SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Dipartimento di / Department of

Statistica e Metodi Quantitativi

Dottorato di Ricerca in / PhD program Statistica e Matematica per la Finanza Ciclo / Cycle XXIX

Curriculum in (se presente / if it is) Statistica

Redundancy Analysis Models with Categorical Endogenous Variables: New Estimation Techniques Based on Vector GLM and Artificial Neural Networks

Cognome / Surname Vacca Nome / Name Gianmarco

Matricola / Registration number 713296

Tutore / Tutor: Pietro Giorgio Lovaglio

Cotutore / Co-tutor:
(se presente / if there is one)

Supervisor:
(se presente / if there is one)

Coordinatore / Coordinator: Giorgio Vittadini

ANNO ACCADEMICO / ACADEMIC YEAR 2016/2017

Redundancy Analysis Models with Categorical Endogenous Variables: New Estimation Techniques Based on Vector GLM and Artificial Neural Networks

Gianmarco Vacca

Department of Statistics and Quantitative Methods

University of Milano-Bicocca

Doctoral School in Statistics and Mathematics for Finance

Ph. D. in Statistics - XXIX Cycle

Contents

Introduction

Structural Equation Models with latent variables have considerably developed in recent years. Starting from the pioneers of the two most prominent ways of defining models with latent variables, namely Covariance Structure Analysis and Component Analysis, with LISREL [J"oreskog, 1970] and Partial Least Squares Path Modeling [Wold, 1975; Lohm"oller, 1989] as the most famous techniques, several extensions and improvements have been put forward. Moreover, for Redundancy Analysis [Van den Wollenberg, 1977] models, which are part of the Component Analysis framework, but have only observed endogenous variables, new methods have been proposed in literature to deal with more than one group of exogenous observed variables, with simple linear equations and a unified optimization problem.

One main criticism, that has been dealt with recently in new strands of literature regarding Structural Equation Modeling, is the partial inability of these systems of linear equations to deal with categorical indicators. Several methods have been proposed, either related to Optimal Scaling [Young, 1981], or adapting the EM algorithm [Dempster et al., 1977] to the particular case under examination.

In the Redundancy Analysis framework, with only observed endogenous variables, the possibility of extending the estimation procedures to a qualitative setting is considerably less hampered by model restrictions, even more so in the Extended Redundancy Analysis [Takane and Hwang, 2005] model, with more than one block of exogenous variables. This work will hence present two novel estimation techniques for Extended Redundancy Analysis models in presence of binary or categorical endogenous variables: one that will make use of a modification of the Iterated Reweighed Least Squares algorithm, and one that will employ the Gradient Descent algorithm with backpropagation in an Artificial Neural Network architecture. For the latter, recent developments in Structural Equation Models in the neural networks setting will be firstly examined, and the new technique will be subsequently introduced. The promising feature of this new methods lies in the advantage of not having to resort to Optimal Scaling as a proxy to redefine categorical endogenous variables, with a proper optimization algorithm tailored for binary, multinomial or

multivariate responses.

The first chapter will give the main definitions in the Structural Equation Models setting, with its ramifications and subcategories, also pointing out the limitations and criticisms of each particular model; The second chapter will present the existing extensions of these models in the case of categorical endogenous variables; the third chapter will present the newly proposed estimation procedures for Extended Redundancy Analysis models with categorical indicators, which is the core novelty of this work; the fourth chapter will present simulation studies on the new proposition, which will examine both the capability of the models to recover the real parameter values, and compare the two estimation algorithms, also in terms of predicted probabilities; the fifth chapter illustrates an empirical example to which the new models presented in this work will be applied, focusing on the comparison of different estimation strategies; the sixth chapter offers conclusions and further possible developments related to the scope of this work.

Part I

Structural Equation Models with Latent Variables: definitions, characteristics and comparisons

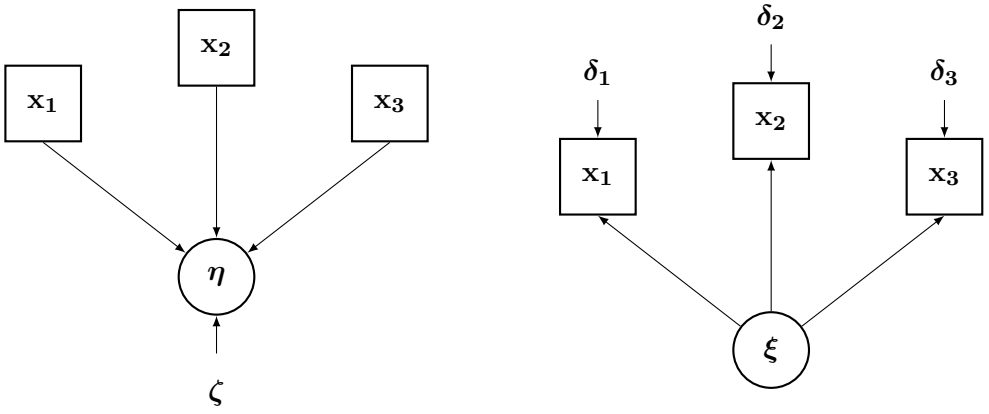
1 Structural Equation Models

1.1 Introduction

Structural Equation Models (SEM) comprise a wide group of methods used to represent hypotheses about means, variances, and covariances of observed data in terms of a smaller number of (structural) parameters defined by a theoretical model. The distinction is between observed variables and latent constructs, which are either measurable variables affected by errors which make them not observable, or theoretical constructs by themselves not measurable or observable (e.g. customer satisfaction). SEM, in their standard formulation, are particular type of linear models which can encompass both relationships among observed variable, as in Path Analysis [Wright, 1921; Wright, 1934; Alwin and Hauser, 1975], and employing also relationships among latent variables [J"oreskog, 1970; Lohm"oller, 1989; Wold, 1966; Wold, 1975; Wold, 1982]. In fact, SEM tries to conciliate both the factor/component analysis framework and the path analysis/simultaneous equation modeling framework.

Latent (unobserved) variables in SEM can be defined either as linear composites of their manifest (observed) measurements or as underlying their manifest measurements. In the first case, the measurement model is defined as formative (left panel of Figure 1), whereas in the second, the measurement model is defined as reflective (right panel of Figure 1) [Fornell and Bookstein, 1982; Bagozzi and Fornell, 1982; Edwards and Bagozzi, 2000].

Formative Scheme *Reflective Scheme*



$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\gamma} + \boldsymbol{\zeta} \qquad \mathbf{X} = \boldsymbol{\xi}\boldsymbol{\lambda}' + \boldsymbol{\Delta}$$

Figure 1: SEM defining schemes

In particular, formative and reflective measurement models have the following differences:

Formative Measurement Model

- Causality from the measures to the latent construct
- High correlation among indicators is unnecessary
- Eliminating an indicator from the construct may affect the construct significance
- The measurement error is at the construct level.

Reflective Measurement Model

- Causality from the latent construct to the measures.
- Usually high correlation among indicators.
- Eliminating an indicator from the construct does not affect the construct significance.
- The measurement error is at the indicator level.

SEM involve linear relationship among those latent variables, in the so called structural model, comprised of exogenous (latent) variables, which are considered a cause of the behavior of the endogenous (latent) variables.

Specifically:

- Exogenous observed variables are denoted with $\mathbf{X}$.
- Endogenous observed variables are denoted with $\mathbf{Y}$.
- Exogenous latent variables are denoted with $\boldsymbol{\xi}$, and measurement coefficients $\boldsymbol{\lambda}^X$.
- Endogenous latent variables are denoted with $\boldsymbol{\eta}$, and measurement coefficients $\boldsymbol{\lambda}^Y$.
- Coefficients between $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are denoted with $\boldsymbol{\gamma}$
- Coefficients between $\boldsymbol{\eta}$ and $\boldsymbol{\eta}$ are denoted with $\boldsymbol{\beta}$
- Errors for $\mathbf{X}$ are denoted with $\boldsymbol{\delta}$, whose covariance matrix is $\boldsymbol{\Theta}^\delta$.
- Errors for $\mathbf{Y}$ are denoted with $\boldsymbol{\epsilon}$, whose covariance matrix is $\boldsymbol{\Theta}^\epsilon$.
- Errors for $\boldsymbol{\eta}$ are denoted with $\boldsymbol{\zeta}$.
- Covariance matrix of $\boldsymbol{\xi}$ is denoted as $\boldsymbol{\Phi}$.

Figure 2 shows an example of structural model representation. The unaccounted causes of exogenous variables are not represented in the model, being unmeasured (or unknown), and exogenous variables have unrestricted covariances (i.e. not defined by a modeled relationship). On the contrary, the presumed measured causes of endogenous variables are explicitly considered in the model and therein estimated. To estimate SEM coefficients, two strands of literature have been established: Covariance Structure Analysis (CSA), where the first and most famous model is the LISREL Model [J“oreskog, 1970] and Partial Least Squares Path Modeling [PLS-PM; Wold, 1975; Lohm“oller, 1989]. CSA and PLS deal with SEM from different viewpoints: the former has its roots in Confirmatory Factor Analysis, whereas the latter falls under Component Analysis (CA) models.

1.2 Covariance-based SEM

The LISREL model [J“oreskog, 1978; J“oreskog and S“orbom, 1982] is the most common approach in CSA, dealing simultaneously with the estimation of the measurement model of the LV and the estimation of the structural parameters' causal

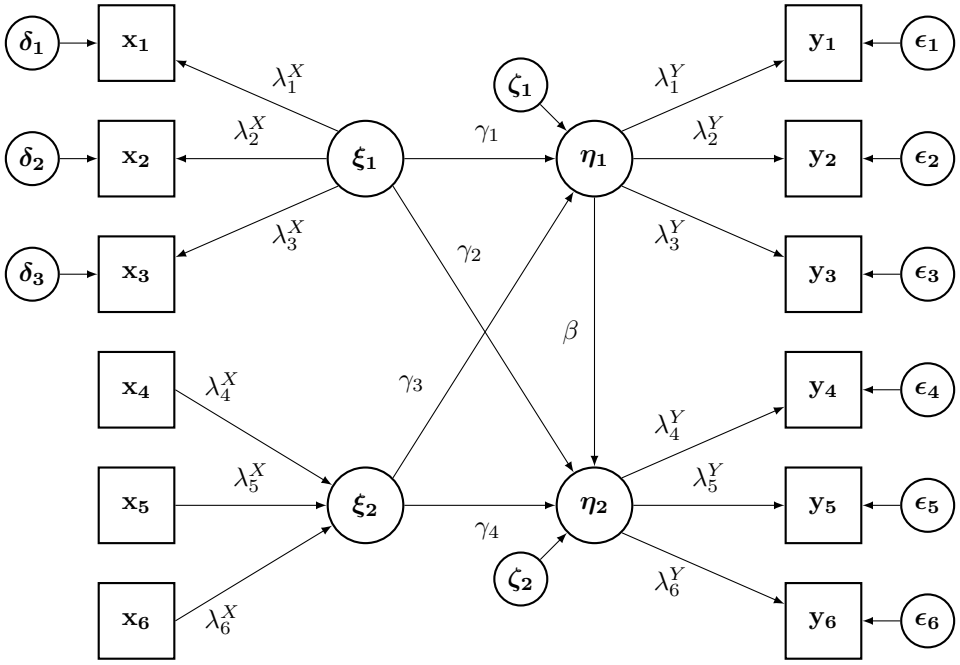


Figure 2: SEM path diagram example

links, using a well defined optimum criterion.

The structural model, using the variables and parameters categorization defined above, is

$$\boldsymbol{\eta} = \mathbf{B}\boldsymbol{\eta} + \boldsymbol{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta} \tag{1}$$

while the measurement model for the observed variables is

$$\mathbf{y} = \boldsymbol{\Lambda}^Y \boldsymbol{\eta} + \boldsymbol{\epsilon} \tag{2}$$

$$\mathbf{x} = \boldsymbol{\Lambda}^X \boldsymbol{\xi} + \boldsymbol{\delta} \tag{3}$$

From the model assumed above, the covariance of $(\mathbf{y}', \mathbf{x}')$ then is

$$\boldsymbol{\Sigma} = \begin{bmatrix} (\boldsymbol{\Lambda}^Y(\mathbf{I}-\mathbf{B})^{-1})(\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Gamma}' + \boldsymbol{\Psi})(\boldsymbol{\Lambda}^Y(\mathbf{I}-\mathbf{B})^{-1})' + \boldsymbol{\Theta}^\epsilon & (\boldsymbol{\Lambda}^Y(\mathbf{I}-\mathbf{B})^{-1})\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Lambda}^X \\ \boldsymbol{\Lambda}^{X'}\boldsymbol{\Phi}\boldsymbol{\Gamma}'(\boldsymbol{\Lambda}^Y(\mathbf{I}-\mathbf{B})^{-1})' & \boldsymbol{\Lambda}^X\boldsymbol{\Phi}\boldsymbol{\Lambda}^{X'} + \boldsymbol{\Theta}^\delta \end{bmatrix}$$

The latter specification permits flexibility in defining fixed or constrained parameters (allowing also for model identification).
The main estimation methods are either based on unweighted least square criterion or maximum likelihood criterion, to find the parameter matrices that best approximate the sample covariance matrix $\mathbf{S}$.

$$f_{ULS} = \frac{1}{2}\text{tr}[(\mathbf{S} - \boldsymbol{\Sigma})'(\mathbf{S} - \boldsymbol{\Sigma})]$$

$$f_{ML} = \log|\boldsymbol{\Sigma}| + \text{tr}[(\mathbf{S}\boldsymbol{\Sigma}^{-1}) - \log|\mathbf{S}| - (p+q)$$

The estimation method is iterative, using the Fletcher-Powell algorithm [Fletcher and Powell, 1963]. Fit indices are either based on chi-square statistic for ML estimation, GFI for ULS estimation, or on the determinant of the fitted covariance matrix (Q index).

$$\chi^2 = (n-1)f_{ML}$$

$$GFI = 1 - \frac{\text{tr}[(\mathbf{S} - \hat{\boldsymbol{\Sigma}})'(\mathbf{S} - \hat{\boldsymbol{\Sigma}})]}{\text{tr}[\mathbf{S}'\mathbf{S}]}$$

$$Q = \frac{|\mathbf{S}|}{|\hat{\boldsymbol{\Sigma}}|}$$

1.3 Component Analysis SEM and PLS Path Modeling

Since the CSA framework require distributional assumptions and has a moderate level of complexity, alternative "soft modelling" approaches in the framework of component analysis [CA; Meredith and Millsap, 1985] have been developed.
Partial Least Squares, the most famous method in CA, firstly introduced by Wold (1975) under the name NIPALS (Nonlinear Iterative PArtial Least Squares), focuses on maximizing the variance of the dependent variables (either observed or unobserved) explained by the independent ones (either observed or unobserved), while CSA aims to reproduce the covariance matrix. PLS-Path Models (PLS-PM) are formally defined by two sets of linear equations: an inner model (the CA equivalent of structural model), focused on the relationships between the LVs, and an outer model (the CA equivalent of measurement model) which models the relation-

ships between each LV and its manifest indicators (MVs); a third component is also present, the weight relationships, which is used to estimate case-values for the LVs [Chin, 1998].

For simplicity, let us suppose that all the LVs are indexed as $\boldsymbol{\xi}_h$, and all the LV coefficients are indexed with $\boldsymbol{\beta}$. The inner model connections directed to $\boldsymbol{\xi}_h$ are defined as

$$\boldsymbol{\xi}_h = \sum_{j \neq h} \boldsymbol{\xi}_j \beta_j + \boldsymbol{\zeta}_h$$

while the outer model for the h-th LV can be either reflective (Mode A) [Tenenhaus and Tenenhaus, 2011]

$$\mathbf{X}_h = \boldsymbol{\xi}_h \mathbf{w}_h' + \boldsymbol{\epsilon}_h$$

or formative (Mode B) [Hanafi, 2007]. Note that formative way is not readily available for LISREL.

$$\boldsymbol{\xi}_h = \mathbf{X}_h \mathbf{w}_h + \boldsymbol{\delta}_h$$

The estimation of PLS-PM fits inner and outer model sequentially, using a fixed point algorithm, until inner and outer LV estimates reach an equilibrium [Tenenhaus et al., 2005]. Then the final estimates of the inner and outer coefficients is carried out.

- Outer estimate: the estimate $\mathbf{y}_h$ of the LV $\boldsymbol{\xi}_h$ is

$$\mathbf{y}_h \propto \sum_j w_{jh} \mathbf{x}_{jh}$$

  For Mode A, the weights are estimated as

$$w_{jh} = \mathbb{C}\text{ov}(\mathbf{x}_{jh}, \mathbf{z}_h)$$

  For Mode B, the weights are estimated as

$$\mathbf{w}_h = (\mathbf{X}_h' \mathbf{X}_h)^{-1} \mathbf{X}_h', \mathbf{z}_h$$

where $\mathbf{z}_h$ is the inner estimate of the LV $\boldsymbol{\xi}_h$.

- Inner estimate: the estimate $\mathbf{z}_h$ of the LV $\boldsymbol{\xi}_h$ is

$$\mathbf{z}_h \propto \sum_{j:\boldsymbol{\xi}_j \text{ is connected to } \boldsymbol{\xi}_h} e_{jh}\mathbf{y}_j$$

hence, the inner weights are estimated via usual OLS regression.

PLS-PM works without distributional assumptions and its path diagrams are estimated with a very easy algorithm. Moreover, it can be used with a few individuals and lots of variables, and the LV estimates have a practical meaning, not suffering from the improper solutions and indeterminacy that plagues CSA based SEM. However, despite evident benefits in both CSA and PLS methodologies to fit SEM, both present unsolved issues that may hamper their applicability.

1.4 Redundancy Analysis Models

Within the CA framework, Redundancy Analysis [Van den Wollenberg, 1977] is the simplest type of structural-equation model between two sets of observed variables, in which latent variables are intended as components. The aim of RA is to extract a series of linear components from a set of exogenous variables in such a way that they are mutually orthogonal and successively account for the maximum variance of a set of endogenous variables. In this framework, RA may be viewed as special type of structural-equation model, where: (1) a formative relationship is always assumed between the unobserved and observed (exogenous) variables, and (2) endogenous variables are always observed ones.

Recently, new methods to estimate Redundancy based SEM have been proposed: Multiblock Redundancy Analysis [MbRa; Bougeard et al., 2011], and the so-called Extended Redundancy Analysis [ERA; Takane and Hwang, 2005; Lovaglio and Vacca, 2016a for the related software], which generalizes RA for more than two blocks. In ERA, the relationships between the observed exogenous variables and the observed endogenous variables are moderated by the presence of linear compos-

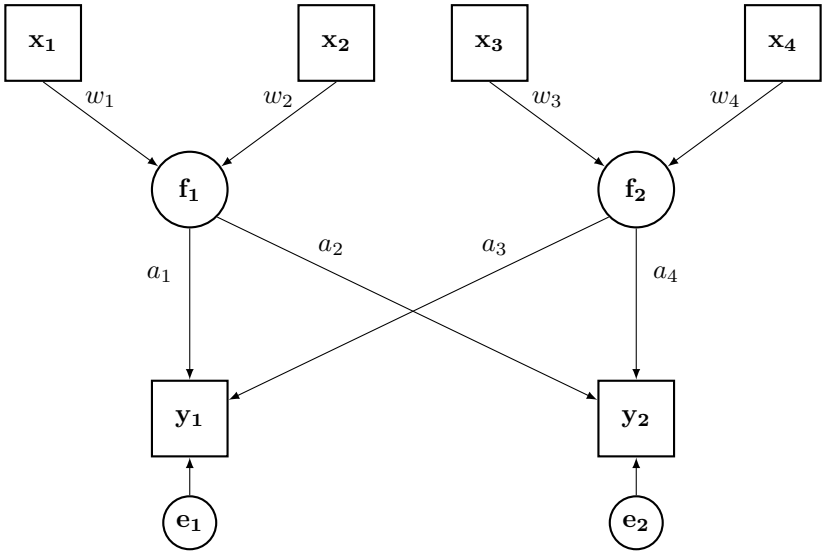


Figure 3: ERA path diagram example

ites (hereinafter LC): LCs are estimated as exact linear combinations of formative indicators, and both component weights and component loadings are estimated by consistently minimizing a single optimum criterion.

1.4.1 ERA Model specification

Suppose we have a $n \times p$ matrix $\mathbf{Y}$ of endogenous variables and a $n \times q$ matrix $\mathbf{X}$ of exogenous variables, both centered and with variance scaled to unit. If a variable is exogenous as well as endogenous it is included in both $\mathbf{Y}$ and $\mathbf{X}$. The relationship between variables is

$$\mathbf{Y} = \mathbf{XWA}' + \mathbf{E} = \mathbf{FA}' + \mathbf{E}$$

subject to rank constraint

$$\mathrm{r}(\mathbf{WA}') \leq \min\{q, p\}$$

where $\mathbf{W}$ is a component weight matrix, $\mathbf{A}'$ is a loading matrix, $\mathbf{E}$ is the residual matrix, and $\mathbf{F}$ is the component scores matrix with diagonal identification restriction, $\mathrm{diag}(\mathbf{F}'\mathbf{F}) = \mathbf{I}$. The model may be exemplified by the graph in Figure 3.

The SSQ objective function (SSQ($\mathbf{X}$)=tr($\mathbf{X}'\mathbf{X}$)) is

$$\begin{cases} \min\limits_{\mathbf{W},\mathbf{A}'} \text{SSQ}(\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{A}') = \min\limits_{\mathbf{W},\mathbf{A}'} \text{SSQ}(\mathbf{Y} - \mathbf{F}\mathbf{A}') \\ \text{sub diag}[(\mathbf{X}\mathbf{W})'(\mathbf{X}\mathbf{W})] = \text{sub diag}[\mathbf{F}'\mathbf{F}] = \mathbf{I} \end{cases}$$

to be minimized with an Alternating Least Squares estimation algorithm [ALS; Kiers and ten Berge, 1989]. The above may be expressed also as

$$\begin{aligned} \text{SSQ}(\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{A}') &= \text{SSQ}\ [\text{vec}(\mathbf{Y}) - \text{vec}(\mathbf{X}\mathbf{W}\mathbf{A}')] && (4) \\ &= \text{SSQ}\ [\text{vec}(\mathbf{Y}) - (\mathbf{A} \otimes \mathbf{X})\text{vec}(\mathbf{W})] && (5) \\ &= \text{SSQ}\ \left[\text{vec}(\mathbf{Y}) - (\mathbf{I} \otimes \mathbf{F})\text{vec}(\mathbf{A}^T)\right] && (6) \end{aligned}$$

where "$\otimes$" denotes the Kronecker product.

