Chapter 2

Regularized Partial Correlation Networks

Abstract

Recent years have seen an emergence of network modeling applied to moods, attitudes, and problems in the realm of psychology. In this framework, psychological variables are understood to directly interact with each other rather than being caused by an unobserved latent entity. In this tutorial, we introduce the reader to estimating the most popularly used network model for psychological data: the partial correlation network. We describe how regularization techniques can be used to efficiently estimate a parsimonious and interpretable network structure on cross-sectional data. We show how to perform these analyses in R and demonstrate the method in an empirical example on post-traumatic stress disorder data. In addition, we discuss the effect of the hyperparameter that needs to be manually set by the researcher and provide a checklist with potential solutions for problems often arise when estimating regularized partial correlation networks. The chapter concludes with a simulation study that shows the performance of the discussed methodology using a plausible psychological network structure.

2.1 Introduction

Recent years have seen the emergence of the use of network modeling for exploratory studies of psychological behavior as an alternative to latent-variable modeling (Borsboom & Cramer, 2013; Schmittmann et al., 2013). In these so-called *psychological networks* (Epskamp, Borsboom, & Fried, 2016), nodes represent psychological variables such as mood states, symptoms or attitudes, and

This chapter has been adapted from: Epskamp, S., and Fried, E.I. (2016). A Tutorial on Regularized Partial Correlation Networks. *arXiv preprint*, arXiv:1607.01367, and: Epskamp, S. (2016). Regularized Gaussian Psychological Networks: Brief Report on the Performance of Extended BIC Model Selection. *arXiv preprint*, arXiv:1606.05771.

links between the nodes represent unknown statistical relationships that need to be estimated. As a result, this class of network models is strikingly different from e.g., social networks in which links are known (Wasserman & Faust, 1994), and poses novel problems of statistical inference. A great body of technical literature exists on the estimation of such network models (e.g., Meinshausen & Bühlmann, 2006; Friedman, Hastie, & Tibshirani, 2008; Hastie, Tibshirani, & Friedman, 2001; Hastie, Tibshirani, & Wainwright, 2015; Foygel & Drton, 2010). However, this line of literature often requires a more technical background than can be expected from psychological researchers and does not focus on the unique problems that come with analyzing psychological data, such as the handling of ordinal data, interpretability of networks based on different samples and attempting to find evidence for an underlying causal mechanism. While this tutorial is aimed at empirical researchers in psychology, it should be noted that the methodology can readily be applied to other fields of research as well.

The main type of model used to estimate psychological methods are so-called pairwise Markov random fields (PMRF; Lauritzen, 1996; Murphy, 2012). The present chapter will focus on the most common PMRF for continuous data: *partial correlation networks*. Partial correlation networks are usually estimated using *regularization*, an important statistical procedure that helps to recover the true network structure of the data. In this chapter, we present a tutorial on estimating such regularized partial correlation networks, using a methodology implemented in the *qgraph* package (Epskamp, Cramer, Waldorp, Schmittmann, & Borsboom, 2012) for the statistical programming language R (R Core Team, 2016). This methodology has already been used in a substantive number of publications in diverse fields, such as psychology, psychiatry, health sciences and more (e.g., Fried, Epskamp, Nesse, Tuerlinckx, & Borsboom, 2016; Isvoranu, van Borkulo, et al., 2016; Isvoranu, Borsboom, van Os, & Guloksuz, 2016; Knefel, Tran, & Lueger-Schuster, 2016; Levine & Leucht, 2016; Jaya, Hillmann, Reininger, Gollwitzer, & Lincoln, 2016; Deserno, Borsboom, Begeer, & Geurts, 2016; McNally, 2016; Kossakowski et al., 2016; Langley, Wijn, Epskamp, & Van Bork, 2015; van Borkulo et al., 2015). However, the methodology itself has not yet been introduced in psychological literature. In addition, because of the novelty of regularized partial correlation networks in psychological research, we are not aware of concise and clear introductions aimed at empirical researchers that explain regularization. The goal of this chapter is thus (1) to provide a short introduction to regularization partial correlation networks, (2) to outline the commands used in R to perform this procedure, and (3) to present a checklist for identifying the most common problems and questions arising from regularized networks. In addition, this chapter will present simulation results that show the described estimation method works well with plausible psychological networks on both continuous and ordinal data.

2.2 Partial Correlation Networks

The most commonly used framework for constructing a psychological network on data that can be assumed to be multivariate normal[1] is to estimate a network of *partial correlation coefficients* (McNally et al., 2015; Borsboom & Cramer, 2013). Such networks can also be termed *concentration graphs* (Cox & Wermuth, 1994) or *Gaussian graphical models* (Lauritzen, 1996). Each link in the network represents a partial correlation coefficient between two variables after conditioning on all other variables in the dataset. These coefficients range from -1 to 1 and encode the remaining association between two nodes after controlling for all other information possible, also known as conditional independence associations. Typically, the connections are visualized using red lines indicating negative partial correlations, green lines indicating positive partial correlations, and wider and more saturated connections indicate partial correlations that are far from zero (see Chapter 9). Whenever the partial correlation is exactly zero, no connection is drawn between two nodes, indicating that two variables are independent after controlling for all other variables in the network. This is of particular interest since such a missing connection indicates one of the two variables could not have caused the other (Pearl, 2000). As such, whenever there is a connection present, it highlights a potential causal pathway between two variables (see also Chapter 6).

Due to sampling variation, we do not obtain partial correlations that are exactly zero when estimating a partial correlation network. Instead, even when in reality two variables are conditionally independent, we still obtain partial correlations that are very small and are represented as very weak edges in the network. These connections are called *spurious* (Costantini, Epskamp, et al., 2015), as they represent relationships that are not true in reality. We wish to control for such spurious connections, especially considering the fact that we estimate a large number of parameters in partial correlation networks that can also lead to false positive associations. One way to do so is to test all partial correlations for statistical significance and remove all connections that fail to reach significance (Drton & Perlman, 2004). However, this poses a problem of multiple testing, and controlling for this problem (e.g., by using a Bonferroni correction) results in a loss of power (Costantini, Epskamp, et al., 2015).

2.3 LASSO Regularization

An increasingly popular method for controlling for spurious connections—as well as to obtain easier interpretable networks that may perform better in cross-validation prediction—is to use statistical *regularization* techniques originating in the field of machine learning. The goal here is to obtain a network structure in which as few connections as possible are required to parsimoniously explain the covariance among variables in the data. Especially prominent is to use of the 'least absolute shrinkage and selection operator' (LASSO; Tibshirani, 1996).

[1] The assumption of normality can be relaxed by applying a transformation when data are continuous but not normal (Liu, Lafferty, & Wasserman, 2009), or by basing the network estimation on polychoric correlations when the data are ordinal.

