



UvA-DARE (Digital Academic Repository)

Network psychometrics

Epskamp, S.

[Link to publication](#)

Citation for published version (APA):
Epskamp, S. (2017). Network psychometrics.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Generalized Network Psychometrics

Abstract

We introduce the network model as a formal psychometric model, conceptualizing the covariance between psychometric indicators as resulting from pairwise interactions between observable variables in a network structure. This contrasts with standard psychometric models, in which the covariance between test items arises from the influence of one or more common latent variables. Here, we present two generalizations of the network model that encompass latent variable structures, establishing network modeling as parts of the more general framework of Structural Equation Modeling (SEM). In the first generalization, we model the covariance structure of latent variables as a network. We term this framework *Latent Network Modeling* (LNM) and show that, with LNM, a unique structure of conditional independence relationships between latent variables can be obtained in an explorative manner. In the second generalization, the residual variance-covariance structure of indicators is modeled as a network. We term this generalization *Residual Network Modeling* (RNM) and show that, within this framework, identifiable models can be obtained in which local independence is structurally violated. These generalizations allow for a general modeling framework that can be used to fit, and compare, SEM models, network models, and the RNM and LNM generalizations. This methodology has been implemented in the free-to-use software package *lwnet*, which contains confirmatory model testing as well as two exploratory search algorithms: stepwise search algorithms for low-dimensional datasets and penalized maximum likelihood estimation for larger datasets. We show in simulation studies that these search algorithms performs adequately in identifying the structure of the relevant residual or latent networks. We further demonstrate the utility of these generalizations in an empirical example on a personality inventory dataset.

This chapter has been adapted from: Epskamp, S., Rhemtulla, M.T., and Borsboom, D. (in press). Generalized Network Psychometrics: Combining Network and Latent Variable Models. *Psychometrika*.

7.1 Introduction

Recent years have seen an emergence of network modeling in psychometrics (Borsboom, 2008; Schmittmann et al., 2013), with applications in clinical psychology (e.g., van Borkulo et al., 2015; McNally et al., 2015; Fried et al., 2015), psychiatry (e.g., Isvoranu, van Borkulo, et al., 2016; Isvoranu, Borsboom, et al., 2016), health sciences (e.g., Kossakowski et al., 2016), social psychology (e.g., Dalege et al., 2016; Cramer, Sluis, et al., 2012), and other fields (see for a review of recent literature Fried & van Borkulo, 2016). This line of literature stems from the network perspective of psychology, which conceptualizes psychological behavior as complex systems in which observed variables interact with one-another (Cramer et al., 2010). As described in previous chapters of this dissertation, network models are used to gain insight into this potentially high-dimensional interplay. In practice, network models can be used as a sparse representation of the joint distribution of observed indicators, and as such these models show great promise in psychometrics by providing a perspective that complements latent variable modeling. Network modeling highlights variance that is unique to pairs of variables, whereas latent variable modeling focuses on variance that is shared across all variables (Costantini, Epskamp, et al., 2015). As a result, network modeling and latent variable modeling can complement—rather than exclude—one-another.

In this chapter, we introduce the reader to this field of *network psychometrics* (Epskamp et al., in press) and formalize the network model for multivariate normal data, the Gaussian Graphical Model (GGM; Lauritzen, 1996), as a formal psychometric model. We contrast the GGM to the Structural Equation Model (SEM; Wright, 1921; Kaplan, 2000) and show that the GGM can be seen as another way to approach modeling covariance structures as is typically done in psychometrics. In particular, rather than modeling the covariance matrix, the GGM models the *inverse* of a covariance matrix. The GGM and SEM are thus very closely related: every GGM model and every SEM model imply a constrained covariance structure. We make use of this relationship to show that, through a reparameterization of the SEM model, the GGM model can be obtained in two different ways: first, as a network structure that relates a number of latent variables to each other, and second, as a network between residuals that remain given a fitted latent variable model. As such, the GGM can be modeled and estimated in SEM, which allows for network modeling of psychometric data to be carried out in a framework familiar to psychometricians and methodologists. In addition, this allows for one to assess the fit of a GGM, compare GGMs to one-another and compare a GGM to a SEM model.

However, the combination of GGM and SEM allows for more than fitting network models. As we will show, the strength of one framework can help overcome shortcomings of the other framework. In particular, SEM falls short in that exploratory estimation is complicated and there is a strong reliance on local independence, whereas the GGM falls short in that it assumes no latent variables. In this chapter, we introduce network models for latent covariances and for residual covariances as two distinct generalized frameworks of both the SEM and GGM. The first framework, Latent Network Modeling (LNM), formulates a network among latent variables. This framework allows researchers to exploratively estimate con-

ditional independence relationships between latent variables through model search algorithms; this estimation is difficult in the SEM framework due to the presence of equivalent models (MacCallum et al., 1993). The second framework, which we denote Residual Network Modeling (RNM), formulates a network structure on the residuals of a SEM model. With this framework, researchers can circumvent critical assumptions of both SEM and the GGM: SEM typically relies on the assumption of local independence, whereas network modeling typically relies on the assumption that the covariance structure among a set of the items is not due to latent variables at all. The RNM framework allows researchers to estimate SEM models without the assumption of local independence (all residuals can be correlated, albeit due to a constrained structure on the inverse residual covariance matrix) as well as to estimate a network structure, while taking into account the fact that the covariance between items may be partly due to latent factors.

While the powerful combination of SEM and GGM allows for confirmative testing of network structures both with and without latent variables, we recognize that few researchers have yet formulated strict confirmatory hypotheses in the relatively new field of network psychometrics. Often, researchers are more interested in exploratively searching a plausible network structure. To this end, we present two exploratory search algorithms. The first is a step-wise model search algorithm that adds and removes edges of a network as long as fit is improved, and the second uses penalized maximum likelihood estimation (Tibshirani, 1996) to estimate a sparse model. We evaluate the performance of these search methods in four simulation studies. Finally, the proposed methods have been implemented in a free-to-use R package, `lvnet`, which we illustrate in an empirical example on personality inventory items (Revelle, 2010).

7.2 Modeling Multivariate Gaussian Data

Let \mathbf{y} be the response vector of a random subject on P items¹. We assume \mathbf{y} is centered and follows a multivariate Gaussian density:

$$\mathbf{y} \sim N_P(\mathbf{0}, \boldsymbol{\Sigma}),$$

In which $\boldsymbol{\Sigma}$ is a $P \times P$ variance-covariance matrix, estimated by some model-implied $\hat{\boldsymbol{\Sigma}}$. Estimating $\hat{\boldsymbol{\Sigma}}$ is often done through some form of *maximum likelihood estimation*. If we measure N independent samples of \mathbf{y} we can formulate the $N \times P$ matrix \mathbf{Y} containing realization \mathbf{y}_i^\top as its i th row. Let \mathbf{S} represent the sample variance-covariance matrix of \mathbf{Y} :

$$\mathbf{S} = \frac{1}{N-1} \mathbf{Y}^\top \mathbf{Y}.$$

¹Throughout this chapter, vectors will be represented with lowercase boldfaced letters and matrices will be denoted by capital boldfaced letters. Roman letters will be used to denote observed variables and parameters (such as the number of nodes) and Greek letters will be used to denote latent variables and parameters that need to be estimated. The subscript i will be used to denote the realized response vector of subject i and omission of this subscript will be used to denote the response of a random subject.

In maximum likelihood estimation, we use \mathbf{S} to compute and minimize -2 times the log-likelihood function to find $\hat{\Sigma}$ (Lawley, 1940; Jöreskog, 1967; Jacobucci, Grimm, & McArdle, 2016):

$$\min_{\hat{\Sigma}} \left[\log \det \left(\hat{\Sigma} \right) + \text{Trace} \left(\mathbf{S} \hat{\Sigma}^{-1} \right) - \log \det \left(\hat{\mathbf{S}} \right) - P \right]. \quad (7.1)$$

To optimize this expression, $\hat{\Sigma}$ should be estimated as closely as possible to \mathbf{S} and perfect fit is obtained if $\hat{\Sigma} = \mathbf{S}$. A properly identified model with the same number of parameters (K) used to form $\hat{\Sigma}$ as there are unique elements in \mathbf{S} ($P(P+1)/2$ parameters) will lead to $\hat{\Sigma} = \mathbf{S}$ and therefore a saturated model. The goal of modeling multivariate Gaussian data is to obtain some model for $\hat{\Sigma}$ with *positive degrees of freedom*, $K < P(P+1)/2$, in which $\hat{\Sigma}$ resembles \mathbf{S} closely.