1.4.2 Parameter estimation

The estimate of $\mathbf{W}$ and $\mathbf{A}$ follows the aforementioned minimization criterion, with an ALS algorithm whose two steps are iterated until convergence:

1. ***Estimate of*** $\mathbf{W}$ ***for fixed*** $\mathbf{A}'$ *(Equation 5)*

   let $\mathbf{w}$ be the vector obtained by eliminating the zero elements of vec($\mathbf{W}$), and let $\mathbf{\Omega}$ be the matrix obtained by eliminating the columns of $\mathbf{A} \otimes \mathbf{X}$ corresponding to the zero elements of vec($\mathbf{W}$). The LS estimate of $\mathbf{w}$ therefore is

   $$\tilde{\mathbf{w}} = (\mathbf{\Omega}'\mathbf{\Omega})^{-1}\mathbf{\Omega}'\text{vec}(\mathbf{X}) \qquad (7)$$

   $\tilde{\mathbf{W}}$ is then reconstructed from $\tilde{\mathbf{w}}$ and $\mathbf{F}$ is normalised to respect the identification restriction.

2. ***Estimate of*** $\mathbf{A}'$ ***for fixed*** $\mathbf{W}$ *(Equation 6)*

   let $\mathbf{a}$ be the vector obtained by eliminating the zero element of vec($\mathbf{A}'$), and let $\mathbf{\Gamma}$ be the matrix obtained by eliminating the columns of $\mathbf{I} \otimes \mathbf{F}$ corresponding

to the zero elements of vec($\mathbf{A}'$). The LS estimate of $\mathbf{a}$ subsequently is

$$\tilde{\mathbf{a}} = (\mathbf{\Gamma}'\mathbf{\Gamma})^{-1}\mathbf{\Gamma}'\text{vec}(\mathbf{Y}) \tag{8}$$

$\tilde{\mathbf{A}}'$ is then reconstructed from $\tilde{\mathbf{a}}$.

The goodness-of-fit is the usual

$$\Psi = 1 - \frac{\text{SSQ}((\mathbf{Y} - \mathbf{X}\tilde{\mathbf{W}}\tilde{\mathbf{A}}'))}{\text{SSQ}(\mathbf{Y})}$$

Bootstrap algorithm may be implemented to assess the reliability of the parameter estimates, as well as to obtain their standard errors [Efron, 1982] or, alternatively, conventional Maximum Likelihood Estimation can be used, assuming matrix-normality on $\mathbf{E}$, employing an Alternating Maximum Likelihood approach.

2 Limits of CSA, PLS-PM and SEM-RA models

2.1 CSA Models limitations

CSA estimates model parameters via Maximum Likelihood, following the assumption of multivariate normality of the variables. However, such a distributional assumption is often violated. More problematically, estimates can be improper (e.g., negative variance estimates), and factor scores or latent variable scores are indeterminate. However, the main limitation of CSA SEM (in the 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 [Steiger and Sch"onemann, 1975; Sch"onemann, 1971; Vittadini, 1989; Vittadini et al., 2007]. Due to this, LVs and errors are not unique even with a precise identification of the model [Steiger and Sch"onemann, 1975; Vittadini, 1989; Hwang and Takane, 2004]. Furthermore, there is no necessary and sufficient condition available for model identification, reason for which it is 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" [J"oreskog, 1988].

2.2 PLS-PM limitations

The main intention of PLS-PM is to predict in situations of low theoretical information, not emphasizing confirmatory analysis [McDonald, 1996; Garthwaite, 1994; Tenenhaus et al., 2005]. Secondly, PLS algorithm, despite achieving the goal of prediction, suffers from logic inconsistency: the weights that define the composite scores are calculated via linear regressions, which may not respect the role of the observed variables in the measurement models [Vittadini et al., 2007; Fattore et al., 2012]. For example, in presence of a LV with reflective indicators, composite scores are obtained by treating the observed variables as formative. Another problem of PLS-PM is the so called "consistency at large". Because the latent scores of the LVs are a linear combination of the MVs that involve measurement error, they must be regarded as biased [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]. In addition to this, PLS-PM does not solve a global optimization problem, which makes difficult the evaluation of an overall model fit, and does not allow for imposing value or equality constraints on path coefficients. A solution to this problem has been presented with the Generalized Structured Component Analysis [GSCA; Hwang and Takane, 2004], but this new model still has to establish itself as a valid alternative to PLS-PM [Henseler, 2012], despite being capable to fit an optimum criterion.

2.3 ERA limitations

The ERA model has been subject to several extensions, among which restriction on the composite weights [DeSarbo et al., 2015], application on functional data [Hwang et al., 2015] and the inclusion of concomitant indicators (i.e. exogenous variables that may have an effect both on LCs and endogenous observed variables) managing to separate the contribution of those variable and the fully exogenous

variables to the formation of the latent composites [Lovaglio and Vittadini, 2014; Lovaglio and Vacca, 2016b for the related software]. However, the current definition of the ERA model is purely descriptive, not relying on inferential aspects if not by aid of bootstrap techniques and, for the constrained version in DeSarbo et al., ridge restrictions on composite weights proposed in the literature deal with all the weights simultaneously, failing to separate internal selection of weights in each latent composite.Furthermore, in presence of categorical endogenous variables, the only estimation of the model existing in literature is achieved via adaptation of the Optimal Scaling algorithm [OS; see Hwang and Takane, 2002; Young, 1981].

Part II

SEM with categorical indicators

1 CSA with categorical indicators

As far as categorical indicators are concerned, CSA and PLS-PM have undergo significant extensions. In the definition of latent variables following completely reflective scheme, sets of only ordinal variables have been modeled either overlooking the approach typical of Latent Class Analysis [McCutcheon, 1987], using maximum likelihood estimation in combination with the EM algorithm [Moustaki, 2000; also accounting for external covariate effect in Moustaki, 2003]. More closely to the LISREL methodology, a maximum likelihood approach considering a mixture of dichotomous, ordinal and continuous indicators as manifestations of the latent variables has been used to estimate CSA models [Muthén, 1984], in a three-stages algorithm. In these situations, having defined the LV as a cause of its manifest indicators, likelihood approaches are indeed favorable, relying on the common definition of manifest variables $\mathbf{Y}$ as

$$f(\mathbf{Y}) = \int_{\mathbb{R}} \dots \int_{\mathbb{R}} g(\mathbf{Y}|\boldsymbol{\xi}) h(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

where $g(.)$ is the distribution of the manifest variables conditional on the latent variable $\boldsymbol{\xi}$, and $h(.)$ is the distribution of the latent variable $\boldsymbol{\xi}$ (in the case of LCA, integrals are replaced with sums.

2 PLS-PM with categorical indicators

For what concerns PLS-PM, given the possibility of having formative LV constructs, the role of categorical variates can be regarded from several points of view:

- Manifest categorical variables that have a moderation effect in the formation of the latent composite;

- Manifest categorical variables that concur in the formation of the LV, possibly altogether with continuous manifest variables.
- Latent categorical variables that create clusters of individual based on unobserved characteristics (Finite Mixture PLS [Herrmann et al., 2002; Trinchera, 2008; Esposito Vinzi et al., 2008]);

It is important to note that these methods, being closely related to PLS-PM, retain both advantages and drawbacks of the main methodology (i.e., lack of proper optimization and logic inconsistency).

2.1 The moderation effect

Moderation effects (e.g. the effect of variables such as gender) have firstly been dealt considering them as separate LVs, accounting also for their interaction with the continuous manifest variables of the other LVs [Chin et al., 2003], in the so-called product indicator approach. Consider for example a formative LV $\boldsymbol{\xi}_X$ with indicators $\mathbf{X}$, and let $\mathbf{m}$ be the moderating effect. To evaluate the effect of $\mathbf{m}$, two new LVs are constructed: $\boldsymbol{\xi}_m$, formed by $\mathbf{m}$, the moderating effect, and $\boldsymbol{\xi}_v$, formed by the product terms $v_{ij} = x_{ij}m_i$.
More appropriately, a two-step approach has been recently proposed in literature [Henseler and Fassott, 2010]: in the first step PLS-PM is fitted, regarding the exogenous and moderating variables as independent in the formation of $\boldsymbol{\xi}_X$ and $\boldsymbol{\xi}_m$; in the second step an interaction among the latent variable scores is produced, namely $\hat{\xi_{vi}} = \hat{\xi_{Xi}} * \hat{\xi_{mi}}$ and a new PLS-PM estimation is produced, incorporating this interaction term in the prediction of the endogenous latent variable.
In both cases, the categorical variable is not considered as a direct participant in the formation of the LV, but only as a confounding factor whose effect needs to be taken into consideration for each of its categories.

2.2 PLS-PM with nominal and ordinal indicators

In these situations, the estimation steps typical of PLS-PM are preceded by a quantification step that transforms the categorical variables into continuous ones

[Trinchera et al., 2008; Russolillo, 2012]. Specifically, in the Non-Metric PLS (NM-PLS) approach, a new quantification step is integrated in the estimation algorithm: let $\mathbf{x}^*$ be the observed nominal or ordinal variable, a scaling (numeric) value is assigned to each category k ($k = 1, ..., K \leq N$) of $\mathbf{x}^*$, such that it is coherent with the chosen scaling level and optimizing the model criterion. Each raw variable is transformed as $\mathbf{x} \propto \mathbf{X}\boldsymbol{\phi}$, where $\boldsymbol{\phi}' = (\phi_1, \ldots, \phi_k)$ is the vector of optimal scaling parameters and the matrix $\tilde{\mathbf{X}}$ defines a space in which the scaling constraint is respected. The scaling step is optimized by means of the ordinary least squares regression coefficients of the LC $\boldsymbol{\gamma}_{\mathbf{x}^*}$ on $\tilde{\mathbf{X}}$, i.e. by projecting the LC on the space spanned by the columns of $\tilde{\mathbf{X}}$. Each level of scaling (nominal or ordinal) has a corresponding scaling function $\mathcal{Q}(.)$, which is the projection operator of the LC in a suitable space spanned the columns of $\tilde{\mathbf{X}}$.

In nominal scaling

$$\mathcal{Q}(\tilde{\mathbf{X}}^n, \boldsymbol{\gamma}_{\mathbf{x}^*}) = \tilde{\mathbf{X}}^n\hat{\boldsymbol{\phi}} = \tilde{\mathbf{X}}^n(\tilde{\mathbf{X}}^{n\prime}\tilde{\mathbf{X}}^n)^{-1}\tilde{\mathbf{X}}^{n\prime}\boldsymbol{\gamma}_{\mathbf{x}^*}$$

Respecting the measurement restriction for every unit

$$(x_i^* \sim x_i'^*) \rightarrow (\hat{x}_i = \hat{x}_i')$$

where "$\sim$" means "belonging to the same category".

In ordinal scaling

$$\mathcal{Q}(\tilde{\mathbf{X}}^o, \boldsymbol{\gamma}_{\mathbf{x}^*}) = \tilde{\mathbf{X}}^o\hat{\boldsymbol{\phi}} = \tilde{\mathbf{X}}^o(\tilde{\mathbf{X}}^{o\prime}\tilde{\mathbf{X}}^o)^{-1}\tilde{\mathbf{X}}^{o\prime}\boldsymbol{\gamma}_{\mathbf{x}^*}$$

Respecting the measurement restrictions for every unit

$$(x_i^* \sim x_i'^*) \rightarrow (\hat{x}_i = \hat{x}_i') \quad \text{and} \quad (x_i^* \prec x_i'^*) \rightarrow (\hat{x}_i \leq \hat{x}_i')$$

where "$\prec$" indicates categorical order.

Outside the Optimal Scaling methods, variations on the original PLS-PM algorithm have been devised specifically to treat nominal categorical variables with reflective

measurement model [Jakobowicz and Derquenne, 2007], substituting linear fitting with ANOVA or logistic model for the outer estimates. To specifically treat ordinal categorical variables, the most recent developments to the PLS-PM methodology make use of the polychoric correlation matrix to fit the model, not having to resort to Optimal Scaling [Cantaluppi, 2012; Schuberth et al., 2016].

3 SEM-RA with categorical indicators

The original Redundancy Analysis model can be applied also in presence of categorical responses, either seeing the model as a particular case of Reduced Rank Regression [Izenman, 1975; Davies, 1982; van der Leeden, 1990], and extending it to the GLM class for multinomial responses [Yee and Hastie, 2003], or applying Optimal Scaling techniques [Israels, 1984].

3.1 ERA with categorical indicators

For the ERA model, only OS methods are avaliable, and the estimation is carried out adding a data transformation step to the ALS algorithm [see Hwang and Takane, 2002]. Specifically, let $\mathbf{Y}^S$ and $\mathbf{X}^S$ be the parametrized versions of the original data $\mathbf{Y}$ and $\mathbf{X}$. All the parameters are divided into model parameters and data parameters, which are alternately updated, and the parametrized variables are updated one at a time, respecting their measurement characteristics.
Let $\mathbf{z}_i$ be a variable either in $\mathbf{Y}$ or $\mathbf{X}$, so that $i = 1, \ldots, p+q$, and let $\mathbf{s}_i$ be a variable either in $\mathbf{Y}^S$ or $\mathbf{X}^S$. The LS criterion becomes

$$f = \text{SSQ}(\mathbf{Y}^S - \mathbf{X}^S\mathbf{W}\mathbf{A}') = \text{SSQ}(\mathbf{Y}^S - \mathbf{X}^S\mathbf{B}) \tag{9}$$

under the constraints $\text{diag}[\mathbf{W}'\mathbf{X}^{\mathbf{S}'}\mathbf{X}^S\mathbf{W}] = \mathbf{I}$, $\mathbf{s}_i'\mathbf{s}_i = 1$ and $\mathbf{s}_i = \xi(\mathbf{z}_i)$, where $\mathbf{B} = \mathbf{W}\mathbf{A}'$. The data transformation phase consists of two steps:

- **Step 1:** the model predictions of $\mathbf{s}_i$ is obtained minimizing Equation 9.
  Let $\mathbf{s}_g^Y$ and $\mathbf{s}_h^X$ be the g-th and the h-th variables in $\mathbf{Y}^S$ and $\mathbf{X}^S$ respectively,

and let $\tilde{\mathbf{s_i}}$ be the model prediction. Equation (4) can be expressed as

$$f = \text{SSQ}(\mathbf{s}_i\boldsymbol{\eta}' - (\boldsymbol{\Delta} - \boldsymbol{\Psi})) \tag{10}$$

Where $\boldsymbol{\eta}$, $\boldsymbol{\Delta}$ and $\boldsymbol{\Psi}$ are defined as follows: if $\mathbf{s}_i$ is shared between $\mathbf{Y}^S$ and $\mathbf{X}^S$, it is placed in the g-th column and in the h-th columns of $\mathbf{Y}^S$ and $\mathbf{X}^S$ respectively. Then, for $\mathbf{Y}^S$

$$\boldsymbol{\Delta} = \begin{cases} \mathbf{X}^S_{(h)}\mathbf{B}_{(h)} & \text{if } \mathbf{s}_i \text{ is shared} \\ \mathbf{X}^S\mathbf{B} & \text{otherwise} \end{cases} ; \boldsymbol{\Psi} = \mathbf{Y}^S_{(g)}; \quad \boldsymbol{\eta}' = \begin{cases} \mathbf{e}'_g - \mathbf{b}'_h & \text{if } \mathbf{s}_i \text{ is shared} \\ \mathbf{e}'_g & \text{otherwise} \end{cases}$$

and, for $\mathbf{X}^S$

$$\boldsymbol{\Delta} = \mathbf{X}^S_{(h)}\mathbf{B}_{(h)}; \qquad \boldsymbol{\Psi} = \mathbf{Y}^S; \qquad \boldsymbol{\eta}' = \mathbf{b}'_h$$

where $\mathbf{X}^S_{(h)}\mathbf{B}_{(h)}$ is the product of $\mathbf{X}^S$ whose h-th column is zero and $\mathbf{B}$ whose h-th row is zero, $\mathbf{Y}^S_{(g)}$ is the matrix $\mathbf{Y}^S$ with the g-th column replaced with zeroes, $\mathbf{e}'_g$ is an all-zeroes vector except for position g, $\mathbf{b}'_h$ is the h-th row of $\mathbf{B}$. The optimal prediction is then obtained by

$$\tilde{\mathbf{s}}_\mathbf{i} = (\boldsymbol{\Delta} - \boldsymbol{\Psi})\boldsymbol{\eta}(\boldsymbol{\eta}'\boldsymbol{\eta})^{-1}$$

- **Step 2:** $\mathbf{s}_i$ is transformed to maximize the relationship between $\mathbf{s}_i$ and the model predictions obtained in the previous step, hence it is transformed to be as close to $\tilde{\mathbf{s}}_\mathbf{i}$ as possible, under the appropriate measurement restriction. This gives the LS estimate of $\mathbf{s_i}$

$$\mathbf{s_i} = \boldsymbol{\Upsilon}_i(\boldsymbol{\Upsilon}'_i\boldsymbol{\Upsilon}_i)^{-1}\tilde{\mathbf{s}}_\mathbf{i}$$

with $\boldsymbol{\Upsilon}_i$ determined by the measurement restriction imposed on the original data columns (i.e., indicator matrix for nominal variables, whose element stands for category membership).

3.2 Limitations of optimal scaling for the ERA model and a new proposition

Estimation of ERA with categorical variables via OS suffers from the typical drawbacks of the method: the quality of scaling, and thus model performance, is susceptible to the number of categories and to the equality of interval width [see Lin, 2009]. Hence, this work is going to completely overcome potential setbacks of Optimal Scaling, applying more appropriate estimation methods. In fact, since the original ERA model consists of alternating linear models applied recursively, the most natural extension would be to consider the case of categorical endogenous variables as an alternating GLM model applied recursively, without resorting to less accurate Optimal Scaling methods. Thus, the core novelty of this work will be twofold:

- The introduction and estimation of a proper parametric version of the ERA model for categorical responses, hereinafter named (Vector) Generalized Linear ERA [(V)GLERA],
- The introduction ad adaptation of gradient descent methods, applied in the Artificial Neural Networks setting, to the ERA model, thereby named ERA-ANN, with no need of restrictive parametric assumptions.

Both model specifications will be formally presented in the next chapter, both in the binary and in the multinomial case, and will subsequently be evaluated with a simulation study that will cover also hybrid MLE-ANN formulations for either weights or loading parameters.

Part III

Extended Redundancy Analysis with Categorical Endogenous Variables

1 Maximum Likelihood Estimation

1.1 The case of one binary endogenous variable (GLERA)

Consider a $n \times q$ matrix $\mathbf{X}$ of exogenous variables which is centered and with variance scaled to unit, and a binary endogenous variable $\mathbf{y}$, such that $Y_i \sim \text{Ber}(\pi)$. Suppose also that the usual logit link function is applied to the probability of success of each element of $\mathbf{y}$ to obtain the linear predictor η_i, with weight matrix $\mathbf{W}$ and loading vector $\mathbf{a}'$, as below:

$$\eta_i = \text{logit}(P(y_i = 1)) = \text{logit}(\pi_i) = (\mathbf{XWa}')_i \tag{11}$$

The likelihood of the model is

$$\mathcal{L}(\mathbf{W}, \mathbf{a}') = \prod_{i=1}^{n} \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}')^{y_i} \ (1 - \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}'))^{1-y_i}$$

Hence the log-likelihood

$$l(\mathbf{W}, \mathbf{a}') = \sum_{i=1}^{n} y_i \log \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}') + (1 - y_i) \log(1 - \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}')) = \tag{12}$$

$$= \sum_{i=1}^{n} y_i \log \left(\frac{\pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}')}{1 - \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}')} \right) + \sum_{i=1}^{n} \log(1 - \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}')) = \tag{13}$$

$$= \sum_{i=1}^{n} y_i \, (\mathbf{XWa}')_i + \sum_{i=1}^{n} \log(1 - \pi_i(\mathbf{X}; \mathbf{W}, \mathbf{a}')) = \tag{14}$$

$$= \sum_{i=1}^{n} y_i \left(\mathbf{XWa}'\right)_i - \sum_{i=1}^{n} \log\left(1 + \exp(\mathbf{XWa}')_i\right)) \tag{15}$$

The log-likelihood in Equations 12 to 15 can be treated either fixing $\mathbf{W}$ or $\mathbf{a}'$, resulting in

$$l_{\mathbf{a}}(\mathbf{W}) = \sum_{i=1}^{n} y_i \left[(\mathbf{a} \otimes \mathbf{X})_i \text{vec}(\mathbf{W})\right] - \sum_{i=1}^{n} \log(1 + \exp\left\{(\mathbf{a} \otimes \mathbf{X})_i \text{vec}(\mathbf{W})\right\}) \tag{16}$$

for fixed $\mathbf{a}'$, and

$$l_{\mathbf{W}}(\mathbf{a}) = \sum_{i=1}^{n} y_i \left[\mathbf{F}_i \mathbf{a}'\right] - \sum_{i=1}^{n} \log(1 + \exp\left\{\mathbf{F}_i \mathbf{a}'\right\}) \tag{17}$$

for fixed $\mathbf{W}$, with the identification restriction $\text{diag}[(\mathbf{XW})'(\mathbf{XW})] = \text{diag}[\mathbf{F}'\mathbf{F}] = \mathbf{I}$. Similarly to the ERA linear model: let $\mathbf{w}^*$ be the vector obtained by eliminating the zero elements of $\text{vec}(\mathbf{W})$, and let $\mathbf{\Omega}_i$ be the vector obtained by eliminating the elements of $(\mathbf{a} \otimes \mathbf{X})_i$ corresponding to the zero elements of $\text{vec}(\mathbf{W})$.
Then Equation 16 becomes

$$l_{\mathbf{a}}(\mathbf{w}^*) = \sum_{i=1}^{n} y_i \left[\mathbf{\Omega}_i \mathbf{w}^*\right] - \sum_{i=1}^{n} \log(1 + \exp\left\{\mathbf{\Omega}_i \mathbf{w}^*\right\}) \tag{18}$$

and Equation 17 remains unchanged.
The estimation of the model parameters is carried out with an Alternating Maximum Likelihood (AML) algorithm, maximizing Equation 18 and Equation 17 alternately until convergence. Standard errors of the parameters can be obtained either with Bootstrap procedures, or parametrically at convergence, using the covariance matrices from the profile likelihoods [Richards, 1961; Yee and Hastie, 2003]. The proposed method is a first step towards modeling categorical endogenous variable, in the simplest case of one binary response.

