are correlated and each of them participate in the formation of the latent variable, the

correspondent parameters would be more bias affected. With larger n the bias in any case tends to “switch off”, it decreases, even if more slowly. Moreover, high degrees of correlation determine not so good performances of GRA algorithm. That’s why further work in this sense is needed: an attempt to lower biases in this sense would be by adding further exogenous variables to the latent composites.

In case of non-normality of the errors instead, it has emerged that a change in skewness and kurtosis parameters does not affect the performance of the model, but, comparing the normal and non-normal scenarios, several estimates with skewed errors provide lower biases than those with the normal scenario. This could be due to a lower variability of the error, besides a higher concentration, but this still remains to be proven.

I decided to undertake the research on this issue as no published empirical research has addressed the behavior of GRA with direct effects and concomitant indicators in previous simulation studies.

The relevance and the actuality of the present work are confirmed by the increasing attention towards the estimation techniques of Structural Equation Models, which have been highly introduced in literature in the last few years.

We can individuate different research path for the future development of the study in this field.

First of all another aspect to outline in the evaluation of GRA methodology in presence of covariate effects is also the presence of concomitant indicators simultaneously linked to different LCs.

A second future perspective of research regard the applicative field, as here we tested GRA for external covariates just with a simulation study, hence without any real data.

Moreover, comparing GRA to other structural model estimation methods, it appears that GRA is not devoid of drawbacks which entail further studies:

1. First of all measurement models for endogenous variables ($\mathbf{f} \rightarrow \mathbf{Y}$) need to be specified, as they are not allowed yet.
2. It would be worth to eliminate the mediation of the direct structural links between an exogenous and an endogenous LC by the formative indicator of the endogenous LCs (e.g., $f_1 \rightarrow \mathbf{X}_2 \rightarrow f_2$).

These issues can be faced only leaving Redundancy Analysis Framework for more general methods, as Generalized Structured Component Analysis (GSCA: Hwang and Takane, 2004) because Redundancy Analysis focus is on the achievement of linear composites of observed exogenous variables.

Finally, accordingly with traditional use of Component Analysis, we can look at GRA as a purely descriptive approach. However, as inferential issues are a growing area in the statistical research, further studies are needed to develop this field.

References

1. Bagozzi, R.P. and Fornell, C. (1982). *Theoretical concepts, measurements, and meaning*. In Fornell, C. (Ed.), *A Second Generation of Multivariate Analysis*, Vol. 1, Praeger, New York, NY, 24-38.
2. Blalock, H. (1964). *Causal Inferences in Nonexperimental Research*. University of North Carolina Press, Chapel Hill.
3. Blalock, H.M., Jr. (1971). *Causal models involving unmeasured variables in stimulus-response situations*. In "Causal Models in the Social Sciences" (H. M. Blalock, Jr., Ed.), chap. 19, Aldine-Atherton, Chicago.
4. Bollen, K. A., and Lennox, R. (1991). *Conventional Wisdom on Measurement: A Structural Equation Perspective*. *Psychological Bulletin*, 110(2): 305-314.
5. Boomsma, A., Hoogland, J.J. (2001). *The robustness of LISREL modeling revisited*. In: R. Cudeck, S. Du Toit and D. Sörbom (Eds), *Structural-equation modeling: Present and Future. A Festschrift in honor of Karl Jöreskog* (139-168). Chicago, IL: Scientific Software International.
6. Bougeard S., Qannari E.M., Lupo C., Hanafi M. (2011). *From multiblock partial least squares to multiblock redundancy analysis. A continuum approach*. *Informatica*; 22(1): 1–16.
7. Bougeard, S., Hanafi, M., Qannari, E.M. (2008). *Continuum redundancy PLS regression: a simple continuum approach. Application to epidemiological data*. *Computational Statistics and Data Analysis*, 52: 3686–3696.
8. Bougeard, S., Qannari, E. M., and Rose, N. (2011). *Multiblock Redundancy Analysis: interpretation tools and application in epidemiology*. *Journal of Chemometrics*, 25: 467-475.
9. Burnham, A.J., Viveros, R., Macgregor, J.F. (1996). *Framework for latent variable multivariate regression*. *Journal of Chemometrics*, 10: 31–45.

10. Chin W., Newsted, P. (1999). *Structural Equation Modeling Analysis with Small Samples Using Partial Least Squares*. In Rick Hoyle (Ed.), *Statistical Strategies for Small Sample Research*, Sage Publications, 307-341.
11. Costner, H. L. (1969). *Theory, deduction, and rules of correspondence*. *Am. J. Sociol.* 75: 245-63.
12. Croon, M., Bolck, A. (1997). *On the use of factor scores in structural equations models* (Tech. Rep. No. 97.10.102/7). Work and Organization Research Centre, Tilburg University. (WORC Paper). The Netherlands.
13. De Leeuw, J., Young, F. W., and Takane, Y. (1976). *Additive structure in qualitative data: An alternating least squares method with optimal scaling features*. *Psychometrika*, 41(4), 471–503.
14. Dijkstra, T. (1983). *Some comments on maximum likelihood and partial least squares methods*. *Journal of Econometrics*, 22: 67–90.
15. Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM, Philadelphia.
16. Efron, B., Tibshirani, R. (1998). *An Introduction to the Bootstrap*. Chapman and Hall, Boca Raton.
17. Fleishman, A. I. (1978). *A method for simulating non-normal distributions*. *Psychometrika*, 43(4): 521-532.
18. Fornell, C, Bookstein, F.L. (1982) *Two structural equation models: LISREL and PLS applied to consumer exit-voice theory*. *Journal of Marketing Research*, XIX, 440-452.
19. Fornell, C. and Cha, J. (1994). *Partial least squares*. In R. P. Bagozzi (Ed.), *Advanced methods of marketing research* (pp. 52–78). Cambridge, England: Blackwell.
20. Garthwite, P.H. (1994). *An interpretation of partial least squares*. *Journal of American Statistical Association*, 89: 122-127.
21. Goldberger, A. S. (1972). *Structural equation models in the social sciences*. *Econometrica*, 40: 979-99

22. Hanafi, M. (2007). *PLS path modelling: computation of latent variables with the estimation mode B*. *Comput. Stat.* 22(2): 275–292.
23. Henseler, J., Fassott, G. (2009). *Testing moderating effects in PLS path models: An illustration of available procedures*. In: V. Esposito Vinzi, W.W. Chin, J. Henseler and H. Wang (Eds.), *Handbook of Partial Least Squares: Concepts, Methods, and Applications* (713-735). Berlin: Springer.
24. Henseler, J., Sarstedt, M. (2013). *Goodness-of-fit indices for partial least squares path modeling*. *Computational Statistics*, 28 (2), 565-580
25. Hwang, H. (2000). *Structural equation modeling by extended redundancy analysis*. PhD thesis, November 2000, Department of Psychology, McGill University, Montreal.
26. Hwang, H., Takane, Y. (2002). *Structural Equation Modeling by Extended Redundancy Analysis*. *Measurement and Multivariate Analysis*, 115-124. Springer-Verlag.
27. Hwang, H., Takane, Y. (2004). *Generalized Structured Component Analysis*. *Psychometrika*, 69: 81-99.
28. Hwang, H., Takane, Y. (2005). *An Extended Redundancy Analysis and its Applications to Two Practical Examples*. *Computational Statistics and Data Analysis*, 49(3): 785-808.
29. Jöreskog, K.J. (1981). *Analysis of covariance structures*. *Scandinavian Journal of Statistics*, 8: 65-92.
30. Kiers, H.A.L., Takane, Y., ten Berge, J.M.F. (1996). *The analysis of multitrait - multimethod matrices via constrained components analysis*. *Psychometrika*, 61: 601-628.
31. Kiers, H.A.L., ten Berge, J.M.F. (1989). *Alternating least squares algorithm for Simultaneous Component Analysis with Equal Component Weight Matrices in Two or More Populations*. *Psychometrika*, 54: 467-473. The Psychometric Society.
32. Lambert Z. V., Wildt A. R., and Durand R. M. (1988). *Redundancy analysis: An alternative to canonical correlation and multivariate multiple regression in exploring interset associations*. *Psychological Bulletin*, 104: 282-289.

33. Lee, S. Y., Shi, J.Q. (2000). *Bayesian analysis of structural equation model with fixed covariates*. Structural Equation Modeling, 7: 411–430.
34. Lee, S.Y., Song, X.Y. (2003). *Model comparison of a nonlinear structural equation model with fixed covariates*. Psychometrika, 68: 27–47.
35. Lee, S.Y., Song, X.Y., Tang, N.S. (2007). *Bayesian methods for analyzing structural equation models with covariates, interaction, and quadratic latent variables*. Structural Equation Modeling: A Multidisciplinary Journal, 14(3): 404-434.
36. Lohmöller, J.B. (1989). *Latent variable path modelling with partial least squares*. Heidelberg, Germany: Physica Verlag.
37. Lorenzo-Seva, U. and ten Berge, J.M.F. (2006). *Tucker's Congruence Coefficient as a Meaningful Index of Factor Similarity*. Methodology, 2: 57-64.
38. Lovaglio, P.G., and Boselli, R., (2013). *Generalized Redundancy Analysis*. In Advances in Latent Variables (pp. 1-8). Vita e Pensiero.
39. Lovaglio, P.G., Vittadini, G. (2013). *Component analysis for structural equation models with concomitant indicators*. In Giudici, P., Ingrassia, S., Vichi, M. (eds.) Statistical Models for Data Analysis, (pp. 209-215), Springer.
40. Lovaglio, P.G., Vittadini, G. (2014). *Structural Equation Models in a Redundancy Analysis Framework with Covariates*. Multivariate Behavioral Research, 49(5): 486-501.
41. MacCallum, R. C., and Browne, M. W. (1993). *The use of causal indicators in covariance structure models: Some practical issues*. Psychological Bulletin, 114: 533-541.
42. McDonald, R.P. (1968). *A unified treatment of the weighting problem*. Psychometrika, 33: 351-381.
43. McDonald, R.P. (1996). *Path analysis with composite variables*. Multivariate Behavioral Research, 31: 239–270.
44. Meredith, W., Millsap, R.E. (1985). *On component analysis*. Psvchometrika, 50: 495-507.

45. Millsap, R.E., Meredith, W. (1988). *Component analysis in cross-sectional and longitudinal data*. *Psychometrika*, 53: 123-134.
46. Moustaki et al. (2004). *Factor models for ordinal variables with covariate effects on the manifest and latent variables: A comparison of LISREL and IRT approaches*. *Structural Equation Modeling*, 11(4): 487-513.
47. Moustaki, I. (2003). *A general class of latent variable methods for ordinal manifest variables with covariate effects on the manifest and latent variables*. *British Journal of Mathematical and Statistical Psychology*, 56: 337–357.
48. Muthén, B.O. (2002) *Beyond SEM: general latent variable modeling*. *Behaviormetrika*, 29(1): 81-117.
49. Pafundi, P.C. and Vacca, G. (2014). *Complex Redundancy Analysis models with covariate effect: a simulation study*. 47th SIS Scientific Meeting of the Italian Statistical Society, 2014. CUEC Cooperativa Universitaria Editrice Cagliari. ISBN: 978-88-8467-874-4. In press.
50. Reinsel, G. C., and R. P. Velu (1998). *Multivariate Reduced-Rank Regression*. Springer, New York.
51. Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer-Verlag, New York, second edition.
52. Sammel, M.D., Ryan, L.M. (1996). *Latent variables with fixed effects*. *Biometrics*, 52: 220-243.
53. SAS® software. <http://www.sas.com>
54. Schonemann, P.H. (1971). *The minimum average correlation between equivalent sets of uncorrelated factors*. *Psychometrika*, 36: 21-30.
55. Schonemann, P.H. and Steiger, J.H. (1976). *Regression component analysis*. *British Journal of Mathematical and Statistical Psychology*, 29: 175-189.
56. Tenenhaus, A. and Tenenhaus, M. (2011). *Regularized generalized canonical correlation analysis*. *Psychometrika*, 76(2): 257–284.
57. Tenenhaus, M., Esposito Vinzi, V., Chatelin, Y.M., Lauro, C. (2005). *PLS Path Modeling*. *Computational Statistics and Data Analysis*, 48: 159-205.

58. Vacca, G. (A.A. 2012-13). *Evaluation of economical impacts of Human Capital with Extended redundancy Analysis*. University of Milano Bicocca, Master Degree Thesis.
59. Van Den Wollenberg, A.L. (1977). *Redundancy Analysis: an Alternative for Canonical Analysis*. *Psychometrika*, 42: 207-219.
60. Velicer, W.F., Jackson, D.N. (1990). *Component analysis vs. common factor analysis: Some further observations*. *Multivariate Behavioral Research*, 25: 95-112.
61. Velu, R. P. (1991). *Reduced rank models with two sets of regressors*. *Applied Statistics*, 40: 159-170.
62. Vittadini G. et al., (2007). *On the relationships among latent variables and residuals in PLS path modeling: The formative-reflective scheme*. In *Computational Statistics & Data Analysis*, Elsevier B.V., 51(12): 5828-5846.
63. Vittadini, G. (1989). *Indeterminacy Problems in the Lisrel Model*. In *Multivariate Behavioral Research*, Forth Worth (Texas), 24(4): 397-414.
64. Wold, H. (1966). *Non-linear estimation by iterative least squares procedures*. In: David FN (Ed.) *Research papers in statistics*. Wiley, London, 411-444.
65. Wold, H. (1974). *Causal flows with latent variables: partings of the ways in the light of NIPALS modelling*. *Eur Econ Rev* 5(1): 67-86.
66. Wold, H. (1975). *Modelling in complex situations with soft information*. In *Third World Congress of Econometric Society*, Toronto, Canada.
67. Wold, H. (1982). *Soft Modeling: the Basic Design and Some Extensions*. In Jöreskog K., Wold H. (Eds.), *Systems Under Indirect Observation: Causality, Structure, Prediction*, pp. 1-54. North Holland.
68. Wold, H. (1985a). *Partial Least Squares*. In *Encyclopedia of Statistical Sciences*, vol. 6, 581-591. Kots S., Johnson N.L. (Eds). Wiley: New York.
69. Wold, H. (1985b). *Partial least squares and LISREL models*. In: Nijkamp, P., Leitner, H., Wrigley, N. (Eds) *Measuring the unmeasurable*. Nijhoff, Dordrecht, Boston, Lancaster, pp 220-251.

70. Wold, H. (1989). *Introduction to the second generation of multivariate analysis*. In: Wold H. (Ed) *Theoretical empiricism. A general rationale for scientific model-building*. Paragon House, New York, pp VIII–XL.
71. Wright, S. (1934). *The Method of Path Coefficients*. *The Annals of Mathematical Statistics* 5(3): 161-215.