In essence, the LASSO shrinks partial correlation coefficients when estimating a network model, which means that small coefficients are estimated to be exactly zero. This results in fewer connections in the network, or in other words, a *sparse* network in which likely spurious connections are removed. The LASSO utilizes a tuning parameter λ (lambda) that needs to be set, controlling this level of sparsity. When the tuning parameter is low, only few connections are removed, likely resulting in too many spurious connections. When the tuning parameter is high, many connections are removed, likely resulting in too many true connections to be removed in addition to all spurious connections. More broadly, when λ equals zero every connection remains in the network and when λ is substantively high no connection remains in the network. As such, the tuning parameter needs to be carefully selected to result in a network structure that minimizes the number of spurious connections while maximizing the number of true connections (Foygel Barber & Drton, 2015; Foygel & Drton, 2010).

Typically, a range of networks is estimated under different values of λ (Zhao & Yu, 2006). The value for λ under which no edges are retained (the empty network), $\lambda_{\max}$, is set to the largest absolute correlation (Zhao et al., 2015). A minimum value can be chosen by multiplying some ratio R with this maximum value[2]:

$$\lambda_{\min} = R\lambda_{\max}.$$

A logarithmically spaced range of tuning parameters (typically 100 different values), ranging from $\lambda_{\min}$ to $\lambda_{\max}$, can be used to estimate different networks. To summarize, the LASSO can be used to estimate a *range* of networks rather than a single network, ranging from a fully connected network to a fully disconnected network. Next, one needs to select the best network out of this range of networks. This selection can be done by optimizing the fit of the network to the data (i.e. by minimizing some information criterion). Minimizing the Extended Bayesian Information Criterion (EBIC; Chen & Chen, 2008) has been shown to work particularly well in retrieving the true network structure (Foygel Barber & Drton, 2015; Foygel & Drton, 2010; van Borkulo et al., 2014), especially when the generating network is sparse (i.e., does not contain many edges). LASSO regularization with EBIC model selection has been shown to have high specificity all-around (i.e., does not estimate edges that are not in the true network) but a varying sensitivity (i.e., estimates edges that are in the true network) based on the true network structure and sample size. For example, sensitivity typically is less when the true network is dense (contains many connections) or features some nodes with many edges (hubs).

Many variants of the LASSO with different methods for selecting the LASSO tuning parameter have been implemented in open-source software (e.g., Krämer, Schäfer, & Boulesteix, 2009; Zhao et al., 2015). We suggest to use the variant termed the ‘graphical LASSO’ (glasso; Friedman et al., 2008), which is a fast variant of the LASSO specifically aimed at estimating partial correlation networks. The glasso algorithm has been implemented in the *glasso* package (Friedman, Hastie, & Tibshirani, 2014) for the statistical programming language R (R Core Team, 2016). An automatic function that uses this package in combination with

[2] The *qgraph* package uses $R = 0.01$ by default.

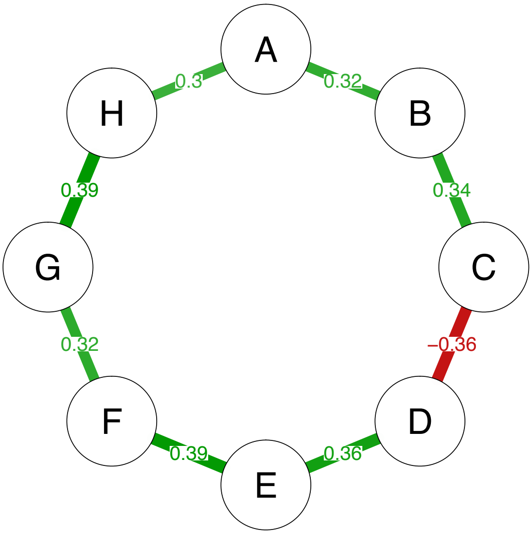


Figure 2.1: True network structure used in simulation example. The network represents a *partial correlation network*: nodes represent observed variables and links represent partial correlations between two variables after conditioning on all other variables. The simulated structure is a *chain graph* in which all absolute partial correlation coefficients were drawn randomly between 0.3 and 0.4.

EBIC model selection as described by Foygel and Drton (2010) has been implemented in the R package *qgraph* (Epskamp et al., 2012). We suggest using this routine because—in addition to simple input commands—it only requires an estimate of the covariance matrix and not the raw data, allowing one to use, e.g., polychoric correlation matrices when the data are ordinal.

The EBIC uses a hyperparameter γ (gamma) that controls how much the EBIC prefers simpler models (fewer connections). This hyperparameter γ should not be confused with the LASSO tuning parameter λ, and needs to be set manually. It typically is set between 0 and 0.5 (Foygel & Drton, 2010, suggest to use 0.5), with higher values indicating that simpler models (more parsimonious models with fewer connections) are preferred. Setting the hyperparameter to 0 errs on the side of discovery: more connections are estimated, including possible spurious ones (the network has a higher specificity). Setting the hyperparameter to 0.5, on the other hand, errs on the side of caution or parsimony: fewer connections are obtained including hardly any spurious connections but also less true connections (the network has a higher sensitivity). It is important to mention that even when setting the hyperparameter to 0, the network will still be sparser compared to a partial correlation network that does not employ any form of regularization; setting γ to 0 indicates that the EBIC reduces to the standard BIC, which is still a criterion that prefers simple models.

To exemplify the above-described method of selecting a best fitting regularized partial correlation network, we simulated a dataset of 100 people and 8 nodes based on the *chain graph* shown in Figure 2.1. Such graphs are particularly suitable for our example because the true network (the one we want to recover with our statistical analysis) only features connections among neighboring nodes visualized in a circle. This makes spurious connections—any edge that connects non-neighboring nodes—easy to identify visually. We used the *qgraph* package to estimate 100 different network structures, based on different values for λ, and compute the EBIC under different values of γ. Figure 2.2 depicts a representative sample of 10 of these networks. As can be seen, networks 1 through 7 feature spurious connections and err on the side of discovery, while networks 9 and 10 recover too few connections and err on the side of caution. For each network, we computed the EBIC based on γ of 0, 0.25 and 0.5 (the parameter the researchers needs to set manually). The boldface values show the best fitting models, indicating which models would be selected using a certain value of γ. When $\gamma = 0$ was used, network 7 was selected that featured three weak spurious connections. When γ was set to 0.25 or 0.5 (the default in *qgraph*) respectively, network 8 was selected, which has the same structure as the true network shown in Figure 2.1. These results show that in our case, varying γ changed the results only slightly. Importantly, this simulation does not imply that $\gamma = 0.5$ always leads to the true model; simulation work has shown that 0.5 is fairly conservative and may result in omitting true edges from the network, while it is very unlikely that spurious ones are obtained (Foygel & Drton, 2010). In sum, the choice of the hyperparameter is somewhat arbitrary and up to the researcher, and depending on the relative importance assigned to caution or discovery (Dziak, Coffman, Lanza, & Li, 2012). Which of these γ values work best is a complex function of the (usually unknown) true network structure.