Structural Equation Modeling

In Confirmatory Factor Analysis (CFA), \mathbf{Y} is typically assumed to be a causal linear effect of a set of M centered latent variables, $\boldsymbol{\eta}$, and independent residuals or error, $\boldsymbol{\varepsilon}$:

$$\mathbf{y} = \mathbf{\Lambda} \boldsymbol{\eta} + \boldsymbol{\varepsilon}.$$

Here, $\mathbf{\Lambda}$ represents a $P \times M$ matrix of *factor loadings*. This model implies the following model for $\hat{\Sigma}$:

$$\hat{\Sigma} = \mathbf{\Lambda} \boldsymbol{\Psi} \mathbf{\Lambda}^{\top} + \boldsymbol{\Theta}, \quad (7.2)$$

in which $\boldsymbol{\Psi} = \text{Var}(\boldsymbol{\eta})$ and $\boldsymbol{\Theta} = \text{Var}(\boldsymbol{\varepsilon})$. In Structural Equation Modeling (SEM), $\text{Var}(\boldsymbol{\eta})$ can further be modeled by adding structural linear relations between the latent variables²:

$$\boldsymbol{\eta} = \mathbf{B} \boldsymbol{\eta} + \boldsymbol{\zeta},$$

in which $\boldsymbol{\zeta}$ is a vector of residuals and \mathbf{B} is an $M \times M$ matrix of regression coefficients. Now, $\hat{\Sigma}$ can be more extensively modeled as:

$$\hat{\Sigma} = \mathbf{\Lambda} (\mathbf{I} - \mathbf{B})^{-1} \boldsymbol{\Psi} (\mathbf{I} - \mathbf{B})^{-1\top} \mathbf{\Lambda}^{\top} + \boldsymbol{\Theta}, \quad (7.3)$$

in which now $\boldsymbol{\Psi} = \text{Var}(\boldsymbol{\zeta})$. This framework can be used to model direct causal effects between observed variables by setting $\mathbf{\Lambda} = \mathbf{I}$ and $\boldsymbol{\Theta} = \mathbf{O}$, which is often called path analysis (Wright, 1934).

The $\boldsymbol{\Theta}$ matrix is, like $\hat{\Sigma}$ and \mathbf{S} , a $P \times P$ matrix; if $\boldsymbol{\Theta}$ is fully estimated—contains no restricted elements—then $\boldsymbol{\Theta}$ alone constitutes a saturated model. Therefore, to make either (7.2) or (7.3) identifiable, $\boldsymbol{\Theta}$ must be strongly restricted. Typically, $\boldsymbol{\Theta}$ is set to be diagonal, a restriction often termed *local independence* (Lord, Novick, & Birnbaum, 1968; Holland & Rosenbaum, 1986) because indicators are independent of each other after conditioning on the set of latent variables. To improve fit, select off-diagonal elements of $\boldsymbol{\Theta}$ can be estimated, but systematic violations of local independence—many nonzero elements in $\boldsymbol{\Theta}$ —are not possible as that will

²We make use here of the convenient all- y notation and do not distinguish between exogenous and endogenous latent variables (Hayduk, 1987).

quickly make (7.2) and (7.3) saturated or even over-identified. More precisely, Θ can *not* be fully-populated—some elements of Θ must be set to equal zero—when latent variables are used. An element of Θ being fixed to zero indicates that two variables are *locally independent* after conditioning on the set of latent variables. As such, local independence is a *critical assumption* in both CFA and SEM; if local independence is systematically violated, CFA and SEM will never result in correct models.

The assumption of local independence has led to critiques of the factor model and its usage in psychology; local independence appears to be frequently violated due to direct causal effects, semantic overlap, or reciprocal interactions between putative indicators of a latent variable (Borsboom, 2008; Cramer et al., 2010; Borsboom et al., 2011; Cramer, Sluis, et al., 2012; Schmittmann et al., 2013). In psychopathology research, local independence of symptoms given a person’s level of a latent mental disorder has been questioned (Borsboom & Cramer, 2013). For example, three problems associated with depression are “fatigue”, “concentration problems” and “rumination”. It is plausible that a person who suffers from fatigue will also concentrate more poorly, as a direct result of being fatigued and regardless of his or her level of depression. Similarly, rumination might lead to poor concentration. In another example, Kossakowski et al. (2016) describe the often-used SF-36 questionnaire (Ware Jr & Sherbourne, 1992) designed to measure health related quality of life. The SF-36 contains items such as “can you walk for more than one kilometer” and “can you walk a few hundred meters”. Clearly, these items can never be locally independent after conditioning on any latent trait, as one item (the ability to walk a few hundred meters) is a *prerequisite* for the other (walking more than a kilometer). In typical applications, the excessive covariance between items of this type is typically left unmodeled, and treated instead by combining items into a subscale or total score that is subsequently subjected to factor analysis; of course, however, this is tantamount to ignoring the relevant psychometric problem rather than solving it.

Given the many theoretically expected violations of local independence in psychometric applications, many elements of Θ in both (7.2) and (7.3) should ordinarily be freely estimated. Especially when violations of local independence are expected to be due to causal effects of partial overlap, residual correlations should not be constrained to zero; in addition, a chain of causal relationships between indicators can lead to all residuals to become correlated. Thus, even when latent factors cause much of the covariation between measured items, fitting a latent variable model that involves local independence may not fully account for correlation structure between measured items. Of course, in practice, many psychometricians are aware of this problem, which is typically addressed by freeing up correlations between residuals to improve model fit. However, this is usually done in an ad-hoc fashion, on the basis of inspection of modification indices and freeing up error covariances one by one, which is post hoc, suboptimal, and involves an uncontrolled journey through the model space. As a result, it is often difficult to impossible to tell how exactly authors arrived at their final reported models. As we will show later in this chapter, this process can be optimized and systematized using network models to connect residuals on top of a latent variable structure.

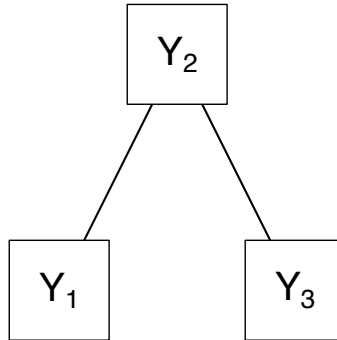


Figure 7.1: Example of a pairwise Markov Random Field model. Edges in this model indicate pairwise interactions, and are drawn using undirected edges to distinguish from (bidirectional) covariances. Rather than a model for marginal associations (such as a network indicating covariances), this is a model for *conditional* associations. The network above encodes that Y_1 and Y_3 are independent after conditioning on Y_2 . Such a model allows all three variables to correlate while retaining one degree of freedom (the model only has two parameters)

Network Modeling

Recent authors have suggested that the potential presence of causal relationships between measured variables may allow the explanation of the covariance structure without the need to invoke any latent variables (Borsboom, 2008; Cramer et al., 2010; Borsboom et al., 2011; Schmittmann et al., 2013). The interactions between indicators can instead be modeled as a *network*, in which indicators are represented as nodes that are connected by edges representing *pairwise interactions*. Such interactions indicate the presence of covariances that cannot be explained by any other variable in the model and can represent—possibly reciprocal—causal relationships. Estimating a network structure on psychometric data is termed *network psychometrics* (Epskamp et al., in press). Such a network of interacting components can generate data that fit factor models well, as is commonly the case in psychology. Van Der Maas et al. (2006) showed that the positive manifold of intelligence—which is commonly explained with the general factor for intelligence, g —can emerge from a network of mutually benefiting cognitive abilities. Borsboom et al. (2011) showed that a network of psychopathological symptoms, in which disorders are modeled as clusters of symptoms, could explain comorbidity between disorders. Furthermore, Epskamp et al. (in press) showed that the Ising model for ferromagnetism (Ising, 1925), which models magnetism as a network of particles, is equivalent to multidimensional item response theory (Reckase, 2009; see Chapter 8).

In network psychometrics, psychometric data are modeled through directed or undirected networks. Directed networks are equivalent to path analysis models. For modeling undirected networks, pairwise Markov Random Fields (Lauritzen, 1996; Murphy, 2012) are used. In these models, each variable is represented by a node, and nodes are connected by a set of edges. If two nodes, y_j and y_k , are not connected by an edge, then this means they are independent after conditioning on the set of all other nodes, $\mathbf{y}^{-(j,k)}$. Whenever two nodes cannot be rendered independent conditional on the other nodes in the system, they are said to feature in a *pairwise interaction*, which is represented by an *undirected* edge—an edge with no arrows—to contrast such an effect from covariances typically represented in the SEM literature with bidirectional edges. Figure 7.1 represents such a network model, in which nodes y_1 and y_3 are independent after conditioning on node y_2 . Such a model can readily arise from direct interactions between the nodes. For example, this conditional independence structure would emerge if y_2 is a common cause of y_1 and y_3 , or if y_2 is the mediator in a causal path between y_1 and y_3 . In general, it is important to note that pairwise interactions are not mere correlations; two variables may be strongly correlated but unconnected (e.g., when both are caused by another variable in the system) and they may be uncorrelated but strongly connected in the network (e.g., when they have a common effect in the system). For instance, in the present example the model does *not* indicate that y_1 and y_3 are uncorrelated, but merely indicates that any correlation between y_1 and y_3 is due to their mutual interaction with y_2 ; a network model in which either directly or indirectly connected paths exist between all pairs of nodes typically implies a fully populated (no zero elements) variance–covariance matrix.