APPENDIX

(In the present appendix we report the SAS Codes used in the simulation studies shown in the thesis.)

APPENDIX 1A– STUDY 1 (SCENARIO I)

```
/*INPUT: MATRICI DI VARIANZE E COVARIANZE DI X, T ED E E RISPETTIVI VETTORI DI
MEDIE*/
/*Le dimensioni devono essere conformi*/
options nomprint nonotes;
data meansx;
input x @@;
cards;
0 0 0 0
;
run;
data varcovx;
input row v1-v4;
cards;
1 1 0.3 0.1 0.1
2 0.3 1 0.1 0.1
3 0.1 0.1 1 0.3
4 0.1 0.1 0.3 1
;
run;

data meanst;
/*togliere un valore alla volta per arrivare da V1 a V2, V3, V4*/
/*es. V2: 0 0 0*/
input x @@;
cards;
0
;
run;
data varcovt;
input row v1;
/* diminuire il range e togliere una riga e colonna dalla matrice per passare
da
V1 a V2, V3, V4*/
/*es. V2: input row v1-v3;
cards;
1 1 0.1 0.1
2 0.1 1 0.1
3 0.1 0.1 1
;*/
cards;
1 1
;
run;

data meanse;
input x @@;
cards;
0 0
;
run;
data varcove;
input row v1-v2;
cards;
1 1 0.1
2 0.1 1
;
run;

/*INPUT: PATH DIAGRAM, Le dimensioni devono essere conformi alle matrici
dichiarate precedentemente
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*Parte esogena verso variabili latenti, X (6 variabili) verso latent
composites (3 componenti)
/*3 coefficienti
w1: x01 -> f1,
w2: x02 -> f1,
w3: x03 -> f2,
w4: x04 -> f2,
w5: x05 -> f3,
w6: x06 -> f3,
NB dim(X)=nrow(varcovx)*/
data dim;
  input vm $3. f01 f02;
cards ;
x01 1 0
x02 1 0
x03 0 1
x04 0 1
;
run ;

data dimt;
  input vm $3. y01 y02;
cards ;
t01 1 0
;
run ;

/*Parte verso variabili endogene, latent composites (2 componenti) verso Y (2
variabili)
/*4 coefficienti
a1: f1 -> y01
a2: f1 -> y02
a3: f2 -> y01
a4: f2 -> y02
a5: f3 -> y01*/
data dime;
  input ve $3. f01 f02;
cards ;
y01 1 1
y02 1 1
;
run ;

/*MACRO: simula il modello
Y=(X*W*AT)+(T*AY)+E;*/
/*Argomenti:
varcov e means: matrici varcov e vettori di medie di X, Y e T
values: valori delle matrici di coefficienti nella popolazione
rate1-rate4: ampiezze di campionamento,
replic: numero di repliche nei 4 campionamenti
iter: iterazioni dell'algoritmo, se 0 si usa criterio di accostamento per
farlo terminare*/

%macro mcsamples(varcovx, meansx,varcovt, meanst, varcove, meanse,valuew,
valuea,valueay, rate1,rate2,rate3,rate4,replic,iter,threshold);

%do r=1 %to 4;

proc iml;
/*PROCEDURE CHE VERRANNO UTILIZZATE NELL'ALGORITMO*/
/*START*/
start delrow(x,i);
/*CANCELLA RIGA DA MATRICE*/
return(x[setdif(1:nrow(x),i),]);
finish;
```



```
start delcol(x,i); /*CANCELLA COLONNA DA MATRICE*/
return(x[,setdif(1:ncol(x),i)]);
finish;

start normalize(x);
/*STANDARDIZZAZIONE (UTILIZZATO PER diag(F'F)=I)*/
p=ncol(x);
n=nrow(x);
mu=j(1,p,0);
mom2=j(1,p,0);
sigma=j(1,p,0);
one=j(n,1,1);
sigmaI=i(p);
do i=1 to p;
  sum=0; sum2=0;
  do j=1 to n;
    sum=sum+x[j,i];
    sum2=sum2+(x[j,i]**2);
  end;
  mu[i]=sum/n;
  mom2[i]=sum2/n;
  sigma[i]=1/(sqrt(mom2[i]-((mu[i])**2)));
  sigmaI[i,i]=sigma[i];
end;
xbar=(x-(one*mu))*sigmaI;
return(xbar);
finish;

start vectorise(x); /*OPERATORE VEC*/
k=1;
do while (k<=ncol(x));
  if k=1 then w=x[,k];
  else w=w//x[,k];
  k=k+1;
end;
return (w);
finish;

start resize2(x,a);
/*RIDIMENSIONAMENTO DELLE MATRICI CON DUPLICATI (UTILIZZATO PER A')*/
k=1; n=ncol(x); b=a; /*inizializzo indice, conto le colonne della matrice di
partenza e salvo in locale la matrice di destinazione*/
do while (k<=n); /*start ciclo esterno*/
  j=k+1;
  do while (j<=n); /*start ciclo interno parte dalla colonna successiva*/
    if any(x[,k]^=x[,j]) then j=j+1; /*se almeno un elemento della colonna è
diverto la colonna non è doppia, vado avanti*/
    else do;
      n=n-1; /*altrimenti parte la procedura di deduplica*/
      x=delcol(x,j);
      b[,k]=b[,k]+b[,j];
      b=delcol(b,j);
    end;
  end;
  k=k+1;
end;
return (b);
finish;

start resize1(x);
/*ridimensionamento di matrice, ritorna la stessa matrice ridimensionata.
Si usa per deduplicare le colonne di Z1*/
k=1; n=ncol(x);
do while (k<=n);
  j=k+1;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
do while (j<=n);
  if any(x[,k]^=x[,j]) then j=j+1;
  else do;
    n=n-1;
    x=delcol(x,j);
  end;
end;
k=k+1;
end;
return (x);
finish;

start update(X,y); /*AGGIORNAMENTO DELLE MATRICI A PARTIRE DAI VETTORI (NEI
PASSAGGI DI STIMA)*/
a=nrow(y);
n=nrow(X);
m=ncol(X);
Z=j(nrow(X),ncol(X),0);
k=1;
do while(k<=a);
  do j=1 to m;
    do i=1 to n;
      if X[i,j]^=0 then do;
        Z[i,j]=y[k]; k=k+1;
      end;
    end;
  end;
end;
return(Z);
finish;
/*PROCEDURE*/
/*END*/

use meansx;
read all into mX;

use meanst;
read all into mT;

use meanse;
read all into mE;

use varcovx;
read all into vX;
vX=delcol(vX,1);

use varcovt;
read all into vT;
vT=delcol(vT,1);

use varcove;
read all into vE;
vE=delcol(vE,1);

call randseed(4891);

Xk = RandNormal(&&rate&r,mX,vX);
Tk = RandNormal(&&rate&r,mT,vT);

X = repeat(Xk,&replic,1);
T = repeat(Tk,&replic,1);
E = RandNormal(&&rate&r*&replic,mE,vE);

sumW=0;
sumA=0;
```

```
sumAY=0;
sumWsqr=0;
sumAsqr=0;
sumAYsqr=0;
sumc=0;
vcongr&r=j(&replc,3,0);

%do k=1 %to &replc;
%PUT replicazione &k in esecuzione, n = &&rate&r;
/*W*/
use dim;
read all var _NUM_ into W;
W=W*(&valuew);

/*A*/
use dime;
read all var _NUM_ into A;
AT=t(A*(&valuea));

/*AY*/
use dimt;
read all var _NUM_ into AY;
AYT=(AY*(&valueay));

Xk=X[1+&&rate&r*(&k-1):&&rate&r*&k,];
Tk=T[1+&&rate&r*(&k-1):&&rate&r*&k,];
Ek=E[1+&&rate&r*(&k-1):&&rate&r*&k,];
Yk = (Xk*W*AT)+(Tk*AYT)+Ek;

/*QR decomposition di X*/
call qr(Q,Rx,piv,lindep,Xk);
Rxx=j(nrow(Xk)-nrow(Rx),ncol(Rx),0);
R=Rx//Rxx;

/*Ortogonalizzazione di T*/
TOrth=Tk-Xk*ginv(Xk)*Tk;

/*START ITERAZIONI*/
ciclo=&iter;
mustexit=0;
do while (mustexit=0);

oldAT=AT;
oldAYT=AYT;
oldW=W;
/*ALGORITMO*/

/*STEP 1: Stima w°*/
/*vec(W)*/
vw=vectorise(W);
A=t(AT);
/*KRONECHER PROD*/
QT=t(Q);
omegad=A@R;
/*tolgo il contributo del legame diretto*/
Ylc=Yk-Tk*AYT;

i=1;
do while(i<=nrow(vw)); /*cancello zeri da vec(W) e rispettive colonne da
omegad*/
if vw[i]^=0 then i=i+1;
else do;
vw=delrow(vw,i);
omegad=delcol(omegad,i);
end;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
end;

QTY=QT*Ylc;
vecQTY=vectorise(QTY);

/*stima del vettore di parametri w°*/
wupd=((inv(t(omegad)*omegad))*t(omegad))*vecQTY;
/*ricostruzione di W*/
W=update(W,wupd);

/*Costruzione di F e sua normalizzazione*/
F=Xk*W;
F=normalize(F);

/*STEP 3: Stima A*/
/*vec(A')*/
ad=vectorise(AT);
Id=I(ncol(AT));
/*KRONECHER PROD*/
gammad=Id@F;
vecYlc=vectorise(Ylc);

i=1;
do while(i<=nrow(ad)); /*cancello zeri da vec(A') e rispettive colonne da
gammad*/
if ad[i]^=0 then i=i+1;
else do;
ad=delrow(ad,i);
gammad=delcol(gammad,i);
end;
end;
/*stima del vettore di parametri a*/
aupd=((inv(t(gammad)*gammad))*t(gammad))*vecYlc;
/*ricostruzione di A'*/
AT=update(AT,aupd);
/*STEP 4: Stima AY*/
/*Sottraggo alla matrice endogena la costruzione tramite LC
F = X W + T WT*/
Y2=Yk-F*AT;

/*Regressione multipla con coefficienti fissi di T su Y*/
vay=vectorise(AYT);
thetad=Id@Tk;
vY2=vectorise(Y2);

i=1;
do while(i<=nrow(vay)); /*cancello zeri da vec(Ay') e rispettive colonne da
thetad*/
if vay[i]^=0 then i=i+1;
else do;
vay=delrow(vay,i);
thetad=delcol(thetad,i);
end;
end;

/*stima del vettore di parametri ay*/
ayupd=((inv(t(thetad)*thetad))*t(thetad))*vY2;
/*ricostruzione di Ay*/
AYT=update(AYT,ayupd);

/*ALGORITMO END*/

/*Criterio di accostamento*/
diffW=abs(W-oldW);
diffAT=abs(AT-oldAT);
```

```
diffAYT=abs (AYT-oldAYT);

if &iter^=0 then do;

flgw=1;flgat=1; flgay=1;
  do i=1 to nrow(diffW);
  do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
  end;
end;
do i=1 to nrow(diffAT);
do j=1 to ncol(diffAT);
  if diffAT[i,j]>&threshold then flgat=0;
end;
end;
do i=1 to nrow(diffAYT);
do j=1 to ncol(diffAYT);
  if diffAYT[i,j]>&threshold then flgay=0;
end;
end;
ciclo=ciclo-1;
if ciclo=0 | (flgw=1 & flgat=1 & flgay=1) then mustexit=1;
end;

if &iter=0 then do;
flgw=1;flgat=1; flgay=1;
  do i=1 to nrow(diffW);
  do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
  end;
end;
do i=1 to nrow(diffAT);
do j=1 to ncol(diffAT);
  if diffAT[i,j]>&threshold then flgat=0;
end;
end;
do i=1 to nrow(diffAYT);
do j=1 to ncol(diffAYT);
  if diffAYT[i,j]>&threshold then flgay=0;
end;
end;
if flgw=1 & flgat=1 & flgay=1 then mustexit=1;
end;

end;

A=t(AT);
AY=t(AYT);

vW=vectorise(W);
vA=vectorise(A);
vAY=vectorise(AY);

i=1;
do while(i<=nrow(vW));
  if vW[i]^=0 then i=i+1;
else do;
  vW=delrow(vW,i);
end;
end;

i=1;
do while(i<=nrow(vA));
  if vA[i]^=0 then i=i+1;
else do;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
vA=delrow(vA,i);
end;
end;

i=1;
do while(i<=nrow(vAY));
  if vAY[i]^=0 then i=i+1;
else do;
  vAY=delrow(vAY,i);
end;
end;

sumW=sumW+vW;
sumA=sumA+vA;
sumAY=sumAY+vAY;
sumWsq=sumWsq+(vW#vW);
sumAsq=sumAsq+(vA#vA);
sumAYsq=sumAYsq+(vAY#vAY);
vpW=vW*0+&valuew;
vpA=vA*0+&valuea;
vpAY=vAY*0+&valueay;

vecmcoef=vW//vA;
vecmcoef=vecmcoef//vAY;
vecp=vpW//vpA;
vecp=vecp//vpAY;

congr=(t(vecp)*vecmcoef)/((sqrt(t(vecp)*vecp))*sqrt(t(vecmcoef)*vecmcoef));

vcongr&r[&k,1]=&k;
vcongr&r[&k,2]=&&rate&r;
vcongr&r[&k,3]=congr;

sumc=sumc+congr;

%end;

MeanvW=sumW/&replc;
MeanvA=sumA/&replc;
MeanvAY=sumAY/&replc;

MeanvWsq=sumWsq/&replc;
MeanvAsq=sumAsq/&replc;
MeanvAYsq=sumAYsq/&replc;

seW=((MeanvWsq-(MeanvW#MeanvW))*((&replc)/(&replc-1)))##0.5;
seA=((MeanvAsq-(MeanvA#MeanvA))*((&replc)/(&replc-1)))##0.5;
seAY=((MeanvAYsq-(MeanvAY#MeanvAY))*((&replc)/(&replc-1)))##0.5;

estimates = MeanvW//MeanvA;
estimates = estimates//MeanvAY;

stderr = seW//seA;
stderr = stderr//seAY;

meancongr=sumc/&replc;
biases=100*abs(estimates-vecp)/vecp;
outputM&&rate&r=estimates||stderr;
outputM&&rate&r=outputM&&rate&r||biases;
num={&&rate&r 0 0};
congruence=meancongr*{1 0 0};
outputM&&rate&r=num//outputM&&rate&r;
outputM&&rate&r=outputM&&rate&r//congruence;
outputM&&rate&r=round(outputM&&rate&r,0.001);
names={'est' 'stderr' 'bias'};
```

```
trail=j(1,3,char(&&rate&r));
collabel=compbl(names + trail);

create results&r from outputM&&rate&r [colname = collabel];
append from outputM&&rate&r;

create results_rep&r from vcongr&r [colname = {'rep' 'n' 'congr'}];
append from vcongr&r;

%put n= &&rate&r eseguita;

%end;
leftcW=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="w"+char(j);
chj=COMPBL(chj);
leftcW[j]=chj;
end;
leftcA=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="a"+char(j);
chj=COMPBL(chj);
leftcA[j]=chj;
end;
leftcAY=j(nrow(meanvAY),1,"xxxx");
do j=1 to nrow(meanvAY);
chj="ay"+char(j);
chj=COMPBL(chj);
leftcAY[j]=chj;
end;

parm="n"//leftcW;
parm=parm//leftcA;
parm=parm//leftcAY;
parm=parm//"congr";

pop=0//vecp;
pop=pop//0;

create results from parm [colname = {'label'}];
append from parm;

create pop from pop [colname = {'pop'}];
append from pop;

data results;
merge results pop;
run;

data results;
merge results results1;
run;
data results;
merge results results2;
run;
data results;
merge results results3;
run;
data results;
merge results results4;
run;

data Res_Simul_GRAEXT; set results;
array change _numeric_;
do over change;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
if change=0 then change="";
end;
run;

data Res_Simul_GRAEXT_rep;
set Results_rep1 Results_rep2 Results_rep3 Results_rep4; run;

proc datasets;
save Res_Simul_GRAEXT Res_Simul_GRAEXT_rep;run;

proc univariate data=Res_Simul_GRAEXT_rep;
var congr;
by n;
run;

%mend;
quit;

%mcsamples(varcovx, meansx, varcovt, meanst, varcove, meanse, 0.6, 0.2, 0.4,
50, 100, 200, 400, 1000, 10000, 0.0001);
```