A note on sparsity. It is important to note that both thresholding networks based on significance of edges or using LASSO regularization will lead to edges being removed from the network (termed a sparse network), but do not present evidence that these edges are, in fact, zero (see Chapter 4). This is because these methods seek to maximize *specificity*; that is, they all aim to include as few *false positives* (edges that are not in the true model) as possible. All these methods will return empty network structures when there is not enough data. It is important to note that observing a structure with missing edges, or even an empty network, is in no way evidence that there are, in fact, missing edges. This is because these methods do not try to keep the number of *false negatives* low, that is, the number of edges that are not present in the estimated network but are present in the true network. This is related to a well-known problem of null hypothesis testing (to which, roughly, all these methods correspond): *Not* rejecting the null-hypothesis is not evidence that the null hypothesis is true (Wagenmakers, 2007). That is, we might not include an edge because the data are too noisy or because the null hypothesis is true; classical tests and LASSO regularization cannot differentiate between these two reasons. Quantifying evidence for edge weights being zero is still a topic of future research (see Chapter 12).

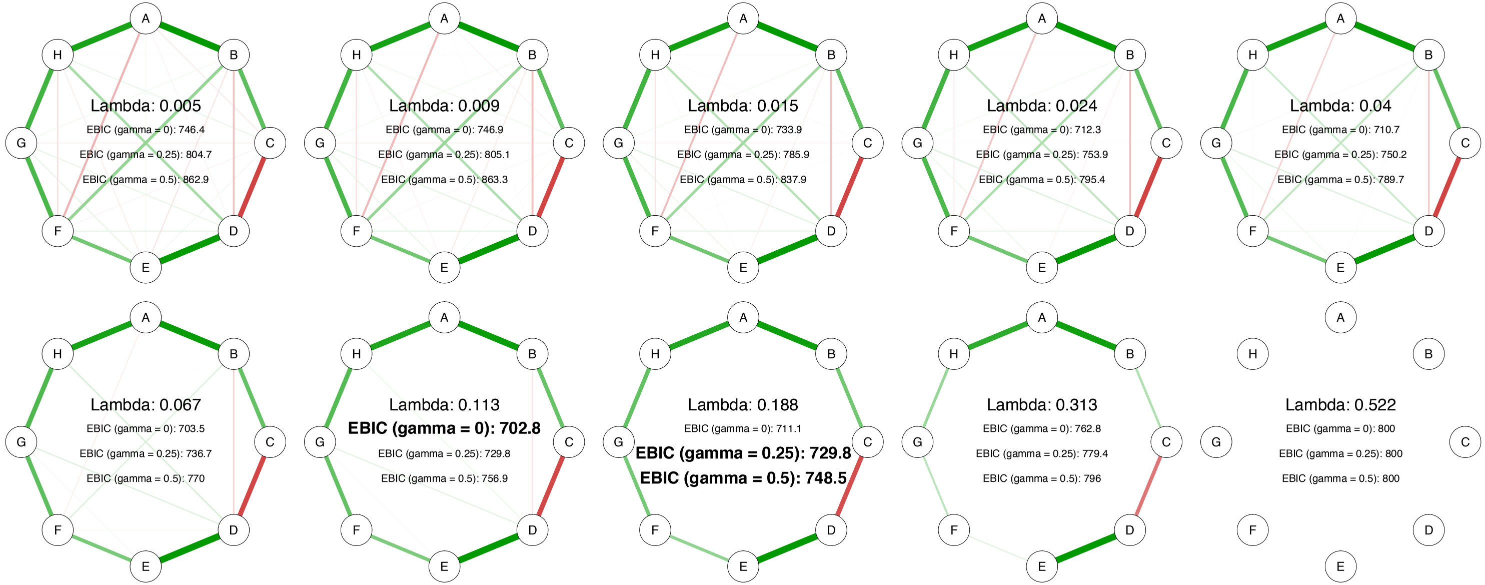


Figure 2.2: Ten different partial correlation networks estimated using LASSO regularization. Setting the LASSO tuningparameter λ that controls sparsity leads to networks ranging from densely connected to fully unconnected. Data were simulated under the network represented in Figure 2.1. The fit of every network was assessed using the EBIC, using hyperparameter γ set to 0, 0.25 or 0.5. The bold-faced EBIC value is the best, indicating the network which would be selected and returned using that γ value.

2.4 Example

In this paragraph, we use an example dataset to estimate a network on data of 221 people with a sub-threshold post-traumatic stress disorder (PTSD) diagnosis; the network features 20 PTSD symptoms. A detailed description of the dataset can be found elsewhere (Armour et al., 2016), and the full R codes for this analysis can be found in the supplementary materials.

The following R codes perform regularized estimation of a partial correlation network using EBIC selection (Foygel & Drton, 2010). These codes make use of the *qgraph* package (Epskamp et al., 2012), which in turns utilizes the *glasso* package for the glasso algorithm (Friedman et al., 2014). These codes assume data is present in R under the object name `Data`.

```
library("qgraph")
corMat <- cor_auto(Data)
graph <- qgraph(corMat,
    graph = "glasso",
    sampleSize = nrow(Data),
    layout = "spring",
    tuning = 0.5)
```