1.2 The case of one multinomial endogenous variables (VGLERA)

Consider a $n \times Q$ matrix $\mathbf{X}$ of exogenous variables which is centered and with variance scaled to unit, and a categorical endogenous variable $\mathbf{y}$ with J categories, such that $Y_{ci} \sim \text{Mult}(\pi_1, \ldots, \pi_J)$ and $\sum_{j=1}^{J} \pi_j = 1$.

Conventionally a baseline category is selected (e.g. the last category J), and it is assumed that the log-odds of each response follows a linear model

$$\eta_{ij} = \log\left(\frac{P(y_{ci}=j)}{P(y_{ci}=J)}\right) = \log\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = (\mathbf{X}\mathbf{W}_j\mathbf{a}_j')_i \tag{19}$$

Furthermore, the conventional rule $\eta_{iJ} = 0$, is adopted when modeling categorical responses.

To adapt the estimation algorithm to the multinomial case, the whole model has to be formulated in Vector GLM notation [VGLM; Agresti, 2002; Yee, 2015], fitting a two-step Alternating IRLS algorithm (AIRLS), fitting $(J-1)$ weight matrices $\mathbf{W}_j$ for fixed loading vectors $\mathbf{a}_j$ in the first step, and fitting $(J-1)$ loading vectors $\mathbf{a}_j$ for fixed weight matrices $\mathbf{W}_j$.

1. *Estimation of* $\mathbf{W}_j$*'s for fixed* $\mathbf{a}_j$*'s.*

   Let $\mathbf{\Omega}_j$ be the $n \times RQ$ matrix such that $\mathbf{\Omega}_j = (\mathbf{a}_j \otimes \mathbf{X})$, for $j = 1, \ldots, J-1$. These matrices have to undergo some manipulation in order to apply VGLM methods. Specifically:

   - Let $\mathbf{\Omega}^{*(q)}$ be the matrix obtained by column-binding the q-th columns of each of the $(J-1)$ $\mathbf{\Omega}_j$ matrices

   $$\mathbf{\Omega}^{*(q)} = \left[\mathbf{\Omega}_1^{(q)}, \ldots, \mathbf{\Omega}_{J-1}^{(q)}\right] \qquad q = 1, \ldots, RQ$$

   - Every row $\boldsymbol{\omega}_i^{*(q)}$ of $\mathbf{\Omega}^{*(q)}$ is transformed into the corresponding $(J-1) \times (J-1)$ diagonal matrix $\mathbf{D}_{\boldsymbol{\omega}_i^{*(q)}}$

   $$\mathbf{D}_{\boldsymbol{\omega}_i^{*(q)}} = \sum_{j=1}^{J-1} \mathbf{e}_j \boldsymbol{\omega}_i^{*(q)} \mathbf{E}_j \qquad i = 1, \ldots, n$$

where $\mathbf{e}_j$ is the canonical basis column vector in $\mathbb{R}^{J-1}$, with 1 in the j-th position and 0 elsewhere, and $\mathbf{E}_j$ is the canonical basis square matrix in $\mathbb{R}^{(J-1)\times(J-1)}$, with 1 in the (j,j)-th position and 0 elsewhere. These diagonal matrices are then stacked vertically to form the $n(J-1)\times(J-1)$ matrix $\mathbf{D}_{\mathbf{\Omega}^{*(q)}}$.

- Finally, the RQ matrices $\mathbf{D}_{\mathbf{\Omega}^{*(q)}}$ are bound columnwise to obtain the $n(J-1)\times RQ(J-1)$ matrix $\mathbf{\Omega}^{\#}$

$$\mathbf{\Omega}^{\#} = [\mathbf{D}_{\mathbf{\Omega}^{*(1)}}|\mathbf{D}_{\mathbf{\Omega}^{*(2)}}|\ldots|\mathbf{D}_{\mathbf{\Omega}^{*(Q)}}]$$

Similarly, Let $\mathbf{W}_j$ be the $Q\times R$ weight matrix, for $j=1,\ldots,J-1$.

- Each $\mathbf{W}_j$ is vectorized into $\mathbf{w}_j$, and the $QR\times(J-1)$ matrix $\widetilde{\mathbf{W}}$ is obtained by column-binding these $(J-1)$ column vectors.
- Every row $\mathbf{w}_q$ of $\widetilde{\mathbf{W}}$ is transformed into the corresponding $(J-1)\times(J-1)$ diagonal matrix $\mathbf{D}_{\mathbf{w}_q}$

$$\mathbf{D}_{\mathbf{w}_q} = \sum_{j=1}^{J-1} \mathbf{e}_j \mathbf{w}_q \mathbf{E}_j \qquad q=1,\ldots,RQ$$

These diagonal matrices are then stacked vertically to form the $RQ(J-1)\times(J-1)$ matrix $\mathbf{D}_{\mathbf{w}}$. Finally, The row-sum vector $\mathbf{w}^{\#}$ is calculated

$$\mathbf{w}^{\#} = \mathbf{D}_{\mathbf{w}}\mathbf{1}_{J-1}$$

where $\mathbf{1}_k$ is a column vector of 1's, of length k.

Furthermore, let $\mathbf{Y}_d$ be the dummy model matrix of the categorical response variable $\mathbf{y}$, leaving out the J-th baseline column, and let $\mathbf{y}^* = \mathrm{vec}(\mathbf{Y}_d')$ be the $n(J-1)\times 1$ VGLM response vector. Then, the following GLS model is fitted recursively

$$\mathbf{z}^{(s-1)} = \mathbf{\Omega}^* \mathbf{w}^{*(s)} + \boldsymbol{\epsilon}^{(s-1)} \tag{20}$$

In Equation 20, $\mathbf{w}^{*(s)}$ is the non-zero subvector of $\mathbf{w}^{\#}$, $\mathbf{\Omega}^*$ is the matrix obtained by eliminating the columns corresponding to the zero elements of $\mathbf{w}^{\#}$, the error covariance matrix is not spherical (the response distribution is intrinsically heteroscedastic) and the $\mathbf{z}$ vector is the "working response" for $\mathbf{y}^*$

$$\mathbf{z}^{(s-1)} = \mathbf{\Omega}^* \mathbf{w}^{*(s-1)} + \mathbf{K}^{-1(s-1)}(\mathbf{y}^* - \mathbf{p}^{(s-1)})$$

with $\mathbf{p}$ as the stacked vector of estimated probabilities for $\mathbf{y}^*$ and $\mathbf{K}^{-1}$ as the $n(J-1) \times n(J-1)$ inverse reweighing matrix for the multinomial case, in which every of its diagonal blocks $\mathbf{K}_i$ is defined as

$$\mathbf{K}_i = \begin{bmatrix} p_{1i}(1-p_{1i}) & & & \\ -p_{1i}p_{2i} & p_{2i}(1-p_{2i}) & & \\ \vdots & \vdots & \ddots & \\ -p_{1i}p_{(J-1)i} & -p_{2i}p_{(J-1)i} & \cdots & p_{(J-1)i}(1-p_{(J-1)i}) \end{bmatrix} \qquad i = 1, \ldots, n$$

According to IRLS, parameter estimates and weight matrix are alternately estimated until convergence. In particular, the estimate of $\mathbf{w}^{*(s)}$ in the s-th step of IRLS is given by

$$\mathbf{w}^{*(s)} = (\mathbf{\Omega}^{*'}\mathbf{K}^{(s-1)}\mathbf{\Omega}^*)^{-1}\mathbf{\Omega}^{*'}\mathbf{K}^{(s-1)}\mathbf{z}^{(s-1)}$$

Subsequently, the $\mathbf{W}_j$'s are reconstructed, and $(J-1)$ latent composite matrices are calculated

$$\mathbf{F}_j = \mathbf{X}\mathbf{W}_j \qquad j = 1, \ldots, J-1$$

under the identification restriction $\text{diag}(\mathbf{F}_j'\mathbf{F}_j) = \mathbf{I}$.

2. *Estimation of* $\mathbf{a}_j$*'s for fixed* $\mathbf{W}_j$*'s.*

   The $\mathbf{F}_j$ matrices have to undergo some manipulation in order to apply VGLM methods. Specifically:

   - Let $\mathbf{F}^{*(r)}$ be the matrix obtained by column-binding the r-th columns of

each of the $(J-1)$ matrices

$$\mathbf{F}^{*(r)} = \left[\mathbf{F}_1^{(r)}, \ldots, \mathbf{F}_{J-1}^{(r)}\right] \qquad r = 1, \ldots, R$$

- Every row $\mathbf{f}_i^{*(r)}$ of $\mathbf{F}^{*(r)}$ is transformed into the corresponding $(J-1) \times (J-1)$ diagonal matrix $\mathbf{D}_{\mathbf{f}_i^{*(r)}}$

$$\mathbf{D}_{\mathbf{f}_i^{*(r)}} = \sum_{j=1}^{J-1} \mathbf{e}_j \mathbf{f}_i^{*(r)} \mathbf{E}_j \qquad i = 1, \ldots, n$$

  These diagonal matrices are then stacked vertically to form the $n(J-1) \times (J-1)$ matrix $\mathbf{D}_{\mathbf{F}^{*(r)}}$.

- Finally, the R matrices $\mathbf{D}_{\mathbf{F}^{*(r)}}$ are bound columnwise to obtain the $n(J-1) \times R(J-1)$ matrix $\mathbf{F}^*$

$$\mathbf{F}^* = [\mathbf{D}_{\mathbf{F}^{*(1)}} | \mathbf{D}_{\mathbf{F}^{*(2)}} | \ldots | \mathbf{D}_{\mathbf{F}^{*(R)}}]$$

Similarly, Let $\mathbf{a}_j$ be the $R \times 1$ loading vector, for $j = 1, \ldots, J-1$. The $(J-1) \times R$ loading matrix $\mathbf{A}$ is constructed

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1' \\ \hline \mathbf{a}_2' \\ \hline \vdots \\ \hline \mathbf{a}_{(J-1)}' \end{bmatrix}$$

and vectorized to get the corresponding $R(J-1) \times 1$ parameter vector $\mathbf{a}^* = \text{vec}(\mathbf{A})$.

Then, the following GLS model is fitted recursively

$$\mathbf{z}^{(s-1)} = \mathbf{F}^* \mathbf{a}^{*(s)} + \boldsymbol{\epsilon}^{(s-1)}$$

In Equation 2, the error covariance matrix is not spherical (the response distribution is intrinsically heteroscedastic) and the $\mathbf{z}$ vector is the "working re-

sponse" for $\mathbf{y}^*$

$$\mathbf{z}^{(s-1)} = \mathbf{F}^* \mathbf{a}^{*(s-1)} + \mathbf{K}^{-1(s-1)}(\mathbf{y}^* - \mathbf{p}^{(s-1)})$$

with $\mathbf{p}$ as the stacked vector of estimated probabilities for $\mathbf{y}^*$ and $\mathbf{K}^{-1}$ as the $n(J-1) \times n(J-1)$ inverse reweighing matrix for the multinomial case, in which every of its diagonal blocks $\mathbf{K}_i$ is defined as

$$\mathbf{K}_i = \begin{bmatrix} p_{1i}(1-p_{1i}) & & & \\ -p_{1i}p_{2i} & p_{2i}(1-p_{2i}) & & \\ \vdots & \vdots & \ddots & \\ -p_{1i}p_{(J-1)i} & -p_{2i}p_{(J-1)i} & \cdots & p_{(J-1)i}(1-p_{(J-1)i}) \end{bmatrix} \qquad i = 1, \ldots, n$$

According to IRLS, parameter estimates and weight matrix are alternately estimated until convergence. In particular, the estimate of $\mathbf{a}^{*(s)}$ in the s-th step of IRLS is given by

$$\mathbf{a}^{*(s)} = (\mathbf{F}^{*'}\mathbf{K}^{(s-1)}\mathbf{F}^*)^{-1}\mathbf{F}^{*'}\mathbf{K}^{(s-1)}\mathbf{z}^{(s-1)}$$

Standard errors of the parameters can be obtained with Bootstrap procedures, and goodness of fit in this preliminary modelization can be obtained calculating the mean absolute error (MAE) [Hyndman and Koehler, 2006] with respect to the observed response, along with usual misclassification rates from the confusion matrix.

The proposed method is a first step towards modeling categorical endogenous variable, in the case of one categorical response, but the above reasoning can also be applied to multinomial endogenous variables: if $\mathbf{y}_k$ has M_k categories, the new variable $\mathbf{y}_c$ will have $J = \prod_{k=1}^{p} M_k$ categories.
The binomial and multinomial formulations can be put in comparison with the Partial Least Squares Discriminant Analysis model [PLS-DA; Barker and Rayens, 2003; Gallo, 2010, in the case of compositional data], that apply the problem of classification via Discriminant Analysis to the PLS Regression framework, highlighting how

the PLS Regression model can be used also for classification purposes.
In the original PLS-R formulation [Wold et al., 1983], suppose $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$, then

$$\max_{\mathbf{a},\mathbf{b};\mathbf{a}'\mathbf{A}=\mathbf{0}'} \left(\frac{[\mathbb{C}\text{ov}(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{y})]^2}{(\mathbf{a}'\mathbf{a})(\mathbf{b}'\mathbf{b})} \right) = \{\mathbf{a}_{k+1}, \mathbf{b}_{k+1}\} \tag{21}$$

where $\mathbf{a}_{k+1}$ is the eigenvector corresponding to the $(k+1)$-th largest eigenvalue of $\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yx}$, and $\mathbf{b}_{k+1} = \mathbf{\Sigma}_{yx}\mathbf{a}_{k+1}$.
Furthermore, PLS can be reformulated as a Canonical Correlation Analysis [CCA; Hotelling, 1936] with PCA in the $\mathbf{X}$ and $\mathbf{Y}$ spaces as the penalties

$$[\mathbb{C}\text{ov}(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{y})]^2 = \mathbb{V}(\mathbf{a}'\mathbf{x})[\mathbb{C}\text{orr}(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{y})]^2\mathbb{V}(\mathbf{b}'\mathbf{y}) \tag{22}$$

with $\mathbf{Y}$ as the dummy matrix representing group membership. In this case, the CCA directions are essentially the same directions of Fisher LDA [Bartlett, 1938]. Reinstating Equation 22 as a PLS-R objective function, but leaving out the constraints on $\mathbf{b}'\mathbf{b}$ since they are not meaningful, pertaining to a dummy variable matrix, leads to the modified version of Equation 21

$$\max_{\mathbf{a},\mathbf{b};\mathbf{a}'\mathbf{A}=\mathbf{0}'} \left(\frac{[\mathbb{C}\text{ov}(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{y})]^2}{\mathbb{V}(\mathbf{b}'\mathbf{y})(\mathbf{a}'\mathbf{a})} \right) = \{\mathbf{a}_{k+1}, \mathbf{b}_{k+1}\}$$

where $\mathbf{a}_{k+1}$ is the eigenvector corresponding to the $(k+1)$-th largest eigenvalue of $\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_y^{-1}\mathbf{\Sigma}_{yx}$, and $\mathbf{b}_{k+1} = \mathbf{\Sigma}_y^{-1}\mathbf{\Sigma}_{yx}\mathbf{a}_{k+1}$. The sample plugin version of this eigenstructure solution has been proved [Barker and Rayens, 2003] to be equivalent to the solution of the discrimination problem that maximizes the among-group sum of squares

$$\frac{1}{n-1}\mathbf{H} = \mathbf{S}_{xy}\mathbf{S}_y^{-1}\mathbf{S}_{yx} = \sum_{j=1}^{J} n_j(\overline{\mathbf{X}}_j - \overline{\mathbf{X}})(\overline{\mathbf{X}}_j - \overline{\mathbf{X}})'$$

The main differences between PLS-DA and ERA for classification are in the model specification and in the formal resolution of the problem: in PLS-DA the classification is carried out without imposing restrictions (zero-fixed values) on the parameters; moreover, the solution of the PLS-DA problem is performed with eigenstructure

decomposition, whilst this new proposition relies on maximum likelihood and least squares optimization.

2 Estimation via Artificial Neural Networks

2.1 Artificial Neural Networks

2.1.1 Introduction

The term "neural network" (NN or ANN) has its origins in biology and neurosciences, attempting to find mathematical representations of information processing in biological systems [McCulloch and Pitts, 1943; Widrow et al., 1960]. This has deeply affected the lexis of the related methodology also in practical applications of pattern recognition and statistical modeling [Werbos, 1974]. A typical neural network architecture is made of input and output neurons (i.e. independent and dependent variables), mediated by so-called layers of hidden (i.e. unobservable) neurons; connections among those layers represent how the input information is combined and transformed towards subsequent layers [Bishop, 1995; Bishop, 2006]. A simple representation of an ANN architecture with one hidden layer is given in Figure 4.

Formally, breaking down the features of the network, the j_t-th hidden neuron in the t-th layer is given by a transformed linear combination of its inputs. For example, considering the above architecture, the hidden layer neurons values are given by:

$$\mathbf{h}_{j_1,1} = \sigma_1 \left(\sum_{j_0=1}^{J_0} \mathbf{x}_{j_0} w_{j_0,0} + b_0 \right) \quad j_1 = 1, \ldots, J_1 \tag{23}$$

then the neurons of the first hidden layer are again combined and transformed, forming the output of the net:

$$\mathbf{o}_{j_2} = \mathbf{h}_{j_2,2} = \sigma_2 \left(\sum_{j_1=1}^{J_1} \mathbf{h}_{j_1,1} w_{j_1,1} + b_1 \right) \quad j_2 = 1, \ldots, J_2 \tag{24}$$

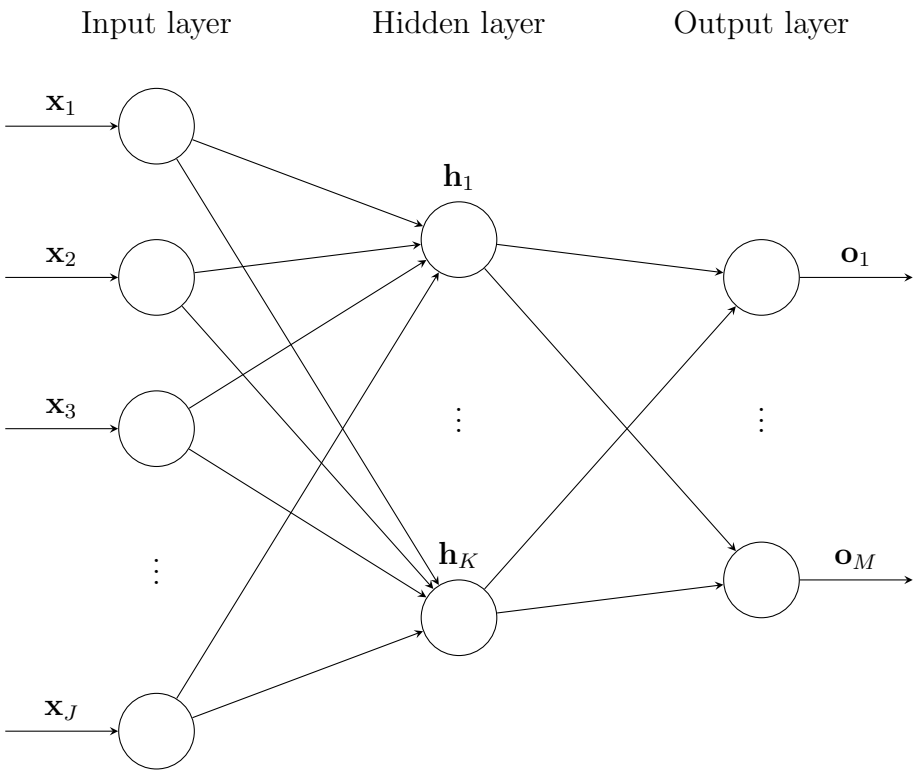


Figure 4: Artificial Neural Network example

Putting all together:

$$\mathbf{o}_{j_2} = \mathbf{h}_{j_2,2} = \sigma_2 \left(\sum_{j_1=1}^{J_1} \left(\sigma_1 \left(\sum_{j_0=1}^{J_0} \mathbf{x}_{j_0} w_{j_0,0} + b_0 \right) \right) w_{j_1,1} + b_1 \right) \quad j_2 = 1, \ldots, J_2 \quad (25)$$

In particular, he first stage neurons are the observed inputs $\mathbf{x}_{j_0},\ j_0 = 1, \ldots, J_0$, while the last stage neurons are the estimated outputs $\mathbf{o}_{j_T},\ j_T = 1, \ldots, J_T$ that will be compared with the observed outputs $\mathbf{y}_{j_T},\ j_T = 1, \ldots, J_T$. Moreover, inputs are not considered as a layer. A schematic representation of the input/output transition is shown in Figure 5.