APPENDIX 1B – STUDY 1 (SCENARIOS II/III)

```
/*INPUT: MATRICI DI VARIANZE E COVARIANZE DI X, T ED E E RISPETTIVI VETTORI DI
MEDIE*/
/*Le dimensioni devono essere conformi*/
options nomprint nonotes;
data meansx;
input x @@;
cards;
0 0 0 0
;
run;
data varcovx;
input row v1-v4;
cards;
1 1 0.3 0.1 0.1
2 0.3 1 0.1 0.1
3 0.1 0.1 1 0.3
4 0.1 0.1 0.3 1
;
run;

data meanst;
/*togliere un valore alla volta per arrivare da V1 a V2, V3, V4*/
/*es. V2: 0 0 0*/
input x @@;
cards;
0
;
run;
data varcovt;
input row v1;
/* diminuire il range e togliere una riga e colonna dalla matrice per passare
da
V1 a V2, V3, V4*/
/*es. V2: input row v1-v3;
cards;
1 1 0.1 0.1
2 0.1 1 0.1
3 0.1 0.1 1
;*/
cards;
1 1
;
run;

data meanse;
input x @@;
cards;
0 0
;
run;
data varcove;
input row v1-v2;
cards;
1 1 0.1
2 0.1 1
;
run;

/*INPUT: PATH DIAGRAM, Le dimensioni devono essere conformi alle matrici
dichiarate precedentemente
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*Parte esogena verso variabili latenti, X (6 variabili) verso latent
composites (3 componenti)
/*3 coefficienti
w1: x01 -> f1,
w2: x02 -> f1,
w3: x03 -> f2,
w4: x04 -> f2,
w5: x05 -> f3,
w6: x06 -> f3,
NB dim(X)=nrow(varcovx)*/
data dim;
  input vm $3. f01 f02;
cards ;
x01 1 0
x02 1 0
x03 0 1
x04 0 1
;
run ;

data dimc;
  input vm $3. f01 f02;
cards ;
t01 1 0
;
run ;

data dimt;
  input vm $3. y01 y02;
cards ;
t01 1 1
;
run ;

/*Parte verso variabili endogene, latent composites (2 componenti) verso Y (2
variabili)
/*4 coefficienti
a1: f1 -> y01
a2: f1 -> y02
a3: f2 -> y01
a4: f2 -> y02
a5: f3 -> y01*/
data dime;
  input ve $3. f01 f02;
cards ;
y01 1 1
y02 1 1
;
run ;

/*MACRO: simula il modello
Y=(X*W*AT)+(T*AY)+E;*/
/*Argomenti:
varcov e means: matrici varcov e vettori di medie di X, Y e T
values: valori delle matrici di coefficienti nella popolazione
rate1-rate4: ampiezze di campionamento,
replic: numero di repliche nei 4 campionamenti
iter: iterazioni dell'algoritmo, se 0 si usa criterio di accostamento per
farlo terminare*/
%macro mcsamples(varcovx, meansx,varcovt, meanst, varcove,
meanse,valuew,valuewt, valuea,valueay,
rate1,rate2,rate3,rate4,replic,iter,threshold);
```

```
%do r=1 %to 4;

proc iml;
/*PROCEDURE CHE VERRANNO UTILIZZATE NELL'ALGORITMO*/
/*START*/
start delrow(x,i);
/*CANCELLA RIGA DA MATRICE*/
return(x[setdif(1:nrow(x),i),]);
finish;

start delcol(x,i); /*CANCELLA COLONNA DA MATRICE*/
return(x[,setdif(1:ncol(x),i)]);
finish;

start normalize(x);
/*STANDARDIZZAZIONE (UTILIZZATO PER diag(F'F)=I)*/
p=ncol(x);
n=nrow(x);
mu=j(1,p,0);
mom2=j(1,p,0);
sigma=j(1,p,0);
one=j(n,1,1);
sigmaI=i(p);
do i=1 to p;
  sum=0; sum2=0;
  do j=1 to n;
    sum=sum+x[j,i];
    sum2=sum2+(x[j,i]**2);
  end;
  mu[i]=sum/n;
  mom2[i]=sum2/n;
  sigma[i]=1/(sqrt(mom2[i]-((mu[i])**2)));
  sigmaI[i,i]=sigma[i];
end;
xbar=(x-(one*mu))*sigmaI;
return(xbar);
finish;

start vectorise(x); /*OPERATORE VEC*/
k=1;
do while (k<=ncol(x));
  if k=1 then w=x[,k];
  else w=w//x[,k];
  k=k+1;
end;
return (w);
finish;

start resize2(x,a);
/*RIDIMENSIONAMENTO DELLE MATRICI CON DUPLICATI (UTILIZZATO PER A')*/
k=1; n=ncol(x); b=a; /*inizializzo indice, conto le colonne della matrice di
partenza e salvo in locale la matrice di destinazione*/
do while (k<=n); /*start ciclo esterno*/
j=k+1;
do while (j<=n); /*start ciclo interno parte dalla colonna successiva*/
  if any(x[,k]^x[,j]) then j=j+1; /*se almeno un elemento della colonna è
diverto la colonna non è doppia, vado avanti*/
  else do;
    n=n-1; /*altrimenti parte la procedura di deduplica*/
    x=delcol(x,j);
    b[,k]=b[,k]+b[,j];
    b=delcol(b,j);
  end;
end;
k=k+1;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
end;
return (b);
finish;

start resize1(x);
/*ridimensionamento di matrice, ritorna la stessa matrice ridimensionata.
Si usa per deduplicare le colonne di Z1*/
k=1; n=ncol(x);
do while (k<=n);
j=k+1;
do while (j<=n);
if any(x[,k]^=x[,j]) then j=j+1;
else do;
n=n-1;
x=delcol(x,j);
end;
end;
k=k+1;
end;
return (x);
finish;

start update(X,y); /*AGGIORNAMENTO DELLE MATRICI A PARTIRE DAI VETTORI (NEI
PASSAGGI DI STIMA)*/
a=nrow(y);
n=nrow(X);
m=ncol(X);
Z=j(nrow(X),ncol(X),0);
k=1;
do while(k<=a);
do j=1 to m;
do i=1 to n;
if X[i,j]^=0 then do;
Z[i,j]=y[k]; k=k+1;
end;
end;
end;
return(Z);
finish;
/*PROCEDURE*/
/*END*/

use meansx;
read all into mX;

use meanst;
read all into mT;

use meanse;
read all into mE;

use varcovx;
read all into vX;
vX=delcol(vX,1);

use varcovt;
read all into vT;
vT=delcol(vT,1);

use varcove;
read all into vE;
vE=delcol(vE,1);

call randseed(4891);
```

```
Xk = RandNormal (&&rate&r,mX,vX);
Tk = RandNormal (&&rate&r,mT,vT);

X = repeat (Xk,&replc,1);
T = repeat (Tk,&replc,1);
E = RandNormal (&&rate&r*&replc,mE,vE);

sumW=0;
sumWT=0;
sumA=0;
sumAY=0;
sumWsqr=0;
sumWTsq=0;
sumAsqr=0;
sumAYsq=0;
sumc=0;
vcongr&r=j (&replc,3,0);

%do k=1 %to &replc;
%PUT replicazione &k in esecuzione, n = &&rate&r;
/*W*/
use dim;
read all var _NUM_ into W;
W=W* (&valuew);

/*WT*/
use dimc;
read all var _NUM_ into WT;
WT=WT* (&valuewt);

/*A*/
use dime;
read all var _NUM_ into A;
AT=t (A* (&valuea));

/*AY*/
use dimt;
read all var _NUM_ into AY;
AYT=(AY* (&valueay));

Xk=X[1+&&rate&r* (&k-1):&&rate&r*&k,];
Tk=T[1+&&rate&r* (&k-1):&&rate&r*&k,];
Ek=E[1+&&rate&r* (&k-1):&&rate&r*&k,];
Yk = (Xk*W*AT) + (Tk*WT*AT) + (Tk*AYT) + Ek;

/*QR decomposition di X*/
call qr (Q,Rx,piv,lndep,Xk);
Rxx=j (nrow (Xk) -nrow (Rx) ,ncol (Rx) ,0);
R=Rx/Rxx;
/*Ortogonalizzazione di T*/
TOrth=Tk-Xk*ginv (Xk) *Tk;

/*START ITERAZIONI*/
ciclo=&iter;
mustexit=0;
do while (mustexit=0);

oldAT=AT;
oldAYT=AYT;
oldW=W;
oldWT=WT;

/*ALGORITMO*/
/*STEP 1: Stima w°*/
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*vec(W)*/
vw=vectorise(W);
A=t(AT);
/*KRONECHER PROD*/
QT=t(Q);
omegad=A@R;
/*tolgo il contributo del legame diretto*/
Ylc=Yk-Tk*AYT;

i=1;
do while(i<=nrow(vw)); /*cancello zeri da vec(W) e rispettive colonne da
omegad*/
  if vw[i]^=0 then i=i+1;
else do;
  vw=delrow(vw,i);
  omegad=delcol(omegad,i);
end;
end;

QTY=QT*Ylc;
vecQTY=vectorise(QTY);

/*stima del vettore di parametri w°*/
wupd=(inv(t(omegad)*omegad))*t(omegad))*vecQTY;
/*ricostruzione di W*/
W=update(W,wupd);

/*STEP 2: Stima Wt*/
/*vec(WT)*/
vwt=vectorise(WT);
A=t(AT);
/*KRONECHER PROD*/
omegadt=A@Torth;

/*tolgo il legame diretto e il legame tramite LC delle variabili esogene X*/
Ylc2=Ylc-(Xk*W*AT);

i=1;
do while(i<=nrow(vwt)); /*cancello zeri da vec(WT) e rispettive colonne da
omegadt*/
  if vwt[i]^=0 then i=i+1;
else do;
  vwt=delrow(vwt,i);
  omegadt=delcol(omegadt,i);
end;
end;

vecYlc2=vectorise(Ylc2);
/*stima del vettore di parametri wt*/
wtupd=(inv(t(omegadt)*omegadt))*t(omegadt))*vecYlc2;
/*ricostruzione di WT*/
WT=update(WT,wtupd);
/*Costruzione di F e sua normalizzazione*/
F=Xk*W+Torth*WT;;
F=normalize(F);

/*STEP 3: Stima A*/
/*vec(A')*/
ad=vectorise(AT);
Id=I(ncol(AT));
/*KRONECHER PROD*/
gammad=Id@F;
vecYlc=vectorise(Ylc);

i=1;
```

```
do while(i<=nrow(ad)); /*cancello zeri da vec(A') e rispettive colonne da
gammad*/
if ad[i]^=0 then i=i+1;
else do;
  ad=delrow(ad,i);
  gammad=delcol(gammad,i);
end;
end;
/*stima del vettore di parametri a*/
aupd=((inv(t(gammad)*gammad))*t(gammad))*vecYlc;
/*ricostruzione di A'*/
AT=update(AT,aupd);

/*STEP 4: Stima AY*/
/*Sottraggo alla matrice endogena la costruzione tramite LC
 $F = XW + TWT$ */
Y2=Yk-F*AT;

/*Regressione multipla con coefficienti fissi di T su Y*/
vay=vectorise(AYT);
thetad=Id@Tk;
vY2=vectorise(Y2);

i=1;
do while(i<=nrow(vay)); /*cancello zeri da vec(Ay') e rispettive colonne da
thetad*/
if vay[i]^=0 then i=i+1;
else do;
  vay=delrow(vay,i);
  thetad=delcol(thetad,i);
end;
end;

/*stima del vettore di parametri ay*/
ayupd=((inv(t(thetad)*thetad))*t(thetad))*vY2;
/*ricostruzione di Ay*/
AYT=update(AYT,ayupd);

/*ALGORITMO END*/
/*Criterio di accostamento*/
diffW=abs(W-oldW);
diffWT=abs(WT-oldWT);
diffAT=abs(AT-oldAT);
diffAYT=abs(AYT-oldAYT);

if &iter^=0 then do;

flgw=1;flgat=1;flgwt=1; flgay=1;
do i=1 to nrow(diffW);
do j=1 to ncol(diffW);
  if diffW[i,j]>&threshold then flgw=0;
end;
end;
do i=1 to nrow(diffWT);
do j=1 to ncol(diffWT);
  if diffWT[i,j]>&threshold then flgwt=0;
end;
end;
do i=1 to nrow(diffAT);
do j=1 to ncol(diffAT);
  if diffAT[i,j]>&threshold then flgat=0;
end;
end;
do i=1 to nrow(diffAYT);
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```

do j=1 to ncol(diffAYT);
    if diffAYT[i,j]>&threshold then flgay=0;
end;
end;
ciclo=ciclo-1;
if ciclo=0 | (flgw=1 & flgat=1 & flgwt=1 & flgay=1) then mustexit=1;
end;
if &iter=0 then do;
flgw=1;flgat=1;flgwt=1; flgay=1;
do i=1 to nrow(diffW);
do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
end;
end;
do i=1 to nrow(diffWT);
do j=1 to ncol(diffWT);
    if diffWT[i,j]>&threshold then flgwt=0;
end;
end;
do i=1 to nrow(diffAT);
do j=1 to ncol(diffAT);
    if diffAT[i,j]>&threshold then flgat=0;
end;
end;
do i=1 to nrow(diffAYT);
do j=1 to ncol(diffAYT);
    if diffAYT[i,j]>&threshold then flgay=0;
end;
end;
if flgw=1 & flgat=1 & flgwt=1 & flgay=1 then mustexit=1;
end;

end;

A=t(AT);
AY=t(AYT);

/*Back Transformation a W da W°*/
XT_BT=(ginv(Xk))*Tk; /*X*T*/
vecWTr=vectorise(WT); /*vec(Wt)*/
vecWr=vectorise(W); /*vec(W°)*/
DiagW=I(ncol(W));
BTr=DiagW@XT_BT; /*I x (X*T), "x" inteso come prodotto di kronecker*/

i=1;
do while(i<=nrow(vecWr));
if vecWr[i]^=0 then i=i+1;
else do;
    vecWr=delrow(vecWr,i); /*cancello righe nulle da vec(W°)*/
    BTr=delrow(BTr,i); /*cancello corrispettive RIGHE da I x (X*T)*/
end;
end;

i=1;
do while(i<=nrow(vecWTr));
if vecWTr[i]^=0 then i=i+1;
else do;
    vecWTr=delrow(vecWTr,i); /*cancello righe nulle da vec(W°)*/
    BTr=delcol(BTr,i); /*cancello corrispettive COLONNE da I x (X*T)*/
end;
end;
end;
vecWnew=vecWr-(BTr)*vecWTr; /*costruisco W con back transformation in forma
vettorizzata*/
W=update(W,vecWnew); /*rimetto i coefficienti al posto corretto nella W back-
transformation*/

