CrossMark

How well do network models predict observations?
On the importance of predictability in network models

Jonas M. B. Haslbeck[1] · Lourens J. Waldorp[1]

Abstract Network models are an increasingly popular way to abstract complex psychological phenomena. While studying the *structure* of network models has led to many important insights, little attention has been paid to how well they *predict* observations. This is despite the fact that predictability is crucial for judging the *practical relevance* of edges: for instance in clinical practice, predictability of a symptom indicates whether an intervention on that symptom through the symptom network is promising. We close this methodological gap by introducing nodewise predictability, which quantifies how well a given node can be predicted by all other nodes it is connected to in the network. In addition, we provide fully reproducible code examples of how to compute and visualize nodewise predictability both for cross-sectional and time series data.

Keywords Network models · Network analysis · Predictability · Clinical relevance

Introduction

Network models graphically describe interactions between a potentially large number variables: each variable is represented as a dot (node) and interactions are represented by lines (edges) connecting the nodes (for an illustration see

Fig. 1a). These models have been a popular way to abstract complex systems in a large variety of disciplines such as statistical mechanics (Albert & Barabasi, 2002), biology (Friedman, Linial, Nachman, & Pe'er, 2000), neuroscience (Huang et al., 2010), and are recently also applied in psychology (Costantini et al., 2015) and psychiatry (Borsboom & Cramer, 2013).

Particularly in psychology, network models are attractive because many psychological phenomena are considered to depend on a large number of variables and interactions between them. In this situation, the graphical representation allows an intuitive interpretation even if the number of variables is large. In addition, network models open up the possibility to study the network structure: for instance, one can use network summary measures like density or centrality to describe the global structure of the network (Newman, 2010). These could allow inferences about the behavior of the whole network that would not be possible when looking at all edge parameters separately. Another possibility is to run generative models on the network, e.g., diffusion models of diseases to explain how symptoms of psychological disorders activate each other (Shulgin, Stone & Agur, 1998).

Currently, most applications are in the field of clinical psychology (e.g., Fried et al., 2015; Fried, Epskamp, Nesse, Tuerlinckx, & Borsboom, 2016; Beard et al., 2016; McNally et al., 2015; Boschloo et al., 2015) but network models are also applied in health psychology (Kossakowski, Epskamp, et al., 2016) and personality psychology (Cramer et al., 2012; Costantini et al., 2015). While initially they were used to model cross-sectional data, there is increasing interest in analyzing data obtained using the experience sampling method (ESM), which consists of repeated measurements of the same person (e.g., Bringmann et al., 2013; Pe et al., 2015). The focus in these papers is the global network structure and the connectedness of specific nodes

✉ Jonas M. B. Haslbeck
  jonashaslbeck@gmail.com
  https://jmbh.github.io/

[1] Psychological Methods, University of Amsterdam, Nieuwe Achtergracht 129-B, 1018 WT, Amsterdam, Netherlands
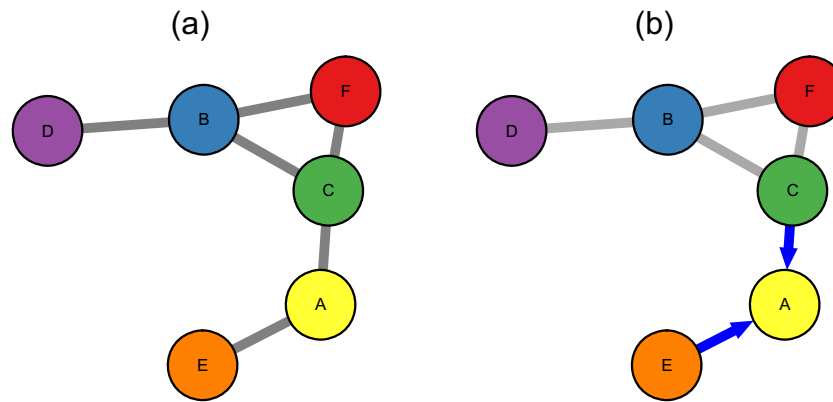
(a) (b)

Fig. 1 a Example network with six nodes. An edge between two nodes indicates a pairwise interaction between those two nodes. **b** Illustration of predicting node A by all its neighboring nodes (C and E)

in the network, which provide a new perspective on many psychological phenomena. For instance, Cramer and colleagues (Cramer et al., 2010) suggested an alternative view on the concept of comorbidity by analyzing how symptoms of different psychological disorders are connected to each other.