The choice of the activation functions depends (also) on the nature of the final output, and on the task of the analysis. The most common functions are the linear activation function, used for linear prediction

$$\sigma_t(\mathbf{w}, \mathbf{x}) = \mathbf{w}'\mathbf{x}$$

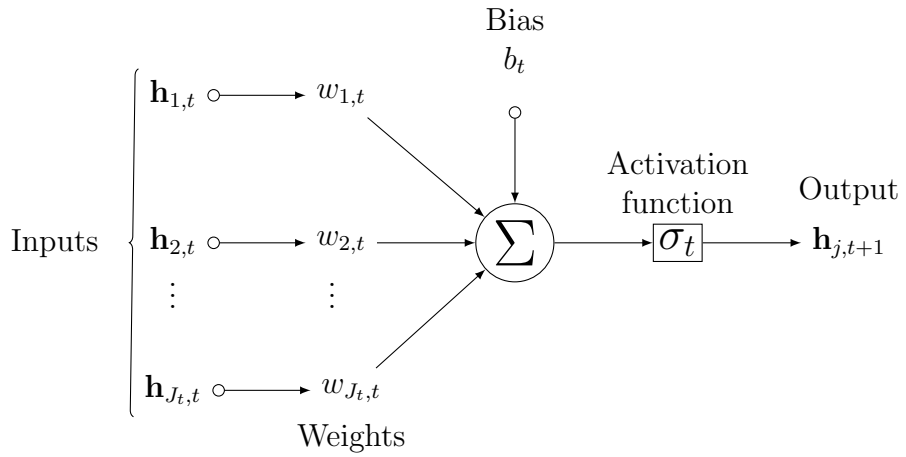


Figure 5: Neural Network input-output transition

the sigmoidal, hyperbolic tangent and hard limit functions, used for non-linear prediction or binary classification

$$\sigma_t(\mathbf{w}, \mathbf{x}) = \text{logit}(\mathbf{w}'\mathbf{x})$$

$$\sigma_t(\mathbf{w}, \mathbf{x}) = \tanh(\mathbf{w}'\mathbf{x})$$

$$\sigma_t(\mathbf{w}, \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}'\mathbf{x} > 0 \\ -1 & \text{otherwise} \end{cases}$$

the softmax function, used for multiclass classification (J categories)

$$\sigma_{tj}(\mathbf{w}_j, \mathbf{x}) = \frac{\exp(\mathbf{w}_\mathbf{j}'\mathbf{x})}{\sum_{k=1}^{J} \exp(\mathbf{w}_\mathbf{k}'\mathbf{x})}$$

the radial basis function, used for classification, similarly to Discriminant Analysis techniques, and weighed kernel approximation.

$$\sigma_t(\mathbf{w}, \mathbf{x}) = \left(\sum_{j=1}^{J_1} w_{j,t} \phi(||\mathbf{x} - \boldsymbol{\mu}_j||) \right)$$

2.1.2 Estimation of Artificial Neural Networks

The objective function to be minimized with respect to the neural network weights, collected in the vector $\mathbf{w}$, is

$$E(\mathbf{w}) = \sum_{n=1}^{N} ||\mathbf{o}_n(\mathbf{x}_n, \mathbf{w}) - \mathbf{y}_n||^2 \tag{26}$$

The above objective function can be used either for prediction or classification purposes, but the cross-entropy error function has been proved to be more efficient and accurate for classification problems than the standard sum of squares criterion [Simard et al., 2003]

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \log o_{nk} + (1 - y_{nk}) \log(1 - o_{nk}) \tag{27}$$

The estimation of neural network models relies on the iteration of two steps:

- **Feedforward**: using the weights estimated in the previous iteration, the network outputs are calculated. For a single observation n, from input neurons i, to output neuron j:

$$a_j = \left(\sum_i w_{ji} h_i \right) \tag{28}$$

$$h_j = \sigma(a_j) \tag{29}$$

  Note that one or more of the variables h_i in the sum in Equation 28 could be an input, and similarly, the unit j in Equation 29 could be an output.

- **Backpropagation of the error** [Rumelhart et al., 1986]: having

$$E_n = \frac{1}{2} \sum_k (o_{nk}(\mathbf{x}_n, \mathbf{w}) - y_{nk})^2$$

  the gradient of the error w.r.t. $\mathbf{w}$ is calculated via the chain rule, "backpropagating" it to every node of the network. For instance, consider the evaluation of the derivative of E_n with respect to a weight w_{ji}, noting that E_n depends

on the weight only via a_j

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} \tag{30}$$

Using the notation $\frac{\partial E_n}{\partial a_j} = \delta_j$, $\frac{\partial a_j}{\partial w_{ji}} = h_i$, Equation 30 becomes

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j h_i$$

hence, the derivative is simply the product of the δ of the neuron at the output side of the weight by the z of the neuron at the input side of the weight. The evaluation of δ_j-s is carried out using the chain rule

$$\delta_j = \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k}\frac{\partial a_k}{\partial a_j} \tag{31}$$

for all k-s to which unit j is connected. Finally, using the above notation and Equations 28 and 29

$$\delta_j = \sigma'(a_j)\sum_k \delta_k w_{kj} \tag{32}$$

Equation 32 is applied recursively, starting from the output units, for which $\delta_k = (o_k - y_k)$, back through the connections of the networks, up until the first set of weights (hence the term "backpropagation").

- **Weights update** by gradient descent, with tuning parameter η.

$$\mathbf{w^{(s+1)}} = \mathbf{w^{(s)}} - \eta \nabla E(\mathbf{w^{(s)}}) \tag{33}$$

The usual gradient descent update can be slow and computationally burdensome. For this reason, quicker update methods have been implemented to drastically increase the algorithm efficiency [Vogl et al., 1988; Rigler et al., 1991; Hagan and Menhaj, 1994], such as variable tuning parameter and resilient backpropagation.

Other methods, such as BFGS Quasi-Newton [Bonnans et al., 2006]

$$\mathbf{w^{(s+1)}} = \mathbf{w^{(s)}} - [\mathbf{H^{(s)}}]^{-1} \nabla E(\mathbf{w^{(s)}})$$

and Lavenberg-Marquardt [Levenberg, 1944; Marquardt, 1963]

$$\mathbf{w^{(s+1)}} = \mathbf{w^{(s)}} - [\mathbf{J^{(s)'}J^{(s)}} + \mu\mathbf{I}]^{-1} \mathbf{J^{(s)'}e^{(s)}}$$

make use also of the Hessian matrix $\mathbf{H}$, either computing it directly or approximating it with the Jacobian matrix of the weights $\mathbf{J}$ (especially useful with big networks).

Usually, a few layers are enough to approximate network outputs and target variables and, with a sufficient number of neurons, any continuous function on a compact input domain can be approximated with great accuracy [Cybenko, 1989; Hornik et al., 1989; Hornik, 1991]; this remains valid for a wide range of hidden layers activation functions, excluding polynomials. Recently there has been a rise of methods involving the estimation of neural networks with many hidden layers (e.g., more than three), referred to as deep learning architectures [Goodfellow et al., 2016]. In the next section, the implementation of ANN in the SEM framework will be discussed and analyzed, leading to a new proposition for the ERA model with categorical endogenous variables.

2.2 SEM and Artificial Neural Networks

To implement an ANN for structural equation models with latent variables, the main challenge is to give a proper definition of latent variable using the typical constructs of the neural network approach.

The most simple case of dimensionality reduction applied to the data, in some sense analogous to Principal Component Analysis, regards the mapping of vectors $\mathbf{x}_n \in \mathbb{R}^d$ onto vectors $\mathbf{z}_n \in \mathbb{R}^m$, with $m << d$ [Rumelhart et al., 1986]. The targets used to train the network are simply the input vectors themselves, hence this ANNs is called autoassociative neural network (see Figure 6). It attempts to map each

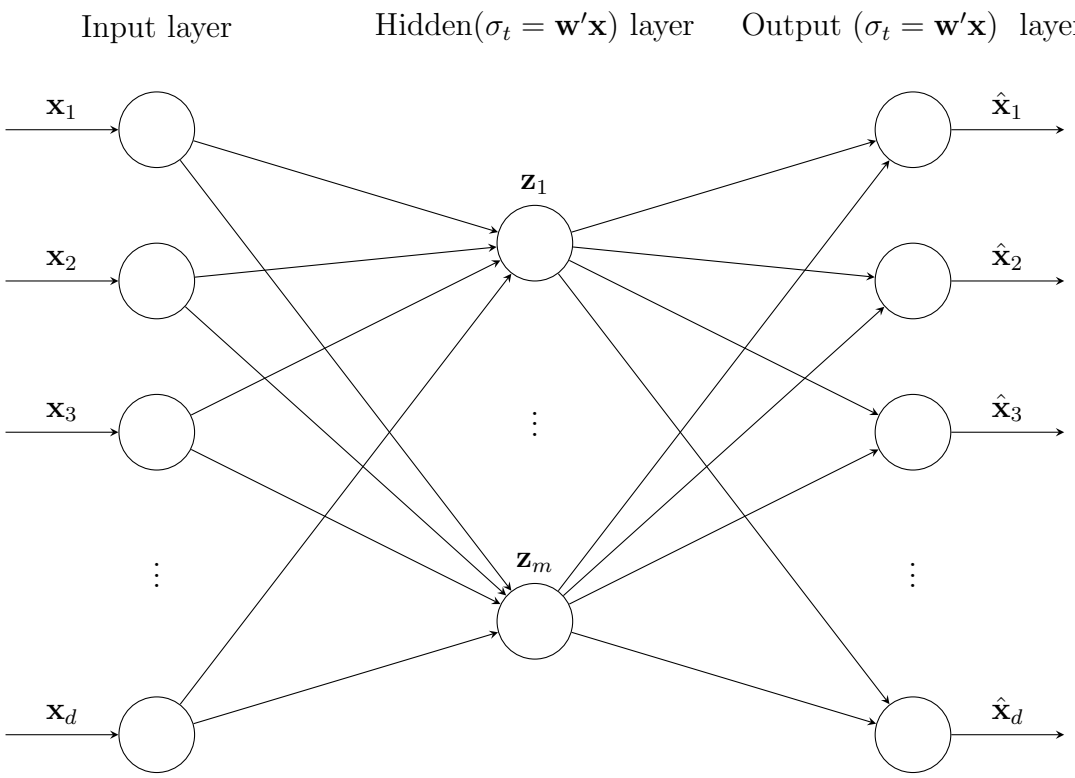


Figure 6: Autoassociative Neural Network

input vector onto itself, with a hidden layer comprised of m neurons, by minimizing a sum-of-squares error of the form

$$E = \frac{1}{2} \sum_{n} \sum_{k=1}^{d} (\hat{x}_k(\mathbf{x}_n, \mathbf{w}) - x_{nk})^2$$

If the hidden units have linear activations functions, then it can be shown that the error function has a unique global minimum, and that at this minimum the network performs a projection onto the m-dimensional subspace spanned by the first m principal components of the data [Bourlard and Kamp, 1988; Baldi and Hornik, 1989]. The vector of weights forms a basis set which spans the principal subspace. If additional hidden layers are permitted in the network, two subsequent functional mappings can be employed with two additional layers, one between the input and the projection, one between the projection layer and the output, both with sigmoid or tanh activation functions. Such a network effectively performs a non-linear principal component analysis [Kramer, 1991].

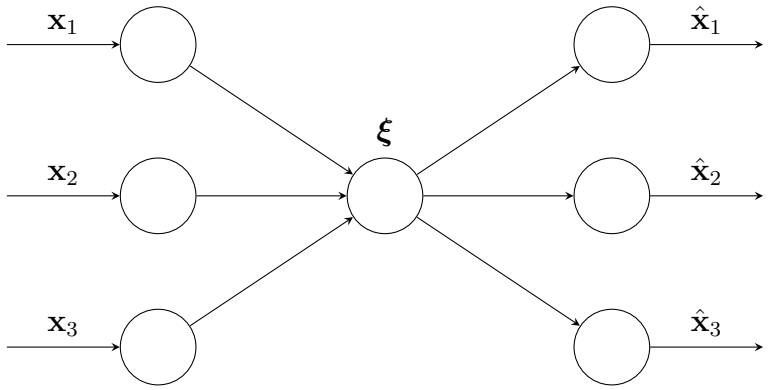


Figure 7: Observation-LV Network

To extend the implementation of ANN to SEM with latent variables, the hidden, mono-neural layers have to be connected one to the other, forming the structural part of the model while, for the measurement models, the left-hand side of the LV neuron contains formative weight, and the right-hand side of the LV neuron contains reflective weights. The measurement model is referred to as "Observation-LV Network" (OLN), depicted in Figure 7, and the corresponding architecture is the so-called "hierarchical ANN topology" [Hsu et al., 2006], shown in Figure 8. This model representation proved to be a good alternative to PLS-PM, and performed better than CSA, albeit tested only with linear activation functions and a fairly simple architecture.

Literature on this topic has also explored the ANN implementation of LISREL, via the same hierarchical ANN topology with limited connectivity, pointing out differences and similarities between the two methods and addressing potential benefits of the ANN implementation, since *"An additional neural estimation assists in assessing the stable and robust relationships, in avoiding overinterpretation, and in reconsidering the policy recommendations"* [Davies et al., 1999]. The potential use of Neural Networks as universal approximators has also allowed to extend the usual SEM specification in a nonlinear setting [Malthouse et al., 1997]. Specifically, in the nonlinear setting the objective function to be minimized is

$$\min_{\mathbf{f},\mathbf{s}_f,\mathbf{h},\mathbf{g},\mathbf{t}_g} \sum_{i=1}^{n} [||\mathbf{x}_i - \mathbf{f}(\mathbf{s}_f(\mathbf{x}_i))||^2 + ||\mathbf{t}_g(\mathbf{y}_i) - \mathbf{h}(\mathbf{s}_f(\mathbf{x}_i))||^2 + ||\mathbf{y}_i - \mathbf{g}(\mathbf{t}_g(\mathbf{y}_i))||^2] \quad (34)$$

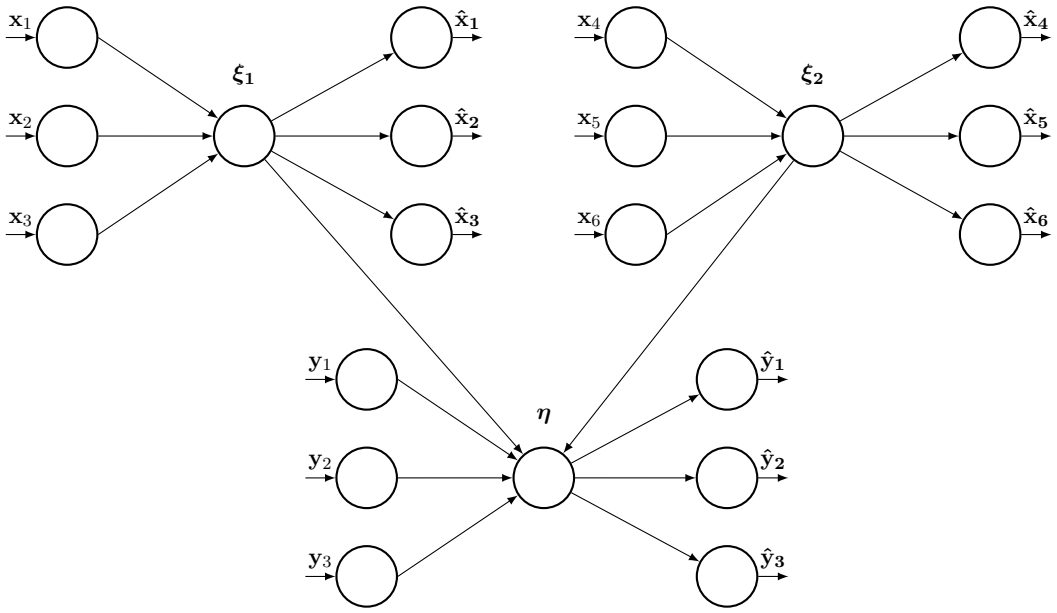


Figure 8: Hierarchical ANN topology for SEM

in Equation 34, the first and last term are the nonlinear autoassociative parts of the network, whereas the second term relate the predictor variable scores to the response variable scores as close as possible. The benefits of this method are clear especially when the observed predictor variables lie on a lower dimensional surface. An example of NLPLS modeled with a neural network architecture is shown in Figure 9.

However, despite their relatively easy implementation, ANN in the SEM framework revolves around observed variables that are identified as targets to be approximated by the network. In ANN, observed endogenous variables are the core of the analysis, while in LISREL or PLS the latent structure is the core of the analysis. For this reason, redundancy analysis models such as ERA are the most favorable setting in which an ANN structure can be developed, since the endogenous variables are all observed. What differentiates the standard ERA specification from the ANN architecture is the estimation method (e.g., ALS/AML vs backpropagation), broadening the spectrum of available estimating methods for the same model and allowing the treatment of categorical endogenous variables for classification purposes with no need of parametric assumptions.

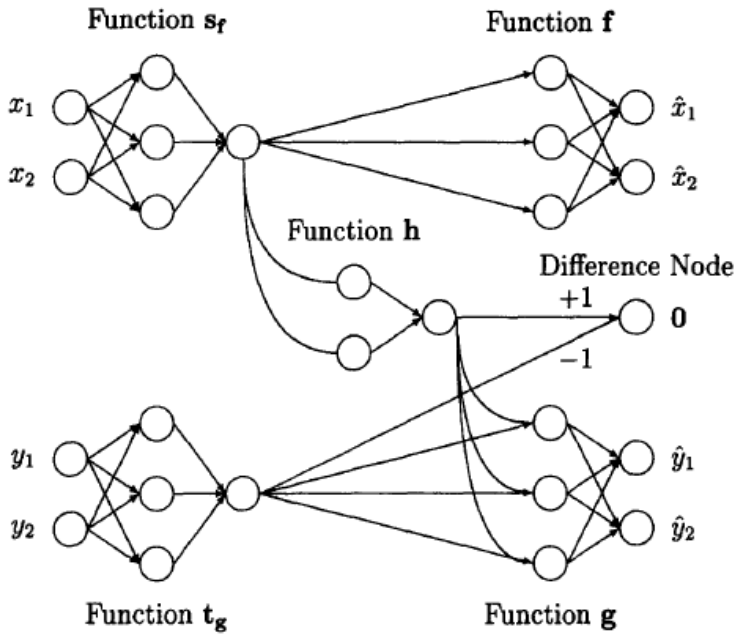


Figure 9: Nonlinear PLS topology [as in Malthouse et al., 1997]

2.3 The ERA-ANN method for one binary endogenous variable: four possible strategies

In the simplest possible structure, the ANN specification of the ERA path diagram is depicted in Figure 10. However, the straightforward implementation of this network leads to parameters unidentifiability, since no restriction is imposed on the hidden layer structure; in practice, any product $\mathbf{wa}'$ of rows and columns of $\mathbf{W}$ and $\mathbf{A}'$ respectively is equally capable of predicting the target values.

For this reason, the AML algorithm for GLERA is converted to its ANN counterpart, with two subnetworks, one fitting the $\mathbf{\Omega}$ matrix onto $\mathbf{y}$, for fixed $\mathbf{a}'$, one fitting the $\mathbf{F}$ matrix onto $\mathbf{y}$, for fixed $\mathbf{W}$.

1. ***Estimate of*** $\mathbf{W}$ ***for fixed*** $\mathbf{a}'$ *(subnetwork 1)*

   let $\mathbf{w}$ be the vector obtained by eliminating the zero elements of vec($\mathbf{W}$) in Equation 5, and let $\mathbf{\Omega}$ be the matrix obtained by eliminating the columns of ($\mathbf{a} \otimes \mathbf{X}$) corresponding to the zero elements of vec($\mathbf{W}$). The ANN for this

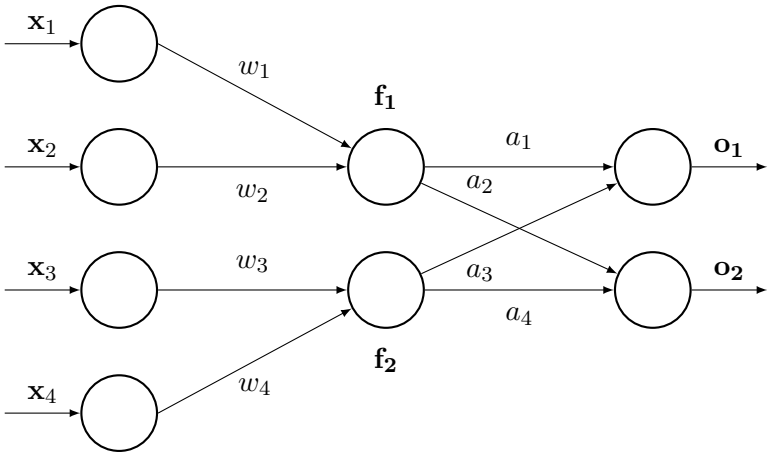


Figure 10: ANN architecture for ERA, unidentifiable model

classification problem is specified by

$$\mathbf{o} = \sigma\left(\sum_{k=1}^{K} \boldsymbol{\omega}_k w_k\right) \tag{35}$$

with sigmoid activation function and the backpropagation algorithm. The weights in $\mathbf{w}$ are thus estimated, reconstructing $\tilde{\mathbf{W}}$, and $\mathbf{F} = \mathbf{X}\tilde{\mathbf{W}}$ is normalised to respect the identification restriction.

2. ***Estimate of*** $\mathbf{a}'$ ***for fixed*** $\mathbf{W}$ *(subnetwork 2)*

   Having now reconstructed and normalized $\mathbf{F}$, the ANN for this classification problem is specified by

$$\mathbf{o} = \sigma\left(\sum_{m=1}^{p} \mathbf{f}_m a_m\right) \tag{36}$$

   with sigmoid activation function and the backpropagation algorithm.

The two subnetworks are iterated until convergence of the estimates, and usual network performance indexes are then calculated to evaluate the model performance. In addition to the GLERA specification in Section 1.1 and the ANN-ERA specification just described, two hybrid methods are estimable: one that estimates weights in $\mathbf{W}$ with MLE (Equation 18) and loadings in $\mathbf{a}'$ with ANN (Equation 36), thereby called MLE / ANN ERA, and one that estimates weights in $\mathbf{W}$ with ANN (Equation

35) and loadings in $\mathbf{a}'$ with MLE (Equation 17), thereby called ANN / MLE ERA. For a graphical representation of the four configurations, see Figure 11.

2.4 The ERA-ANN method for one multinomial endogenous variable: a "One-versus-All" approach

The multinomial version of the model in the VGLERA specification cannot be evaluated by ANN, since ANN is not able to handle varying predictor matrices at each estimation step, hence a two-step One-versus-All strategy will be employed (thereby 2S-ANN), fitting $(J-1)$ categories against the baseline one at a time, with $(J-1)$ separate binary logistic regressions [Agresti, 2002], one for each column of the dummy model matrix of $\mathbf{y}$. For a graphical representation of VGLERA and 2S-ANN, see Figure 12

1. ***Estimate of*** $\mathbf{W}_j$***'s for fixed*** $\mathbf{a}'_j$***'s***

   For every $j = 1, \ldots, J-1$ let $\mathbf{w}_j$ be the vector obtained by eliminating the zero elements of $\text{vec}(\mathbf{W}_j)$, and let $\mathbf{\Omega}_j$ be the matrix obtained by eliminating the columns of $\mathbf{a}_j \otimes \mathbf{X}$ corresponding to the zero elements of $\text{vec}(\mathbf{W}_j)$. The ANN for each classification problem is specified by

   $$\mathbf{o}_j = \sigma\left(\sum_{k=1}^{K} \boldsymbol{\omega}_{kj} w_{kj}\right) \qquad j = 1, \ldots, J-1 \tag{37}$$

   with sigmoid activation function and the backpropagation algorithm.