```



```
vW=vectorise(W);
vWT=vectorise(WT);
vA=vectorise(A);
vAY=vectorise(AY);

i=1;
do while(i<=nrow(vW));
  if vW[i]^=0 then i=i+1;
else do;
  vW=delrow(vW,i);
end;
end;

i=1;
do while(i<=nrow(vWT));
  if vWT[i]^=0 then i=i+1;
else do;
  vWT=delrow(vWT,i);
end;
end;

i=1;
do while(i<=nrow(vA));
  if vA[i]^=0 then i=i+1;
else do;
  vA=delrow(vA,i);
end;
end;

i=1;
do while(i<=nrow(vAY));
  if vAY[i]^=0 then i=i+1;
else do;
  vAY=delrow(vAY,i);
end;
end;

sumW=sumW+vW;
sumWT=sumWT+vWT;
sumA=sumA+vA;
sumAY=sumAY+vAY;
sumWsq=sumWsq+(vW#vW);
sumWTsq=sumWTsq+(vWT#vWT);
sumAsq=sumAsq+(vA#vA);
sumAYsq=sumAYsq+(vAY#vAY);
vpW=vW*0+&valuew;
vpWT=vWT*0+&valuewt;
vpA=vA*0+&valuea;
vpAY=vAY*0+&valueay;

vecmcoef=vW//vWT;
vecmcoef=vecmcoef//vA;
vecmcoef=vecmcoef//vAY;
vecp=vpW//vpWT;
vecp=vecp//vpA;
vecp=vecp//vpAY;

congr=(t(vecp)*vecmcoef)/((sqrt(t(vecp)*vecp))*(sqrt(t(vecmcoef)*vecmcoef)));
vcongr&r[&k,1]=&k;
vcongr&r[&k,2]=&rate&r;
vcongr&r[&k,3]=congr;
sumc=sumc+congr;

%end;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
MeanvW=sumW/&replc;
MeanvWT=sumWT/&replc;
MeanvA=sumA/&replc;
MeanvAY=sumAY/&replc;

MeanvWsq=sumWsq/&replc;
MeanvWTsq=sumWTsq/&replc;
MeanvAsq=sumAsq/&replc;
MeanvAYsq=sumAYsq/&replc;

seW=( (MeanvWsq- (MeanvW#MeanvW)) * ((&replc)/(&replc-1)))##0.5;
seWT=( (MeanvWTsq- (MeanvWT#MeanvWT)) * ((&replc)/(&replc-1)))##0.5;
seA=( (MeanvAsq- (MeanvA#MeanvA)) * ((&replc)/(&replc-1)))##0.5;
seAY=( (MeanvAYsq- (MeanvAY#MeanvAY)) * ((&replc)/(&replc-1)))##0.5;

estimates = MeanvW//MeanvWT;
estimates = estimates//MeanvA;
estimates = estimates//MeanvAY;

stderr = seW//seWT;
stderr = stderr//seA;
stderr = stderr//seAY;

meancongr=sumc/&replc;
biases=100*abs(estimates-vecp)/vecp;
outputM&&rate&r=estimates||stderr;
outputM&&rate&r=outputM&&rate&r||biases;
num={&&rate&r 0 0 };
congruence=meancongr*{1 0 0};
outputM&&rate&r=num//outputM&&rate&r;
outputM&&rate&r=outputM&&rate&r//congruence;
outputM&&rate&r=round(outputM&&rate&r,0.001);

names={'est' 'stderr' 'bias'};
trail=j(1,3,char(&&rate&r));
collabel=compbl(names + trail);

create results&r from outputM&&rate&r [colname = collabel];
append from outputM&&rate&r;

create results_rep&r from vcongr&r [colname = {'rep' 'n' 'congr'}];
append from vcongr&r;

%put n= &&rate&r eseguita;

%end;
leftcW=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="w"+char(j);
chj=COMPBL(chj);
leftcW[j]=chj;
end;
leftcWT=j(nrow(meanvWT),1,"xxxx");
do j=1 to nrow(meanvWT);
chj="wt"+char(j);
chj=COMPBL(chj);
leftcWT[j]=chj;
end;
leftcA=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="a"+char(j);
chj=COMPBL(chj);
leftcA[j]=chj;
end;
leftcAY=j(nrow(meanvAY),1,"xxxx");
```

```
do j=1 to nrow(meanvAY);
  chj="ay"+char(j);
  chj=COMPBL(chj);
  leftcAY[j]=chj;
end;

parm="n"//leftcW;
parm=parm//leftcWT;
parm=parm//leftcA;
parm=parm//leftcAY;
parm=parm//"congr";

pop=0//vecp;
pop=pop//0;

create results from parm [colname = {'label'}];
append from parm;
create pop from pop [colname = {'pop'}];
append from pop;

data results;
merge results pop;
run;
data results;
merge results results1;
run;
data results;
merge results results2;
run;
data results;
merge results results3;
run;
data results;
merge results results4;
run;

data Res_Simul_GRA; set results;
array change _numeric_;
do over change;
if change=0 then change="";
end;
run;

data Res_Simul_GRA_rep;
set Results_rep1 Results_rep2 Results_rep3 Results_rep4; run;

proc datasets;
save Res_Simul_GRA Res_Simul_GRA_rep;run;
proc univariate data=Res_Simul_GRA_rep;
var congr;
by n;
run;

%mend;
quit;

%mcsamples(varcovx, meansx, varcovt, meanst,varcove, meanse,0.6,0.3,0.2,0.4,
50,100,200,400,1000,10000,0.0003);
```