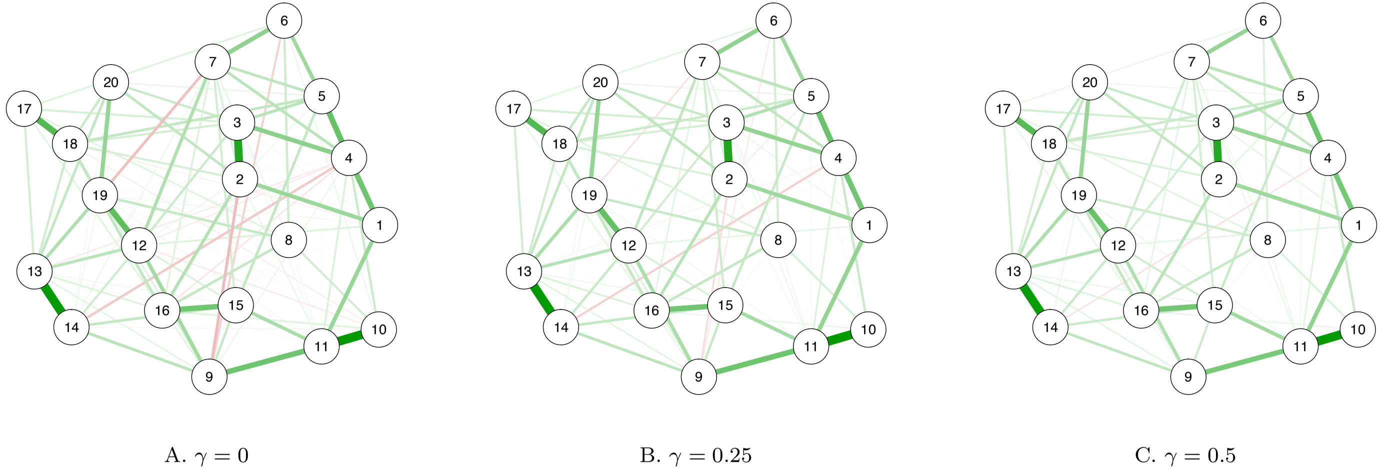
In these codes, `library("qgraph")` loads the package into R and the `cor_auto` function detects ordinal variables (variables with up to 7 unique integer values) and uses the *lavaan* package (Rosseel, 2012) to estimate polychoric, polyserial and Pearson correlations. The `qgraph` function estimates and plots the network structure. The argument `graph` specified that we want to use the glasso algorithm with EBIC model selection, the argument `sampleSize` specifies the sample size of the data, the argument `layout` specifies the node placement and the argument tuning specified the EBIC hyperparameter. The hyperparameter is here set to 0.5, which is also the current default value used in *qgraph*. For more control on the estimation procedure, one can use the `EBICglasso` function, which is automatically called when using `qgraph(..., graph = "glasso")`. Finally, the estimated weights matrix can be obtained either directly using `EBICglasso` or by using the `getWmat` function on the output of `qgraph`:

```
getWmat(graph)
```

Figure 2.3 shows the resulting network estimated under three different values of the hyperparameter 0, 0.25, and 0.5. Table 2.1 shows the description of the nodes. If we investigate the number of edges, we would expect that the network with the largest hyperparameter of 0.5 has the fewest connections. This is indeed the case: the network features 105 edges with $\gamma = 0$, 95 edges with $\gamma = 0.25$, and 87 edges with $\gamma = 0.5$.

We can further investigate properties of the network structures by investigating how important nodes are in the network using measures called centrality indices. A plot of these indices can be obtained as followed:

```
centralityPlot(graph)
```



A. $\gamma = 0$

B. $\gamma = 0.25$

C. $\gamma = 0.5$

Figure 2.3: Partial correlation networks estimated on responses of 221 subjects on 20 PTSD symptoms, with increasing levels of the LASSO hyperparameter γ (from left to right: panel A = 0, panel B = 0.25, panel C = 0.5).

Table 2.1: Description of nodes shown in Figure 2.3

Node	Description
1	Intrusive Thoughts
2	Nightmares
3	Flashbacks
4	Emotional cue reactivity
5	Psychological cue reactivity
6	Avoidance of thoughts
7	Avoidance of reminders
8	Trauma-related amnesia
9	Negative beliefs
10	Blame of self or others
11	Negative trauma-related emotions
12	Loss of interest
13	Detachment
14	Restricted affect
15	Irritability/anger
16	Self-destructive/reckless behavior
17	Hypervigilance
18	Exaggerated startle response
19	Difficulty concentrating
20	Sleep disturbance

An overview of these measures and their interpretation can be found in Chapter 1 and Chapter 10. All measures indicate how important nodes are in a network, with higher values indicating that nodes are more important. Figure 2.4 was made using `centralityPlot` and shows the resulting centrality of all three networks shown in Figure 2.3. For a substantive interpretation of the network model obtained from this dataset we refer the reader to Armour et al. (2016).

2.5 Common Problems and Questions

The estimation of regularized networks is not always without problems and can sometimes lead to network structures that are hard to interpret. Here, we list several common problems and questions encountered when estimating these models.

1. The estimated network has no or very few edges. This can occur in the unlikely case when variables of interest do not exhibit partial correlations. More likely, it occurs when the sample size is too low for the number of nodes in the network. The EBIC penalizes edge weights based on sample size to avoid false positive associations, which means that with increasing sample size, the partial correlation network will be more and more similar to the regularized partial correlation network. The smaller the sample, however, the stronger the impact of the regularization on the network in terms of parsimony. Figure 2.5 (panel A) shows a network estimated on the same data