The key idea of this paper is to analyze the *predictability* of nodes in the network *in addition* to the network structure. By predictability of node A we mean how well node A can be predicted by all its neighboring nodes in the network (see Fig. 1b). The predictability of nodes is important for several reasons:

1. The edges connected to node A should be interpreted taking into account how much of the variance of A is explained by the edges connected to A. For instance, edges will be interpreted differently, depending on whether 0.5% or 50% of the variance of A is explained. This issue is particularly important for networks estimated on a large number of observations, where small edge weights can be detected that might be practically meaningless.
2. In many areas of psychology, the goal is to design effective interventions. Using the predictability measure of node A, one can estimate to which extent we can influence A by intervening on nodes that are connected to it.
3. Predictability across nodes tells us whether a (part of a) network is largely determined by itself through strong mutual interactions between nodes (high predictability) or whether it is mostly determined by other factors that are not included in the network (low predictability).

The problem addressed here is similar to the problem of modeling only the covariance matrix in structural equation modeling (SEM) (Byrne, 2013): one might find a model that perfectly fits the covariance matrix, but if the variance

of variables is much larger than their covariance, the model might be meaningless in practice.

Predictability in general cannot be inferred by the network structure but has to be computed from the network model and the data. Unfortunately, currently there is no easy-to-use tool available to researchers to compute and present predictability in network models. In the present paper, we close this methodological gap by making the following contributions:

1. We present a method to compute easy-to-interpret nodewise predictability measures for state-of-the-art network models ("Methods").
2. We provide a step-by-step description of how to use the R-packages *mgm* and *qgraph* to compute and visualize nodewise predictability, both for cross-sectional ("Predictability in cross-sectional networks") and time-series networks ("Predictability in temporal networks"). The provided code is fully reproducible, which means that the reader can run the code and reproduce all figures while reading. The data in our applications are from two published studies and will be downloaded automatically with the provided code.

Methods

In order to determine the predictability of a given node A, we need to know which nodes are connected to A in the network model. Therefore the first step is to estimate a network model, which we describe in "Network models". In a second step, we use the network model to predict the given node A by the nodes that are connected to it (its neighbors). In "Making predictions", we describe in detail how to compute these predictions. Finally, we quantify how close these predictions are to the actual values of A. The closer the predictions are to the actual values, the higher the predictability

of A. A description of predictability measures for both continuous and categorical variables is given in "Quantifying predictability". In "Predictability and model parameters" we discuss the relationship between the predictability and the parameters of the network model. Finally we describe the data "Application to datasets" that is used in the application examples in "Predictability in cross-sectional networks" and "Predictability in temporal networks".

Network models

We model cross-sectional data using pairwise Mixed Graphical Models (MGMs) (Yang, Baker, Ravikumar, Allen, & Liu, 2014; Haslbeck & Waldorp, 2015b), which generalize well-known exponential family distributions such as the multivariate Gaussian distribution or the Ising model (Wainwright & Jordan, 2008). This is the model used in all papers mentioned in the introduction. MGMs are estimated via ℓ_1-regularized (LASSO) neighborhood regression as implemented in the R-package mgm by the authors (Haslbeck & Waldorp, 2015a). In this approach, one estimates the neighborhood of each node and combines all neighborhoods to obtain the complete graph (network) (Meinshausen & Bühlmann, 2006). The *neighborhood* of a node is the set of nodes that is connected to that node. For example, in Fig. 1a, the neighborhood of node A consists of the nodes E and C. The ℓ_1 regularization ensures that spurious edge-parameters are put to exactly zero, which makes the network model easier to interpret. The parameter that controls the strength of the regularization is selected via 10-fold cross validation.

For time-series data, we use the Vector Autoregressive (VAR) model, which is a popular model for multivariate time series in many disciplines (see e.g., Hamilton, 1994; Pfaff, 2008). The VAR model is different from the MGM in that associations are now defined between time-lagged variables. Specifically, in its simplest form with a time-lag of order one, in this model all variables X_i^{t-1} at time $t-1$ are regressed on *each* of the variables X_i^t at time t, where i indexes different variables. Note that this also includes the variable X_s itself at an earlier time point: that is, one predicts X_s^t at time t by itself and all other variables at time $t-1$. For the analyses in this paper, we use the implementation of mixed VAR models in the R-package mgm (Haslbeck & Waldorp, 2015a).

Making predictions

We are interested in how well a node can be predicted by all adjacent nodes in the network. This means that we would like to compute the mean of the conditional distribution of the node at hand given all its neighbors. We illustrate this by showing how to compute predictability for the node A in

Fig. 1b, for (i) the case of A being a continuous-Gaussian variable and (ii) the case of A being binary.