APPENDIX 2 – STUDY 2

```
/*INPUT: MATRICI DI VARIANZE E COVARIANZE DI X, T ED E E RISPETTIVI VETTORI DI
MEDIE*/
/*Le dimensioni devono essere conformi*/
options nomprint nonotes;
data meansxt;
input x @@;
cards;
0 0 0 0 0
;
run;
data varcovxt;
input row v1-v5;
cards;
1 1 0.3 0.1 0.1 0
2 0.3 1 0.1 0.1 0
3 0.1 0.1 1 0.3 0.7
4 0.1 0.1 0.3 1 0.7
5 0 0 0.7 0.7 1
;
run;

data meanse;
input x @@;
cards;
0 0
;
run;
data varcove;
input row v1-v2;
cards;
1 1 0.1
2 0.1 1
;
run;

/*INPUT: PATH DIAGRAM, Le dimensioni devono essere conformi alle matrici
dichiarate precedentemente
/*Parte esogena verso variabili latenti, X (6 variabili) verso latent
composites (3 componenti)
/*3 coefficienti
w1: x01 -> f1,
w2: x02 -> f1,
w3: x03 -> f2,
w4: x04 -> f2,
w5: x05 -> f3,
w6: x06 -> f3,
NB dim(X)=nrow(varcovx)*/
data dim;
input vm $3. f01 f02;
cards ;
x01 1 0
x02 1 0
x03 0 1
x04 0 1
;
run ;

data dimc;
input vm $3. f01 f02;
cards ;
```

```
t01 1 0
;
run ;
data dimt;
  input vm $3. y01 y02;
cards ;
t01 1 0
;
run ;

/*Parte verso variabili endogene, latent composites (2 componenti) verso Y (2
variabili)
/*4 coefficienti
a1: f1 -> y01
a2: f1 -> y02
a3: f2 -> y01
a4: f2 -> y02
a5: f3 -> y01*/
data dime;
  input ve $3. f01 f02;
cards ;
y01 1 1
y02 1 1
;
run ;

/*MACRO: simula il modello
Y=(X*W*AT)+(T*AY)+E;*/
/*Argomenti:
varcov e means: matrici varcov e vettori di medie di X, Y e T
values: valori delle matrici di coefficienti nella popolazione
rate1-rate4: ampiezze di campionamento,
replic: numero di replicazioni nei 4 campionamenti
iter: iterazioni dell' algoritmo, se 0 si usa criterio di accostamento per
farlo terminare*/

%macro mcsamples(varcovxt, meansxt, varcove, meanse, nx, valuew, valuewt,
valuea, valueay, rate1, rate2, rate3, rate4, replic, iter, threshold);

%do r=1 %to 4;

proc iml;
/*PROCEDURE CHE VERRANNO UTILIZZATE NELL'ALGORITMO*/
/*START*/
start delrow(x,i);
/*CANCELLA RIGA DA MATRICE*/
return(x[setdif(1:nrow(x),i),]);
finish;

start delcol(x,i); /*CANCELLA COLONNA DA MATRICE*/
return(x[,setdif(1:ncol(x),i)]);
finish;

start normalize(x);
/*STANDARDIZZAZIONE (UTILIZZATO PER diag(F'F)=I)*/
p=ncol(x);
n=nrow(x);
mu=j(1,p,0);
mom2=j(1,p,0);
sigma=j(1,p,0);
one=j(n,1,1);
sigmaI=i(p);
do i=1 to p;
  sum=0; sum2=0;
  do j=1 to n;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
sum=sum+x[j,i];
sum2=sum2+((x[j,i])**2);
end;
mu[i]=sum/n;
mom2[i]=sum2/n;
sigma[i]=1/(sqrt(mom2[i]-((mu[i])**2)));
sigmaI[i,i]=sigma[i];
end;
xbar=(x-(one*mu))*sigmaI;
return(xbar);
finish;

start vectorise(x); /*OPERATORE VEC*/
k=1;
do while (k<=ncol(x));
  if k=1 then w=x[,k];
  else w=w//x[,k];
  k=k+1;
end;
return (w);
finish;

start resize2(x,a);
/*RIDIMENSIONAMENTO DELLE MATRICI CON DUPLICATI (UTILIZZATO PER A')*/
k=1; n=ncol(x); b=a; /*inializzo indice, conto le colonne della matrice di
partenza e salvo in locale la matrice di destinazione*/
do while (k<=n); /*start ciclo esterno*/
j=k+1;
do while (j<=n); /*start ciclo interno parte dalla colonna successiva*/
  if any(x[,k]^=x[,j]) then j=j+1; /*se almeno un elemento della colonna è
diverto la colonna non è doppia, vado avanti*/
  else do;
    n=n-1; /*altrimenti parte la procedura di deduplica*/
    x=delcol(x,j);
    b[,k]=b[,k]+b[,j];
    b=delcol(b,j);
  end;
end;
k=k+1;
end;
return (b);
finish;

start resize1(x);
/*ridimensionamento di matrice, ritorna la stessa matrice ridimensionata.
Si usa per deduplicare le colonne di Z1*/
k=1; n=ncol(x);
do while (k<=n);
j=k+1;
do while (j<=n);
  if any(x[,k]^=x[,j]) then j=j+1;
  else do;
    n=n-1;
    x=delcol(x,j);
  end;
end;
k=k+1;
end;
return (x);
finish;

start update(X,y); /*AGGIORNAMENTO DELLE MATRICI A PARTIRE DAI VETTORI (NEI
PASSAGGI DI STIMA)*/
a=nrow(y);
n=nrow(X);
```

```
m=ncol(X);
Z=j(nrow(X),ncol(X),0);
k=1;
do while(k<=a);
  do j=1 to m;
    do i=1 to n;
      if X[i,j]^=0 then do;
        Z[i,j]=y[k]; k=k+1;
      end;
    end;
  end;
end;
return(Z);
finish;
/*PROCEDURE*/
/*END*/

use meansxt;
read all into mXT;

use meanse;
read all into mE;

use varcovxt;
read all into vXT;
vXT=delcol(vXT,1);

use varcove;
read all into vE;
vE=delcol(vE,1);

call randseed(4891);

XTk = RandNormal(&&rate&r,mXT,vXT);

XT = repeat(XTk,&replic,1);
E = RandNormal(&&rate&r*&replic,mE,vE);

sumW=0;
sumWT=0;
sumA=0;
sumAY=0;
sumWsq=0;
sumWTSq=0;
sumAsq=0;
sumAYSq=0;
sumc=0;
vcongr&r=j(&replic,3,0);

%do k=1 %to &replic;
%PUT replicazione &k in esecuzione, n = &&rate&r;
/*W*/
use dim;
read all var _NUM_ into W;
W=W*(&valuew);

/*WT*/
use dimc;
read all var _NUM_ into WT;
WT=WT*(&valuewt);

/*A*/
use dime;
read all var _NUM_ into A;
AT=t(A*(&valuea));
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*AY*/
use dimt;
read all var _NUM_ into AY;
AYT=(AY*(&valueay));

Xk=XT[1+&&rate&r*(&k-1):&&rate&r*&k,1:&nk];
Tk=XT[1+&&rate&r*(&k-1):&&rate&r*&k,&nk+1:ncol(XT)];
Ek=E[1+&&rate&r*(&k-1):&&rate&r*&k,];
Yk = (Xk*W*AT)+(Tk*WT*AT)+(Tk*AYT)+Ek;

/*QR decomposition di X*/
call qr(Q,Rx,piv,lindep,Xk);
Rxx=j(nrow(Xk)-nrow(Rx),ncol(Rx),0);
R=Rx//Rxx;

/*Ortogonalizzazione di T*/
TOrth=Tk-Xk*ginv(Xk)*Tk;

/*START ITERAZIONI*/
ciclo=&iter;
mustexit=0;
do while (mustexit=0);

oldAT=AT;
oldAYT=AYT;
oldW=W;
oldWT=WT;

/*ALGORITMO*/
/*STEP 1: Stima w°*/
/*vec(W)*/
vw=vectorise(W);
A=t(AT);
/*KRONECHER PROD*/
QT=t(Q);
omegad=A@R;
/*tolgo il contributo del legame diretto*/
Ylc=Yk-Tk*AYT;

i=1;
do while(i<=nrow(vw)); /*cancello zeri da vec(W) e rispettive colonne da
omegad*/
  if vw[i]^=0 then i=i+1;
  else do;
    vw=delrow(vw,i);
    omegad=delcol(omegad,i);
  end;
end;

QTY=QT*Ylc;
vecQTY=vectorise(QTY);

/*stima del vettore di parametri w°*/
wupd=((inv(t(omegad)*omegad))*t(omegad))*vecQTY;
/*ricostruzione di W*/
W=update(W,wupd);

/*STEP 2: Stima Wt*/
/*vec(WT)*/
vwt=vectorise(WT);
A=t(AT);
/*KRONECHER PROD*/
omegadt=A@TOrth;
```



```
/*tolgo il legame diretto e il legame tramite LC delle variabili esogene X*/
Ylc2=Ylc-(Xk*W*AT);
i=1;
do while(i<=nrow(vwt)); /*cancello zeri da vec(WT) e rispettive colonne da
omegad*/
  if vwt[i]^=0 then i=i+1;
else do;
  vwt=delrow(vwt,i);
  omegadt=delcol(omegad,i);
end;
end;

vecYlc2=vectorise(Ylc2);
/*stima del vettore di parametri wt*/
wtupd=((inv(t(omegad)*omegad))*t(omegad))*vecYlc2;
/*ricostruzione di WT*/
WT=update(WT,wtupd);

/*Costruzione di F e sua normalizzazione*/
F=Xk*W+Torth*WT;;
F=normalize(F);

/*STEP 3: Stima A*/
/*vec(A')*/
ad=vectorise(AT);
Id=I(ncol(AT));
/*KRONECHER PROD*/
gammad=Id@F;
vecYlc=vectorise(Ylc);

i=1;
do while(i<=nrow(ad)); /*cancello zeri da vec(A') e rispettive colonne da
gammad*/
  if ad[i]^=0 then i=i+1;
else do;
  ad=delrow(ad,i);
  gammad=delcol(gammad,i);
end;
end;
/*stima del vettore di parametri a*/
aupd=((inv(t(gammad)*gammad))*t(gammad))*vecYlc;
/*ricostruzione di A'*/
AT=update(AT,aupd);

/*STEP 4: Stima AY*/
/*Sottraggo alla matrice endogena la costruzione tramite LC
F = X W + T WT*/
Y2=Yk-F*AT;

/*Regressione multipla con coefficienti fissi di T su Y*/
vay=vectorise(AYT);
thetad=Id@Tk;
vY2=vectorise(Y2);

i=1;
do while(i<=nrow(vay)); /*cancello zeri da vec(Ay') e rispettive colonne da
thetad*/
  if vay[i]^=0 then i=i+1;
else do;
  vay=delrow(vay,i);
  thetad=delcol(thetad,i);
end;
end;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*stima del vettore di parametri ay*/
ayupd=(inv(t(thetad)*thetad))*(t(thetad))*vY2;
/*ricostruzione di Ay*/
AYT=update(AYT,ayupd);

/*ALGORITMO END*/
/*Criterio di accostamento*/
diffW=abs(W-oldW);
diffWT=abs(WT-oldWT);
diffAT=abs(AT-oldAT);
diffAYT=abs(AYT-oldAYT);

if &iter^=0 then do;

flgw=1;flgat=1;flgwt=1; flgay=1;
  do i=1 to nrow(diffW);
  do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
  end;
end;
  do i=1 to nrow(diffWT);
  do j=1 to ncol(diffWT);
    if diffWT[i,j]>&threshold then flgwt=0;
  end;
end;
  do i=1 to nrow(diffAT);
  do j=1 to ncol(diffAT);
    if diffAT[i,j]>&threshold then flgat=0;
  end;
end;
  do i=1 to nrow(diffAYT);
  do j=1 to ncol(diffAYT);
    if diffAYT[i,j]>&threshold then flgay=0;
  end;
end;
  ciclo=ciclo-1;
  if ciclo=0 | (flgw=1 & flgat=1 & flgwt=1 & flgay=1) then mustexit=1;
end;

if &iter=0 then do;
flgw=1;flgat=1;flgwt=1; flgay=1;
  do i=1 to nrow(diffW);
  do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
  end;
end;
  do i=1 to nrow(diffWT);
  do j=1 to ncol(diffWT);
    if diffWT[i,j]>&threshold then flgwt=0;
  end;
end;
  do i=1 to nrow(diffAT);
  do j=1 to ncol(diffAT);
    if diffAT[i,j]>&threshold then flgat=0;
  end;
end;
  do i=1 to nrow(diffAYT);
  do j=1 to ncol(diffAYT);
    if diffAYT[i,j]>&threshold then flgay=0;
  end;
end;
if flgw=1 & flgat=1 & flgwt=1 & flgay=1 then mustexit=1;
end;
```

```
end;
A=t(AT);
AY=t(AYT);

/*Back Transformation a W da W°*/
XT_BT=(ginv(Xk))*Tk; /*X*T*/
vecWTr=vectorise(WT); /*vec(Wt)*/
vecWr=vectorise(W); /*vec(W°)*/
DiagW=I(ncol(W));
BTr=DiagW@XT_BT; /*I x (X*T), "x" inteso come prodotto di kronecker*/

i=1;
do while(i<=nrow(vecWr));
if vecWr[i]^=0 then i=i+1;
else do;
vecWr=delrow(vecWr,i); /*canello righe nulle da vec(W°)*/
BTr=delrow(BTr,i); /*canello corrispettive RIGHE da I x (X*T)*/
end;
end;

i=1;
do while(i<=nrow(vecWTr));
if vecWTr[i]^=0 then i=i+1;
else do;
vecWTr=delrow(vecWTr,i); /*canello righe nulle da vec(W°)*/
BTr=delcol(BTr,i); /*canello corrispettive COLONNE da I x (X*T)*/
end;
end;

vecWnew=vecWr-(BTr)*vecWTr; /*costruisco W con back transformation in forma
vettorizzata*/
W=update(W,vecWnew); /*rimetto i coefficienti al posto corretto nella W back-
transformation*/

vW=vectorise(W);
vWT=vectorise(WT);
vA=vectorise(A);
vAY=vectorise(AY);

i=1;
do while(i<=nrow(vW));
if vW[i]^=0 then i=i+1;
else do;
vW=delrow(vW,i);
end;
end;

i=1;
do while(i<=nrow(vWT));
if vWT[i]^=0 then i=i+1;
else do;
vWT=delrow(vWT,i);
end;
end;

i=1;
do while(i<=nrow(vA));
if vA[i]^=0 then i=i+1;
else do;
vA=delrow(vA,i);
end;
end;

i=1;
do while(i<=nrow(vAY));
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
if vAY[i]^=0 then i=i+1;
else do;
  vAY=delrow(vAY,i);
end;
end;

sumW=sumW+vW;
sumWT=sumWT+vWT;
sumA=sumA+vA;
sumAY=sumAY+vAY;
sumWsq=sumWsq+(vW#vW);
sumWTsq=sumWTsq+(vWT#vWT);
sumAsq=sumAsq+(vA#vA);
sumAYsq=sumAYsq+(vAY#vAY);
vpW=vW*0+&valuew;
vpWT=vWT*0+&valuewt;
vpA=vA*0+&valuea;
vpAY=vAY*0+&valueay;

vecmcoef=vW/vWT;
vecmcoef=vecmcoef/vA;
vecmcoef=vecmcoef/vAY;
vecp=vpW/vpWT;
vecp=vecp/vpA;
vecp=vecp/vpAY;

congr=(t(vecp)*vecmcoef)/((sqrt(t(vecp)*vecp))*(sqrt(t(vecmcoef)*vecmcoef)));

vcongr&r[&k,1]=&k;
vcongr&r[&k,2]=&rate&r;
vcongr&r[&k,3]=congr;

sumc=sumc+congr;

%end;

MeanvW=sumW/&replc;
MeanvWT=sumWT/&replc;
MeanvA=sumA/&replc;
MeanvAY=sumAY/&replc;

MeanvWsq=sumWsq/&replc;
MeanvWTsq=sumWTsq/&replc;
MeanvAsq=sumAsq/&replc;
MeanvAYsq=sumAYsq/&replc;

seW=((MeanvWsq-(MeanvW#MeanvW))*((&replc)/(&replc-1)))##0.5;
seWT=((MeanvWTsq-(MeanvWT#MeanvWT))*((&replc)/(&replc-1)))##0.5;
seA=((MeanvAsq-(MeanvA#MeanvA))*((&replc)/(&replc-1)))##0.5;
seAY=((MeanvAYsq-(MeanvAY#MeanvAY))*((&replc)/(&replc-1)))##0.5;

estimates = MeanvW//MeanvWT;
estimates = estimates//MeanvA;
estimates = estimates//MeanvAY;

stderr = seW//seWT;
stderr = stderr//seA;
stderr = stderr//seAY;

meancongr=sumc/&replc;
biases=100*abs(estimates-vecp)/vecp;
outputM&rate&r=estimates||stderr;
outputM&rate&r=outputM&rate&r||biases;
num={&rate&r 0 0 };
congruence=meancongr*{1 0 0};
```

```
outputM&&rate&r=num//outputM&&rate&r;
outputM&&rate&r=outputM&&rate&r//congruence;
outputM&&rate&r=round(outputM&&rate&r,0.001);

names={'est' 'stderr' 'bias'};
trail=j(1,3,char(&&rate&r));
collabel=compbl(names + trail);

create results&r from outputM&&rate&r [colname = collabel];
append from outputM&&rate&r;

create results_rep&r from vcongr&r [colname = {'rep' 'n' 'congr'}];
append from vcongr&r;

%put n= &&rate&r eseguita;

%end;
leftcW=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="w"+char(j);
chj=COMPBL(chj);
leftcW[j]=chj;
end;
leftcWT=j(nrow(meanvWT),1,"xxxx");
do j=1 to nrow(meanvWT);
chj="wt"+char(j);
chj=COMPBL(chj);
leftcWT[j]=chj;
end;
leftcA=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="a"+char(j);
chj=COMPBL(chj);
leftcA[j]=chj;
end;
leftcAY=j(nrow(meanvAY),1,"xxxx");
do j=1 to nrow(meanvAY);
chj="ay"+char(j);
chj=COMPBL(chj);
leftcAY[j]=chj;
end;

parm="n"//leftcW;
parm=parm//leftcWT;
parm=parm//leftcA;
parm=parm//leftcAY;
parm=parm//"congr";

pop=0//vecp;
pop=pop//0;

create results from parm [colname = {'label'}];
append from parm;

create pop from pop [colname = {'pop'}];
append from pop;

data results;
merge results pop;
run;

data results;
merge results results1;
run;
data results;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
merge results results2;
run;
data results;
merge results results3;
run;
data results;
merge results results4;
run;

data Res_Simul_GRACORR; set results;
array change _numeric_;
do over change;
if change=0 then change="";
end;
run;

data Res_Simul_GRACORR_rep;
set Results_rep1 Results_rep2 Results_rep3 Results_rep4; run;

proc datasets;
save Res_Simul_GRACORR Res_Simul_GRACORR_rep;run;

proc univariate data=Res_Simul_GRACORR_rep;
var congr;
by n;
run;

%mend;
quit;

%mcsamples(varcovxt, meansxt, varcove, meanse, 4,0.6,0.3,0.2,0.4,
250,300,350,400,1000,10000,0.0003);
```