In the case of multivariate Gaussian data this model is termed the Gaussian Graphical Model (GGM; Lauritzen, 1996). In the case of multivariate normality, the *partial correlation coefficient* is sufficient to test the degree of conditional independence of two variables after conditioning on all other variables; if the partial correlation coefficient is zero, there is conditional independence and hence no edge in the network. As such, partial correlation coefficients can directly be used in the network as *edge weights*; the strength of connection between two nodes³. Such a network is typically encoded in a symmetrical and real valued $p \times p$ *weight matrix*, $\mathbf{\Omega}$, in which element ω_{jk} represents the edge weight between node j and node k :

$$\text{Cor} \left(y_j, y_k \mid \mathbf{y}^{-(j,k)} \right) = \omega_{jk} = \omega_{kj}.$$

The partial correlation coefficients can be directly obtained from the inverse of variance–covariance matrix $\hat{\mathbf{\Sigma}}$, also termed the *precision matrix* $\hat{\mathbf{K}}$ (Lauritzen, 1996):

$$\text{Cor} \left(y_j, y_k \mid \mathbf{y}^{-(j,k)} \right) = -\frac{\kappa_{jk}}{\sqrt{\kappa_{kk}}\sqrt{\kappa_{jj}}}.$$

Thus, element κ_{jk} of the precision matrix is proportional to the partial correlation coefficient of variables y_j and y_k after conditioning on all other variables. Since this process simply involves standardizing the precision matrix, we propose

³A saturated GGM is also called a partial correlation network because it contains the sample partial correlation coefficients as edge weights.

the following model⁴:

$$\hat{\Sigma} = \hat{K}^{-1} = \Delta (\mathbf{I} - \Omega)^{-1} \Delta, \quad (7.4)$$

in which Δ is a diagonal matrix with $\delta_{jj} = \kappa_{jj}^{-\frac{1}{2}}$ and Ω has zeroes on the diagonal. This model allows for confirmative testing of the GGM structures on psychometric data. Furthermore, the model can be compared to a saturated model (fully populated off-diagonal values of Ω) and the independence model ($\Omega = \mathbf{O}$), allowing one to obtain χ^2 fit statistics as well as fit indices such as the RMSEA (Browne & Cudeck, 1992) and CFI (Bentler, 1990). Such methods of assessing model fit have not yet been used in network psychometrics.

Similar to CFA and SEM, the GGM relies on a critical assumption; namely, that covariances between observed variables are *not* caused by any latent or unobserved variable. If we estimate a GGM in a case where, in fact, a latent factor model was the true data generating structure, then generally we would expect the GGM to be saturated—i.e., there would be no missing edges in the GGM (Chandrasekaran, Parrilo, & Willsky, 2012). A missing edge in the GGM indicates the presence of conditional independence between two indicators given all other indicators; we do not expect indicators to become independent given subsets of other indicators (see also Ellis & Junker, 1997; Holland & Rosenbaum, 1986). Again, this critical assumption might not be plausible. While variables such as “Am indifferent to the feelings of others” and “Inquire about others’ well-being” quite probably interact with each other, it might be far-fetched to assume that no unobserved variable, such as a personality trait, in part also causes some of the variance in responses on these items.

7.3 Generalizing Factor Analysis and Network Modeling

We propose two generalizations of both SEM and the GGM that both allow the modeling of network structures in SEM. In the first generalization, we adopt the CFA⁵ decomposition in (7.2) and model the variance–covariance matrix of latent variables as a GGM:

$$\Psi = \Delta_{\Psi} (\mathbf{I} - \Omega_{\Psi})^{-1} \Delta_{\Psi}.$$

This framework can be seen as modeling conditional independencies between latent variables not by directed effects (as in SEM) but as an undirected network. As such, we term this framework *latent network modeling* (LNM).

In the second generalization, we adopt the SEM decomposition of the variance–covariance matrix in (7.3) and allow the residual variance–covariance matrix Θ to be modeled as a GGM:

$$\Theta = \Delta_{\Theta} (\mathbf{I} - \Omega_{\Theta})^{-1} \Delta_{\Theta}.$$

⁴To our knowledge, the GGM has not yet been framed in this form. We chose this form because it allows for clear modeling and interpretation of the network parameters.

⁵We use the CFA framework instead of the SEM framework here as the main application of this framework is in exploratively estimating relationships between latent variables.

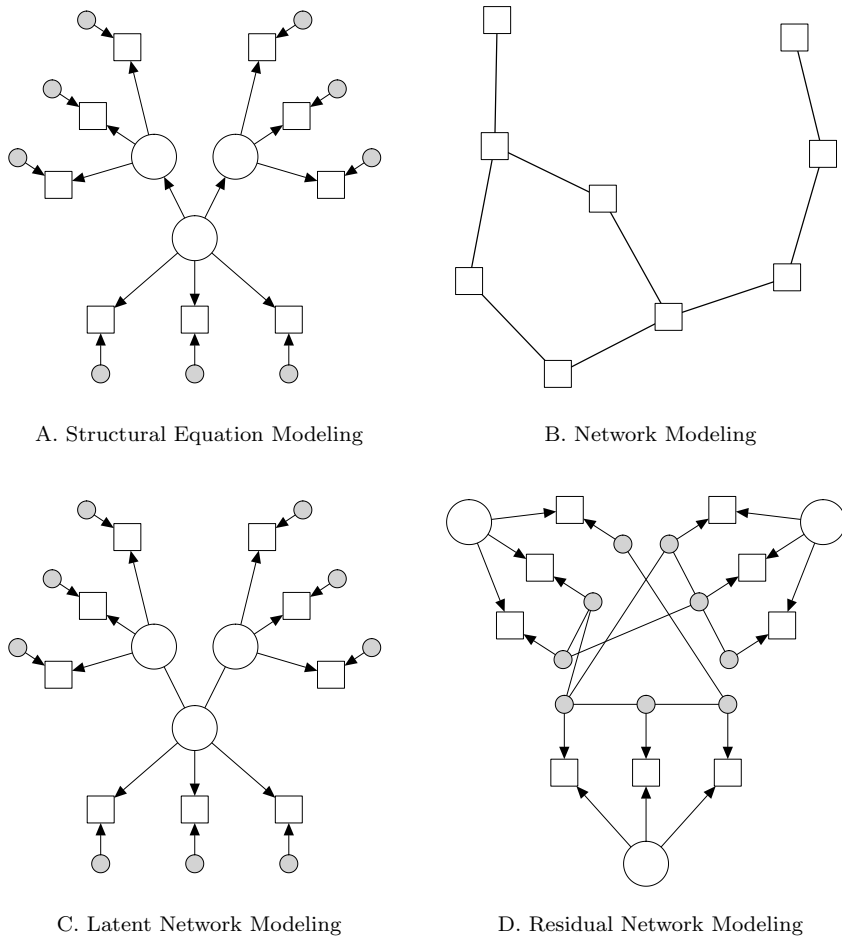


Figure 7.2: Examples of possible models under four different modeling frameworks. Circular nodes indicate latent variables, square nodes indicate manifest variables and gray nodes indicate residuals. Directed edges indicate factor loadings or regression parameters and undirected edges indicate pairwise interactions. Note that such undirected edges do *not* indicate covariances, which are typically denoted with bidirectional edges. Replacing covariances with interactions is where the network models differ from typical SEM.

Because this framework conceptualizes associations between residuals as pairwise *interactions*, rather than correlations, we term this framework *Residual Network Modeling* (RNM). Using this framework allows—as will be described below—for a powerful way of fitting a confirmatory factor structure even though local independence is systematically violated and all residuals are correlated.

Figure 7.2 shows four different examples of possible models that are attainable

under the SEM, LNM and RNM frameworks. Panel A shows a typical SEM model in which one latent variable functions as a common cause of two others. Panel B shows a network model which can be estimated using both the RNM and the LNM frameworks. Panel C shows a completely equivalent LNM model to the SEM model of Panel A in which the direction of effect between latent variables is not modeled. Finally, panel D shows a model in which three exogenous latent variables underlie a set of indicators of which the residuals form a network. The remainder of this section will describe RNM and LNM in more detail and will outline the class of situations in which using these models is advantageous over CFA or SEM.

Latent Network Modeling

The LNM framework models the latent variance–covariance matrix of a CFA model as a GGM:

$$\hat{\Sigma} = \Lambda \Delta_{\Psi} (\mathbf{I} - \Omega_{\Psi})^{-1} \Delta_{\Psi} \Lambda^{\top} + \Theta. \quad (7.5)$$

This allows researchers to model conditional independence relationships between latent variables without making the implicit assumptions of directionality or acyclicity. In SEM, \mathbf{B} is typically modeled as a directed acyclic graph (DAG), meaning that elements of \mathbf{B} can be represented by directed edges, and following along the path of these edges it is not possible to return to any node (latent variable). The edges in such a DAG can be interpreted as causal, and in general they imply a specific set of conditional independence relationships between the nodes (Pearl, 2000).

While modeling conditional independence relationships between latent variables as a DAG is a powerful tool for testing strictly confirmatory hypotheses, it is less useful for more exploratory estimation. Though there have been recent advances in exploratory estimation of DAGs within an SEM framework (e.g., Gates & Molenaar, 2012; Rosa, Friston, & Penny, 2012), many equivalent DAGs can imply the same conditional independence relationships, and thus fit the data equally well even though their causal interpretation can be strikingly different (MacCallum et al., 1993). Furthermore, the assumption that the generating model is acyclic—which, in practice, often is made on purely pragmatic grounds to identify a model—is problematic in that much psychological behavior can be assumed to have at least some cyclic and complex behavior and feedback (Schmittmann et al., 2013). Thus, the true conditional independence relationships in a dataset can lead to many equivalent compositions of \mathbf{B} , and possibly none of them are the true model.