   Each $\tilde{\mathbf{w}}_j$ is thus estimated, reconstructing $\tilde{\mathbf{W}}_j$, and each $\mathbf{F}_j = \mathbf{X}\tilde{\mathbf{W}}_j$ is normalised to respect the identification restriction.

2. ***Estimate of*** $\mathbf{a}'_j$***'s for fixed*** $\mathbf{W}_j$***'s***

   Having now reconstructed and normalized $\mathbf{F}_j$, for every $j = 1, \ldots, J-1$, the ANN for this classification problem is specified by

   $$\mathbf{o}_j = \sigma\left(\sum_{m=1}^{p} \mathbf{f}_{mj} a_{mj}\right) \qquad j = 1, \ldots, J-1 \tag{38}$$

   with sigmoid activation function and the backpropagation algorithm.

The estimated parameters will be different from the ones obtained with VGLERA, but the two approaches will yield comparable results in terms of classification.
The models presented in this chapter will be evaluated in a simulation study that will define their capability to recover the underlying population parameters or, in the case of the 2S-ANN strategy, its performance in terms of prediction.

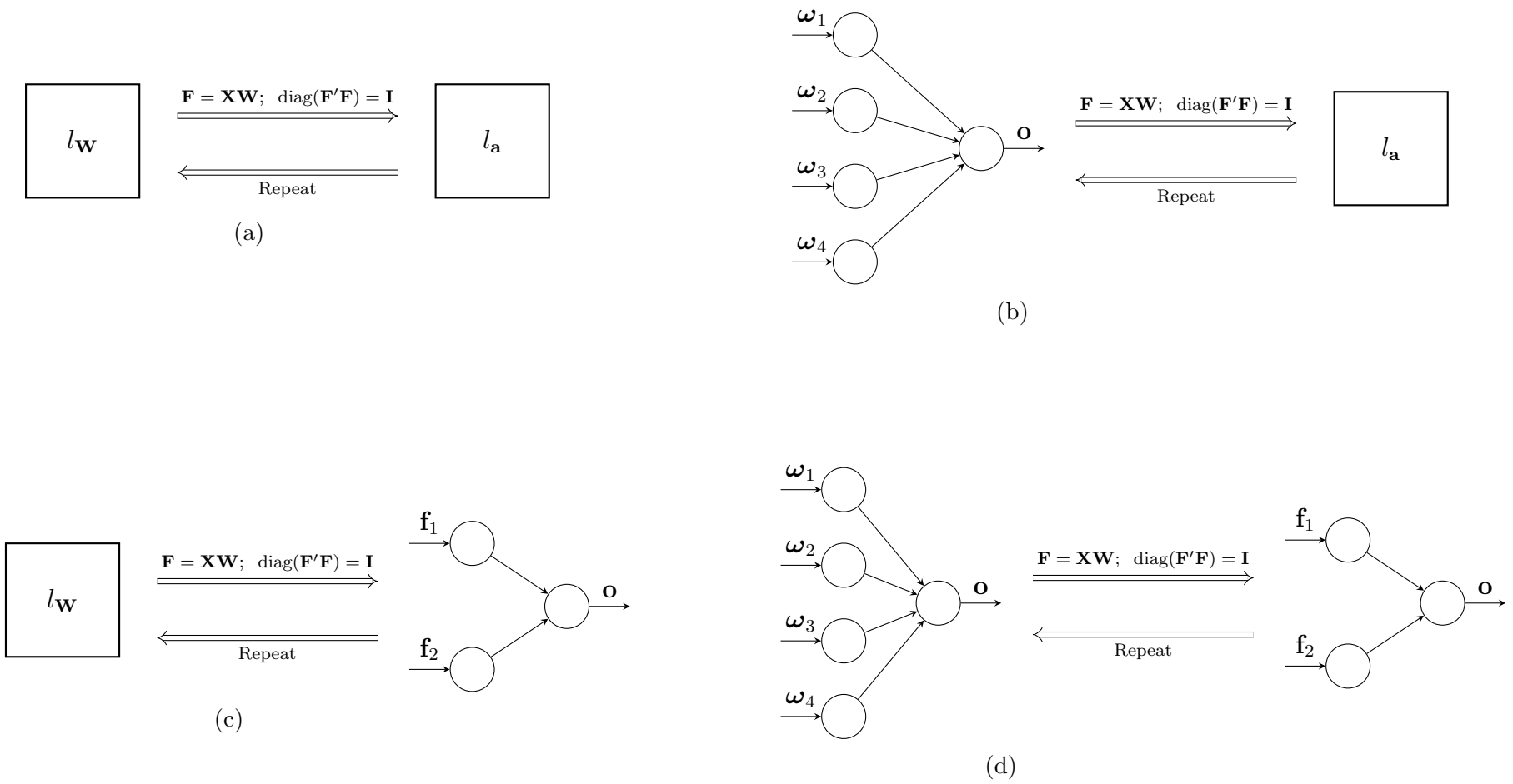


Figure 11: Four strategies to fit ERA with one binary endogenous variable. From top left to bottom right: pure GLERA (a), ANN / MLE ERA (b), MLE / ANN ERA (c), pure ANN-ERA (d).

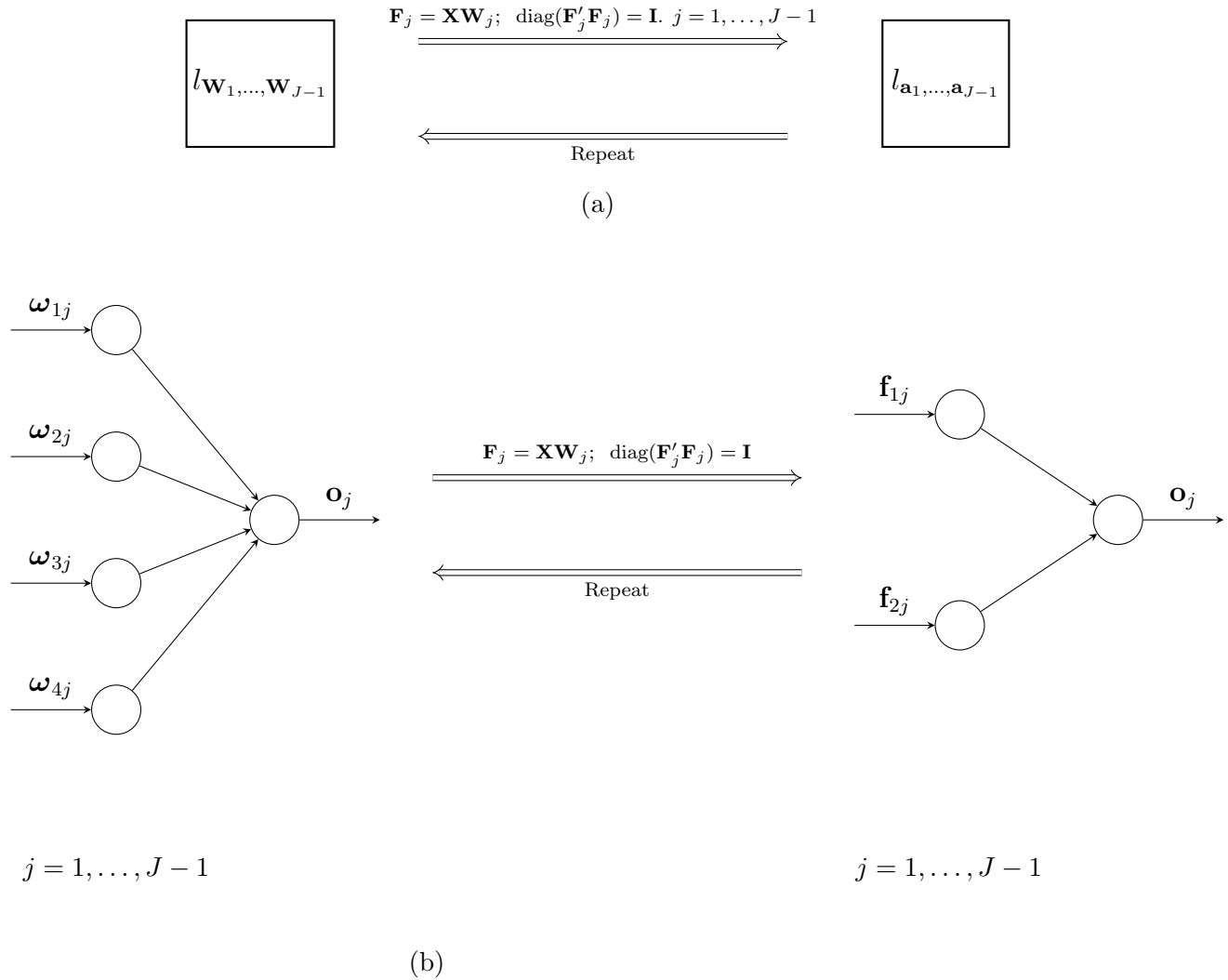


Figure 12: Two strategies to fit ERA with one categorical endogenous variable. Top panel VGLERA (a), bottom panel 2S-ANN (b). Note that in panel (a) parameters are estimated jointly with one model, while in panel (b) $(J-1)$ 2-Stage binary networks are estimated separately.

Part IV

A Simulation Study

1 Introduction

This chapter will cover a simulation study on the models introduced in the previous chapters. Specifically:

- For the case of one binary endogenous variable, the recovery of one set of parameters will be evaluated in four different models: a pure GLERA strategy (AML for both weights and loadings), a pure ANN-ERA strategy for both weights and loading and the two hybrid strategies MLE / ANN ERA and ANN / MLE ERA, as in Figure 11.
- For the case of one multinomial endogenous variable, the recovery of two sets of parameters will be evaluated using VGLERA (Figure 12, panel (a)), whereas the comparison with the 2S-ANN estimation (Figure 12, panel (b)) will be made on the accuracy in estimating the predicted probabilities, compared with the probabilities that generated the data, for both the sets of parameters.
- Both VGLERA and 2S-ANN prediction capabilities will be put in comparison with the standard classification problem with one set of coefficients $\mathbf{b} = \mathbf{W}\mathbf{a}'$ employing ANN with softmax activation function and no latent composites.

The data will be generated, in all cases, for sample size $n = \{50, 100, 200, 500, 1000\}$. The exogenous variables in $\mathbf{X}$ are fixed for each sample size, across $R = 1000$ replications of the response variable. The exogenous variables are randomly generated from $\mathbf{X} \sim \mathcal{N}_Q(\mathbf{0}, \mathbf{\Sigma})$, where

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & & & \\ .3 & 1 & & \\ .1 & .1 & 1 & \\ .1 & .1 & .3 & 1 \end{bmatrix}$$

This $\mathbf{\Sigma}$ matrix has been chosen to give a higher level of correlation among variables in

the same LC with respect to the other variables, also avoiding to induce collinearity that may affect the estimation. To evaluate the quality of the recovery of the parameters $\boldsymbol{\theta}$ across the replications, two indicators are used: the relative bias γ_k in absolute value for each estimated value $\hat{\theta}_k$, calculated as

$$\gamma_k = 100\,\frac{|\theta_k - \hat{\theta}_k|}{|\theta_k|}$$

where

$$\theta_k = \frac{1}{R}\sum_{r=1}^{R}\hat{\theta}_{kr}$$

and the congruence index ρ between the parameter vector and the estimated vector [Tucker, 1951; Lorenzo-Seva and Ten Berge, 2006], calculated as

$$\rho = \frac{\boldsymbol{\theta}'\hat{\boldsymbol{\theta}}}{\sqrt{(\hat{\boldsymbol{\theta}}'\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}'\boldsymbol{\theta})}}$$

To evaluate the predicted probability, the mean absolute error (MAE) across the R replications, between the estimated probabilities and the true probabilities is calculated, for each of the J categories

$$\text{MAE}_j = \frac{1}{Rn}\sum_{r=1}^{R}\sum_{i=1}^{n}|\pi_{ij} - p_{irj}| \qquad j = 1, \ldots, J-1$$

where π_{ij} is the true probability for subject i to belong in class j, according to the simulated model

$$\pi_{ij} = \frac{\exp(\mathbf{x}_i\mathbf{W}_j\mathbf{a}_j')}{1 + \sum_{j=1}^{J-1}\exp(\mathbf{x}_i\mathbf{W}_j\mathbf{a}_j')}$$

For the baseline category J, $\exp(\mathbf{x}_i\mathbf{W}_j\mathbf{a}_j') = 1$.

2 Simulation Results

2.1 The case of one binary endogenous variable

For this simulation study, the parameter matrices are

$$\mathbf{W} = \begin{bmatrix} .6 & 0 \\ .6 & 0 \\ 0 & .6 \\ 0 & .6 \end{bmatrix} \qquad \mathbf{a} = \begin{bmatrix} .3 \\ 3.8 \end{bmatrix}$$

Tables 2 and 3 shows the simulation results for the case of one binary endogenous variable, in all four estimation strategies. To obtain a satisfying parameter recovery n has to be at least equal to 200 in all cases. Standard errors and biases for $n = 50$ give the worst results: due to the dramatic difference between the loading parameters, such a small sample size is not enough to obtain satisfying standard errors and biases. Nevertheless, biases and std. errors patterns are generally decreasing with an increase in n, indicating a successful recovery of the true parameter values, and the congruence index shows an increasing pattern in all four estimation strategies. Among the four strategies there is no clear best choice, at least for higher values of the sample size, whereas if $n \leq 100$, the MLE-ANN strategy yields comparatively better results for both standard errors and biases. Furthermore, the full ANN strategy is the only one by which parameters biases have similar magnitude among themselves, whereas in the other strategies very high biases are only related to loadings. Figure 13 shows biases decrease patterns for weights and loadings in each estimation strategy.

2.2 The case of one multinomial endogenous variable

For this simulation study, a three-categories response variable is modeled with VGLERA and with 2S-ANN, holding $j = 3$ as the reference category. The parameter matrices for the first setting are

$$\mathbf{W}_1 = \mathbf{W}_2 = \begin{bmatrix} .7 & 0 \\ .7 & 0 \\ 0 & .7 \\ 0 & .7 \end{bmatrix} \qquad \mathbf{a}_1 = \begin{bmatrix} .4 \\ .3 \end{bmatrix} \qquad \mathbf{a}_2 = \begin{bmatrix} .6 \\ .6 \end{bmatrix}$$

Setting 1	MAE - 2S-ANN				MAE - VGLERA				MAE - ANN (Classic Softmax)			
n	Cat. 1	Cat. 2	Cat. 3	Overall	Cat. 1	Cat. 2	Cat. 3	Overall	Cat. 1	Cat. 2	Cat. 3	Overall
50	0.108	0.113	0.087	0.103	0.101	0.101	0.097	0.010	0.340	0.478	0.301	0.373
100	0.075	0.078	0.078	0.077	0.072	0.072	0.069	0.071	0.395	0.460	0.289	0.381
200	0.056	0.064	0.075	0.065	0.049	0.051	0.048	0.049	0.377	0.470	0.307	0.385
500	0.043	0.052	0.077	0.057	0.032	0.033	0.031	0.032	0.345	0.490	0.313	0.383
1000	0.038	0.05	0.077	0.055	0.022	0.023	0.022	0.022	0.354	0.538	0.341	0.411

Setting 2	MAE - 2S-ANN				MAE - VGLERA				MAE - ANN (Classic Softmax)			
n	Cat. 1	Cat. 2	Cat. 3	Overall	Cat. 1	Cat. 2	Cat. 3	Overall	Cat. 1	Cat. 2	Cat. 3	Overall
50	0.076	0.074	0.066	0.072	0.076	0.074	0.088	0.079	0.063	0.083	0.102	0.083
100	0.06	0.061	0.059	0.060	0.052	0.053	0.063	0.056	0.095	0.089	0.112	0.099
200	0.043	0.041	0.038	0.041	0.037	0.037	0.045	0.040	0.109	0.099	0.118	0.109
500	0.036	0.037	0.033	0.035	0.023	0.023	0.028	0.025	0.099	0.094	0.122	0.105
1000	0.032	0.032	0.03	0.031	0.016	0.016	0.019	0.017	0.076	0.067	0.105	0.083

Table 1: Comparison between 2S-ANN (gradient descent), VGLERA and classic softmax ANN with no hidden layers in the recovery of each category probability. Top panel for Setting 1, bottom panel for Setting 2. Sample sizes $n = \{50, 100, 200, 500, 1000\}$.

whereas the parameter matrices for the second setting are

$$\mathbf{W}_1 = \mathbf{W}_2 = \begin{bmatrix} .7 & 0 \\ .7 & 0 \\ 0 & .7 \\ 0 & .7 \end{bmatrix} \qquad \mathbf{a}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \mathbf{a}_2 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

For the softmax architecture and no latent composite, the parameter vectors for the first setting are

$$\mathbf{b}_1 = \mathbf{W}_1\mathbf{a}_1' = \begin{bmatrix} 0.7 \\ 1.4 \end{bmatrix} \qquad \mathbf{b}_2 = \mathbf{W}_2\mathbf{a}_2' = \begin{bmatrix} -0.7 \\ -1.4 \end{bmatrix}$$

whereas the parameter vectors for the second setting are

$$\mathbf{b}_1 = \mathbf{W}_1\mathbf{a}_1' = \begin{bmatrix} 0.21 \\ 0.42 \end{bmatrix} \qquad \mathbf{b}_2 = \mathbf{W}_2\mathbf{a}_2' = \begin{bmatrix} 0.28 \\ 0.42 \end{bmatrix}$$

Tables 4 and 5 show the simulation results for the VGLERA model. To obtain a satisfying parameter recovery, with bias below 10%, n has to be at least equal to 500 in the first setting, and 200 in the second setting, meaning that a more efficient recovery is possible if the loadings discriminate the groups more evidently. Biases and std. errors are generally decreasing with an increase in n, indicating a successful recovery of the true parameter values regardless of how far apart are the loadings. The congruence index shows an increasing and excellent pattern in all parameters settings. Figure 15 shows biases decrease patterns for weights and loadings. In

comparison with 2S-ANN estimation method, as shown in Table 1, VGLERA yields slightly more precise estimates of the probabilities generating the response. This is probably due to the restriction in using ANN in this contexts, that prevents the model from considering covariances between the dummy variables. Furthermore, MAE for the baseline category does not decrease as the MAE for the other categories for ANN, in the first setting, remaining stable around 0.07. Nevertheless, both estimation techniques are effective in recovering the true probabilities as the sample size increases, even more efficiently in the second parameter setting. Figure 14 shows the comparison between the two methods for the chosen sample sizes. Furthermore, comparing both methods with a standard softmax ANN, the two new methods always perform better than the architecture without the presence of latent composites, even more evidently in the first parameter setting, when the parameters are close between categories. Explanation for this behaviour is due partly to the double optimization process present in VGLERA and 2S-ANN (hence thanks to the presence of latent composites), partly to the tendency of the architecture of assigning drastically higher probability to the category chosen by the network.

			$n = 50$			$n = 100$			$n = 200$			$n = 500$			$n = 1000$		
method	label	par	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%
MLE	w_1	0.6	1.382	17.639	130.333	0.743	0.98	23.9	0.615	0.316	2.557	0.648	0.334	7.921	0.558	0.259	7.037
	w_2	0.6	1.378	19.601	129.673	0.737	0.553	22.899	0.648	0.337	7.942	0.675	0.414	12.449	0.571	0.256	4.859
	w_3	0.6	0.813	9.302	35.495	0.634	0.099	5.645	0.664	0.046	10.747	0.615	0.032	2.449	0.627	0.023	4.567
	w_4	0.6	0.631	3.992	5.115	0.565	0.09	5.849	0.665	0.052	10.858	0.614	0.036	2.414	0.627	0.023	4.521
MLE	a_1	0.3	5.923	84.936	1874.444	0.553	0.317	84.289	0.445	0.237	48.41	0.349	0.148	16.250	0.322	0.102	7.304
	a_2	3.8	40.882	675.238	975.848	4.249	1.085	11.808	4.065	0.669	6.980	3.883	0.359	2.171	3.844	0.261	1.158
				$\rho = 0.966$			$\rho = 0.998$			$\rho \approx 1$			$\rho \approx 1$			$\rho \approx 1$	

			$n = 50$			$n = 100$			$n = 200$			$n = 500$			$n = 1000$		
method	label	par	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%
ANN	w_1	0.6	0.79	1.079	31.644	0.597	0.308	0.42	0.647	0.328	7.901	0.611	0.293	1.84	0.599	0.285	0.115
	w_2	0.6	0.788	1.355	31.351	0.615	0.308	2.433	0.648	0.34	7.965	0.563	0.273	6.216	0.574	0.264	4.302
	w_3	0.6	0.665	1.05	10.799	0.638	0.094	6.269	0.556	0.054	7.329	0.624	0.028	4.073	0.595	0.023	0.874
	w_4	0.6	0.597	0.949	0.446	0.651	0.085	8.523	0.551	0.055	8.208	0.621	0.031	3.463	0.591	0.024	1.509
MLE	a_1	0.3	1.458	14.154	385.842	0.568	0.303	89.257	0.439	0.22	46.397	0.35	0.142	16.728	0.322	0.104	7.177
	a_2	3.8	23.976	553.861	530.946	4.234	1.062	11.421	3.996	0.615	5.162	3.898	0.378	2.566	3.838	0.26	1.002
				$\rho = 0.970$			$\rho = 0.998$			$\rho = 0.999$			$\rho \approx 1$			$\rho \approx 1$	

Table 2: Simulation study for the case of one binary endogenous variable. The top panel displays pure GLERA estimation (Figure 11, panel (a)), the bottom panel displays ANN / MLE estimation (Figure 11, panel (b)). Columns show estimates (est), standard errors (s.e.) and relative bias ($\times 100$) in absolute value (bias%). Sample sizes $n = \{50, 100, 200, 500, 1000\}$.