APPENDIX 3 – STUDY 3 (SCENARIO I)

```
/*INPUT: MATRICI DI VARIANZE E COVARIANZE DI X, T ED E E RISPETTIVI VETTORI DI
MEDIE*/
/*Le dimensioni devono essere conformi*/
data meansx;
input x @@;
cards;
0 0 0 0
;
run;
data varcovx;
input row v1-v4;
cards;
1 1 0.3 0.1 0.1
2 0.3 1 0.1 0.1
3 0.1 0.1 1 0.3
4 0.1 0.1 0.3 1
;
run;

data skewkurt;
input E skewness kurtosis;
cards;
1 2 5.5
2 2 5.5
;

data meanst;
/*togliere un valore alla volta per arrivare da V1 a V2, V3, V4*/
/*es. V2: 0 0 0*/
input x @@;
cards;
0
;
run;
data varcovt;
input row v1;
/* diminuire il range e togliere una riga e colonna dalla matrice per passare
da
V1 a V2, V3, V4*/
/*es. V2: input row v1-v3;
cards;
1 1 0.1 0.1
2 0.1 1 0.1
3 0.1 0.1 1
;*/
cards;
1 1
;
run;

data meanse;
input x @@;
cards;
0 0
;
run;
data varcove;
input row v1-v2;
cards;
1 1 0.1
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
2 0.1 1
;
run;
proc means data =varcovx max noprint ; var row ;output out =m(keep= max ) max
= max ; run ;
data m;set m;call symput( "dime",max); run ;

/*INPUT: PATH DIAGRAM, Le dimensioni devono essere conformi alle matrici
dichiarate precedentemente
/*Parte esogena verso variabili latenti, X (6 variabili) verso latent
composites (3 componenti)
/*3 coefficienti
w1: x01 -> f1,
w2: x02 -> f1,
w3: x03 -> f2,
w4: x04 -> f2,
w5: x05 -> f3,
w6: x06 -> f3,
NB dim(X)=nrow(varcovx)*/
data dim;
  input vm $3. f01 f02;
cards ;
x01 1 0
x02 1 0
x03 0 1
x04 0 1
;
run ;

data dimt;
  input vm $3. y01 y02;
cards ;
t01 1 0
;
run ;

/*Parte verso variabili endogene, latent composites (2 componenti) verso Y (2
variabili)
/*4 coefficienti
a1: f1 -> y01
a2: f1 -> y02
a3: f2 -> y01
a4: f2 -> y02
a5: f3 -> y01*/
data dime;
  input ve $3. f01 f02;
cards ;
y01 1 1
y02 1 1
;
run ;

%macro mcsamples(varcovx, meansx,varcovt, meanst, varcovx, meanse,
skewkurt,valuew, valuea,valueay,
ratel,rate2,rate3,rate4,replic,iter,threshold);

%macro rename_e(old,new);
%let k=1;
  %do %while(&k <= &dime);
    %let p= %eval(&k);
    rename &old.&p = &new.&k;
    %let k = %eval(&k + 1);
  %end;
%mend;

XXXIV
```



```

%macro rename_t(old,new);
%let k=1;
  %do %while(&k <= &nt);
    %let p= %eval(&k+&nx);
    rename &old.&p = &new.&k;
    %let k = %eval(&k + 1);
  %end;
%mend;

%do r=1 %to 4;

proc iml;

start Fleishman(coef);
b = coef[1]; c = coef[2]; d = coef[3];
b2 = b##2; c2 = c##2; d2 = d##2; bd = b#d;
var = b2 + 6#bd + 2#c2 + 15#d2; /* variance */
skew = 2#c#(b2 + 24#bd + 105#d2 + 2); /* skewness */
kurt = 24#(bd + c2#(1 + b2 + 28#bd) + d2 #
(12 + 48#bd + 141#c2 + 225#d2)); /* excess kurtosis */
return( var // skew // kurt );
finish;

start FlFunc(x) global (g_target); /* g_target=(skewness, kurtosis) */
return ( Fleishman(x) - (1 // g_target[1] // g_target[2]) );
finish FlFunc;

start FlDeriv(x);
b = x[1]; c = x[2]; d = x[3];
b2 = b##2; c2 = c##2; d2 = d##2; bd = b#d;
df1db = 2#b + 6#d;
df1dc = 4#c;
df1dd = 6#b + 30#d;
df2db = 4#c#(b + 12#d);
df2dc = 2#(b2 + 24#bd + 105#d2 + 2);
df2dd = 4#c#(12#b + 105#d);
df3db = 24#(d + c2#(2#b + 28#d) + 48#d##3);
df3dc = 48#c#(1 + b2 + 28#bd + 141#d2);
df3dd = 24#(b + 28#b#c2 + 2#d#(12 + 48#bd + 141#c2 + 225#d2)
+ d2#(48#b + 450#d));
J = (df1db || df1dc || df1dd) //
(df2db || df2dc || df2dd) //
(df3db || df3dc || df3dd);
return( J );
finish FlDeriv;

start Newton(x, x0, optn);
maxIter = optn[1]; converge = optn[2];
x = x0;
f = FlFunc(x);
do iter = 1 to maxIter while(max(abs(f)) > converge);
J = FlDeriv(x);
delta = -solve(J, f); /* correction vector */
x = x + delta; /* new approximation */
f = FlFunc(x);
end;
/* return missing if no convergence */
if iter > maxIter then x = j(nrow(x0),ncol(x0),.);
finish Newton;

start FleishmanIC(skew, kurt);
c = j(3,1);
c[1] = 0.95357 -0.05679*kurt + /* c1 = Quad(skew, kurt) */
0.03520*skew##2 + 0.00133*kurt##2;
c[2] = 0.10007*skew + 0.00844*skew##3; /* c2 = Cubic(skew) */

```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
c[3] = 0.30978 -0.31655*c[1]; /* c3 = Linear(c1) */
return (c);
finish;

start FitFleishmanFromSK(skew, kurt) global (g_target);
/* 1. check that (skew,kurt) is in the feasible region */
if kurt < -1.13168+1.58837*skew##2 then return({. . . .});
/* 2. Initial guess for nonlinear root finding */
x0 = FleishmanIC(skew, kurt);
optn = {25, 1e-5}; /* maximum iterations, convergence criterion */
/* 3. Find cubic coefficients (c1, c2, c3) so that
-c2+c1*Z+c2*Z##2+c3*Z##3 has target (skew,kurt). */
g_target = skew || kurt; /* set global variable */
run Newton(coef, x0, optn); /* find (c1, c2, c3) */
return( -coef[2] // coef );
finish;

start RandFleishman(Z,skew, kurt);
coef=FitFleishmanFromSK(skew, kurt);
X = coef[1] + Z#(coef[2] + Z#(coef[3]+Z#coef[4]));
return( X );
finish;

start delrow(x,i);
/*CANCELLA RIGA DA MATRICE*/
return(x[setdif(1:nrow(x),i),]);
finish;

start delcol(x,i); /*CANCELLA COLONNA DA MATRICE*/
return(x[,setdif(1:ncol(x),i)]);
finish;

start normalize(x);
/*STANDARDIZZAZIONE (UTILIZZATO PER diag(F'F)=I)*/
p=ncol(x);
n=nrow(x);
mu=j(1,p,0);
mom2=j(1,p,0);
sigma=j(1,p,0);
one=j(n,1,1);
sigmaI=i(p);
do i=1 to p;
  sum=0; sum2=0;
  do j=1 to n;
    sum=sum+x[j,i];
    sum2=sum2+((x[j,i])**2);
  end;
  mu[i]=sum/n;
  mom2[i]=sum2/n;
  sigma[i]=1/(sqrt(mom2[i]-((mu[i])**2)));
  sigmaI[i,i]=sigma[i];
end;
xbar=(x-(one*mu))*sigmaI;
return(xbar);
finish;

start vectorise(x); /*OPERATORE VEC*/
k=1;
do while (k<=ncol(x));
  if k=1 then w=x[,k];
  else w=w//x[,k];
  k=k+1;
end;
return (w);
finish;
```

```
start resize2(x,a);
/*RIDIMENSIONAMENTO DELLE MATRICI CON DUPLICATI (UTILIZZATO PER A')*/
k=1; n=ncol(x); b=a; /*inizializzo indice, conto le colonne della matrice di
partenza e salvo in locale la matrice di destinazione*/
do while (k<=n); /*start ciclo esterno*/
j=k+1;
do while (j<=n); /*start ciclo interno parte dalla colonna successiva*/
if any(x[,k]^=x[,j]) then j=j+1; /*se almeno un elemento della colonna è
diverto la colonna non è doppia, vado avanti*/
else do;
n=n-1; /*altrimenti parte la procedura di deduplica*/
x=delcol(x,j);
b[,k]=b[,k]+b[,j];
b=delcol(b,j);
end;
end;
k=k+1;
end;
return (b);
finish;
```

```
start resize1(x);
/*ridimensionamento di matrice, ritorna la stessa matrice ridimensionata.
Si usa per deduplicare le colonne di Z1*/
k=1; n=ncol(x);
do while (k<=n);
j=k+1;
do while (j<=n);
if any(x[,k]^=x[,j]) then j=j+1;
else do;
n=n-1;
x=delcol(x,j);
end;
end;
k=k+1;
end;
return (x);
finish;
```

```
start update(X,y); /*AGGIORNAMENTO DELLE MATRICI A PARTIRE DAI VETTORI (NEI
PASSAGGI DI STIMA)*/
a=nrow(y);
n=nrow(X);
m=ncol(X);
Z=j(nrow(X),ncol(X),0);
k=1;
do while(k<=a);
do j=1 to m;
do i=1 to n;
if X[i,j]^=0 then do;
Z[i,j]=y[k]; k=k+1;
end;
end;
end;
return(Z);
finish;
/*PROCEDURE*/
/*END*/
```

```
use meansx;
read all into mX;

use meanst;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
read all into mT;
use meanse;
read all into mE;

use varcovx;
read all into vX;
vX=delcol(vX,1);

use varcovt;
read all into vT;
vT=delcol(vT,1);

use varcove;
read all into vE;
vE=delcol(vE,1);

use &skewkurt;
read all into skewkurt;

call randseed(4891*(&r)**(-2));

Xk = RandNormal(&&rate&r,mX,vX);
Tk = RandNormal(&&rate&r,mT,vT);
X = repeat(Xk,&replic,1);
T = repeat(Tk,&replic,1);
Z1M = RandNormal(1000000,mE,vE);
create Z1M from Z1M ;
append from Z1M;

E1M=j(nrow(Z1M),ncol(Z1M),0);

do c=1 to ncol(Z1M);
E1M[,c]=RandFleishman(Z1M[,c],skewkurt[c,2],skewkurt[c,3]);
end;

create E1M from E1M ;
append from E1M;

E=j(&&rate&r*&replic,ncol(E1M));

do j=1 to &&rate&r*&replic;
rnd=ranuni(3479)*1000000;
E[j,]=E1M[rnd,];
end;

create E&r from E ;
append from E;

sumW=0;
sumA=0;
sumAY=0;
sumWsq=0;
sumWTsq=0;
sumAsq=0;
sumAYsq=0;
sumc=0;
vcongr&r=j(&replic,3,0);

%do k=1 %to &replic;
%PUT replicazione &k in esecuzione, n = &&rate&r;
/*W*/
use dim;
read all var _NUM_ into W;
W=W*(&valuew);
```

```
/*A*/
use dime;
read all var _NUM_ into A;
AT=t(A*(&valuea));

/*AY*/
use dimt;
read all var _NUM_ into AY;
AYT=(AY*(&valueay));

Xk=X[1+&&rate&r*(&k-1):&&rate&r*&k,];
Tk=T[1+&&rate&r*(&k-1):&&rate&r*&k,];
Ek=E[1+&&rate&r*(&k-1):&&rate&r*&k,];
Yk = (Xk*W*AT)+(Tk*AYT)+Ek;

/*QR decomposition di X*/
call qr(Q,Rx,piv,lindexp,Xk);
Rxx=j(nrow(Xk)-nrow(Rx),ncol(Rx),0);
R=Rx//Rxx;
/*Ortogonalizzazione di T*/
TOrth=Tk-Xk*ginv(Xk)*Tk;

/*START ITERAZIONI*/
ciclo=&iter;
mustexit=0;
do while (mustexit=0);

oldAT=AT;
oldAYT=AYT;
oldW=W;

/*ALGORITMO*/
/*STEP 1: Stima w°*/
/*vec(W)*/
vw=vectorise(W);
A=t(AT);
/*KRONECHER PROD*/
QT=t(Q);
omegad=A@R;
/*tolgo il contributo del legame diretto*/
Ylc=Yk-Tk*AYT;

i=1;
do while(i<=nrow(vw)); /*cancello zeri da vec(W) e rispettive colonne da
omegad*/
  if vw[i]^=0 then i=i+1;
else do;
  vw=delrow(vw,i);
  omegad=delcol(omegad,i);
end;
end;

QTY=QT*Ylc;
vecQTY=vectorise(QTY);

/*stima del vettore di parametri w°*/
wupd=((inv(t(omegad)*omegad))*t(omegad))*vecQTY;
/*ricostruzione di W*/
W=update(W,wupd);

/*Costruzione di F e sua normalizzazione*/
F=Xk*W;
F=normalize(F);

/*STEP 2: Stima A*/
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*vec(A')*/
ad=vectorise(AT);
Id=I(ncol(AT));
/*KRONECHER PROD*/
gammad=Id@F;
vecYlc=vectorise(Ylc);

i=1;
do while(i<=nrow(ad)); /*canello zeri da vec(A') e rispettive colonne da
gammad*/
if ad[i]^=0 then i=i+1;
else do;
  ad=delrow(ad,i);
  gammad=delcol(gammad,i);
end;
end;
/*stima del vettore di parametri a*/
aupd=((inv(t(gammad)*gammad))*(t(gammad)))*vecYlc;
/*ricostruzione di A'*/
AT=update(AT,aupd);
/*STEP 4: Stima AY*/
/*Sottraggo alla matrice endogena la costruzione tramite LC
 $F = XW + TWT$ */
Y2=Yk-F*AT;

/*Regressione multipla con coefficienti fissi di T su Y*/
vay=vectorise(AYT);
thetad=Id@Tk;
vY2=vectorise(Y2);

i=1;
do while(i<=nrow(vay)); /*canello zeri da vec(Ay') e rispettive colonne da
thetad*/
if vay[i]^=0 then i=i+1;
else do;
  vay=delrow(vay,i);
  thetad=delcol(thetad,i);
end;
end;

/*stima del vettore di parametri ay*/
ayupd=((inv(t(thetad)*thetad))*(t(thetad)))*vY2;
/*ricostruzione di Ay*/
AYT=update(AYT,ayupd);

/*ALGORITMO END*/
/*Criterio di accostamento*/
diffW=abs(W-oldW);
diffAT=abs(AT-oldAT);
diffAYT=abs(AYT-oldAYT);

if &iter^=0 then do;

flgw=1;flgat=1; flgay=1;
  do i=1 to nrow(diffW);
    do j=1 to ncol(diffW);
      if diffW[i,j]>&threshold then flgw=0;
    end;
  end;
  do i=1 to nrow(diffAT);
    do j=1 to ncol(diffAT);
      if diffAT[i,j]>&threshold then flgat=0;
    end;
  end;
  do i=1 to nrow(diffAYT);
```

```
        do j=1 to ncol(diffAYT);
            if diffAYT[i,j]>&threshold then flgay=0;
        end;
    end;
    end;
    ciclo=ciclo-1;
    if ciclo=0 | (flgw=1 & flgat=1 & flgay=1) then mustexit=1;
end;

if &iter=0 then do;
flgw=1;flgat=1; flgay=1;
    do i=1 to nrow(diffW);
        do j=1 to ncol(diffW);
            if diffW[i,j]>&threshold then flgw=0;
        end;
    end;
    do i=1 to nrow(diffAT);
        do j=1 to ncol(diffAT);
            if diffAT[i,j]>&threshold then flgat=0;
        end;
    end;
    do i=1 to nrow(diffAYT);
        do j=1 to ncol(diffAYT);
            if diffAYT[i,j]>&threshold then flgay=0;
        end;
    end;
end;

if flgw=1 & flgat=1 & flgay=1 then mustexit=1;
end;

end;

A=t(AT);
AY=t(AYT);

vW=vectorise(W);
vA=vectorise(A);
vAY=vectorise(AY);

i=1;
do while(i<=nrow(vW));
    if vW[i]^=0 then i=i+1;
else do;
    vW=delrow(vW,i);
end;
end;

i=1;
do while(i<=nrow(vA));
    if vA[i]^=0 then i=i+1;
else do;
    vA=delrow(vA,i);
end;
end;

i=1;
do while(i<=nrow(vAY));
    if vAY[i]^=0 then i=i+1;
else do;
    vAY=delrow(vAY,i);
end;
end;

sumW=sumW+vW;
sumA=sumA+vA;
sumAY=sumAY+vAY;
sumWsq=sumWsq+(vW#vW);
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
sumAsq=sumAsq+ (vA#vA) ;
sumAYsq=sumAYsq+ (vAY#vAY) ;
vpW=vW*0+&valuew;
vpA=vA*0+&valuea;
vpAY=vAY*0+&valueay;

vecmcoef=vW;
vecmcoef=vecmcoef//vA;
vecmcoef=vecmcoef//vAY;
vecp=vpW;
vecp=vecp//vpA;
vecp=vecp//vpAY;

congr=(t (vecp) *vecmcoef) / ((sqrt (t (vecp) *vecp)) * (sqrt (t (vecmcoef) *vecmcoef)));
vcngr&r[&k,1]=&k;
vcngr&r[&k,2]=&&rate&r;
vcngr&r[&k,3]=congr;

sumc=sumc+congr;