In psychometrics and SEM the GGM representation has not been very prominent, even though it has some manifest benefits over the attempt to identify DAGs directly. For example, by modeling conditional independence relationships between latent variables as a GGM, many relationships can be modeled in a simpler way as compared to a DAG. In addition, in the GGM each set of conditional independence relations only corresponds to one model: there are no equivalent GGMs with the same nodes. Figure 7.3 shows a comparison of several conditional independence relations that can be modeled equivalently or not by using a GGM

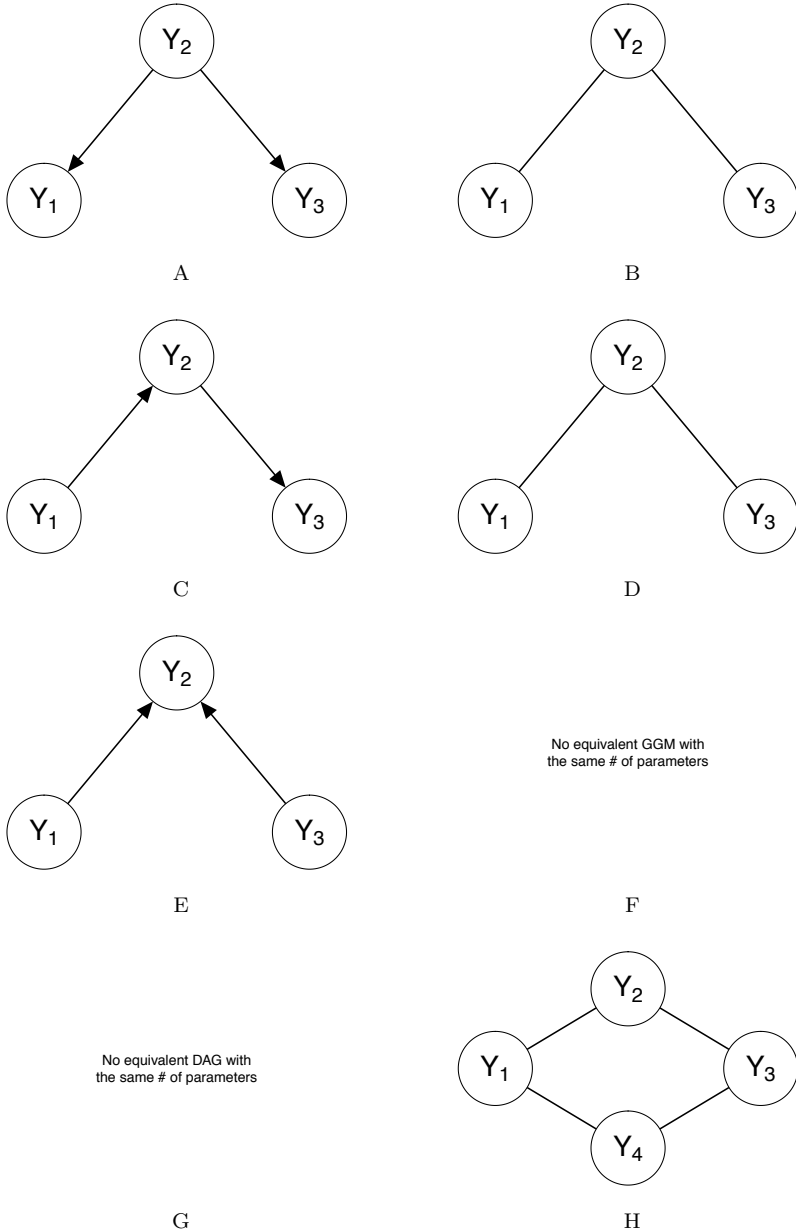


Figure 7.3: Equivalent models between directed acyclic graphs (DAG; left) and Gaussian graphical models (GGM; right). Each row of graphs show two models that are equivalent.

or by using a DAG. Panel A and Panel C show two DAGs that represent the same conditional independence relations, $y_1 \perp\!\!\!\perp y_3 \mid y_2$, which can both be represented by the same GGM shown in Panel B and Panel D. There are some conditional independence relations that a GGM cannot represent in the same number of parameters as a DAG; Panel E shows a collider structure that cannot be exactly represented by a GGM (the best fitting GGM would feature three edges instead of two). On the other hand, there are also conditional independence relationships that a GGM can represent and a DAG cannot; the cycle of Panel H cannot be represented by a DAG. Further equivalences and differences between GGMs and DAGs are beyond the scope of this chapter, but have been well described in the literature (e.g., chapter 3 of Lauritzen, 1996; Koller & Friedman, 2009; Kolaczyk, 2009). In sum, the GGM offers a natural middle ground between zero-order correlations and DAGs: every set of zero-order correlations implies exactly one GGM, and every DAG implies exactly one GGM. In a sense, the road from correlations to DAGs (including hierarchical factor models) thus always must pass through the realm of GGMs, which acts as a bridge between the correlational and causal worlds.

Because there are no equivalent undirected models possible, LNM offers a powerful tool for exploratory estimation of relationships between latent variables. For example, suppose one encounters data generated by the SEM model in Figure 7.2, Panel A. Without prior theory on the relations between latent variables, exploratory estimation on this dataset would lead to three completely equivalent models: the one shown in Figure 7.2, Panel C and two models in which the common cause instead is the middle node in a causal chain. As the number of latent variables increases, the potential number of equivalent models that encode the same conditional independence relationships grows without bound. The LNM model in Panel C of Figure 7.2 portrays the same conditional independence relationship as the SEM model in Panel A of Figure 7.2, while having no equivalent model. Exploratory estimation could easily find this model, and portrays the retrieved relationship in a clear and unambiguous way.

A final benefit of using LNM models is that they allow network analysts to construct a network while taking measurement error into account. So far, networks have been constructed based on single indicators only and no attempt has been made to remediate measurement error. By forming a network on graspable small concepts measured by a few indicators, the LNM framework can be used to control for measurement error.

Residual Network Modeling

In the RNM framework the residual structure of SEM is modeled as a GGM:

$$\hat{\Sigma} = \mathbf{\Lambda} (\mathbf{I} - \mathbf{B})^{-1} \mathbf{\Psi} (\mathbf{I} - \mathbf{B})^{-1\top} \mathbf{\Lambda}^\top + \mathbf{\Delta}_\mathbf{e} (\mathbf{I} - \mathbf{\Omega}_\mathbf{e})^{-1} \mathbf{\Delta}_\mathbf{e}. \quad (7.6)$$

This modeling framework conceptualizes latent variable and network modeling as two sides of the same coin, and offers immediate benefits to both. In latent variable modeling, RNM allows for the estimation of a factor structure (possibly including structural relationships between the latent variables), while having no uncorrelated

errors and thus no local independence. The error-correlations, however, are still highly structured due to the residual network structure. This can be seen as a compromise between the ideas of network analysis and factor modeling; while we agree that local independence is plausibly violated in many psychometric tests, we think the assumption of no underlying latent traits and therefore a sparse GGM may often be too strict. For network modeling, RNM allows a researcher to estimate a sparse network structure while taking into account that some of the covariation between items was caused by a set of latent variables. Not taking this into account would lead to a saturated model (Chandrasekaran et al., 2012), whereas the residual network structure can be sparse.

To avoid confusion between residual correlations, we will denote edges in Ω_{Θ} *residual interactions*. Residual interactions can be understood as pairwise linear effects, possibly due to some causal influence or partial overlap between indicators that is left after controlling for the latent structure. Consider again the indicators for agreeableness “Am indifferent to the feelings of others” and “Inquire about others’ well-being”. It seems clear that we would not expect these indicators to be locally independent after conditioning on agreeableness; being indifferent to the feelings of others will cause one to not inquire about other’s well-being. Thus, we could expect these indicators to feature a residual interaction; some degree of correlation between these indicators is expected to remain, even after conditioning on the latent variable and all other indicators in the model.

The RNM framework in particular offers a new way of improving the fit of confirmatory factor models. In contrast to increasingly popular methods such as exploratory SEM (ESEM; Marsh, Morin, Parker, & Kaur, 2014) or LASSO regularized SEM models (Jacobucci et al., 2016), the RNM framework improves the fit by adding residual interactions rather than allowing for more cross-loadings. The factor structure is kept exactly intact as specified in the confirmatory model. Importantly, therefore, the interpretation of the latent factor does not change. This can be highly valuable in the presence of a strong theory on the latent variables structure underlying a dataset even in the presence of violations of local independence.

7.4 Exploratory Network Estimation

Both the LNM and RNM modeling frameworks allow for confirmative testing of network structures. Confirmatory estimation is straightforward and similar to estimating SEM models, with the exception that instead of modeling Ψ or Θ now the latent network Ω_{Ψ} or Ω_{Θ} is modeled. Furthermore, both modeling frameworks allow for the confirmatory fit of a network model. In LNM, a confirmatory network structure can be tested by setting $\Lambda = \mathbf{I}$ and $\Theta = \mathbf{O}$; in RNM, a confirmatory network model can be tested by omitting any latent variables. We have developed the R package `lvnet`⁶, which utilizes `OpenMx` (Neale et al., 2016) for confirmative testing of RNM and LNM models (as well as a combination of the two). The `lvnet` function can be used for this purpose by specifying the fixed and the free elements

⁶github.com/sachaepskamp/lvnet

of model matrices. The package returns model fit indices (e.g., the RMSEA, CFI and χ^2 value), parameter estimates, and allows for model comparison tests.