			$n = 50$			$n = 100$			$n = 200$			$n = 500$			$n = 1000$		
method	label	par	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%
MLE	w_1	0.6	1.013	7.583	68.835	0.643	0.316	7.198	0.636	0.637	6.003	0.624	0.453	3.954	0.589	0.263	1.87
	w_2	0.6	0.832	3.255	38.699	0.612	0.296	2.043	0.615	0.318	2.45	0.627	0.385	4.493	0.587	0.258	2.194
	w_3	0.6	0.73	5.457	21.683	0.578	0.087	3.599	0.606	0.048	0.966	0.606	0.034	1.012	0.634	0.022	5.716
	w_4	0.6	0.728	4.379	21.284	0.578	0.079	3.729	0.596	0.054	0.672	0.608	0.032	1.327	0.637	0.023	6.097
ANN	a_1	0.3	0.969	3.117	223.023	0.636	0.405	111.988	0.444	0.226	48.07	0.35	0.152	16.643	0.322	0.098	7.218
	a_2	3.8	5.622	15.511	47.944	4.415	1.185	16.191	4.021	0.661	5.815	3.891	0.39	2.408	3.831	0.255	0.813
				$\rho = 0.995$			$\rho = 0.997$			$\rho = 0.999$			$\rho \approx 1$			$\rho \approx 1$	

			$n = 50$			$n = 100$			$n = 200$			$n = 500$			$n = 1000$		
method	label	par	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%
ANN	w_1	0.6	2.205	8.517	267.471	0.684	0.371	14.055	0.668	0.331	11.268	0.596	0.29	0.723	0.607	0.254	1.209
	w_2	0.6	1.909	7.814	218.237	0.641	0.327	6.904	0.656	0.324	9.347	0.57	0.285	4.974	0.579	0.251	3.433
	w_3	0.6	1.456	5.177	142.644	0.58	0.091	3.326	0.618	0.046	3.013	0.575	0.033	4.095	0.612	0.023	1.925
	w_4	0.6	1.440	4.84	139.966	0.579	0.107	3.456	0.616	0.055	2.659	0.572	0.035	4.589	0.614	0.023	2.255
ANN	a_1	0.3	2.081	11.553	593.628	0.592	0.358	97.312	0.44	0.227	46.556	0.355	0.134	18.306	0.317	0.103	5.532
	a_2	3.8	12.406	49.075	226.472	4.419	1.234	16.302	4.054	0.65	6.696	3.894	0.386	2.472	3.832	0.251	0.831
				$\rho = 0.995$			$\rho = 0.998$			$\rho \approx 1$			$\rho \approx 1$			$\rho \approx 1$	

Table 3: Simulation study for one binary endogenous variable. The top panel displays MLE / ANN estimation (Figure 11, panel (c)), the bottom panel displays pure ANN-ERA estimation (Figure 11, panel (c)). Columns show estimates (est), standard errors (s.e.) and relative bias ($\times 100$) in absolute value (bias%). Sample sizes $n = \{50, 100, 200, 500, 1000\}$.

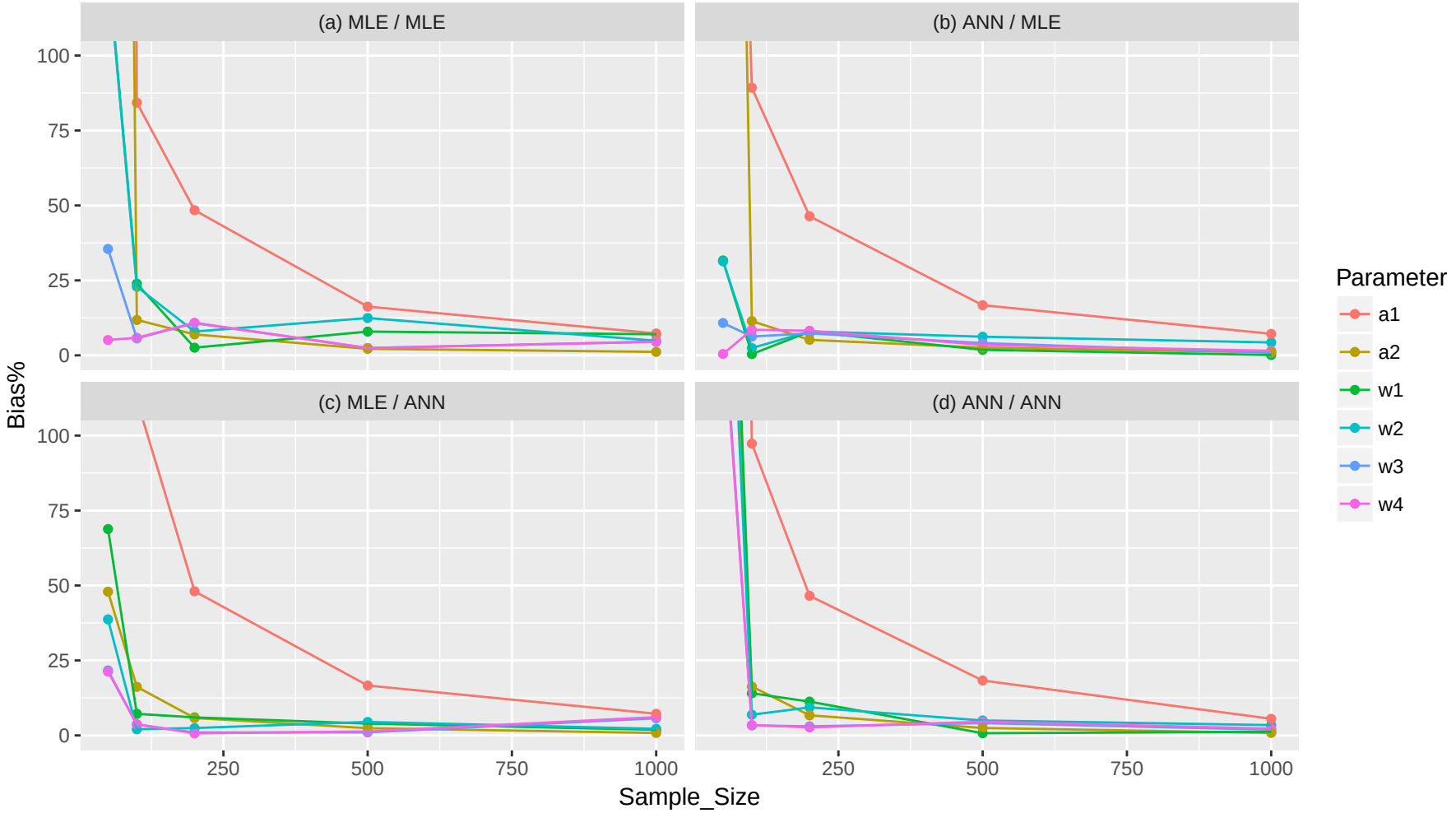


Figure 13: Simulation study for the case of one binary endogenous variable. Panel (a) displays pure GLERA estimation, panel (b) displays ANN / MLE estimation, panel (c) displays MLE / ANN estimation, panel (d) displays pure ANN-ERA estimation (See Figure 11 for the corresponding diagrams).

			$n = 50$			$n = 100$			$n = 200$			$n = 500$			$n = 1000$		
method		par	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%
VGLERA	$w_{1.1}$	0.7	0.422	0.592	39.663	0.534	0.459	23.744	0.605	0.386	13.604	0.685	0.205	2.171	0.705	0.142	0.651
ref: 3	$w_{1.2}$	0.7	0.337	0.642	51.918	0.633	0.34	9.641	0.687	0.222	1.9	0.703	0.138	0.482	0.711	0.093	1.513
	$w_{2.1}$	0.7	0.413	0.593	41.064	0.554	0.471	20.904	0.579	0.412	17.218	0.679	0.202	3.004	0.695	0.145	0.712
	$w_{2.2}$	0.7	0.325	0.648	53.563	0.63	0.341	10.002	0.67	0.227	4.236	0.695	0.133	0.747	0.706	0.093	0.924
	$w_{3.1}$	0.7	0.571	0.465	18.488	0.428	0.58	38.812	0.526	0.462	24.828	0.647	0.285	7.536	0.68	0.19	2.845
	$w_{3.2}$	0.7	0.548	0.477	21.746	0.645	0.332	7.807	0.669	0.222	4.444	0.692	0.136	1.213	0.693	0.098	0.935
	$w_{4.1}$	0.7	0.528	0.485	24.558	0.398	0.586	43.1	0.582	0.437	16.845	0.631	0.296	9.909	0.667	0.197	4.756
	$w_{4.2}$	0.7	0.553	0.472	20.968	0.617	0.332	11.869	0.696	0.213	0.555	0.682	0.139	2.616	0.692	0.098	1.103
	$a_{1.1}$	0.4	0.605	0.356	51.176	0.522	0.246	30.6	0.432	0.229	7.901	0.419	0.114	4.668	0.411	0.081	2.74
	$a_{1.2}$	0.6	0.76	0.39	26.677	0.695	0.294	15.794	0.651	0.192	8.509	0.615	0.117	2.561	0.611	0.084	1.771
	$a_{2.1}$	0.3	0.537	0.351	78.836	0.412	0.257	37.338	0.373	0.171	24.195	0.326	0.111	8.766	0.313	0.083	4.396
	$a_{2.2}$	0.6	0.763	0.405	27.198	0.687	0.275	14.509	0.651	0.182	8.501	0.619	0.117	3.23	0.609	0.086	1.525
				$\rho = 0.932$			$\rho = 0.974$			$\rho = 0.993$			$\rho \approx 1$			$\rho \approx 1$	

Table 4: Simulation study for VGLERA, first parameter setting. Columns show estimates (est), standard errors (s.e.) and relative bias ($\times 100$) in absolute value (bias%). Sample sizes $n = \{50, 100, 200, 500, 1000\}$.

			$n = 50$			$n = 100$			$n = 200$			$n = 500$			$n = 1000$		
method		par	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%	est	s.e.	bias%
VGLERA	$w_{1.1}$	0.7	0.628	0.368	10.344	0.663	0.267	5.295	0.677	0.192	3.330	0.674	0.108	3.704	0.692	0.078	1.137
ref: 3	$w_{1.2}$	0.7	0.568	0.413	18.854	0.598	0.384	14.614	0.685	0.173	2.095	0.686	0.107	2.001	0.696	0.074	0.586
	$w_{2.1}$	0.7	0.618	0.370	11.709	0.670	0.258	4.356	0.694	0.177	0.789	0.688	0.107	1.682	0.696	0.079	0.562
	$w_{2.2}$	0.7	0.625	0.406	10.693	0.610	0.388	12.842	0.691	0.176	1.269	0.676	0.110	3.414	0.693	0.074	0.984
	$w_{3.1}$	0.7	0.694	0.188	0.902	0.709	0.126	1.226	0.696	0.091	0.531	0.695	0.057	0.762	0.719	0.035	2.759
	$w_{3.2}$	0.7	0.678	0.258	3.210	0.618	0.370	11.646	0.699	0.090	0.172	0.694	0.054	0.787	0.718	0.035	2.521
	$w_{4.1}$	0.7	0.698	0.189	0.313	0.708	0.126	1.086	0.700	0.090	0.050	0.699	0.057	0.178	0.718	0.035	2.629
	$w_{4.2}$	0.7	0.677	0.233	3.238	0.614	0.373	12.348	0.697	0.089	0.373	0.699	0.053	0.079	0.720	0.035	2.873
	$a_{1.1}$	1	1.495	0.855	49.532	1.177	0.476	17.741	1.076	0.312	7.588	1.032	0.179	3.231	1.022	0.126	2.174
	$a_{1.2}$	-1	-1.376	0.830	37.620	-1.107	0.706	10.673	-1.082	0.301	8.163	-1.031	0.175	3.060	-1.016	0.121	1.560
	$a_{2.1}$	2	2.731	1.334	36.540	2.300	0.664	15.015	2.121	0.432	6.066	2.051	0.259	2.526	2.023	0.176	1.130
	$a_{2.2}$	-2	-2.599	1.317	29.950	-2.057	1.343	2.872	-2.133	0.445	6.635	-2.048	0.255	2.417	-2.022	0.170	1.119
				$\rho = 0.987$			$\rho = 0.996$			$\rho \approx 1$			$\rho \approx 1$			$\rho \approx 1$	

Table 5: Simulation study for VGLERA, second parameter setting. Columns show estimates (est), standard errors (s.e.) and relative bias ($\times 100$) in absolute value (bias%). Sample sizes $n = \{50, 100, 200, 500, 1000\}$.

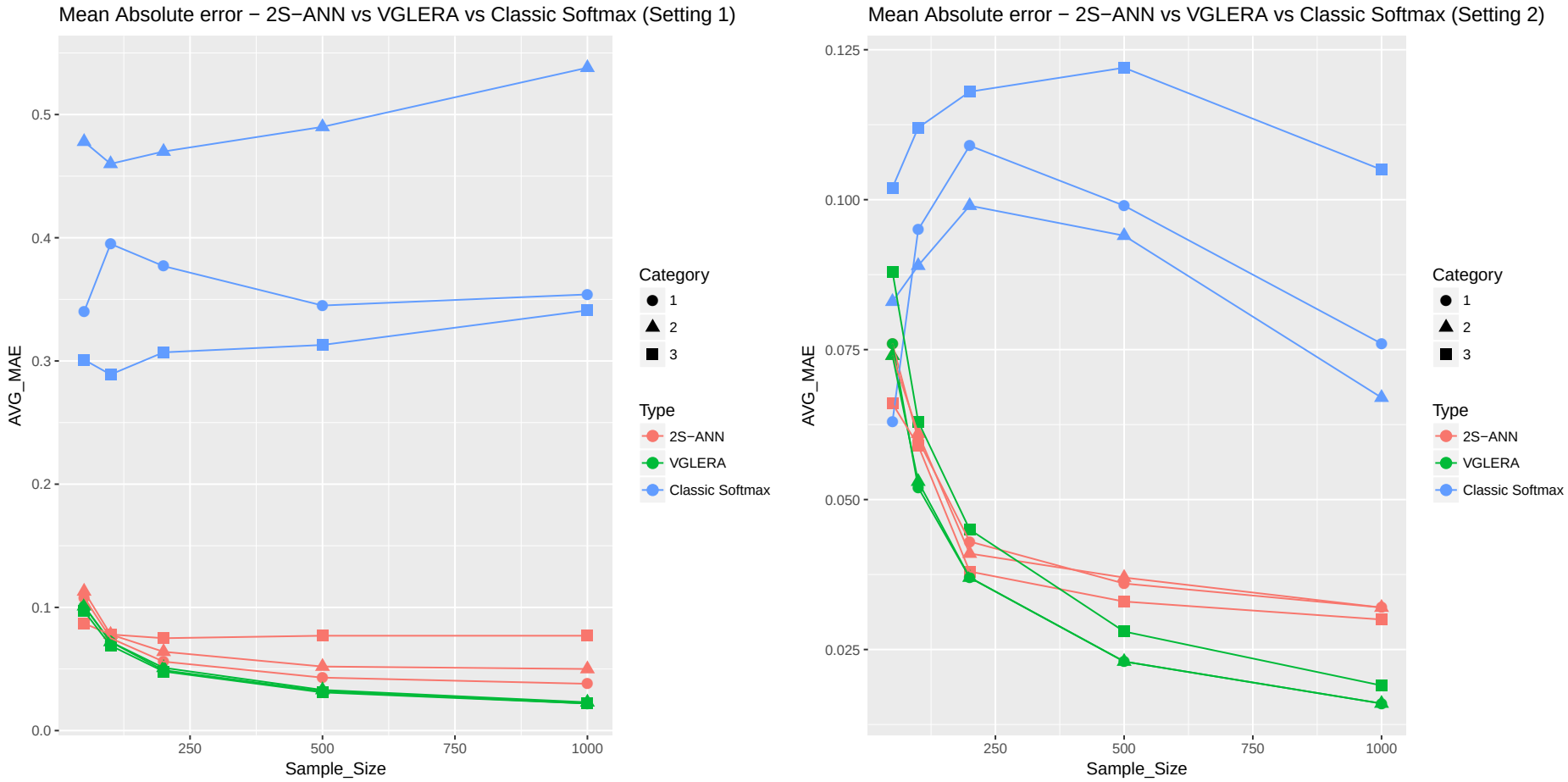


Figure 14: Comparison between VGLERA, 2S-ANN and standard Softmax neural network (no hidden layers) in terms of Mean Absolute Error (MAE), along with sample size. Left panel for setting 1, Right panel for setting 2.

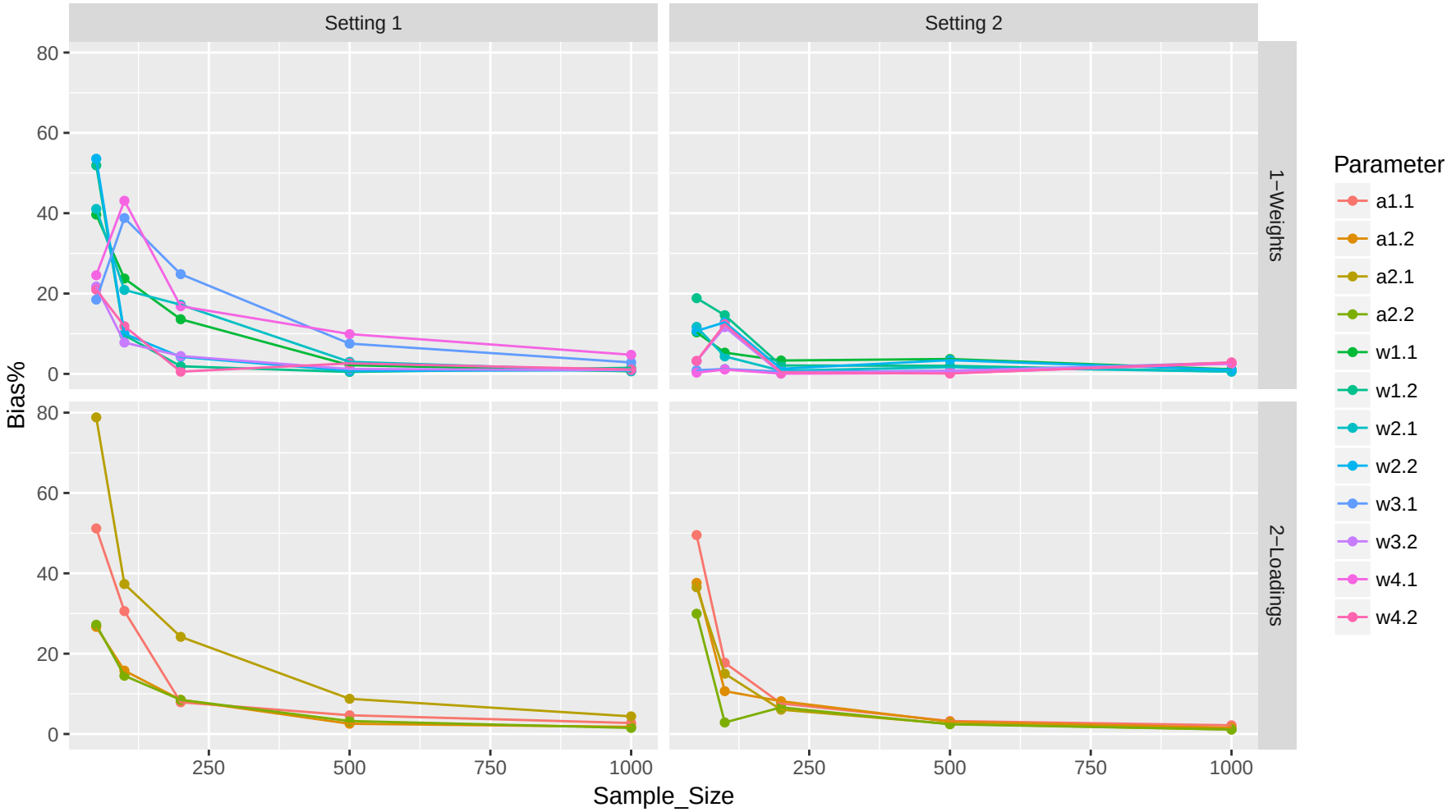


Figure 15: Simulation study for VGLERA. Variation of weights and loadings bias% along with sample size. Top panels are weight parameters, bottom panels are loading parameters, left panels for setting 1, right panels for setting 2.

Part V

Application: Two Practical Examples

1 Example 1: An Application in Marketing Research

The GLERA and VGLERA models, together with their ANN counterparts will now be illustrated, using a classic data set in marketing research, revisited with categorical coding of the endogenous variable. For a study in an industrial sales force in the framework of the classic LISREL model, Bagozzi (1980) examined self-performance appraisals for ($n = 122$) salespersons,with two endogenous latent constructs: a self-fulfilment latent construct, including sales performance which is hypothesized to be a surrogate, and a job satisfaction latent construct, composed of two related manifest indicators. The researcher further believes that exogenous background factors such as achievement, motivation, task specific self-esteem and verbal intelligence may have an impact on the aforementioned job satisfaction and performance measures. The endogenous blocks collected performance as dollar volume of sales achieved by each salesperson (Sales) and two measures of self-satisfaction: the first concerns the degree of satisfaction with promotion, pay, and the overall work situation (Sat1), the second the degree of satisfaction with opportunity to demonstrate ability and initiative and sense of accomplishment (Sat2). Regarding the exogenous blocks, one is related to self-esteem, measuring each salesperson's attributions (in relation to other salespeople in the company) in the firm (Self1, Self2), the second is made by two measures of salesperson's motivations (Motiv1, Motiv2), and the third is a single observed variable measuring cognitive ability to accurately and efficiently perceive, attend and process information associated with the job (Verbal). In that paper the main question was whether sales performance influences satisfaction, or vice-versa, taking into account the exogenous background factors.

In this context, to show a simple application of the new proposition in this work, the two exogenous blocks will be retained, whereas the endogenous categorical variable will be a recoding of (Sat1) in two or three categories based on the quantiles of the original numerical variable, to show how GLERA (and its ANN variations) and VGLERA (and its 2S-ANN variation) respectively perform. For the dichotomization, the median has been chosen as the cut-point, while the first and third quartiles were chosen for the recoding in three categories. In addition, parameters standard errors will be calculated with bootstrap, and their significance will be evaluated using the critical ratio (CR), dubbing a coefficient as significant if its $|\text{CR}| > 2$.

1.1 Results - GLERA

Table 6 and Figure 16 show the results for the GLERA model. Looking at the estimated coefficients it is to be noted that, with respect to the baseline of low satisfaction, all the parameters are strictly positive, meaning that all the exogenous variables have a positive impact on the response variable. However, the only significant ones concern (Motiv2) and the Motivation latent composite.