We begin with (i): the conditional mean of A given its neighbors C and E, which is given by

$$P(A = x | C, E) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \qquad (1)$$

where the mean $\mu = \beta_0 + \beta_C C + \beta_E E$ is a linear combination of the two neighbors C and E. This conditional distribution is obtained from the multivariate exponential family distribution of the MGM. For details see Yang et al. (2014) and Haslbeck and Waldorp (2015b). This prediction problem corresponds to the familiar linear regression problem with Gaussian noise. Now, how can one make predictions? Let's say the intercept is $\beta_0 = 0.25$ and $\beta_C = 0.1$, $\beta_E = -0.5$. Then, if the ith case in the sample is $C_i = 2, E_i = 1$, then for the ith sample of A we predict $A_i = 0.25 + 0.1 \times 2 - 0.5 \times 1 = -0.05$. A measure of predictability should evaluate how close this is the actual observation for node A_i.

In example (ii), where A is categorical, we compute a predicted probability for each category using a multinomial distribution

$$P(A = k | C, E) = \frac{\exp\{\mu_k\}}{\sum_{l=1}^{K} \exp\{\mu_k\}}, \qquad (2)$$

where k indicates the category, K is the number of categories, and $\mu_k = \beta_{0k} + \beta_{Ck} C + \beta_{Ek} E$. Now let's assume A is binary ($K = 2$) and we have $\beta_{01} = 0, \beta_{C1} = 0.5, \beta_{E1} = 1$ and $\beta_{02} = 0, \beta_{C2} = -0.5, \beta_{E2} = -1$ and if for the ith cases we have $C_i = 1$ and $E_i = 1$. When filling in the numbers in Eq. (2) we get $P(A = 1 | C, E) \approx 0.95$ and $P(A = 2 | C, E) \approx 0.05$, and predict category $k = 1$ for the ith sample of A, because $0.95 > \frac{1}{2}$. Of course, all probabilities have to add up to 1, so we have $1 - P(A = 1 | C, E) = P(F = 2 | C, E)$. This direct approach of modeling the probabilities of categories is possible due to the regularization used in estimation (see e.g., Hastie, Tibshirani, & Wainwright, 2015), otherwise this model would not be identified. Note that predicting A by all its neighbors is the same as predicting A by all nodes in the network. This is because all nodes that are *not* in the neighborhood of A have a zero weight associated to them in the regression equation on A in (1) or (2) and can hence be dropped.

In the case of other exponential family distributions, such as Poisson or exponential, one similarly uses the univariate conditional distribution to make predictions (Yang et al., 2014). Importantly, the joint distribution of the MGM can be represented as a factorization of p conditional distributions and hence our method to compute predictions is based on a proper representation of the joint distribution. Indeed, this factorization is used when estimating the MGM in the neighborhood regression approach (see "Network models").

Quantifying predictability

After computing predictions, we would like to know how close these are to the observed values in the data. Because it is of interest how well a given node can be predicted by all *other* nodes in the network, we need to remove any effects of the intercept (continuous variables) and the marginal (categorical variables). The marginal indicates the probabilities of categories, when ignoring all other variables. For example, the marginal of a binary variable is described by relative frequency of observing category 1, e.g., $P(X = 1) = 0.7$.

Predictability of continuous variables

For continuous data, we choose the proportion of explained variance as predictability measure since it is well known in the literature and easy to interpret:

$$R_A^2 = 1 - \frac{var(\hat{A} - A)}{var(A)},$$

where var is the variance, $\hat{A}$ is a vector of predictions for A as described in "Making predictions", and A is the vector of observed values in the data. In order to remove any influences of the intercepts, all variables are centered to mean zero. Hence, all intercepts will be zero and cannot affect the predictability measure. Thus, we can interpret R^2 as follows: a value of 0 means that a node cannot be predicted at all by its neighboring nodes in the network, whereas a value of 1 means that a node can be perfectly predicted by its neighboring nodes.

Predictability of categorical variables

For categorical variables, it is slightly more difficult to get a measure with the same interpretation as the R^2 for continuous variables because there is no way to center categorical variables. The following example shows that it is, however, important to somehow take the marginal into account: let's say we have 100 observations of a binary variable A and observe 10 0s and 90 1s. This means that the marginal probabilities of A are $p_0 = 0.1$ and $p_1 = 0.9$. Now, if all other nodes contribute nothing to predicting whether there is a 0 or 1 present in case A_i, one can just predict a 1 for all cases and get a proportion of correct classification (or accuracy, see below) of 90%. For our purpose of determining how well a node can be predicted by all other nodes, this is clearly misleading, because actually *nothing* is predicted by all other nodes. We therefore compute a *normalized accuracy* that removes the accuracy that is achieved by the trivial prediction using marginal of the variable ($p_1 = 0.9$) alone.