%end;

MeanvW=sumW/&replc;
MeanvA=sumA/&replc;
MeanvAY=sumAY/&replc;

MeanvWsq=sumWsq/&replc;
MeanvAsq=sumAsq/&replc;
MeanvAYsq=sumAYsq/&replc;

seW= ( (MeanvWsq- (MeanvW#MeanvW)) * ((&replc) / (&replc-1)))##0.5;
seA= ( (MeanvAsq- (MeanvA#MeanvA)) * ((&replc) / (&replc-1)))##0.5;
seAY= ( (MeanvAYsq- (MeanvAY#MeanvAY)) * ((&replc) / (&replc-1)))##0.5;

estimates = MeanvW;
estimates = estimates//MeanvA;
estimates = estimates//MeanvAY;

stderr = seW;
stderr = stderr//seA;
stderr = stderr//seAY;

meancongr=sumc/&replc;
biases=100*abs (estimates-vecp) /vecp;
outputM&&rate&r=estimates||stderr;
outputM&&rate&r=outputM&&rate&r||biases;
num={&&rate&r 0 0 };
congruence=meancongr*{1 0 0 };
outputM&&rate&r=num//outputM&&rate&r;
outputM&&rate&r=outputM&&rate&r//congruence;
outputM&&rate&r=round (outputM&&rate&r,0.001);

names={'est' 'stderr' 'bias'};
trail=j (1,3,char (&&rate&r));
collabel=compbl (names + trail);

create results&r from outputM&&rate&r [colname = collabel];
append from outputM&&rate&r;
create results_rep&r from vcngr&r [colname = {'rep' 'n' 'congr'}];
append from vcngr&r;

%put n= &&rate&r eseguita;

%end;
leftcW=j (nrow (meanvW) ,1, "xxx");
```



```
do j=1 to nrow(meanvW);
chj="w"+char(j);
chj=COMPBL(chj);
leftcW[j]=chj;
end;
leftcA=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="a"+char(j);
chj=COMPBL(chj);
leftcA[j]=chj;
end;
leftcAY=j(nrow(meanvAY),1,"xxxx");
do j=1 to nrow(meanvAY);
chj="ay"+char(j);
chj=COMPBL(chj);
leftcAY[j]=chj;
end;

parm="n"//leftcW;
parm=parm//leftcA;
parm=parm//leftcAY;
parm=parm//"congr";

pop=0//vecp;
pop=pop//0;

create results from parm [colname = {'label'}];
append from parm;

create pop from pop [colname = {'pop'}];
append from pop;
run;
quit;

data results;
merge results pop;
run;

data results;
merge results results1;
run;
data results;
merge results results2;
run;
data results;
merge results results3;
run;
data results;
merge results results4;
run;

data Res_Simul_GRAEXT_NNORM; set results;
array change _numeric_;
do over change;
if change=0 then change="";
end;
run;

data Res_Simul_GRAEXT_NNORM_rep;
set Results_rep1 Results_rep2 Results_rep3 Results_rep4; run;

data E1; set E1;
%rename_e(COL,e0);
n=1;
RUN;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
data E2; set E2;
%rename_e(COL,e0);
n=2;
RUN;
data E3; set E3;
%rename_e(COL,e0);
n=3;
RUN;
data E4; set E4;
%rename_e(COL,e0);
n=4;
RUN;

proc datasets;
save Res_Simul_GRAEXT_NNORM Res_Simul_GRAEXT_NNORM_rep E1 E2 E3 E4 E1M
Z1M;run;

data E;
set E1 E2 E3 E4;
run;

proc univariate data=Res_Simul_GRAEXT_NNORM_rep;
var congr;
by n;
run;

proc univariate data=e;
histogram e01 e02/ normal;
var e01 e02;
by n;
run;

quit;
%mend;

%mcsamples(varcovx, meansx, varcovt, meanst, varcove, meanse,
skewkurt,0.6,0.2,0.4, 100, 200, 400, 600,1000,10000,0.0003);
```

APPENDIX 3 – STUDY 3 (SCENARIO II)

```
/*INPUT: MATRICI DI VARIANZE E COVARIANZE DI X, T ED E E RISPETTIVI VETTORI DI
MEDIE*/
/*Le dimensioni devono essere conformi*/
data meansx;
input x @@;
cards;
0 0 0 0
;
run;
data varcovx;
input row v1-v4;
cards;
1 1 0.3 0.1 0.1
2 0.3 1 0.1 0.1
3 0.1 0.1 1 0.3
4 0.1 0.1 0.3 1
;
run;

data skewkurt;
input E skewness kurtosis;
cards;
1 1 0.5
2 1 0.5
;
proc means data =varcove max noprint ; var row ;output out =m(keep= max ) max
= max ; run ;
data m;set m;call symput( "dime",max); run ;

data meanst;
/*togliere un valore alla volta per arrivare da V1 a V2, V3, V4*/
/*es. V2: 0 0 0*/
input x @@;
cards;
0
;
run;
data varcovt;
input row v1;
/* diminuire il range e togliere una riga e colonna dalla matrice per passare
da
V1 a V2, V3, V4*/
/*es. V2: input row v1-v3;
cards;
1 1 0.1 0.1
2 0.1 1 0.1
3 0.1 0.1 1
;*/
cards;
1 1
;
run;

data meanse;
input x @@;
cards;
0 0
;
run;
data varcove;
input row v1-v2;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
cards;
1 1 0.1
2 0.1 1
;
run;

/*INPUT: PATH DIAGRAM, Le dimensioni devono essere conformi alle matrici
dichiarate precedentemente
/*Parte esogena verso variabili latenti, X (6 variabili) verso latent
composites (3 componenti)
/*3 coefficienti
w1: x01 -> f1,
w2: x02 -> f1,
w3: x03 -> f2,
w4: x04 -> f2,
w5: x05 -> f3,
w6: x06 -> f3,
NB dim(X)=nrow(varcovx)*/
data dim;
  input vm $3. f01 f02;
cards ;
x01 1 0
x02 1 0
x03 0 1
x04 0 1
;
run ;

data dimc;
  input vm $3. f01 f02;
cards ;
t01 1 0
;
run ;

data dimt;
  input vm $3. y01 y02;
cards ;
t01 1 0
;
run ;

/*Parte verso variabili endogene, latent composites (2 componenti) verso Y (2
variabili)
/*4 coefficienti
a1: f1 -> y01
a2: f1 -> y02
a3: f2 -> y01
a4: f2 -> y02
a5: f3 -> y01*/
data dime;
  input ve $3. f01 f02;
cards ;
y01 1 1
y02 1 1
;
run ;

%macro mcsamples(varcovx, meansx, varcovt, meanst, varcove, meanse,
skewkurt, valuew, valuewt, valuea, valueay,
ratel, rate2, rate3, rate4, replic, iter, threshold);

%macro rename_e(old, new);
%let k=1;
  %do %while(&k <= &dime);
```

```

    %let p= %eval(&k);
    rename &old.&p = &new.&k;
    %let k = %eval(&k + 1);
    %end;
%mend;

%macro rename_t(old,new);
%let k=1;
    %do %while(&k <= &nt);
        %let p= %eval(&k+&nx);
        rename &old.&p = &new.&k;
        %let k = %eval(&k + 1);
    %end;
%mend;

%do r=1 %to 4;

proc iml;

start Fleishman(coef);
b = coef[1]; c = coef[2]; d = coef[3];
b2 = b##2; c2 = c##2; d2 = d##2; bd = b#d;
var = b2 + 6#bd + 2#c2 + 15#d2; /* variance */
skew = 2#c#(b2 + 24#bd + 105#d2 + 2); /* skewness */
kurt = 24#(bd + c2#(1 + b2 + 28#bd) + d2 #
(12 + 48#bd + 141#c2 + 225#d2)); /* excess kurtosis */
return( var // skew // kurt );
finish;

start FlFunc(x) global (g_target); /* g_target=(skewness, kurtosis) */
return ( Fleishman(x) - (1 // g_target[1] // g_target[2]) );
finish FlFunc;

start FlDeriv(x);
b = x[1]; c = x[2]; d = x[3];
b2 = b##2; c2 = c##2; d2 = d##2; bd = b#d;
df1db = 2#b + 6#d;
df1dc = 4#c;
df1dd = 6#b + 30#d;
df2db = 4#c#(b + 12#d);
df2dc = 2#(b2 + 24#bd + 105#d2 + 2);
df2dd = 4#c#(12#b + 105#d);
df3db = 24#(d + c2#(2#b + 28#d) + 48#d##3);
df3dc = 48#c#(1 + b2 + 28#bd + 141#d2);
df3dd = 24#(b + 28#b#c2 + 2#d#(12 + 48#bd + 141#c2 + 225#d2)
+ d2#(48#b + 450#d));
J = (df1db || df1dc || df1dd) //
(df2db || df2dc || df2dd) //
(df3db || df3dc || df3dd);
return( J );
finish FlDeriv;