Binary Response. Reference Category: Sat1 = 0				
Par	Est.	Boot Mean	Boot SE	CR
Motiv1 → Motivation	0.296	0.230	0.325	0.911
Motiv2 → Motivation	**0.857**	0.799	0.234	3.664
Self1 → Self-Esteem	0.438	0.370	0.549	0.798
Self2 → Self-Esteem	0.696	0.573	0.520	1.338
Motivation→ Sat1	**0.611**	0.666	0.236	2.587
Self-Esteem→ Sat1	0.381	0.460	0.213	1.788

Table 6: Marketing data example with the GLERA model. "Est." column contains parameter estimates for the original dataset, "Boot Mean", "Boot SE" and "Boot CR" show bootstrap mean estimates, bootstrapped standard errors and Critical Ratios, respectively. Significant parameters ($|\text{CR}| > 2$) in bold.

Table 7 show performance comparisons between GLERA and full or hybrid ANN specifications, in terms of misclassification rate. Class membership is selected based on the highest predicted probability value. All the specifications yield similar results.

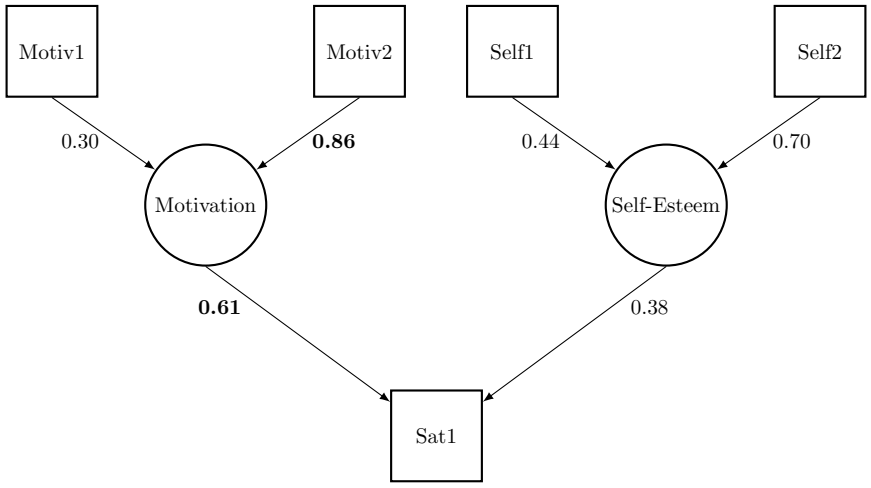


Figure 16: Marketing data example. GLERA Application path diagram. Significant parameters ($|CR| > 2$) in bold.

Method	Miscl. Rate (LOOCV: Overall)	Miscl. Rate (LOOCV: Sat1 = 0)	Miscl. Rate (LOOCV: Sat1 = 1)
MLE / MLE (a)	0.387	0.373	0.402
ANN on $\mathbf{W}$ (b)	0.385	0.368	0.404
ANN on $\mathbf{A}'$ (c)	0.385	0.368	0.404
ANN / ANN (d)	0.386	0.368	0.404

Table 7: Marketing data example. Comparison of prediction capabilities for the four different specifications of the GLERA model (letters (a) to (d) as in Figure 11, evaluated with Leave One Out Cross Validation (LOOCV).

1.2 Results - VGLERA / 2S-ANN

Table 8 and Figure 17 show the results for the VGLERA model. Looking at the estimated coefficients it is to be noted that, compared to the lowest level of (Sat1), variables (Motiv2) and (Self1) play an increasing role in the level of satisfaction, going from 0.692 for an average level of satisfaction, to 1.04 for a high level of satisfaction. Variables (Motiv1) and (Self2) show a decreasing, yet non-significant pattern in determining the level of satisfaction. In fact, only the comparison with the highest level of (Sat1) shows significant results, and only for the increasing coefficients, i.e. (Motiv2) and (Self1). Concerning the loadings, only the comparison with the highest level of (Sat1) shows significant results, with a slight prevalence of the self-esteem latent composite, that has a higher loading. Compared to GLERA, adding a new category provides more insight on the behavior of exogenous variables

Categorical Response (Ref: Sat1 = 1)	Par	Est.	Boot Mean	Boot SE	Boot CR
Sat1 = 2	**Motiv1 → Motivation**	0.517	0.386	0.584	0.886
	Motiv2 → Motivation	0.692	0.472	0.596	1.161
	Self1 → Self-Esteem	0.269	0.186	0.593	0.454
	Self2 → Self-Esteem	0.831	0.707	0.492	1.69
	Motivation→ Sat1	0.34	0.454	0.252	1.348
	Self-Esteem → Sat1	0.461	0.551	0.263	1.75
Sat1 = 3	**Motiv1 → Motivation**	-0.114	-0.103	0.379	-0.3
	Motiv2 → Motivation	**1.04**	0.965	0.179	5.818
	Self1 → Self-Esteem	**1.003**	0.927	0.286	3.506
	Self2 → Self-Esteem	0.002	0.004	0.414	0.005
	Motivation→ Sat1	**0.551**	0.605	0.213	2.591
	Self-Esteem → Sat1	**0.606**	0.672	0.244	2.485

Table 8: Marketing data example with the VGLERA model. "Est." column contains parameter estimates for the original dataset, "Boot Mean", "Boot SE" and "Boot CR" show bootstrap mean estimates, bootstrapped standard errors and Critical Ratios, respectively. Significant parameters ($|CR| > 2$) in bold.

and latent composites, turning the results in favor of Self-Esteem condition in the prediction of the level of satisfaction.

Table 9 shows performance comparisons among VGLERA, 2S-ANN and two standard classification networks without forming latent composites (i.e. a Softmax ANN and a more general pattern recognition network composed of one hidden layer with 10 neurons and one softmax layer). The confusion matrix for each model has been calculated, selecting class membership based on the highest predicted probability value. The 2S-ANN specification performs comparatively better w.r.t. Multinomial Logit and VGLERA, when classifying the medium category, which accounts for 50% of the total observations; the VGLERA specification, conversely, performs better with the low and high categories, but does not manage to recover the true group membership for the medium category as efficiently. Multinomial Logit performs worse than 2S-ANN and VGLERA for this dataset, and the standard classification network, while performing comparatively better than all the other methods in general, fails completely to predict individuals in the third category.

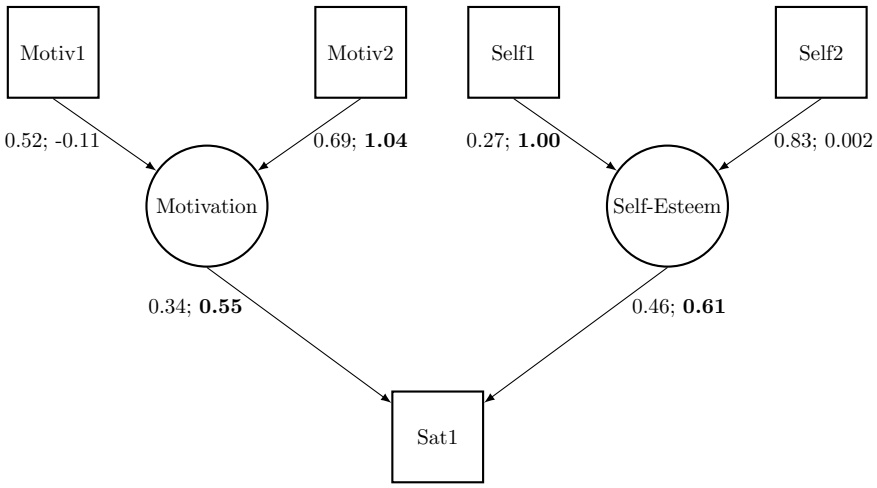


Figure 17: Marketing data example. VGLERA Application path diagram. Estimates of second and third category w.r.t. baseline (Sat1 = 1) are separated by a semicolon. Significant parameters (|CR| > 2) in bold.

VGLERA				Multinomial Logit		
	$\widehat{\text{Sat1}} = 1$	$\widehat{\text{Sat1}} = 2$	$\widehat{\text{Sat1}} = 3$	$\widehat{\text{Sat1}} = 1$	$\widehat{\text{Sat1}} = 2$	$\widehat{\text{Sat1}} = 3$
Sat1 = 1	17	7	7	17	7	7
Sat1 = 2	22	17	21	23	16	21
Sat1 = 3	8	5	18	9	4	18
LOOCV Acc.%		0.429			0.431	

2S-ANN				Hidden Sigmoid + Softmax ANN		
	$\widehat{\text{Sat1}} = 1$	$\widehat{\text{Sat1}} = 2$	$\widehat{\text{Sat1}} = 3$	$\widehat{\text{Sat1}} = 1$	$\widehat{\text{Sat1}} = 2$	$\widehat{\text{Sat1}} = 3$
Sat1 = 1	15	8	8	8	23	0
Sat1 = 2	19	23	18	6	54	0
Sat1 = 3	8	9	14	1	30	0
LOOCV Acc.%		0.442			0.529	

Table 9: Marketing data example. Comparison between the confusion matrices on the original dataset of VGLERA (top-left table), 2S-ANN (bottom-left table), Multinomial Logit (top-right table) and Softmax Neural Network with preceding hidden layer - 10 neurons - (bottom-right table). The bottom line of each table provides LOOCV Accuracy of classification.

2 Example 2: The Iris Dataset

This famous dataset consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. For the purpose of illustrating the VGLERA and 2S-ANN prediction capabilities, two latent composites are created: the petal LC, whose manifest variables are (Petal Length, Petal Width), and the Sepal LC, whose manifest variable are (Sepal Length, Sepal Width). The response variable is the (Species) of each iris. Parameter estimates will not be analyzed, being not statistically significant with the use of a preliminary multinomial logistic regression, thus the focus will be on correct classification of each species.

2.1 Results - VGLERA / 2S-ANN

Table 10 shows performance comparisons among VGLERA, 2S-ANN and two standard classification methods without forming latent composites (i.e. a multinomial logistic regression and a pattern recognition network composed of one hidden layer with 10 neurons and one Softmax layer). For each of the models, the confusion matrix for each model has been calculated, having class membership selected based on the highest predicted probability value. The 2S-ANN specification performs comparatively better when classifying Versicolor, and worse when classifying the Virginica, with respect to VGLERA; Multinomial Logit performs worse than VGLERA and 2S-ANN for Versicolor, but better than both for Virginica; in general, neural network models tend to wrongly predict Virginica species, while compensating by correctly predicting Versicolor species. Lastly, between 2S-ANN and Softmax neural network with one hidden layer, the latter performs better, thanks to the additional hidden layer. For a graphical representation of the VGLERA classification performance, which has offered the best correct classification rate, see Figure 18.

VGLERA			
	$\widehat{\text{Setosa}}$	$\widehat{\text{Versicolor}}$	$\widehat{\text{Virginica}}$
Setosa	50	0	0
Versicolor	0	33	17
Virginica	0	5	45
LOOCV Acc.%		0.850	

Multinomial Logit			
	$\widehat{\text{Setosa}}$	$\widehat{\text{Versicolor}}$	$\widehat{\text{Virginica}}$
Setosa	50	0	0
Versicolor	0	30	20
Virginica	0	3	47
LOOCV Acc.%		0.848	

2S-ANN			
	$\widehat{\text{Setosa}}$	$\widehat{\text{Versicolor}}$	$\widehat{\text{Virginica}}$
Setosa	50	1	0
Versicolor	0	38	12
Virginica	0	17	33
LOOCV Acc.%		0.824	

Hidden Sigmoid + Softmax ANN			
	$\widehat{\text{Setosa}}$	$\widehat{\text{Versicolor}}$	$\widehat{\text{Virginica}}$
Setosa	49	1	0
Versicolor	4	42	4
Virginica	0	14	36
LOOCV Acc.%		0.735	

Table 10: Iris example. Comparison between the confusion matrices on the original dataset of VGLERA (top-left table), 2S-ANN (bottom-left table), Multinomial Logit (top-right table) and Softmax Neural Network with preceding hidden layer - 10 neurons - (bottom-right table). The bottom line of each table provides LOOCV Accuracy of classification.

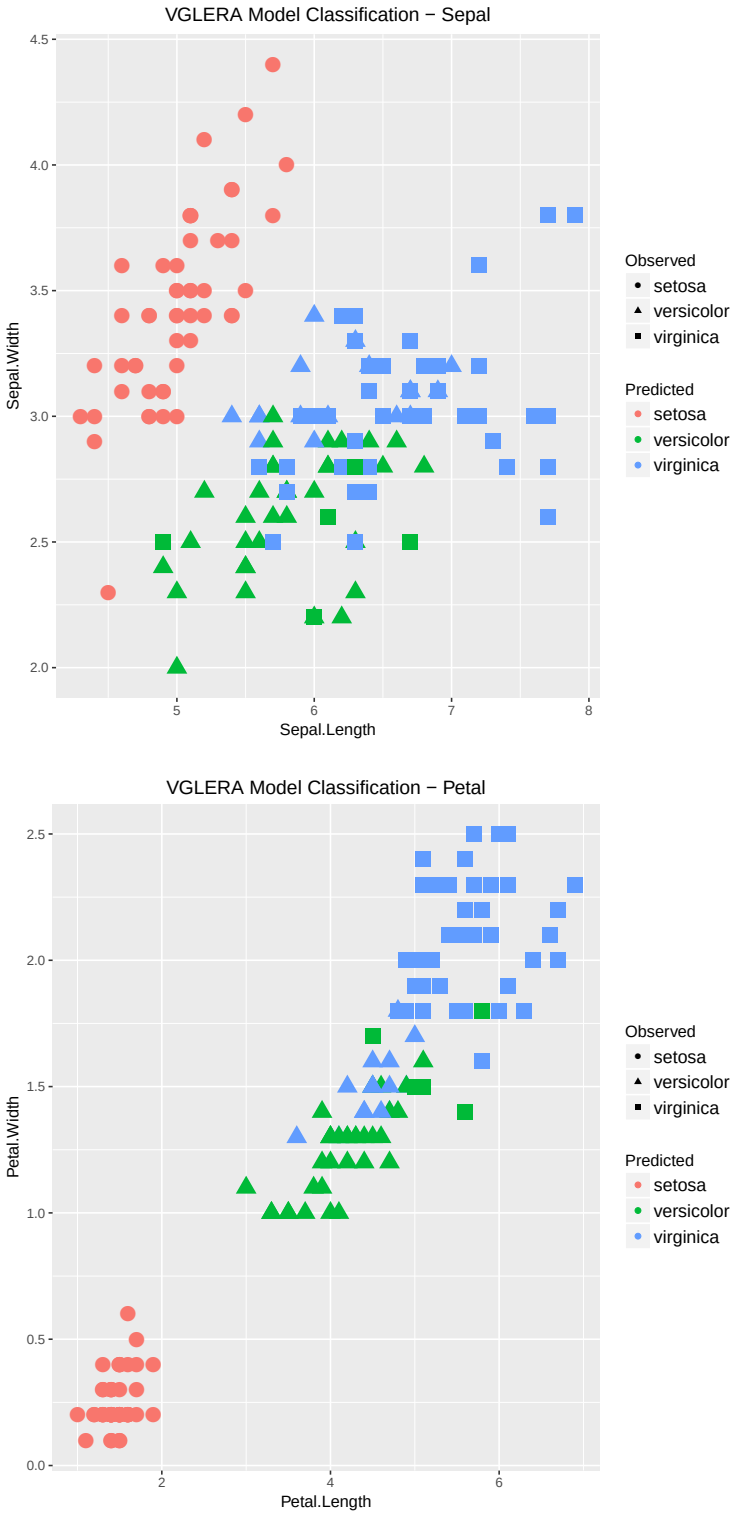


Figure 18: Classification results for the iris dataset, VGLERA model. Sepal variables on the top panel, petal variables on the bottom panel.

Part VI

Conclusion

Structural Equation Models with latent variables have recently undergone considerable development, in terms of estimation techniques, extensions and inferential capability. Several methods are available when dealing with categorical or ordinal observed variables: for the LISREL model, they make use of adaptations of the EM algorithm, whereas for PLS-PM they are mostly related to Optimal Scaling of the categorical variables into pseudo-quantitative variables. In the Redundancy Analysis framework, with only observed endogenous variables, the possibility of extending the estimation procedures to a qualitative setting meet favorable conditions, not having strong model restrictions and a unified optimization criterion. In fact, this work has proved that, especially for the Extended Redundancy Analysis model, the GLERA extension herein proposed makes resorting to Optimal Scaling and its potential flaws in the transformation of categorical variables into numerical surrogates unnecessary.

The simulation results give credit to the potential of this new model in the qualitative setting, with a large enough sample. Considering also the Artificial Neural Network setting, in which SEM have still found limited applicability, ERA can be readily adapted for classification problems, with three other competing strategies that yielded results comparable to GLERA. In fact, bias values rapidly decrease with higher sample size and have the tendency to uniform, regardless of which of the four strategy is chosen. The only occasion in which ANN estimation seems to perform better than MLE is the MLE / ANN strategy, but the ability to recover the true parameter values is still in question for lower sample sizes.

In the multinomial case, VGLERA has been presented as a natural extension of GLERA, exploiting the possibilities offered by Vector GLM models and Iterated Reweighed Least Squares combined with the Alternating Least Squares algorithm already used in the ERA model. The simulations have shown that VGLERA manages to recover the true parameter values efficiently with reasonable sample size, in

two parameter settings reflecting different capability of the underlying model to separate the response categories. Considering also Artificial Neural Networks, VGLERA outperforms 2S-ANN in all sample sizes for all categories, yet both modeling techniques offer satisfying prediction capabilities. However, if the focus is on parameter values, VGLERA offers category-vs-baseline estimates which are more directly interpretable, whereas in 2S-ANN predicted probabilities are the core feature. The standard softmax neural network, modeling directly the response onto the exogenous variables without the use of latent composites, offers drastically worse performance: reasons for this may be found either in the absence of the latent composites, that if present manage to provide better estimates, by separating weights and loadings parameters, or in the tendency of neural networks to give unbalanced probability towards the chosen category.

The application on a classic dataset used in the SEM framework has shown clear and interpretable results, with additional insights depending on the number of categories and different prediction capabilities depending on the chosen methodology. For this dataset, and for dichotomous response, GLERA and its ANN variations offer virtually the same prediction power, whereas in the multinomial setting 2S-ANN does not manage to generally outperform existing classification methods based on neural networks such as Softmax with one hidden layer. There is however no method clearly emerging as preferable: for the marketing data example, general prediction based on Softmax network gives better results, but misclassifies one entire category; for the iris example, VGLERA generally outperforms the other three methods, and Softmax behaves better than 2S-ANN, thanks to the presence of the hidden layer.

These results offer also the opportunity to further extend the capabilities of ERA, both in the GLM and in the ANN frameworks: firstly, this work deals with logistic links, but can be readily extended to different link functions in future developments; secondly, these new classification models have yet to be adapted and extended with ordinal response or in presence of external covariate, which may affect directly the endogenous variable(s), or affect both the latent composites and the endogenous variable(s) simultaneously; thirdly, the possibility of having a mix of categorical and quantitative endogenous variables has yet to be explored; then, looking at the

marketing example and the iris example, noticing how they offer different results in term of correct classification, further studies are necessary to investigate the model under unbalanced designs; finally, especially looking at the iris example and noticing how hidden layers improve the performance, the potential of neural network analysis shows at its best when dealing with non-linearities in the data, or in non-parametric analyses, hence ERA for neural networks can be also employed for classification with more hidden layers, mapping latent composites onto high dimensional feature spaces before fitting them with the response variable.

Appendix: Code Excerpts in R and Matlab ®

This appendix offers supplementary information on the scripts used for GLERA and VGLERA, coded in R [R Core Team, 2016]. The ANN variations are coded in MATLAB ® [2016], creating custom neural networks for; excerpts of the ANN-ANN configuration in the binary case will be shown.

1 R Code for the GLERA model

The code below invokes the GLERA algorithm for binary response

```
attach(jobsat)
glera.oneshot=glera(
               y = sat.cat,                          #binary response variable
               X = list(X1 = cbind(motiv1,motiv2), #list of blocks of exogenous
                   variables
                        X2 = cbind(self1,self2)),
               winit = 0.5, ainit = 1,               #starting value for W and a'
               maxiter = 100,                        #maximum number of GLERA
                   iterations
               tol = 0.00001,                        #threshold value
```

Inside the function, the two steps of GLERA are iterated until convergence.

```
      # STEP 1 = W
      vW = vec(W)[vec(W) != 0, drop=FALSE] #W matrix is vectorized and zeroes are
          deleted
      gamma = (t((A)) %x% Z)[,vec(W) != 0,drop=FALSE] #corresponding columns of A x
           Z are deleted
      df = data.frame(gamma = gamma)
      y2 = relevel(y, ref = base)  #set baseline according to specified input
      mw = glm(y2 ~ 0 + ., data = df, family = "binomial") #fit logistic with no
          intercept
      w = mw$coefficients  #extract coefficients
      W = update(W,t(w))
```

Estimate of **W**

```
1       FZ = Z %*% W #compute F and normalize
2       FZ = normLC(FZ)
3
4       #STEP 2 = A
5       df2 = data.frame(FZ = FZ)
6       y2 = relevel(y, ref = base) #set baseline according to specified input
7       ma = glm(y2 ~ 0 + ., data = df2, family = "binomial") #fit logistic with no
            intercept
8       A = ma$coefficients #extract coefficients
```

Estimate of $\mathbf{a}'$

The algorithm stops if either the maximum number of iterations is reached or the difference between subsequent estimates is lower than the specified threshold.

```
1       iter = iter + 1  #iteration number increases
2       if ((abs(OldW - W) < tol && abs(OldA - A) < tol) | iter > maxiter) #stop
            criterion
3       { break }
```

Stopping criteria

The output is a list containing the estimated parameter matrices and the predicted probabilities for the outcome.

```
1   #output
2   out = list(weights = W,  #weight matrix W
3              loadings = A,   #loading matrix a'
4              fitted.values = ma$fitted.values) #predicted probabilities
5   return(out)
```

Output generation

2 R Code for the VGLERA model

The code below invokes the VGLERA algorithm for multinomial response

```
attach(iris)
oneshot.vglera=vglera(y = Species,           #categorical response variable
                      X = list(X1 = cbind(Sepal.Length, Sepal.Width), #list of
                          blocks of exogenous variables
                               X2 = cbind(Petal.Length, Petal.Width)),
                      winit = 2, ainit = -3, #starting value for W's and a's
                      maxiter = 100, #maximum number of ALS iterations
                      tol1 = 0.0001, #threshold value for IRLS
                      tol2 = 0.0001, #threshold  value for ALS
                      base = 1,  #baseline category
                      eps=0.002, #adjustment for initial BFGS hessian matrix
                      boot=FALSE) #TRUE for reduced output in case of bootstrap
```

Inside the function, the two steps of VGLERA are iterated until convergence. The code chunk that transforms the initial input (as in Part III, Section 1.2) is omitted. Specifically, IRLS is fitted for $\mathbf{W}$ and $\mathbf{a}'$, alternately. BFGS algorithm used to update the individual $\mathbf{K}_i$ matrices.