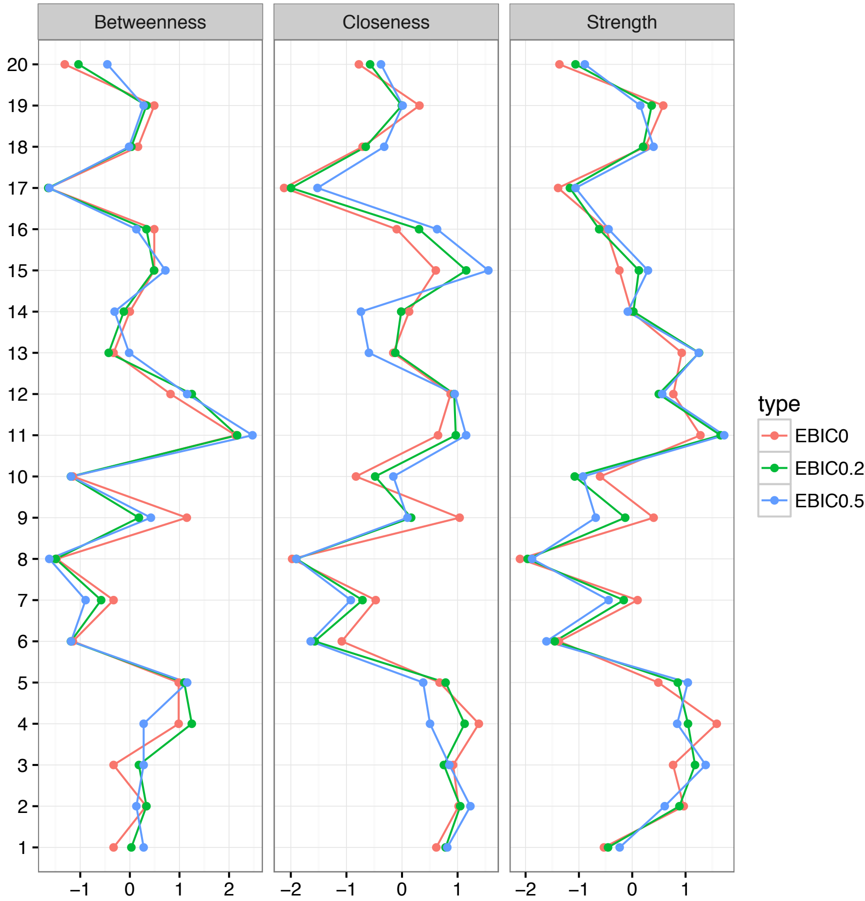


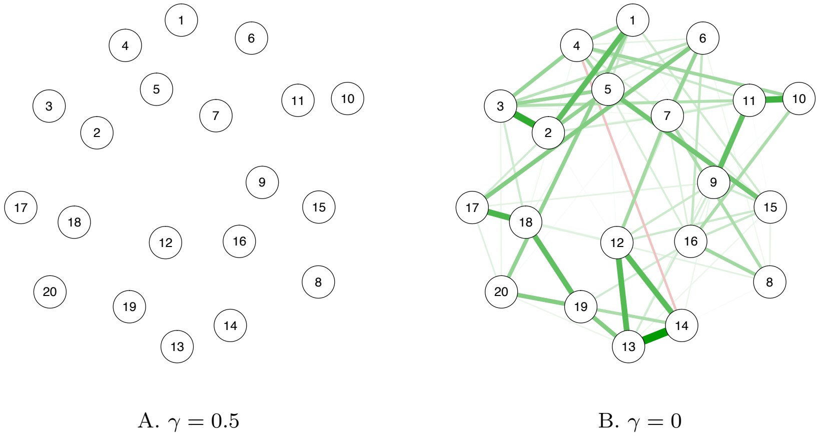
Figure 2.4: Closeness, betweenness, and degree centrality of the three networks described in Figure 2.3 with increasing levels of the LASSO hyperparameter γ. Values are standardized to z-scores.

as Figure 2.3, but this time with only 50 instead of the 221 participants. A way to remediate this problem is by setting the hyperparameter lower (e.g., 0; see Figure 2.5 panel B). Note that this likely leads to spurious connections. An alternative solution is to make a selection of the variables of interest and estimate a network based only on a subset of variables, as less nodes in the network leads to less edges to be estimated, resulting in more observations per parameter to be estimated.

2. The network is densely connected (i.e., many edges) including many unexpected negative connections and, in particular, including many implausibly high partial correlations (e.g., higher than 0.8). As the LASSO aims to remove connections and return a relatively sparse network, we would not expect densely connected networks. In addition, we would not expect many partial correlations to be so high, as (partial) correlations above 0.8 indicate near-perfect collinearity between variables. These structures can occur when the correlation matrix used as input is not *positive definite*, which in turn can be a result of a too small sample size, or of the estimation of polychoric correlations. In the case of a non-positive definite correlation matrix, `cor_auto` will warn the user and attempt to correct for this by searching for a nearest positive definite matrix. This matrix, however, can still lead to wildly unstable results. When the network looks very strongly connected with few (if any) missing connections and partial correlations near 1 and -1, the network structure is likely resulting from such a problem and should not be interpreted. We suggest that researchers always compare networks based on polychoric correlations with networks based on Spearman correlations (they should look somewhat similar) to rule out if estimating the polychoric correlations are the source of this problem.

3. While in general the graph looks as expected (i.e., relatively sparse), some connections are extremely high and/or unexpectedly extremely negative. This problem is related to the previous problem. The estimation of polychoric correlations relies on the pairwise cross-tables of variables in the dataset. When the sample size is relatively low, some cells in the cross-tables could be zero (e.g., nobody was observed that scored a 2 on one item and a 1 on another item). This can lead to unstable estimated polychoric correlations, and in turn to unstable partial correlations. Again, the network based on polychoric correlations should be compared to a network based on Spearman correlations. Obtaining very different networks indicates that the estimation of the polychoric correlations may not be trustworthy.

4. Negative connections are found between variables where one would expect positive connections. For example, two symptoms of the same disorder could, unexpectedly, feature a negative partial correlation rather than a positive one. This can occur artificially when one conditions on a *common effect* (Pearl, 2000). Suppose one measures students' grades of a recent test, their motivation, and the easiness of that test (Koller & Friedman, 2009). We expect the grade to be positively influenced by the easiness of the test and the motivation of the student, and we do not expect any correlation between



A. $\gamma = 0.5$

B. $\gamma = 0$

Figure 2.5: Network of 20 PTSD symptoms. Instead of the full data like in Figure 2.3 (221 subjects), only 50 subjects were used. Panel A: LASSO hyperparameter γ set to the default of 0.5; panel B: γ set to 0 for discovery.

motivation and easiness: knowing a student is motivated does not help us predict the easiness of a test. However, if we only look at students who obtained an A for the test (i.e., *conditioning* on grades), we now *can* predict that if the student is not at all motivated, the test must have been very easy. By conditioning on the common effect (grade) we artificially created a *negative* partial correlation between test easiness and student motivation. Because partial correlation networks indicate such conditional relationships, these negative relationships can occur when common effect relationships are present, and unexpected negative relationships might indicate common effect structures. Another way these relationships can occur is if the network is based on a subsample of the population, and that subsample is a common effect of the nodes in the network. For example, when one splits the sample based on the *sum score* of variables used also in the network, negative relationships could be induced. We recommend results based on such subsamples to be interpreted with care.

In addition to the above-mentioned problems, some questions are often encountered in network analysis:

1. How large does my sample have to be for a given number of nodes? Or in other words, how stable are the estimated network structures and centrality indices to sampling size? This topic goes beyond the scope of this chapter, and is further discussed in Chapter 3. In summary, networks are complicated models using many parameters, which can be unstable given relatively low sample sizes. The LASSO remedies this problem somewhat, and stable networks can be obtained with much smaller samples compared to unregularized networks. Nonetheless, network models estimate a large number of

parameters, implying that even when the LASSO is used, the models need considerable power to obtain stable parameter estimates. It is therefore advisable to always check for the accuracy and stability of edge weights and centrality measures when these are reported and substantively interpreted (c.f., Chapter 3).

2. Can we compare two different groups of people (e.g., clinical patients and healthy controls) regarding the connectivity or density of their networks (i.e. the number of connections)? The answer depends on the differences in sample size. As mentioned before, the EBIC is a function of the sample size: the lower the sample size, the more parsimonious the network structure. This means that comparing the connectivity of two networks is meaningful if they were estimated on roughly the same sample size, but that differences should not be compared if this assumption is not met (e.g., see Rhemtulla et al., 2016). A statistical test for comparing networks based on different sample sizes is currently being developed (Van Borkulo et al., 2016)[3].

3. Does the network structure provide evidence that the data are indeed causally interacting and derive from a true network model, and not from a common cause model where the covariance of symptoms is explained by one or more underlying latent variables (Schmittmann et al., 2013)? The short answer is no. While psychological networks have been introduced as an alternative modeling framework to latent variable modeling, and are capable of strongly changing the point of focus from the common shared variance to unique variance between variables (Costantini, Epskamp, et al., 2015), they do not necessarily disprove the latent variable model. There is a direct equivalence between network models and latent variable models (see Chapter 7 and Chapter 8), and if we generate data based on a true latent variable model, the corresponding network model will be fully connected. However, this does not mean that when the resulting network is not fully connected, the latent variable model must be false. LASSO estimation will *always* return a sparse network with at least some missing connections. As such, observing that there are missing connections does not indicate that the true model was a model without missing connections. Because of the equivalence stated above, observing a model with missing connections cannot be taken for evidence that a latent variable model was not true. A more detailed discussion on this topic can be found in Chapter 4 and a methodology on statistically comparing fit of a network model and latent variable model is described in Chapter 7. In addition, statistical tests to distinguish sparse networks from latent variable models are currently being developed (Van Bork, 2015).