Let $\mathcal{A} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y_i = \hat{y}_i)$ be the proportion of correct predictions (or accuracy) and let $p_0, p_1, \ldots p_m$ be the marginal probabilities of the categories, where $\mathbb{I}$ is the indicator function for the event $F_i = \hat{F}_i$. In the binary case, the latter are p_0 and $p_1 = 1 - p_0$. We then define normalized accuracy as

$$\mathcal{A}_{norm} = \frac{\mathcal{A} - \max\{p_0, p_1, \ldots, p_m\}}{1 - \max\{p_0, p_1, \ldots, p_m\}}.$$

Hence, $\mathcal{A}_{norm}$ indicates how much the node at hand can be predicted by all other nodes in the network, *beyond* what is trivially predicted by the marginal distribution. $\mathcal{A}_{norm} = 0$ means that none of the other nodes adds anything to the marginal in predicting the node at hand, while $\mathcal{A}_{norm} = 1$ means that all other nodes perfectly predict the node at hand (together with the marginal).

Let's return to the above example: in contrast to the high accuracy of $\mathcal{A} = 0.9$, the normalized accuracy $\mathcal{A}_{norm}$ is zero, indicating that the node at hand cannot be predicted by other nodes in the network. However, notice that both $\mathcal{A}$ and $\mathcal{A}_{norm}$ are important for interpretation. For instance, if we have a marginal of $p_1 = .9$ in a binary variable, then it is less impressive if all other predictors account for 80% of the remaining accuracy that can be achieved (.98 instead of .9) than in a situation where $p_1 = .5$, where accounting 80% of the remaining accuracy would mean an improvement from .5 to .9. We therefore visualize both $\mathcal{A}$ and $\mathcal{A}_{norm}$ for the binary variable in Fig. 2.

Predictability and model parameters

Given the above definition of measures of predictability, it is evident that there is a close relationship between the parameters of the network model and predictability: if a node is not connected to any other node, then the explained variance/ normalized accuracy of this node *has* to be 0. Also, the more edges are connected to a node, the higher predictability tends to be. There is a strong linear relationship between predictability and edge parameters for Gaussian graphical models (GGM), where the edge parameters (partial correlation) are restricted to $[-1, 1]$. This linear relationship is much weaker for models including categorical variables, where the model parameters are only constrained to be finite.

This implies that centrality measures (like degree centrality), which are a function of edge parameters, are also strongly correlated with predictability for GGMs, but much less for MGMs (e.g., Haslbeck & Fried, 2017). However, note that even if a given centrality measure would correlate perfectly with predictability, it would not be a substitute, because it would only allow us to order nodes by predictability but would *not* tell us the predictability of any node. Hence, while centrality measures are related to predictability, they are not a good proxy for predictability.

Application to datasets

We illustrate how to compute and visualize nodewise predictability for network models for both cross-sectional and time-series data. We use a cross-sectional dataset from Fried et al. (2015) ($N = 515$) with 11 variables on the relationship on bereavement and depressive symptoms. In order to illustrate how to compute predictability for VAR models we use a dataset consisting of up to ten daily measurements of nine variables related to mood over a long period of time ($N = 1478$) of a single individual (Wichers, Groot, Psychosystems, & Group, 2016). A detailed description of the time-series data can be found in Kossakowski, Groot, Haslbeck, Borsboom, and Wichers (2016).

Predictability in cross-sectional networks

Here we show how to obtain the proposed predictability measures using the *mgm* package. We will give the code below so all steps can be reproduced exactly by the reader.

First, we download the preprocessed data. The raw data and the preprocessing file can be found in the Github repository https://github.com/jmbh/NetworkPrediction.

```
library(httr)
url="https://tinyurl.com/y8cqhb9c"
GET(url, write_disk("Fried2015.RDS"))
dlist <- readRDS("Fried2015.RDS")
```

Next, we fit a MGM using the mgm-package:

```
library(mgm)
set.seed(1)
fit_obj <- mgm(data = dlist$data,
               type = c(rep("g", 11), "c"),
               level = c(rep(1, 11), 2),
               ruleReg = "OR",
               k = 2,
               binarySign = TRUE)
```