As initial $\mathbf{K}_i$'s, the linear predictor based on the initial parameter values is considered, and it is multiplied by `eps` to ensure matrix inversion.

```
   #STEP 1: W FOR FIXED A
   omega = (Omegavlm)[,lambdav != 0, drop=FALSE] #drop zero columns from Omega
   lambdav = lambdav[lambdav != 0, drop=FALSE]  #drop zero elements from vector of
       weights

   #initial values for BFGS
   #linear predictor
   eta = omega %*% lambdav * eps

   listeta = list() #list of individual linear predictors
   listx = list()   #list of individual covariates
   listy = list()   #list of individual response pattern

   for (i in 1 : nrow(omega0))
      {
         low = (2 * (i - 1)) + 1
         up = 2 * i
         listx[[i]] = omega[low:up,]
```

```
        listeta[[i]] = eta[low:up]
        listy[[i]] = vy[low:up]
      }

  #computing concatenation of individual probabilities with logistic function
  p = as.matrix(vec(sapply(listeta, FUN = function(x){exp(x) / (1 + sum(exp(x)))})
      ))
  p.bis = as.matrix(t(sapply(listeta, FUN = function(x){exp(x)/(1 + sum(exp(x)))})
      ))

  #vertical concatenation of individual hessian matrices
  pmat = hadamard.prod(( - p.bis %x% rep(1, 2) + rep(1, nrow(p.bis)) %x% diag(2)),
                         (matrix(p,nrow = nrow(p), ncol = 2)))

  listch = list() #list of individual hessian matrices
  listu = list()  #list of individual gradients
  for (i in 1 : nrow(omega0))
     {
       low = (2 * (i - 1)) + 1
       up = 2 * i
       listu[[i]] = listy[[i]] - p[low:up]
       listch[[i]] = pmat[low:up,]
     }

  listz = list()  #list of individual adjusted responses
  for (i in 1 : nrow(omega0))
     {
       listz[[i]] = listeta[[i]] + chol2inv(listch[[i]]) %*% listu[[i]]
```

Estimate of $\mathbf{W}$'s: initial value for individual $\mathbf{K}_i$'s.

Then IRLS iterates to obtain the estimate of $\mathbf{W}$, for fixed $\mathbf{a}'$.

```
   repeat{
     #save old quantities
     old.listeta = listeta
     old.listu = listu
     old.listch = listch

     #individual estimates
     listb1 = list()
     listb2 = list()
     for (i in 1 : nrow(omega0))
     {
        listb1[[i]] = t(listx[[i]]) %*% listch[[i]] %*% listx[[i]]
        listb2[[i]] = t(listx[[i]]) %*% listch[[i]] %*% listz[[i]]
```

```
14      }
15
16      #new parameter estimate
17      lambdanew = chol2inv(Reduce('+', listb1)) %*% Reduce('+', listb2)
18
19      #break loop if difference between subsequent iteration is lower than
20      #the threshold value
21      if(mean(lambdav - lambdanew) < tol1)
22         {break}
23
24      #update quantities
25      lambdav=lambdanew
26
27      #compute BFGS updates:
28      #linear predictor
29      eta=omega%*%lambdav
30
31      listeta=list()   #list of individual linear predictors
32      listx = list()   #list of individual covariates
33      listy = list()   #list of individual response pattern
34
35
36      for (i in 1 : nrow(omega0))
37      {
38         low = (2 * (i - 1)) + 1
39         up = 2 * i
40         listx[[i]] = omega[low:up,]
41         listeta[[i]] = eta[low:up]
42         listy[[i]] = vy[low:up]
43      }
44
45      #computing concatenation of individual probabilities with logistic function
46      p = as.matrix(vec(sapply(listeta, FUN = function(x){exp(x) / (1 + sum(exp(x))
            )})))
47      p.bis = as.matrix(t(sapply(listeta, FUN = function(x){exp(x) / (1 + sum(exp(x
            )))})))
48
49      #vertical concatenation of individual hessian matrices
50      pmat = hadamard.prod(( - p.bis %x% rep(1, 2) + rep(1, nrow(p.bis)) %x% diag
            (2)),
51                           (matrix(p,nrow = nrow(p),ncol = 2)))
52
53      listu = list() #list of individual gradients
54
55      for (i in 1 : nrow(omega0))
56      {
57        low = (2 * (i - 1)) + 1
```

```
58          up = 2 * i
59          listu[[i]] = listy[[i]] - p[low:up]
60        }
61
62        #BFGS updates
63
64        listq = list() #difference between subsequent estimates of gradient
65        lists = list() #difference between subsequent estimates of linear predictor
66
67        for(i in 1 : length(listeta)){
68
69           listq[[i]] = - (listu[[i]] - old.listu[[i]])
70           lists[[i]] = listeta[[i]] - old.listeta[[i]]
71
72           #BFGS update of Ki's
73           listch[[i]] = old.listch[[i]] +
74              (t(t(listq[[i]])) %*% t(listq[[i]])) /
75              matrix(rep(t(lists[[i]]) %*% t(t(listq[[i]])), 4), nrow = 2, ncol = 2)
                   -
76              (old.listch[[i]] %*% lists[[i]] %*% t(lists[[i]]) %*% old.listch[[i]])
                   /
77              matrix(rep((t(lists[[i]]) %*% old.listch[[i]] %*% lists[[i]]), 4), nrow
                    = 2, ncol = 2)
78           }
79
80        listz=list()
81
82        for (i in 1 : nrow(omega0))
83        {
84          listz[[i]] = listeta[[i]] + chol2inv(listch[[i]]) %*% listu[[i]]
85        }
86      }
```

IRLS for $\mathbf{W}$ with BFGS updates.

The same method is applied for $\mathbf{a}'$, for fixed $\mathbf{W}$. The algorithm stops if either the maximum number of iterations is reached or the difference between subsequent estimates is lower than the specified threshold (as in GLERA).

The output is a list containing the estimated parameter matrices and the predicted probabilities for the outcome (as in GLERA).

3 Matlab ® Code for ANN sections

The code below fits the ANN-ANN configuration for the binary case, giving out parameter matrices and predicted category.

```
%create network for W
netW = network( ...
1, ... % numInputs, number of inputs,
1, ... % numLayers, number of layers
[0], ... % biasConnect, numLayers-by-1 Boolean vector,
[1], ... % inputConnect, numLayers-by-numInputs Boolean matrix,
[0], ... % layerConnect, numLayers-by-numLayers Boolean matrix
[1] ... % outputConnect, 1-by-numLayers Boolean vector
);
% number of hidden layer neurons
netW.layers{1}.size = 1;
% hidden layer transfer function
netW.layers{1}.transferFcn = 'logsig';

% create network for a'
netA = network( ...
1, ... % numInputs, number of inputs,
1, ... % numLayers, number of layers
[0], ... % biasConnect, numLayers-by-1 Boolean vector,
[1], ... % inputConnect, numLayers-by-numInputs Boolean matrix,
[0], ... % layerConnect, numLayers-by-numLayers Boolean matrix
[1] ... % outputConnect, 1-by-numLayers Boolean vector
);
% number of hidden layer neurons
netA.layers{1}.size = 1;
% hidden layer transfer function
netA.layers{1}.transferFcn = 'logsig';

%initial values for W and a'
ww = 0.6;
aa = 1;
W = [ww, 0;
     ww, 0;
     0, ww;
     0, ww];

A = [aa; aa];

%input matrix X
X = [motiv1 motiv2 self1 self1];
```

```
41
42 edges = [-Inf 0 Inf];
43
44 sat.cat = discretize(sat1, edges);
45
46 %ANN-ANN fitting
47 exit = 0; %loop exit flag
48 iter = 0; %iteration count
49 tol = 0.00001 %threshold value
50 while exit == 0
51
52 if cyc > 0
53 W = Wupd;
54 A = Aupd;
55 end
56 cyc = cyc+1;
57
58 %STEP 1: estimate of W
59 vW = W( : ); %vector of W
60 vWs = W( : );
61 Omega = kron(A', X); % A kronecker X
62
63 %delete columns of Omega corresponding to zeroes in vec(W)
64 Omega( :, find(~vW)) = [];
65 %delete zeroes in vec(W)
66 vW( find(~vW), : ) = [];
67
68 %configure input and outputs for network
69 inputs = {Omega'};
70 outputss = {sat.cat'};
71
72 netW.trainFcn = 'traingdx'; %gradient descent algorithm
73 netW.performFcn = 'crossentropy'; %cross-entropy objective function
74 netW = configure(netW,inputs,outputss); %net configuration
75
76 netW.IW{1,1} = vW( find(vW), : )'; %initial net weights
77 netW.trainParam.showWindow = 0;
78 netW = train(netW, inputs, outputss); %net training
79
80 PAR = getwb(netW); %extract weigths
81 Wupd = zeros([8,1]);
82 Wupd(find(vWs)) = PAR;
83 Wupd = reshape(Wupd,[4,2]);
84
85 %compute F and normalize
86 F = X * Wupd;
87 F = zscore(F);
```

```
88
89 %STEP 2: estimate of a'
90 vA = A( : );
91 vAs = A( : );
92 Gamma = kron(eye(1), F);
93
94 Gamma( :, find(~vA)) = [];
95 vA( find(~vA), : ) = [];
96
97 %configure input and outputs for network
98 inputs = {Gamma'};
99 outputss = {sat.cat'};
100
101 netA.trainFcn = 'traingdx';
102 netA.performFcn = 'crossentropy';
103 netA = configure(netA, inputs, outputss);
104
105 netA.IW{1,1} = vA( find(vA), : )';
106 netA.trainParam.showWindow = 0;
107 netA = train(netA, inputs, outputss);
108
109 PAR = getwb(netA); %extract loadings
110 Aupd = reshape(PAR,[2,1]);
111
112 if cyc > 50 || (all(abs(W( : ) - Wupd( : )) < tol) && all(abs(A( : ) - Aupd( : )) <
       tol))
113 exit = 1;
114 end
115 end
116
117 %parameter estimates
118 Wupd
119 Aupd
120
121 %predicted category
122 predY = netA(F') > 0.5
```

MATLAB ® code for the ANN-ANN configuration

References

A. Agresti. *Categorical Data Analysis: 2nd Edition.* Wiley, 2002.

D. F. Alwin and R. M. Hauser. The decomposition of effects in path analysis. *American Sociological Review*, 40(1):37–47, 1975.

R. P. Bagozzi. Performance and satisfaction in an industrial sales force: An examination of their antecedents and simultaneity. *the Journal of Marketing*, pages 65–77, 1980.

R. P. Bagozzi and C. Fornell. *A second generation of multivariate analysis: Measurement and evaluation.* Praeger, 1982.

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

M. Barker and W. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17(3):166–173, 2003.

M. S. Bartlett. Further aspects of the theory of multiple regression. *Mathematical Proceedings of the Cambridge Philosophical Society*, 34(1):33–40, 1938.

C. M. Bishop. *Neural networks for pattern recognition.* Oxford university press, 1995.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag New York, 2006.

J. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects.* Springer Science & Business Media, 2006.

S. Bougeard, E. M. Qannari, C. Lupo, and M. Hanafi. From multiblock partial least squares to multiblock redundancy analysis. a continuum approach. *Informatica*,

22(1):11–26, 2011.

H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, 1988.

G. Cantaluppi. A partial least squares algorithm handling ordinal variables also in presence of a small number of categories. *ArXiv e-prints*, 2012.

W. W. Chin. The partial least squares approach to structural equation modeling. *Modern methods for business research*, 295(2):295–336, 1998.

W. W. Chin, B. L. Marcolin, and P. R. Newsted. A partial least squares latent variable modeling approach for measuring interaction effects: Results from a monte carlo simulation study and an electronic-mail emotion/adoption study. *Information systems research*, 14(2):189–217, 2003.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

F. Davies, M. Goode, J. Mazanec, and L. Moutinho. Lisrel and neural network modelling: Two comparison studies. *Journal of Retailing and Consumer Services*, 6(4):249–261, 1999.

P. T. Davies. Procedures for reduced-rank regression. *Applied Statistics*, pages 244–255, 1982.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

W. S. DeSarbo, H. Hwang, A. S. Blank, and E. Kappe. Constrained stochastic extended redundancy analysis. *Psychometrika*, 80(2):516–534, 2015.

J. R. Edwards and R. P. Bagozzi. On the nature and direction of relationships

between constructs and measures. *Psychological methods*, 5(2):155, 2000.

B. Efron. *The jackknife, the bootstrap and other resampling plans*. Society for industrial and applied mathematics Philadelphia, 1982.

V. Esposito Vinzi, Trinchera L., S. Squillacciotti, and M. Tenenhaus. Rebus-pls: A response-based procedure for detecting unit segments in pls path modelling. *Applied Stochastic Models in Business and Industry*, 24(5):439–458, 2008.

M. Fattore, M. Pelagatti, and G. Vittadini. Inconsistencies of the pls-pm approach to structural equation models with formative-reflective schemes. *Electronic Journal of Applied Statistical Analysis: EJASA*, 5(3):333–338, 2012.

R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 1963.

C. Fornell and F. L. Bookstein. Two structural equation models: Lisrel and pls applied to consumer exit-voice theory. *Journal of Marketing Research*, 19(4): 440–452, 1982.

C. Fornell and J. Cha. Partial least squares. *Advanced methods of marketing research*, 407(3):52–78, 1994.

M. Gallo. Discriminant partial least squares analysis on compositional data. *Statistical Modelling*, 10(1):41–56, 2010.

P. H. Garthwaite. An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425):122–127, 1994.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

M. T. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

M. Hanafi. Pls path modelling: computation of latent variables with the estimation mode b. *Computational Statistics*, 22(2):275–292, 2007.

J. Henseler. Why generalized structured component analysis is not universally preferable to structural equation modeling. *Journal of the Academy of Marketing Science*, 40(3):402–413, 2012.

J. Henseler and G. Fassott. Testing moderating effects in pls path models: An illustration of available procedures. In *Handbook of partial least squares*, pages 713–735. Springer, 2010.

A. Herrmann, C. H. Hahn, M. D. Johnson, and F. Huber. Capturing customer heterogeneity using a finite mixture pls approach. *Schmalenbach Business Review (SBR)*, 54, 2002.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

S. Hsu, W. Chen, and M. Hsieh. Robustness testing of pls, lisrel, eqs and ann-based sem for measuring customer satisfaction. *Total Quality Management & Business Excellence*, 17(3):355–372, 2006.

H. Hwang and Y. Takane. Structural equation modeling by extended redundancy analysis. In *Measurement and multivariate analysis*, pages 115–124. Springer, 2002.

H. Hwang and Y. Takane. Generalized structured component analysis. *Psychometrika*, 69(1):81–99, 2004.

H. Hwang, H. W. Suk, Y. Takane, J. Lee, and J. Lim. Generalized functional extended redundancy analysis. *Psychometrika*, 80(1):101–125, 2015.

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

A. Z. Israels. Redundancy analysis for qualitative variables. *Psychometrika*, 49(3): 331–346, 1984.

A. J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248 – 264, 1975.

E. Jakobowicz and C. Derquenne. A modified pls path modeling algorithm handling reflective categorical variables and a new model building strategy. *Computational Statistics & Data Analysis*, 51(8):3666–3678, 2007.

K. G. J"oreskog. A general method for estimating a linear structural equation system. *ETS Research Bulletin Series*, 1970(2):i–41, 1970.

K. G. J"oreskog. Structural analysis of covariance and correlation matrices. *Psychometrika*, 43(4):443–477, 1978.

K. G. J"oreskog. Analysis of covariance structures. In *Handbook of multivariate experimental psychology*, pages 207–230. Springer, 1988.

K. G. J"oreskog and D. S"orbom. Recent developments in structural equation modeling. *Journal of marketing research*, pages 404–416, 1982.

H. A. L. Kiers and J. M. F. ten Berge. Alternating least squares algorithms for simultaneous components analysis with equal component weight matrices in two or more populations. *Psychometrika*, 54(3):467–473, 1989.

M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

J. Lin. Does optimal scaling of ordinal variables by linearization improve estimation of the underlying continuous correlation? Master's thesis, University of California, Los Angeles, 2009.

J. Lohm"oller. *Latent variable path modeling with partial least squares*. Physica-Verlag Heidelberg, 1989.

U. Lorenzo-Seva and J. M. F. Ten Berge. Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology*, 2(2):57–64, 2006.

P. G. Lovaglio and G. Vacca. %era: A sas macro for extended redundancy analysis. *Journal of Statistical Software*, 74(1):1–19, 2016a.

P. G. Lovaglio and G. Vacca. %gra: an sas macro for generalized redundancy analysis. *Journal of Statistical Computation and Simulation*, 0(0):1–13, 2016b.

P. G. Lovaglio and G. Vittadini. Structural equation models in a redundancy analysis framework with covariates. *Multivariate Behavioral Research*, 49(5):486–501, 2014.

E. C. Malthouse, A. C. Tamhane, and R. S. H. Mah. Nonlinear partial least squares. *Computers & Chemical Engineering*, 21(8):875–890, 1997.

D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

MATLAB. *version 9.0.0 (R2016a)*. The MathWorks Inc., 2016.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

A. L. McCutcheon. *Latent class analysis*. Sage, 1987.

R. P. McDonald. Path analysis with composite variables. *Multivariate Behavioral Research*, 31(2):239–270, 1996.

W. Meredith and R. E. Millsap. On component analyses. *Psychometrika*, 50(4): 495–507, 1985.

I. Moustaki. A latent variable model for ordinal variables. *Applied psychological measurement*, 24(3):211–223, 2000.

I. Moustaki. A general class of latent variable models for ordinal manifest variables with covariate effects on the manifest and latent variables. *British Journal of Mathematical and Statistical Psychology*, 56(2):337–357, 2003.

B. Muthén. A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika*, 49(1):115–132, 1984.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2016. URL `https://www.R-project.org/`.

F. S. G. Richards. A method of maximum-likelihood estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 469–475, 1961.

A. K. Rigler, J. M. Irvine, and T. P. Vogl. Rescaling of variables in back propagation learning. *Neural Networks*, 4(2):225–229, 1991.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

G. Russolillo. Non-metric partial least squares. *Electronic Journal of Statistics*, 6: 1641–1669, 2012.

P. H. Sch"onemann. The minimum average correlation between equivalent sets of

uncorrelated factors. *Psychometrika*, 36(1):21–30, 1971.

F. Schuberth, J. Henseler, and T. K. Dijkstra. Partial least squares path modeling using ordinal categorical indicators. *Quality & Quantity*, pages 1–27, 2016.

P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

J. H. Steiger and P. H. Sch"onemann. A history of factor indeterminacy. *Unpublished*, 1975.

Y. Takane and H. Hwang. An extended redundancy analysis and its applications to two practical examples. *Comput. Stat. Data Anal.*, 49(3):785–808, 2005.

A. Tenenhaus and M. Tenenhaus. Regularized generalized canonical correlation analysis. *Psychometrika*, 76(2):257, 2011.

M. Tenenhaus, V. Esposito Vinzi, Chatelin Y., and N. C. Lauro. Pls path modeling. *Computational Statistics & Data Analysis*, 48(1):159 – 205, 2005.

L. Trinchera. *Unobserved Heterogeneity in Structural Equation Models: a new approach to latent class detection in PLS Path Modeling*. PhD thesis, Università degli Studi di Napoli Federico II, 2008.

L. Trinchera, G. Russolillo, and N. C. Lauro. Using categorical variables in pls path modeling to build system of composite indicators. *Statistica Applicata*, 20(3-4): 309–330, 2008.

L. R. Tucker. A method for synthesis of factor analysis studies, 1951.

A. L. Van den Wollenberg. Redundancy analysis an alternative for canonical correlation analysis. *Psychometrika*, 42(2):207–219, 1977.

R. van der Leeden. *Reduced rank regression with structured residuals*. DSWO Press,

1990.

G. Vittadini. Indeterminacy problems in the lisrel model. *Multivariate Behavioral Research*, 24(4):397–414, 1989.

G. Vittadini, S. Minotti, M. Fattore, and P. G. Lovaglio. On the relationships among latent variables and residuals in pls path modeling: The formative-reflective scheme. *Computational Statistics & Data Analysis*, 51(12):5828–5846, 2007.

T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. Accelerating the convergence of the back-propagation method. *Biological cybernetics*, 59(4-5): 257–263, 1988.

P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

B. Widrow, M. E. Hoff, et al. Adaptive switching circuits. In *IRE WESCON convention record*, volume 4, pages 96–104, 1960.

H. Wold. *Estimation of Principal Components and Related Models by Iterative Least squares*, pages 391–420. Academic Press, 1966.

H. Wold. Soft modeling by latent variables; the nonlinear iterative partial least squares approach. *Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett*, 1975.

H. Wold. Soft modeling : the basic design and some extensions. *Systems under indirect observation : causality, structure, prediction*, 2:1–54, 1982.

S. Wold, H. Martens, and H. Wold. *The multivariate calibration problem in chemistry solved by the PLS method*, pages 286–293. Springer Berlin Heidelberg, 1983.

S. Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557–585, 1921.

S. Wright. The method of path coefficients. *The Annals of Mathematical Statistics*, 5(3):161–215, 1934.

T. W. Yee. *Vector Generalized Linear and Additive Models With an Implementation in R*. Springer-Verlag New York, 2015.

T. W. Yee and T. J. Hastie. Reduced-rank vector generalized linear models. *Statistical Modelling*, 3(1):15–41, 2003.

F. W. Young. Quantitative analysis of qualitative data. *Psychometrika*, 46(4): 357–388, 1981.