2.6 Simulation Study

While partial correlation network estimation using EBIC model selection has already been shown to work well in retrieving the GGM structure (Foygel & Drton,

[3] `github.com/cvborkulo/NetworkComparisonTest.`

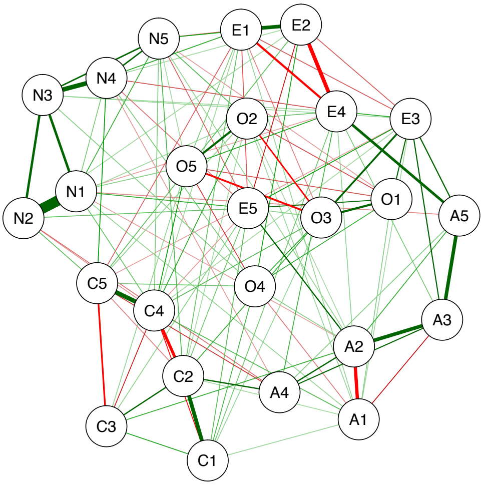


Figure 2.6: True Gaussian graphical model used in simulation study. The network was obtained by computing the (unregularized) sample partial correlation network on the BFI personality dataset from the psych package in R, followed by removing absolute edges below 0.05.

2010), it has not been validated in plausible scenarios for psychological networks. In addition, no simulation study has assessed the performance of using a polychoric correlation matrix in this methodology. To this end, this report presents a simulation study that assesses the performance in a plausible psychological network structure. In addition, the simulation study varied R and γ in order to provide recommendations of these parameters in estimating psychological networks. The simulation study makes use of the *qgraph* package, using the R codes described above.

Methods

To obtain a representative psychological network structure, the `bfi` dataset from the *psych* package (Revelle, 2010) was used on the Big 5 personality traits (Benet-Martinez & John, 1998; Digman, 1989; Goldberg, 1990a, 1993; McCrae & Costa, 1997). The `bfi` dataset consists of 2,800 observations of 25 personality inventory items. The network structure was obtained by computing the sample partial correlation coefficients (negative standardized inverse of the sample variance–covariance matrix; Lauritzen, 1996). Next, to create a sparse network all absolute edge weights below 0.05 were set to zero, thus removing edges from the network. Figure 2.6 shows the resulting network structure. In this network, 125 out of 300 possible edges were nonzero (41.6%). While this network is not the most appropriate network based on this dataset, it functions well as a proxy for psychological network structures as it is both sparse (has missing edges) and has parameter values that are not shrunken by the LASSO.

In the simulation study, data was generated based on the network of Figure 2.6. Following, the network was estimated using the the *qgraph* codes described above. Sample size was varied between 50, 100, 250, 500, 1,000, and 2,500, γ was varied between 0, 0.25, 0.5, 0.75, and 1, and R was varied between 0.001, 0.01 and 0.1. The data was either simulated to be multivariate normal, in which case Pearson correlations were used in estimation, or ordinal, in which case polychoric correlations were used in the estimation. Ordinal data was created by sampling four thresholds for every variable from the standard normal distribution, and next using these thresholds to cut each variable in five levels. To compute polychoric correlations, the `cor_auto` function was used, which uses the `lavCor` function of the *lavaan* package (Rosseel, 2012). The number of different λ values used in generating networks was set to 100 (the default in *qgraph*).

For each simulation, in addition to the correlation between estimated and true edge weights, the sensetivity and specificity were computed (van Borkulo et al., 2014). The *sensitivity*, also termed the true-positive rate, indicates the proportion of edges in the true network that were estimated to be nonzero:

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

Specificity, also termed the true negative rate, indicates the proportion of true missing edges that were also estimated to be missing:

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

When specificity is high, there are not many false positives (edges detected to be nonzero that are zero in the true network) in the estimated network.

Results

Each of the conditions was replicated 1,000 times, leading to 180,000 simulated datasets. Figure 2.7 shows the sensitivity of the analyses. This figure shows that sensitivity increases with sample size and is high for large sample sizes. When $\gamma > 0$, small sample sizes are likely to result in empty networks (no edges), indicating a sensitivity of 0. When ordinal data is used, small sample sizes (50 and 100) resulted in far too densely connected networks that are hard to interpret. Setting γ to be higher remediated this by estimating empty networks. At higher sample sizes, γ does not play a role and sensitivity is comparable in all conditions. Using $R = 0.1$ remediates the poor performance of polychoric correlations in lower sample sizes, but also creates an upper bound to sensitivity at higher sample sizes.

Figure 2.8 shows the specificity of the analyses, which was all-around high except for the lower sample sizes in ordinal data using $R = 0.01$ or $R = 0.001$. Some outliers indicate that fully connected networks were estimated in ordinal data even when setting $\gamma = 0.25$ in small sample sizes. In all other conditions specificity was comparably high, with higher γ values only performing slightly