Often the network structure, either at the residual or the latent level, is unknown and needs to be estimated. To this end, the package includes two exploratory search algorithms described below: step-wise model search and penalized maximum likelihood estimation. For both model frameworks and both search algorithms, we present simulation studies to investigate the performance of these procedures. As is typical in simulation studies investigating the performance of network estimation techniques, we investigated the *sensitivity* and *specificity* (van Borkulo et al., 2014). These measures investigate the estimated edges versus the edges in the true model, with a ‘positive’ indicating an estimated edge and a ‘negative’ indicating an edge that is estimated to be zero. Sensitivity, also termed the true positive rate, gives the ratio of the number of true edges that were detected in the estimation versus the total number of edges in the true model:

$$\text{sensitivity} = \frac{\# \text{ true positives}}{\# \text{ true positives} + \# \text{ of false negatives}}$$

Specificity, also termed the true negative rate, gives the ratio of true missing edges detected in the estimation versus the total number of absent edges in the true model:

$$\text{specificity} = \frac{\# \text{ true negatives}}{\# \text{ true negatives} + \# \text{ false positives}}$$

The specificity can be seen as a function of the number of false positives: a high specificity indicates that there were not many edges detected to be nonzero that are zero in the true model. To favor degrees of freedom, model sparsity and interpretability, specificity should be high all-around—estimation techniques should not result in many false positives—whereas sensitivity should increase as a function of the sample size.

Simulating Gaussian Graphical models

In all simulation studies reported here, networks were constructed in the same way as done by Yin and Li (2011) in order to obtain a positive definite inverse-covariance matrix \mathbf{K} . First, a network structure was generated without weights. Next, weights were drawn randomly from a uniform distribution between 0.5 and 1, and made negative with 50% probability. The diagonal elements of \mathbf{K} were then set to 1.5 times the sum of all absolute values in the corresponding row, or 1 if this sum was zero. Next, all values in each row were divided by the diagonal value, ensuring that the diagonal values become 1. Finally, the matrix was made symmetric by averaging the lower and upper triangular elements. In the chain graphs used in the following simulations, this algorithm created networks in which the non-zero partial correlations had a mean of 0.33 and a standard deviation of 0.04.

Stepwise Model Search

In exploratory search, we are interested in recovering the network structure of either Ω_{Ψ} in LNM or Ω_{Θ} in RNM. This can be done through a step-wise model search, either based on χ^2 difference tests (Algorithm 1) or on minimization of some information criterion (Algorithm 2) such as the Akaike information criterion (AIC), Bayesian information criterion (BIC) or the extended Bayesian information criterion (EBIC; Chen & Chen, 2008) which is now often used in network estimation (van Borkulo et al., 2014; Foygel & Drton, 2010). In LNM, removing edges from Ω_{Ψ} cannot improve the fit beyond that of an already fitting CFA model. Hence, model search for Ω_{Ψ} should start at a fully populated initial setup for Ω_{Ψ} . In RNM, on the other hand, a densely populated Ω_{Θ} would lead to an over-identified model, and hence the step-wise model search should start at an empty network $\Omega_{\Theta} = \mathbf{O}$. The function `lvnetSearch` in the `lvnet` package can be used for both search algorithms.

Algorithm 1 Stepwise network estimation by χ^2 difference testing.

```

Start with initial setup for  $\Omega$ 
repeat
  for all Unique elements of  $\Omega$  do
    Remove edge if present or add edge if absent
    Fit model with changed edge
  end for
  if Adding an edge significantly improves fit ( $\alpha = 0.05$ ) then
    Add edge that improves fit the most
  else if Removing an edge does not significantly worsen fit ( $\alpha = 0.05$ ) then
    Remove edge that worsens fit the least
  end if
until No added edge significantly improves fit and removing any edge significantly worsens fit

```

Algorithm 2 Stepwise network estimation by AIC, BIC or EBIC optimization.

```

Start with initial setup for  $\Omega$ 
repeat
  for all Unique elements of  $\Omega$  do
    Remove edge if present or add edge if absent
    Fit model with changed edge
  end for
  if Any changed edge improved AIC, BIC or EBIC then
    Change edge that improved AIC, BIC or EBIC the most
  end if
until No changed edge improves AIC, BIC or EBIC

```

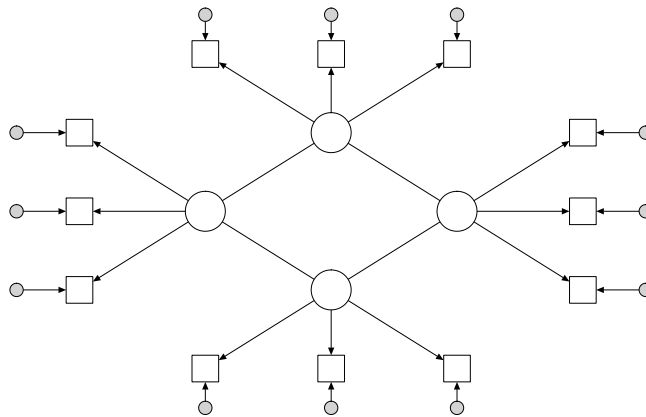


Figure 7.4: Model used in simulation study 1: step-wise model search in latent network modeling. Four latent variables were each measured by three items. Latent variables covary due to the structure of a latent Gaussian graphical model in which edges indicate partial correlation coefficients. This model has the form of a chain graph, which cannot be represented in a structural equation model. Factor loadings, residual variances and latent variances were set to 1 and the latent partial correlations had an average of 0.33 with a standard deviation of 0.04.

Simulation Study 1: Latent Network Modeling

We performed a simulation study to assess the performance of the above mentioned step-wise search algorithms in LNM models. Figure 7.4 shows the LNM model under which we simulated data. In this model, four latent factors with three indicators each were connected in a latent network. The latent network was a chain network, leading all latent variables to be correlated according to a structure that cannot be represented in SEM. Factor loadings and residual variances were set to 1, and the network weights were simulated as described in the section “Simulating Gaussian Graphical models”. The simulation study followed a 5×4 design: the sample size was varied between 50, 100, 250, 500 and 1000 to represent typical sample sizes in psychological research, and the stepwise evaluation criterion was either χ^2 difference testing, AIC, BIC or EBIC (using a tuning parameter of 0.5). Each condition was simulated 1000 times, resulting in 20000 total simulated datasets.

Figure 7.5 shows the results of the simulation study. Data is represented in standard boxplots (McGill, Tukey, & Larsen, 1978): the box shows the 25th, 50th (median) 75th quantiles, the whiskers range from the largest values in 1.5 times the inter-quantile range (75th - 25th quantile) and points indicate outliers outside that range. In each condition, we investigated the sensitivity and specificity. The top panel shows that sensitivity improves with sample size, with AIC performing best and EBIC worst. From sample sizes of 500 and higher all estimation criterion

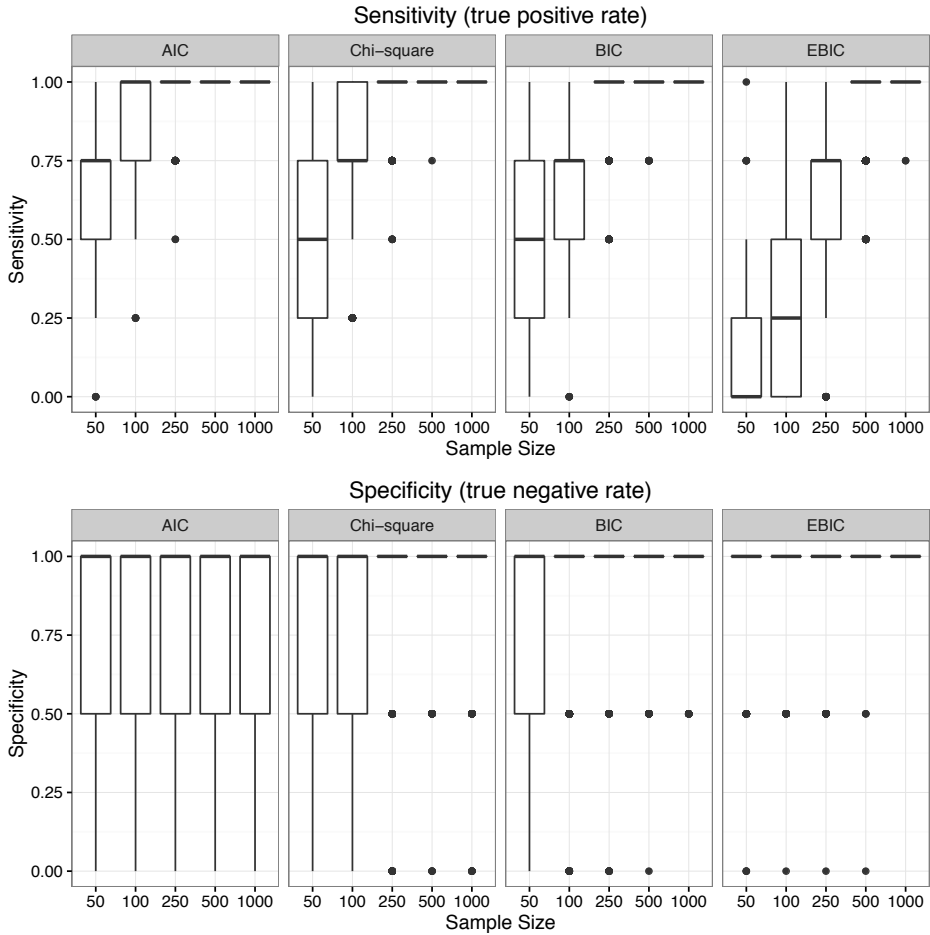


Figure 7.5: Simulation results of simulation study 1: step-wise model search in latent network modeling. Each condition was replicated 1000 times, leading to 20 000 total simulated datasets. High sensitivity indicates that the method is able to detect edges in the true model, and high specificity indicates that the method does not detect edges that are zero in the true model.