start Newton(x, x0, optn);
maxIter = optn[1]; converge = optn[2];
x = x0;
f = FlFunc(x);
do iter = 1 to maxIter while(max(abs(f)) > converge);
J = FlDeriv(x);
delta = -solve(J, f); /* correction vector */
x = x + delta; /* new approximation */
f = FlFunc(x);
end;
/* return missing if no convergence */
if iter > maxIter then x = j(nrow(x0),ncol(x0),.);
finish Newton;

```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
start FleishmanIC(skew, kurt);
c = j(3,1);
c[1] = 0.95357 -0.05679*kurt + /* c1 = Quad(skew, kurt) */
0.03520*skew##2 + 0.00133*kurt##2;
c[2] = 0.10007*skew + 0.00844*skew##3; /* c2 = Cubic(skew) */
c[3] = 0.30978 -0.31655*c[1]; /* c3 = Linear(c1) */
return (c);
finish;

start FitFleishmanFromSK(skew, kurt) global (g_target);
/* 1. check that (skew, kurt) is in the feasible region */
if kurt < -1.13168+1.58837*skew##2 then return({. . .});
/* 2. Initial guess for nonlinear root finding */
x0 = FleishmanIC(skew, kurt);
optn = {25, 1e-5}; /* maximum iterations, convergence criterion */
/* 3. Find cubic coefficients (c1, c2, c3) so that
-c2+c1*Z+c2*Z##2+c3*Z##3 has target (skew, kurt). */
g_target = skew || kurt; /* set global variable */
run Newton(coef, x0, optn); /* find (c1, c2, c3) */
return( -coef[2] // coef );
finish;

start RandFleishman(Z, skew, kurt);
coef=FitFleishmanFromSK(skew, kurt);
X = coef[1] + Z#(coef[2] + Z#(coef[3]+Z#coef[4]));
return( X );
finish;

start delrow(x,i);
/*CANCELLA RIGA DA MATRICE*/
return(x[setdif(1:nrow(x),i),]);
finish;

start delcol(x,i); /*CANCELLA COLONNA DA MATRICE*/
return(x[,setdif(1:ncol(x),i)]);
finish;

start normalize(x);
/*STANDARDIZZAZIONE (UTILIZZATO PER diag(F'F)=I)*/
p=ncol(x);
n=nrow(x);
mu=j(1,p,0);
mom2=j(1,p,0);
sigma=j(1,p,0);
one=j(n,1,1);
sigmaI=i(p);
do i=1 to p;
  sum=0; sum2=0;
  do j=1 to n;
    sum=sum+x[j,i];
    sum2=sum2+(x[j,i]**2);
  end;
  mu[i]=sum/n;
  mom2[i]=sum2/n;
  sigma[i]=1/(sqrt(mom2[i]-((mu[i])**2)));
  sigmaI[i,i]=sigma[i];
end;
xbar=(x-(one*mu))*sigmaI;
return(xbar);
finish;

start vectorise(x); /*OPERATORE VEC*/
k=1;
do while (k<=ncol(x));
  if k=1 then w=x[,k];
```

```
else w=w//x[,k];
k=k+1;
end;
return (w);
finish;

start resize2(x,a);
/*RIDIMENSIONAMENTO DELLE MATRICI CON DUPLICATI (UTILIZZATO PER A')*/
k=1; n=ncol(x); b=a; /*inizializzo indice, conto le colonne della matrice di
partenza e salvo in locale la matrice di destinazione*/
do while (k<=n); /*start ciclo esterno*/
j=k+1;
do while (j<=n); /*start ciclo interno parte dalla colonna successiva*/
if any(x[,k]^x[,j]) then j=j+1; /*se almeno un elemento della colonna è
diverto la colonna non è doppia, vado avanti*/
else do;
n=n-1; /*altrimenti parte la procedura di deduplica*/
x=delcol(x,j);
b[,k]=b[,k]+b[,j];
b=delcol(b,j);
end;
end;
k=k+1;
end;
return (b);
finish;

start resize1(x);
/*ridimensionamento di matrice, ritorna la stessa matrice ridimensionata.
Si usa per deduplicare le colonne di Z1*/
k=1; n=ncol(x);
do while (k<=n);
j=k+1;
do while (j<=n);
if any(x[,k]^x[,j]) then j=j+1;
else do;
n=n-1;
x=delcol(x,j);
end;
end;
k=k+1;
end;
return (x);
finish;

start update(X,y); /*AGGIORNAMENTO DELLE MATRICI A PARTIRE DAI VETTORI (NEI
PASSAGGI DI STIMA)*/
a=nrow(y);
n=nrow(X);
m=ncol(X);
Z=j(nrow(X),ncol(X),0);
k=1;
do while(k<=a);
do j=1 to m;
do i=1 to n;
if X[i,j]^=0 then do;
Z[i,j]=y[k]; k=k+1;
end;
end;
end;
return(Z);
finish;
/*PROCEDURE*/
/*END*/
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
use meansx;
read all into mX;

use meanst;
read all into mT;

use meanse;
read all into mE;

use varcovx;
read all into vX;
vX=delcol(vX,1);

use varcovt;
read all into vT;
vT=delcol(vT,1);

use varcove;
read all into vE;
vE=delcol(vE,1);

use &skewkurt;
read all into skewkurt;

call randseed(4891*(&r)**(-2));

Xk = RandNormal(&&rate&r,mX,vX);
Tk = RandNormal(&&rate&r,mT,vT);
X = repeat(Xk,&replic,1);
T = repeat(Tk,&replic,1);
Z1M = RandNormal(1000000,mE,vE);
create Z1M from Z1M ;
append from Z1M;

E1M=j(nrow(Z1M),ncol(Z1M),0);

do c=1 to ncol(Z1M);
E1M[,c]=RandFleishman(Z1M[,c],skewkurt[c,2],skewkurt[c,3]);
end;

create E1M from E1M ;
append from E1M;

E=j(&&rate&r*&replic,ncol(E1M));

do j=1 to &&rate&r*&replic;
rnd=ranuni(3479)*1000000;
E[j,]=E1M[rnd,];
end;

create E&r from E ;
append from E;

sumW=0;
sumWT=0;
sumA=0;
sumAY=0;
sumWsq=0;
sumWTsq=0;
sumAsq=0;
sumAYSq=0;
sumc=0;
vcongr&r=j(&replic,3,0);
%do k=1 %to &replic;
%PUT replicazione &k in esecuzione, n = &&rate&r;
```



```
/*W*/
use dim;
read all var _NUM_ into W;
W=W*(&valuew);

/*WT*/
use dimc;
read all var _NUM_ into WT;
WT=WT*(&valuewt);

/*A*/
use dime;
read all var _NUM_ into A;
AT=t(A*(&valuea));

/*AY*/
use dimt;
read all var _NUM_ into AY;
AYT=(AY*(&valueay));

Xk=X[1+&&rate&r*(&k-1):&&rate&r*&k,];
Tk=T[1+&&rate&r*(&k-1):&&rate&r*&k,];
Ek=E[1+&&rate&r*(&k-1):&&rate&r*&k,];
Yk = (Xk*W*AT) + (Tk*WT*AT) + (Tk*AYT) + Ek;

/*QR decomposition di X*/
call qr(Q,Rx,piv,lindep,Xk);
Rxx=j(nrow(Xk)-nrow(Rx),ncol(Rx),0);
R=Rx//Rxx;
/*Ortogonalizzazione di T*/
TOrth=Tk-Xk*ginv(Xk)*Tk;

/*START ITERAZIONI*/
ciclo=&iter;
mustexit=0;
do while (mustexit=0);

oldAT=AT;
oldAYT=AYT;
oldW=W;
oldWT=WT;

/*ALGORITMO*/
/*STEP 1: Stima w°*/
/*vec(W)*/
vw=vectorise(W);
A=t(AT);
/*KRONECHER PROD*/
QT=t(Q);
omegad=A@R;
/*tolgo il contributo del legame diretto*/
Ylc=Yk-Tk*AYT;

i=1;
do while(i<=nrow(vw)); /*cancello zeri da vec(W) e rispettive colonne da
omegad*/
if vw[i]^=0 then i=i+1;
else do;
vw=delrow(vw,i);
omegad=delcol(omegad,i);
end;
end;

QTY=QT*Ylc;
vecQTY=vectorise(QTY);
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
/*stima del vettore di parametri w°*/
wupd=((inv(t(omegad)*omegad))*(t(omegad)))*vecQTY;
/*ricostruzione di W*/
W=update(W,wupd);

/*STEP 2: Stima Wt*/
/*vec(WT)*/
vwt=vectorise(WT);
A=t(AT);
/*KRONECHER PROD*/
omegadt=A@Torth;

/*tolgo il legame diretto e il legame tramite LC delle variabili esogene X*/
Ylc2=Ylc-(Xk*W*AT);

i=1;
do while(i<=nrow(vwt)); /*canello zeri da vec(WT) e rispettive colonne da
omegadt*/
  if vwt[i]^=0 then i=i+1;
else do;
  vwt=delrow(vwt,i);
  omegadt=delcol(omegadt,i);
end;
end;

vecYlc2=vectorise(Ylc2);
/*stima del vettore di parametri wt*/
wtupd=((inv(t(omegadt)*omegadt))*(t(omegadt)))*vecYlc2;
/*ricostruzione di WT*/
WT=update(WT,wtupd);

/*Costruzione di F e sua normalizzazione*/
F=Xk*W+Torth*WT;;
F=normalize(F);

/*STEP 3: Stima A*/
/*vec(A')*/
ad=vectorise(AT);
Id=I(ncol(AT));
/*KRONECHER PROD*/
gammad=Id@F;
vecYlc=vectorise(Ylc);

i=1;
do while(i<=nrow(ad)); /*canello zeri da vec(A') e rispettive colonne da
gammad*/
  if ad[i]^=0 then i=i+1;
else do;
  ad=delrow(ad,i);
  gammad=delcol(gammad,i);
end;
end;
/*stima del vettore di parametri a*/
aupd=((inv(t(gammad)*gammad))*(t(gammad)))*vecYlc;
/*ricostruzione di A'*/
AT=update(AT,aupd);
/*STEP 4: Stima AY*/
/*Sottraggo alla matrice endogena la costruzione tramite LC
F = X W + T WT*/
Y2=Yk-F*AT;

/*Regressione multipla con coefficienti fissi di T su Y*/
vay=vectorise(AYT);
thetad=Id@Tk;
```

```
vY2=vectorise(Y2);

i=1;
do while(i<=nrow(vay)); /*cancello zeri da vec(Ay') e rispettive colonne da
thetad*/
if vay[i]^=0 then i=i+1;
else do;
  vay=delrow(vay,i);
  thetad=delcol(thetad,i);
end;
end;

/*stima del vettore di parametri ay*/
ayupd=((inv(t(thetad)*thetad))*(t(thetad)))*vY2;
/*ricostruzione di Ay*/
AYT=update(AYT,ayupd);

/*ALGORITMO END*/
/*Criterio di accostamento*/
diffW=abs(W-oldW);
diffWT=abs(WT-oldWT);
diffAT=abs(AT-oldAT);
diffAYT=abs(AYT-oldAYT);

if &iter^=0 then do;

flgw=1;flgat=1;flgwt=1; flgay=1;
  do i=1 to nrow(diffW);
  do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
  end;
end;
  do i=1 to nrow(diffWT);
  do j=1 to ncol(diffWT);
    if diffWT[i,j]>&threshold then flgwt=0;
  end;
end;
  do i=1 to nrow(diffAT);
  do j=1 to ncol(diffAT);
    if diffAT[i,j]>&threshold then flgat=0;
  end;
end;
  do i=1 to nrow(diffAYT);
  do j=1 to ncol(diffAYT);
    if diffAYT[i,j]>&threshold then flgay=0;
  end;
end;
  ciclo=ciclo-1;
  if ciclo=0 | (flgw=1 & flgat=1 & flgwt=1 & flgay=1) then mustexit=1;
end;

if &iter=0 then do;
flgw=1;flgat=1;flgwt=1; flgay=1;
  do i=1 to nrow(diffW);
  do j=1 to ncol(diffW);
    if diffW[i,j]>&threshold then flgw=0;
  end;
end;
  do i=1 to nrow(diffWT);
  do j=1 to ncol(diffWT);
    if diffWT[i,j]>&threshold then flgwt=0;
  end;
end;
  do i=1 to nrow(diffAT);
  do j=1 to ncol(diffAT);
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
        if diffAT[i,j]>&threshold then flgat=0;
    end;
    end;
    do i=1 to nrow(diffAYT);
    do j=1 to ncol(diffAYT);
        if diffAYT[i,j]>&threshold then flgay=0;
    end;
    end;
if flgw=1 & flgat=1 & flgwt=1 & flgay=1 then mustexit=1;
end;

end;

A=t(AT);
AY=t(AYT);

/*Back Transformation a W da W°*/
XT_BT=(ginv(Xk)*Tk; /*X*T*/
vecWTr=vectorise(WT); /*vec(Wt)*/
vecWr=vectorise(W); /*vec(W°)*/
DiagW=I(ncol(W));
BTr=DiagW@XT_BT; /*I x (X*T), "x" inteso come prodotto di kronecker*/

i=1;
do while(i<=nrow(vecWr));
if vecWr[i]^=0 then i=i+1;
else do;
    vecWr=delrow(vecWr,i); /*cancello righe nulle da vec(W°)*/
    BTr=delrow(BTr,i); /*cancello corrispettive RIGHE da I x (X*T)*/
end;
end;

i=1;
do while(i<=nrow(vecWTr));
if vecWTr[i]^=0 then i=i+1;
else do;
    vecWTr=delrow(vecWTr,i); /*cancello righe nulle da vec(W°)*/
    BTr=delcol(BTr,i); /*cancello corrispettive COLONNE da I x (X*T)*/
end;
end;

vecWnew=vecWr-(BTr)*vecWTr; /*costruisco W con back transformation in forma
vettorizzata*/
W=update(W,vecWnew); /*rimetto i coefficienti al posto corretto nella W back-
transformation*/

vW=vectorise(W);
vWT=vectorise(WT);
vA=vectorise(A);
vAY=vectorise(AY);

i=1;
do while(i<=nrow(vW));
    if vW[i]^=0 then i=i+1;
else do;
    vW=delrow(vW,i);
end;
end;

i=1;
do while(i<=nrow(vWT));
    if vWT[i]^=0 then i=i+1;
else do;
    vWT=delrow(vWT,i);
end;
end;
```

```
end;

i=1;
do while(i<=nrow(vA));
  if vA[i]^=0 then i=i+1;
else do;
  vA=delrow(vA,i);
end;
end;

i=1;
do while(i<=nrow(vAY));
  if vAY[i]^=0 then i=i+1;
else do;
  vAY=delrow(vAY,i);
end;
end;

sumW=sumW+vW;
sumWT=sumWT+vWT;
sumA=sumA+vA;
sumAY=sumAY+vAY;
sumWsqr=sumWsqr+(vW#vW);
sumWTsqr=sumWTsqr+(vWT#vWT);
sumAsqr=sumAsqr+(vA#vA);
sumAYsqr=sumAYsqr+(vAY#vAY);
vpW=vW*0+&valuew;
vpWT=vWT*0+&valuewt;
vpA=vA*0+&valuea;
vpAY=vAY*0+&valueay;

vecmcoef=vW/vWT;
vecmcoef=vecmcoef/vA;
vecmcoef=vecmcoef/vAY;
vecp=vpW/vpWT;
vecp=vecp/vpA;
vecp=vecp/vpAY;

congr=(t(vecp)*vecmcoef)/((sqrt(t(vecp)*vecp))*(sqrt(t(vecmcoef)*vecmcoef)));
vcongr&r[&k,1]=&k;
vcongr&r[&k,2]=&rate&r;
vcongr&r[&k,3]=congr;

sumc=sumc+congr;

%end;

MeanvW=sumW/&replc;
MeanvWT=sumWT/&replc;
MeanvA=sumA/&replc;
MeanvAY=sumAY/&replc;

MeanvWsqr=sumWsqr/&replc;
MeanvWTsqr=sumWTsqr/&replc;
MeanvAsqr=sumAsqr/&replc;
MeanvAYsqr=sumAYsqr/&replc;

seW=((MeanvWsqr-(MeanvW#MeanvW))*((&replc)/(&replc-1)))##0.5;
seWT=((MeanvWTsqr-(MeanvWT#MeanvWT))*((&replc)/(&replc-1)))##0.5;
seA=((MeanvAsqr-(MeanvA#MeanvA))*((&replc)/(&replc-1)))##0.5;
seAY=((MeanvAYsqr-(MeanvAY#MeanvAY))*((&replc)/(&replc-1)))##0.5;

estimates = MeanvW//MeanvWT;
estimates = estimates//MeanvA;
estimates = estimates//MeanvAY;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
stderr = seW//seWT;
stderr = stderr//seA;
stderr = stderr//seAY;

meancongr=sumc/&replic;
biases=100*abs(estimated-vecp)/vecp;
outputM&&rate&r=estimated||stderr;
outputM&&rate&r=outputM&&rate&r||biases;
num={&&rate&r 0 0 };
congruence=meancongr*{1 0 0};
outputM&&rate&r=num//outputM&&rate&r;
outputM&&rate&r=outputM&&rate&r//congruence;
outputM&&rate&r=round(outputM&&rate&r,0.001);

names={'est' 'stderr' 'bias'};
trail=j(1,3,char(&&rate&r));
collabel=compbl(names + trail);

create results&r from outputM&&rate&r [colname = collabel];
append from outputM&&rate&r;
create results_rep&r from vcongr&r [colname = {'rep' 'n' 'congr'}];
append from vcongr&r;

%put n= &&rate&r eseguita;

%end;
leftcW=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="w"+char(j);
chj=COMPBL(chj);
leftcW[j]=chj;
end;
leftcWT=j(nrow(meanvWT),1,"xxxx");
do j=1 to nrow(meanvWT);
chj="wt"+char(j);
chj=COMPBL(chj);
leftcWT[j]=chj;
end;
leftcA=j(nrow(meanvW),1,"xxx");
do j=1 to nrow(meanvW);
chj="a"+char(j);
chj=COMPBL(chj);
leftcA[j]=chj;
end;
leftcAY=j(nrow(meanvAY),1,"xxxx");
do j=1 to nrow(meanvAY);
chj="ay"+char(j);
chj=COMPBL(chj);
leftcAY[j]=chj;
end;

parm="n"//leftcW;
parm=parm//leftcWT;
parm=parm//leftcA;
parm=parm//leftcAY;
parm=parm//"congr";

pop=0//vecp;
pop=pop//0;

create results from parm [colname = {'label'}];
append from parm;

create pop from pop [colname = {'pop'}];
```

```
append from pop;

run;
quit;

data results;
merge results pop;
run;

data results;
merge results results1;
run;
data results;
merge results results2;
run;
data results;
merge results results3;
run;
data results;
merge results results4;
run;

data Res_Simul_GRA_NNORM; set results;
array change _numeric_;
do over change;
if change=0 then change="";
end;
run;

data Res_Simul_GRA_NNORM_rep;
set Results_rep1 Results_rep2 Results_rep3 Results_rep4; run;

data E1; set E1;
%rename_e(COL,e0);
n=1;
RUN;
data E2; set E2;
%rename_e(COL,e0);
n=2;
RUN;
data E3; set E3;
%rename_e(COL,e0);
n=3;
RUN;
data E4; set E4;
%rename_e(COL,e0);
n=4;
RUN;

proc datasets;
save Res_Simul_GRA_NNORM Res_Simul_GRA_NNORM_rep E1 E2 E3 E4 E1M Z1M;run;

data E;
set E1 E2 E3 E4;
run;

proc univariate data=Res_Simul_GRA_NNORM_rep;
var congr;
by n;
run;

proc univariate data=e;
histogram e01 e02/ normal;
var e01 e02;
by n;
```

Specification and Estimation of Complex Structural Models with Covariate Effects

```
run;
```

```
quit;
```

```
%mend;
```

```
%mcsamples(varcovx, meansx, varcovt, meanst, varcove, meanse,  
skewkurt, 0.6, 0.3, 0.2, 0.4, 100, 200, 400, 600, 1000, 10000, 0.0003);
```