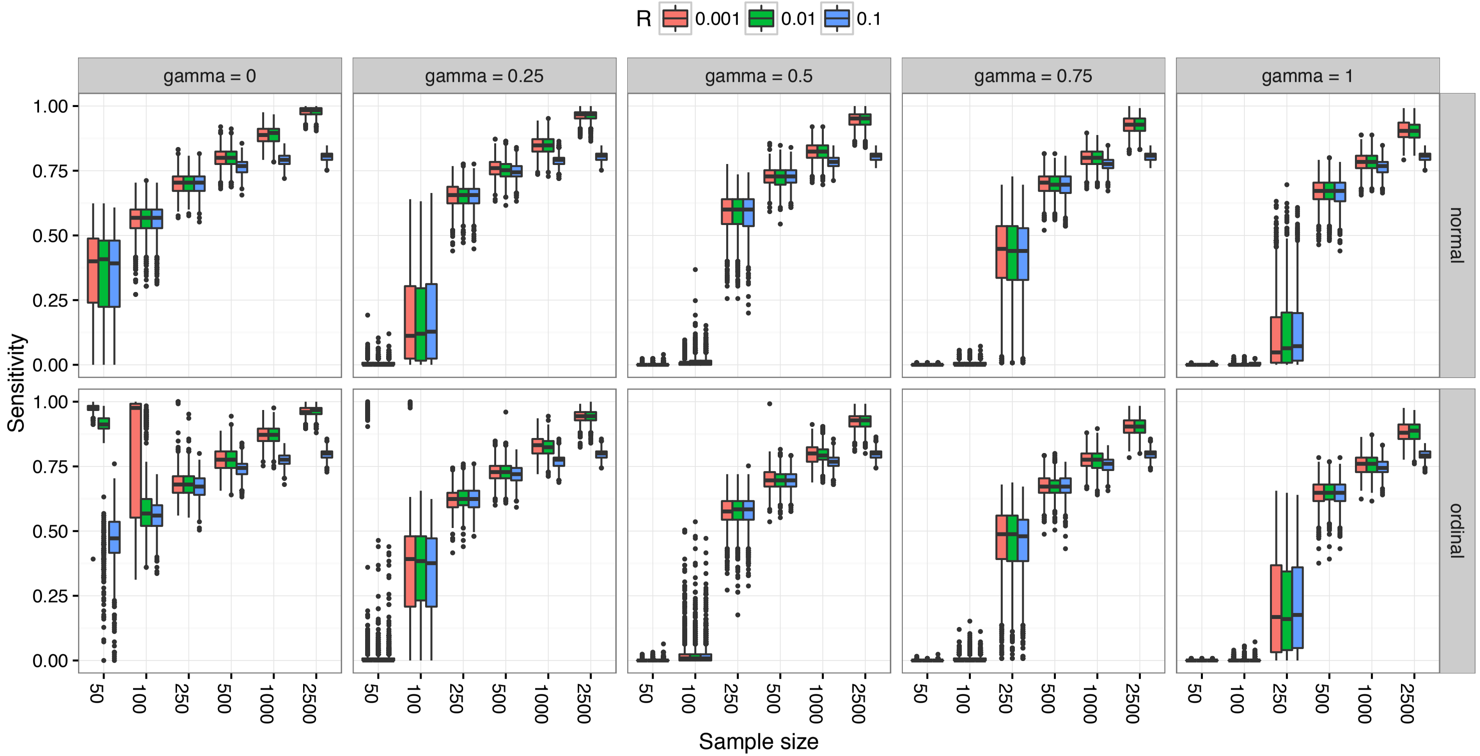


Figure 2.7: Sensitivity of the simulated datasets. Data is represented in standard boxplots. Horizontal panels indicate different EBIC hyperparameter values, vertical panels indicate if data was normal (Pearson correlations) or ordinal (polychoric correlations) and the color of the boxplots indicate the different ratio values used in setting the LASSO tuning parameter range. When sensitivity is high, true edges are likely to be detected.

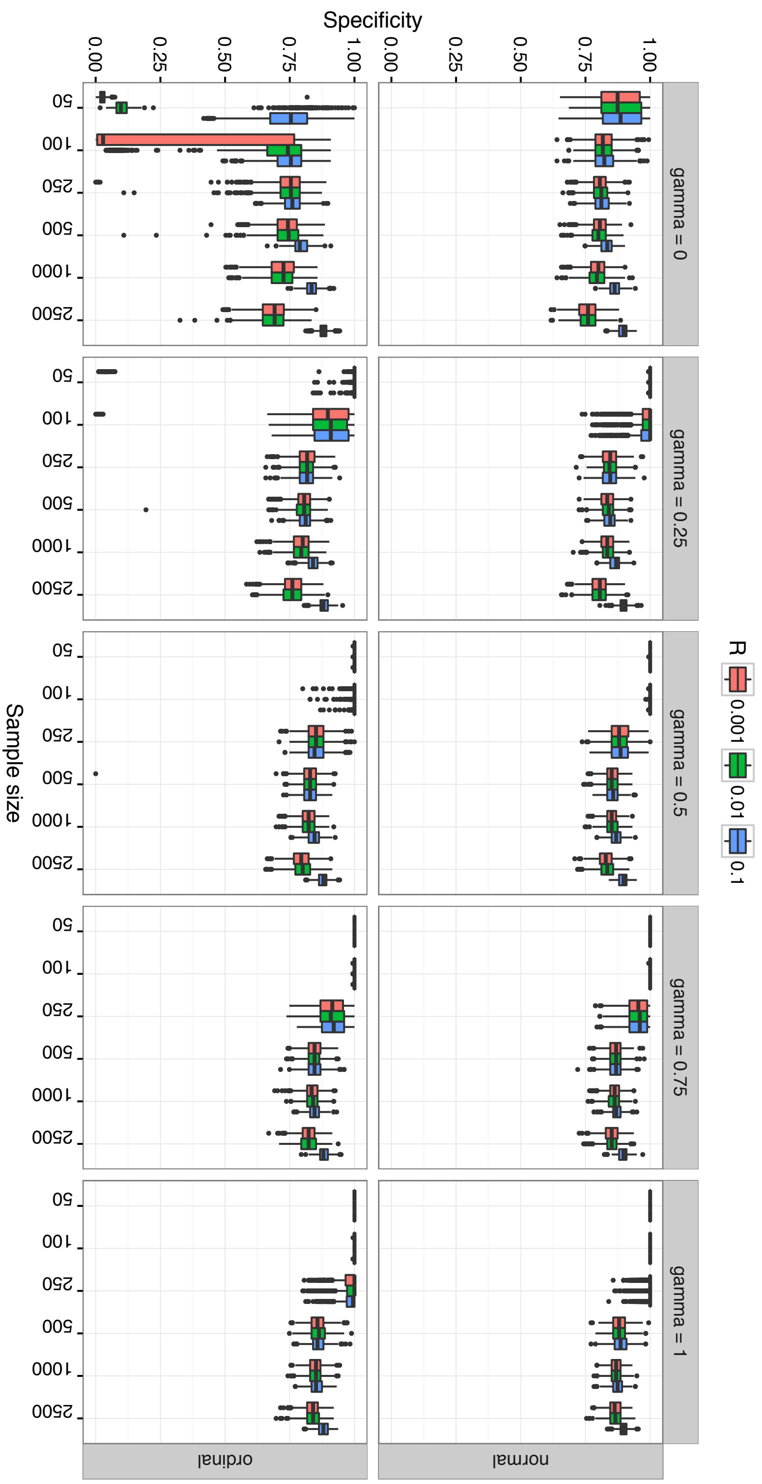


Figure 2.8: The specificity of the simulated datasets. When specificity is high, there are not many edges in the estimated network that are not present in the true network. See caption of Figure 2.7 for more details.