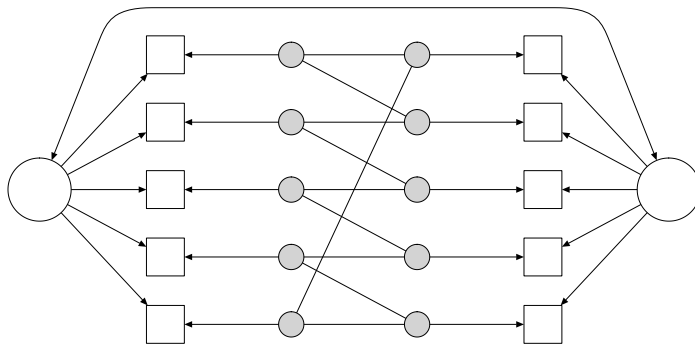


Figure 7.6: Model used in simulation study 2: step-wise model search in residual network modeling. Two latent variables were each measured by five items; a Gaussian graphical model, in which edges indicate partial correlation coefficients, leads to all residuals to be correlated due to a chain graph between residuals, which cannot be represented in a structural equation model. Factor loadings, residual variances and latent variances were set to 1, the factor covariance was set to 0.25 and the latent partial correlations had an average of 0.33 with a standard deviation of 0.04.

performed well in retrieving the edges. The bottom panel shows that specificity is generally very high, with EBIC performing best and AIC worst. These results indicate that the step-wise procedure is conservative and prefers simpler models to more complex models; missing edges are adequately detected but present edges in the true model might go unnoticed except in larger samples. With sample sizes over 500, all four estimation methods show both a high sensitivity and specificity.

Simulation Study 2: Residual Network Modeling

We conducted a second simulation study to assess the performance of step-wise model selection in RNM models. Figure 7.7 shows the model under which data were simulated: two latent variables with 5 indicators each. The residual network was constructed to be a chain graph linking a residual of an indicator of one latent variable to two indicators of the other latent variable. This structure cannot be represented by a DAG and causes all residuals to be connected, so that Θ is fully populated. Factor loadings and residual variances were set to 1, the factor covariance was set to 0.25, and the network weights were simulated as described in the section “Simulating Gaussian Graphical models”.

The simulation study followed a 5×4 design; sample size was again varied between 50, 100, 250, 500 and 1000, and models were estimated using either χ^2 significance testing, AIC, BIC or EBIC. Factor loadings and factor variances were set to 1 and the factor correlation was set to 0.25. The weights in $\Omega_{\mathbf{e}}$ were chosen as described in the section “Simulating Gaussian Graphical models”. Each condition was simulated 1,000 times, leading to 20 000 total datasets.

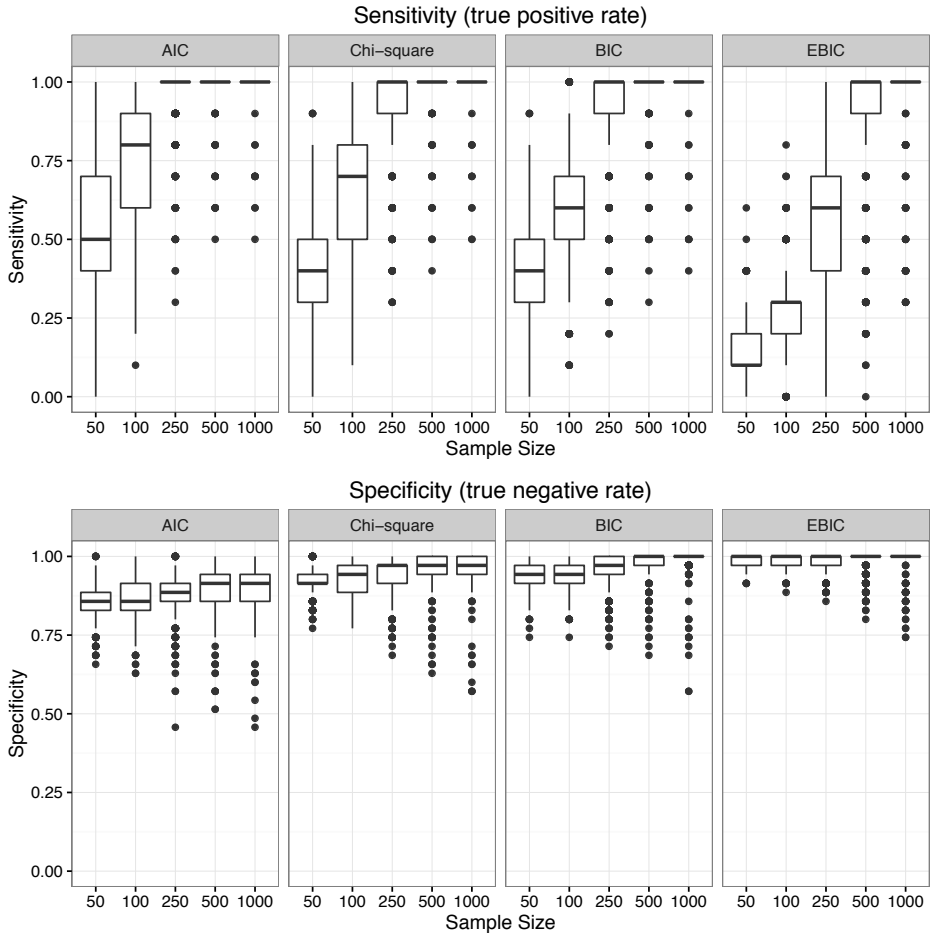


Figure 7.7: Simulation results of simulation study 2: step-wise model search in residual network modeling. Each condition was replicated 1000 times, leading to 20 000 total simulated datasets. High sensitivity indicates that the method is able to detect edges in the true model, and high specificity indicates that the method does not detect edges that are zero in the true model.

Figure 7.7 shows the results of the simulation study. The top panel shows that sensitivity increases with sample size and performs best when using AIC as the criterion. BIC performed comparably in sensitivity to χ^2 testing and EBIC performed the worst. The bottom panel shows that specificity was very high for all sample sizes and all criteria, with EBIC performing best and AIC worst. These results indicate that the number of false positives is very low and that the method is on average well capable of discovering true edges for sample size larger than 250. In sum, all four criteria perform well with EBIC erring on the side of caution and AIC erring on the side of discovery.

LASSO Regularization

While the step-wise model selection algorithms perform well in retrieving the correct network structure, they are very slow when the number of nodes in the network increases (e.g., more than 10 nodes). This is particularly important in the context of RNM, in which the number of indicators can be larger than 10 even in small models. A popular method for fast estimation of high-dimensional network structures is by applying the least absolute shrinkage and selection operator (LASSO; Tibshirani, 1996). LASSO regularization has also recently been introduced in the SEM literature (Jacobucci et al., 2016) as a method for obtaining sparser structures of \mathbf{A} and \mathbf{B} . In the LASSO, instead of optimizing the likelihood function as described in (7.1), a penalized likelihood is optimized (Jacobucci et al., 2016):

$$\min_{\hat{\Sigma}} \left[\log \det \left(\hat{\Sigma} \right) + \text{Trace} \left(\mathbf{S} \hat{\Sigma}^{-1} \right) - \log \det \left(\hat{\mathbf{S}} \right) - P + \nu \text{Penalty} \right], \quad (7.7)$$

in which ν denotes a tuning parameter controlling the level of penalization. The penalty here is taken to be the sum of absolute parameters:

$$\text{Penalty} = \sum_{\langle i,j \rangle} |\omega_{ij}|,$$

in which ω_{ij} denotes an element from either $\mathbf{\Omega}_{\Psi}$ or $\mathbf{\Omega}_{\Theta}$. Other penalty functions may be used as well—such as summing the squares of parameter estimates (ridge regression; Hoerl & Kennard, 1970) or combining both absolute and squared values (elastic net; Zou & Hastie, 2005)—but these are not currently implemented in `lvnet`. The benefit of the LASSO is that it returns models that perform better in cross-validation. In addition, the LASSO yields sparse models in which many relationships are estimated to be zero.