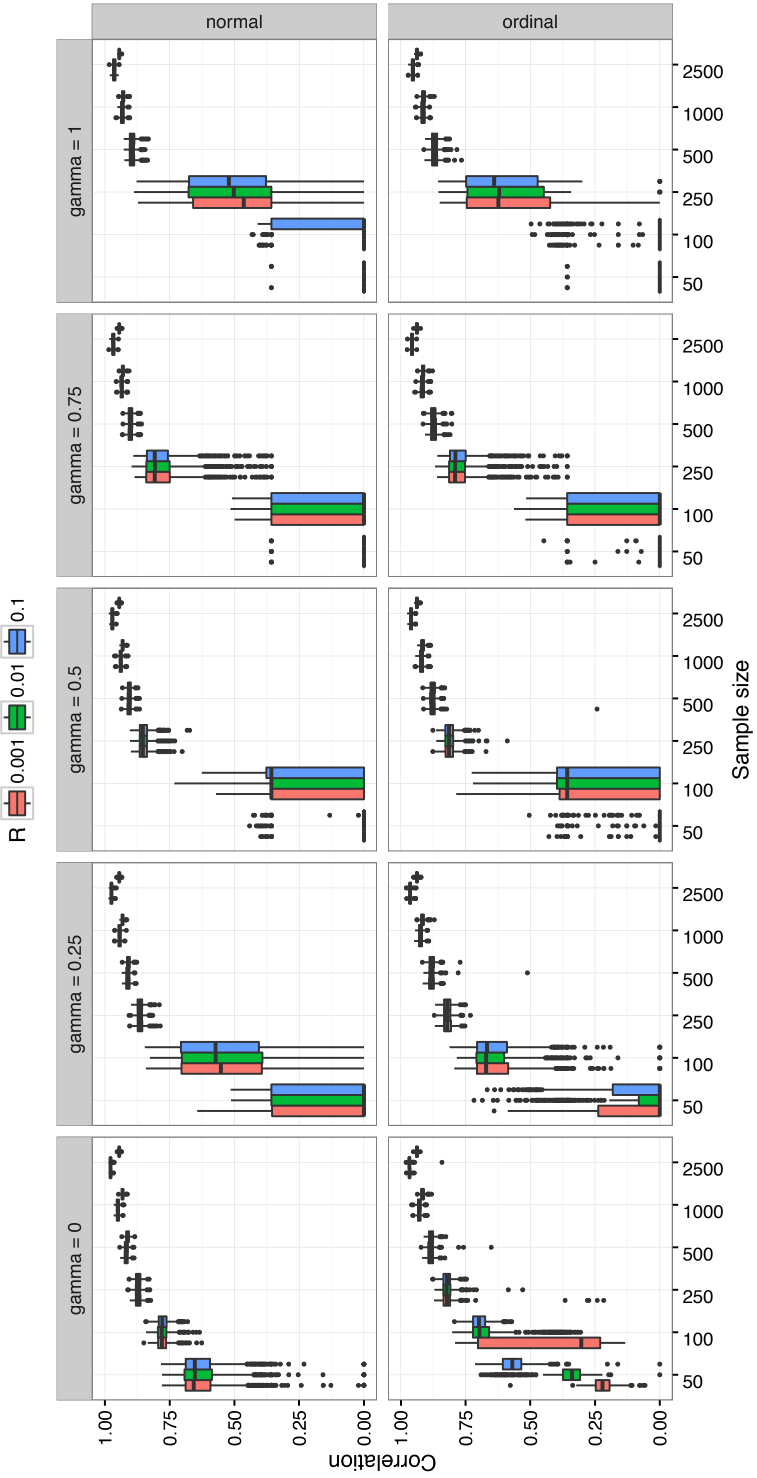


Figure 2.9: Correlation between true edge weights and estimated edge weights. See caption of Figure 2.7 for more details.

better. Figure 2.9 shows the correlation between true and estimated edge weights. This figure shows a comparable good performance from sample sizes of 250 and higher in all conditions, with γ values up to 0.5 outperforming the higher γ values. It should be noted that the correlation was set to zero if the estimated network had no edges (all edge weights were then zero).

2.7 Conclusion

This chapter presented tutorial on how to estimate psychological networks using a popular estimation technique: LASSO regularization with the EBIC model selection. The resulting network is a network of partial correlation coefficients controlled for spurious connections. One possibility to do so is provided by the *qgraph* R package that allows the estimation of network structure based on the correlation matrix of the data. The method also allows constructing partial correlation networks of ordered-categorical data by estimating the appropriate (in this case, polychoric) correlation matrix. The performance was assessed on 180,000 simulated datasets using a plausible psychological network structure. Results indicate that partial correlation networks could be well retrieved using either Pearson correlations or polychoric correlations. The default setup of *qgraph* uses $\gamma = 0.5$ and $R = 0.01$, which are shown to work well in all conditions. Setting $\gamma = 0.25$ improved the detection rate, but sometimes led to poorly estimated networks based on polychoric correlations. γ can be set to 0 to err more on the side of discovery (Dziak et al., 2012), but should be done with care in low sample polychoric correlation matrices. All conditions showed increasing sensitivity with sample size and a high specificity all-around. This is comparable to other network estimation techniques (van Borkulo et al., 2014), and shows that even though a network does not contain all true edges, the edges that are returned can usually be expected to be genuine. The high correlation furthermore indicated that the strongest true edges are usually estimated to be strong as well.

Many other estimation techniques exist. Regularized estimation of partial correlation networks can also be performed using the *huge* (Zhao et al., 2015) and *parcor* (Krämer et al., 2009) packages. When all variables are binary, one can estimate the Ising Model using, for instance, the *IsingFit* R package (van Borkulo & Epskamp, 2014). The resulting network has a similar interpretation as partial correlation networks, and is also estimated using LASSO with EBIC model selection (van Borkulo et al., 2014). When the data consist of both categorical and continuous variables, a state-of-the-art methodology is implemented in the *mgm* package (Haslbeck & Waldorp, 2016a) also making use of LASSO estimation with EBIC model selection. The *bootnet* package can subsequently be used to assess the accuracy of the estimated network structure obtained via *qgraph* or any of the other packages mentioned above (see also Chapter 3).

Important to note is that the methods described in this chapter are only appropriate to use when the cases in the data (the rows of the spreadsheet) can reasonably be assumed to be independent of one-another. Such is the case in cross-sectional analysis—where cases represent people that are measured only once—but not in longitudinal data where one person is measured on several occasions. In this case, temporal information needs to be taken into account when estimating network structures. One way to do so is by using the *graphical vector-autoregression* model (graphical VAR; Wild et al., 2010). LASSO regularization making use of glasso in an iterative algorithm has been developed to estimate the network structures (Abegaz & Wit, 2013; Rothman, Levina, & Zhu, 2010). EBIC model selection using these routines has been implemented in the R packages *sparseTSCGM* (Abegaz & Wit, 2015; aimed at estimating genetic networks) and *graphicalVAR* (Epskamp, 2015; aimed at estimating $n = 1$ psychological networks).

In conclusion, while psychological network analysis is a novel field that is rapidly changing and developing, we have not seen an accessible description of the most commonly used estimation procedure in the literature: LASSO regularization using EBIC model selection to estimate a sparse partial correlation network. This chapter aimed to provide a short overview of this common and promising method.