Algorithm 3 LASSO estimation for exploratory network search.

for all Sequence of tuning parameters ν_1, ν_2, \dots **do**
 Estimate LASSO regularized model using given tuning parameter
 Count the number of parameters for which the absolute estimate is larger than ϵ
 Determine information criterion AIC or BIC given fit and number of parameters
end for
 Select model with best AIC, BIC or EBIC
 Refit this model without LASSO in which absolute parameters smaller than ϵ are fixed to zero

The `lvnet` function allows for LASSO regularization for a given model matrix ($\mathbf{\Omega}_\Theta$, $\mathbf{\Omega}_\Psi$, $\mathbf{\Theta}$, $\mathbf{\Psi}$, $\mathbf{\Lambda}$ or \mathbf{B}) and a given value for the tuning parameter ν . The optimizer used in `lvnet` does not return exact zeroes. To circumvent this issue, any absolute parameter below some small value ϵ (by default $\epsilon = 0.0001$) is treated as zero in counting the number of parameters and degrees of freedom (Zou, Hastie, Tibshirani, et al., 2007). The `lvnetLasso` function implements the search algorithm described in Algorithm 3 to automatically choose an appropriate tuning parameter, use that for model selection and rerun the model to obtain a comparable fit to non-regularized models. In this algorithm, a sequence of tuning parameters is tested, which is set by default to a logarithmically spaced sequence of 20 values between 0.01 and 1.

Simulation Study 3: Latent Network Modeling

We studied the performance of LASSO penalization in estimating the latent network structure in a similar simulation study to the study of the step-wise procedure described above. Data were simulated under a similar model to the one shown in Figure 7.4, except that now 8 latent variables were used leading to a total of 24 observed variables. All parameter values were the same as in simulation study 1. The simulation followed a 5×3 design. Sample size was varied between 100, 250, 500, 1 000 and 2 500, and for each sample size 1 000 datasets were simulated leading to a total of 5 000 generated datasets. On these datasets the best model was selected using either AIC, BIC or EBIC, leading to 15 000 total replications. In each replication, sensitivity and specificity were computed. Figure 7.8 shows that AIC had a relatively poor specificity all-around, but a high sensitivity. EBIC performed well with sample sizes of 500 and higher.

Simulation Study 4: Residual Network Modeling

To assess the performance of LASSO in estimating the residual network structure we simulated data as in Figure 7.6, except that in this case four latent variables were used, each with 5 indicators, the residuals of which were linked via a chain graph. All parameter values were the same as in simulation study 2. The design was the same as in simulation study 3, leading to 5 000 generated datasets on which AIC, BIC or EBIC were used to select the best model. While Figure 7.9

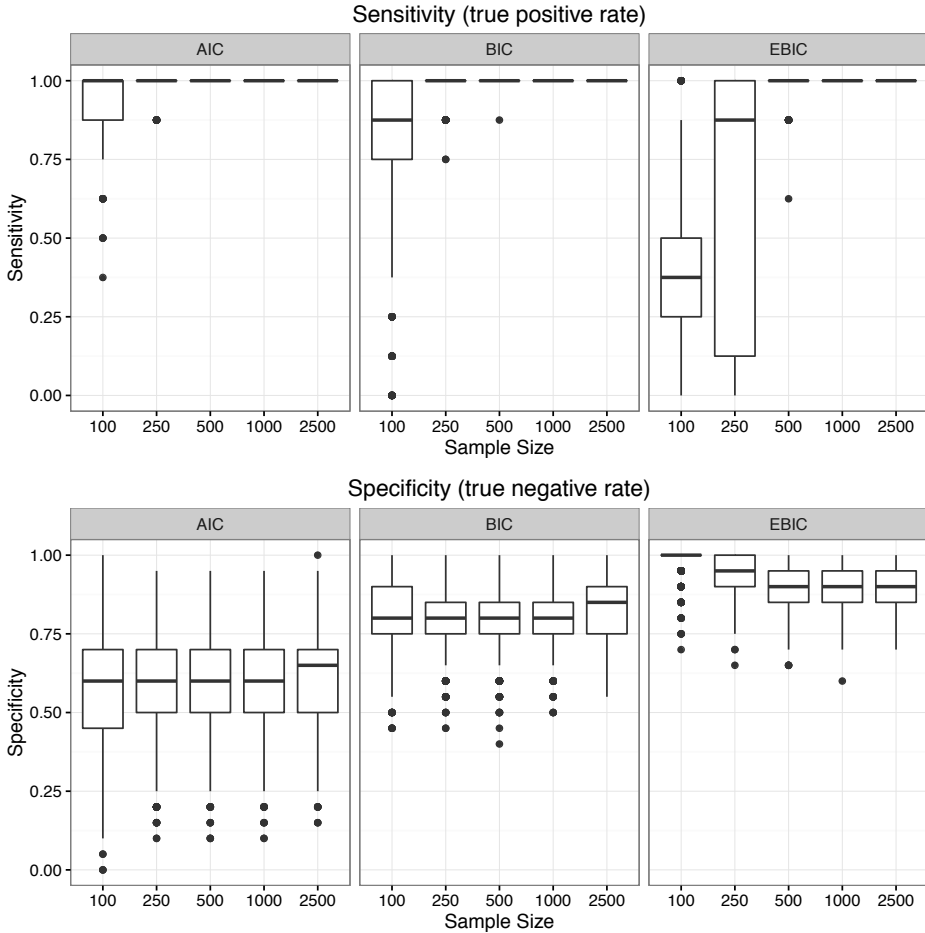


Figure 7.8: Simulation results of simulation study 3: model selection via penalized maximum likelihood estimation in latent network modeling. The same model as in Figure 7.4 was used except now with 4 latent variables leading to 24 observed variables. For each sample size 1000 datasets were generated, leading to 5000 total simulated datasets on which AIC, BIC or EBIC was used to select the best model. High sensitivity indicates that the method is able to detect edges in the true model, and high specificity indicates that the method does not detect edges that are zero in the true model.

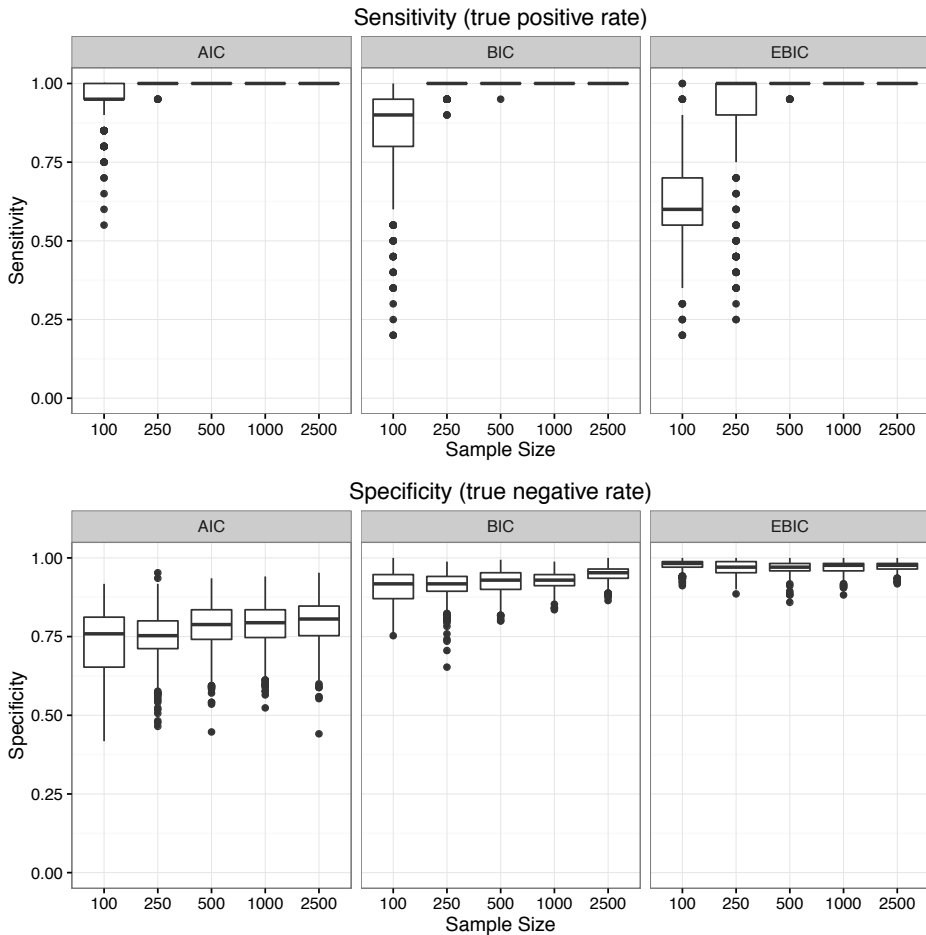


Figure 7.9: Simulation results of simulation study 4: model selection via penalized maximum likelihood estimation in residual network modeling. The same model as in Figure 7.6 was used except now with 4 latent variables leading to 20 observed variables. For each sample size 1000 datasets were generated, leading to 5000 total simulated datasets on which AIC, BIC or EBIC was used to select the best model. High sensitivity indicates that the method is able to detect edges in the true model, and high specificity indicates that the method does not detect edges that are zero in the true model.

shows good performance of the LASSO in retrieving the residual network structure and similar results as before: AIC performs the worst in specificity and EBIC the best.

7.5 Empirical Example: Personality Inventory

In this section, we demonstrate LNM and RNM models by confirmative testing a model and exploratively searching a residual and a latent network structure. We use the `lvnet` package, which can be installed directly from Github using the `devtools` package in R:

```
> library("devtools")
> install_github("sachaepskamp/lvnet")
> library("lvnet")
```

To exemplify the method, we will use a dataset from the `psych` package (Revelle, 2010) on the Big 5 personality traits (Benet-Martinez & John, 1998; Digman, 1989; Goldberg, 1990a, 1993; McCrae & Costa, 1997). This dataset consists of 2800 observations of 25 items designed to measure the 5 central personality traits with 5 items per trait. We estimated the CFA model on the BFI dataset. Next, we used LASSO estimation to the RNM model using 100 different tuning parameters and using EBIC as criterion to maximize specificity and search for a sparse residual network. The fully correlated covariance matrix of latent variables is equivalent to a fully connected latent network structure. Thus, after fitting a RNM model, we can again apply LASSO to estimate a latent network in the resulting model, which we abbreviate here to an RLNM model. The R code used for this analysis can be found in the supplementary materials.

	df	χ^2	AIC	BIC	EBIC	RMSEA	TLI	CFI
CFA	265	4713.94	183233.7	183589.9	184542.4	0.08	0.75	0.78
RNM	172	806.63	179511.0	180419.4	182848.2	0.04	0.94	0.97
RLNM	176	843.18	179539.5	180424.2	182789.5	0.04	0.94	0.97

Table 7.1: Fit measures for three models estimated on the BFI dataset in the `psych` R package. CFA is the correlated five-factor model. RNM is the same model as the CFA model with a residual network. RLNM denotes the same model as the RNM model in which edges of the latent network have been removed.

Table 7.1 shows the fit of the three models. The CFA model fits poorly. The RNM model has substantively improved fit and resulted in good fit indices overall. The estimated RLNM model showed that 5 edges could be removed from the latent network after taking residual interactions into account. Figure 7.10 shows the factor structure and residual network of the final RLNM model. It can be seen that Agreeableness is now only connected to extraversion: after taking into account someone's level of extraversion agreeableness is independent of the other three personality traits. Extraversion is the most central node in this network and the only trait that is directly linked to all other traits. The residual network shows many meaningful connections. While seemingly densely connected, this network only has 30% of all possible edges in a network of that size, leading the model to have 176 degrees of freedom. The corresponding residual covariance structure is fully populated with no zero elements.

It should be noted that the procedures used in this example are highly explorative. While Table 7.1 shows that the RNM fits better than the CFA model,

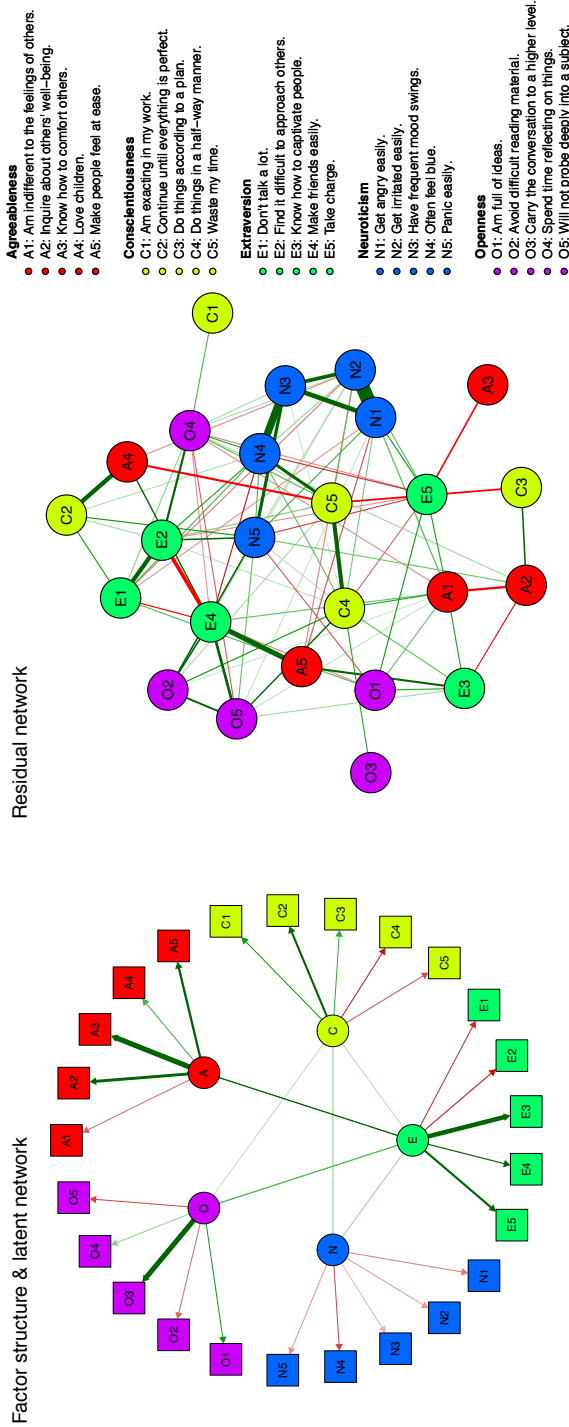


Figure 7.10: Visualization of the factor structure and latent network (left) and the residual network (right) of the BFI personality dataset from the psych package in R. LASSO estimation with 100 different tuning parameters in combination with EBIC model selection was used to first estimate the residual network structure and following the latent network structure.

the CFA model is solely based on theory whereas the RNM model was found through high-dimensional model search. As a result, to substantially interpret the structure found in Figure 7.10 it should first be replicated in independent samples.

7.6 Conclusion

In this chapter we introduced a formal psychometric model for network modeling of multivariate normal data. We contrasted this model with latent variable models as commonly used in CFA and SEM. Furthermore, using the CFA and SEM frameworks, we proposed two generalizations of the network model to encompass latent variable structures within the network paradigm. In the first generalization, LNM, we construct a network among the latent variables, whereas in the second generalization, RNM, a network is formed among residuals of indicators. Both frameworks offer powerful benefits over both latent variable and network modeling. From the perspective of latent variable modeling, the LNM framework allows one to exploratively search for conditional independence relationships between latent variables without the need for prior theory, and the RNM framework allows one to model latent common causes without assuming local independence. From the perspective of network modeling, the LNM framework allows one to model network structures while taking measurement error into account, and the RNM framework allows one to estimate a network structure, even when all nodes are in part caused by unobserved or latent variables. In addition, both frameworks allow for network models to be fitted and compared to SEM models. The discussed methodology has been implemented in the freely available R package `lvnet`.

Simulation studies showed that step-wise search and penalized maximum likelihood estimation of the residual or latent network structures resulted in high specificity all around—the methods did not result often in false positives—and rapidly increasing sensitivity as a function of the sample size; the higher the sample size, the more true edges were detected in the algorithm. These numbers are comparable to state-of-the-art network estimation techniques in sample and model sizes that are plausible in psychological settings (van Borkulo et al., 2014). In all four simulation studies, using AIC as the model selection criterion led to the best sensitivity and using EBIC led to the best specificity. However, it is important to note that the choice of a particular information criterion cannot be argued by these numbers alone, and depends on the relative importance one assigns to the side of discovery (optimizing sensitivity) or the side of caution (optimizing specificity; Dziak et al., 2012). Furthermore, it should be noted that these simulation results are specific for the particular model setup and sample sizes used; results might be different for other kinds of models or sample size ranges.

In addition to the LNM and RNM frameworks, other combinations of CFA, SEM and network modeling are possible as well. For example, a framework can be constructed which contains both a latent and a residual network (as shown in our empirical example), or directed regression paths as in the SEM model can be added to the LNM model. While these models are all estimable in the `lvnet` software, in the current chapter we chose to focus on the distinct benefits that modeling a residual or latent network presents. Thus, in this chapter, we only described the

modeling of multivariate normal data. More advanced models are possible, but not yet implemented in the `lvnet` software. In the case of binary variables, the appropriate model to use is the Ising model, which has been shown to be equivalent to multivariate item response models (see Chapter 8). Future research could aim at constructing Ising models among binary latent variables in latent class analysis, or constructing residual networks in models with binary indicators. Finally, the expressions optimized in Equations (7.1) and (7.7) are based on summary statistics and therefore only truly applicable to complete data. With incomplete data, the appropriate estimation method is to use full-information maximum likelihood (FIML; Arbuckle, Marcoulides, & Schumacker, 1996); however, FIML has not yet been implemented in the `lvnet` software.

In our view, the presented modeling framework is a versatile and promising addition to the spectrum of psychometric models. The GGM, which has a central place in this modeling framework, acts as a natural interface between correlation and causality, and we think this representation should receive more attention in psychometrics. From the point of view afforded by the current chapter, the typical attempt to determine directed SEMs from correlation structures in fact appears somewhat haphazard in psychology, a historical accident in a field that has been prematurely directed to hypothesis testing at the expense of systematic exploration. Perhaps, psychometrics as a field should consider taking a step back to focus on the consistent identification of GGMs, instead of wanting to jump to the causal conclusion immediately. In this regard, the fact that GGMs do not have equivalent models would appear to be a major benefit, as they allow us to focus on charting connections between variables systematically, without being forced to adhere to one particular causal interpretation or another. In addition, because the GGM does not specify the nature or direction of interactions between variables, it appears a natural model for research situations where no temporal information or experimental interventions are present, so that associations may arise for a multitude of reasons: the GGM can be consistently interpreted regardless of whether associations arise from direct causal relations, reciprocal causation, latent common causes, semantic overlap between items, or homeostatic couplings of parameters. While this can be seen as a downside of the GGM—the lack of directionality leads to less falsifiable hypotheses—it can also be a major asset in a field like psychology, where strong causal theory is sparse and the identification of DAGs often appears a bridge too far.