In addition to the data, one has to specify the type and the number of categories for each variable. The remaining arguments are tuning parameters and are selected such that the original results in Fried et al. (2015) are reproduced. For the general usage of the *mgm* package, see Haslbeck and Waldorp (2015a). After estimating the model, which is saved in `fit_obj`, we use the `predict()` function to compute the predictability for each node in the network. For categorical variables, we specify the predictability measures accuracy/correct classification (`"CC"`) and normalized accuracy (`"nCC"`). In addition, we request the accuracy of the intercept (marginal) model (`"CCmarg"`), which we will use to visualize the decomposition of the total accuracy in

intercept model and the contribution of other variables. For continuous variables, we specify explained variance (`"R2"`) as predictability measure.

```
p_obj <- predict(fit_obj, dlist$data,
                 errorCat = c("CC","nCC","CCmarg"),
                 errorCon = c("R2"))
```

To display both the accuracy of the intercept model and the normalized accuracy (contribution by other variables), we require a list for the ring-segments and a list for the corresponding colors:

```
error_list <- list() # List for ring-segments
for(i in 1:11)  error_list[[i]] <- p_obj$errors[i,2]
beyondmarg <- p_obj$errors[12,3]-p_obj$errors[12,5]
error_list[[12]] <- c(p_obj$errors[12,5],beyondmarg)

color_list <- list() # List for Colors
for(i in 1:11) color_list[[i]] <- "#90B4D4"
color_list[[12]] <- c("#ffa500", "#ff4300")
```

We now provide the weighted adjacency matrix and the list containing the nodewise predictability measures to qgraph, resulting in Fig. 2:

```
library(qgraph)
set.seed(1)
qgraph(fit_obj$pairwise$wadj, pie = error_list,
       layout="spring", labels = dlist$names,
       pieColor = color_list, label.cex = .9,
       edge.color = fit_obj$pairwise$edgecolor,
       curveAll = TRUE, curveDefault = .6,
       cut = 0, labels = dlist$names)
```

The color of the ring around the node can be controlled using the `pieColor` argument. The remaining arguments are not necessary but improve the visualization. `layout="spring"` specifies that the placement of the nodes in the visualization is determined by the force-directed Fruchterman–Reingold algorithm (Fruchterman & Reingold, 1991). Note that there is no analytic relation between the distance of nodes in the plotted layout and model parameters, however, the algorithm tends to group strongly connected nodes together in order to avoid edge crossings. Green and red edges indicate positive and negative relationships, respectively, and the width of the edges is proportional to the absolute value of the edge-weight. For a detailed description of the qgraph package, see Epskamp et al. (2012).

This code returns a network that is very similar to the one in the original paper (Fried et al., 2015). Note that the network is not identical as we did not dichotomize ordinal variables but treat them as continuous instead. For the 11 continuous variables, the percentage of explained variance is indicated by the blue part of the ring. For the single binary variable, the colors in the ring indicate the accuracy of the intercept model (orange) and the full accuracy (orange

+ red). The normalized accuracy is the ratio red / (red + white).

As expected, nodes with more/stronger edges can be predicted better (e.g., *lonely*) than nodes with fewer/weaker edges (e.g., unfriendly *unfr*). While this trivially follows from the construction of the predictability measure (see "Predictability and model parameters"), this does not mean that one can use the network structure to infer the predictability of a node: by looking at the network visualization in Fig. 2, we are quite certain that predictability of *lonely* is higher than of *unfr*. However, we do not know *how* high predictability is in either of the two nodes (0.55 and 0.13, respectively), which is highly relevant for interpretation and practical applications.

Because we used the same data for estimating the network and calculating the predictability (or error) measures, we estimated the *within sample* prediction error. In order to see how well the model *generalizes*, one has to calculate the *out of sample* prediction error. This can be done by splitting the data into two parts (or using a cross-validation scheme) and providing one part to the estimation function, and the other part to the prediction function.

Predictability in temporal networks

In this section we show how to compute nodewise predictability measures for VAR models. Note that the interpretation of predictability is slightly different for VAR networks because we predict each node by *all* nodes at the previous time point, which also includes the predicted node itself.

We begin again by downloading the example dataset:

```
url="https://tinyurl.com/yau3xms2"
GET(url, write_disk("Wicherts2016_Mood.RDS"))
dlist_ts <- readRDS("Wicherts2016_Mood.RDS")
```

Next, we provide the data and the type and number of categories of variables as input. In addition, we specify that we would like to estimate a VAR model with lag 1

```
set.seed(1)
var_obj <- mvar(data = dlist_ts$data_mood,
                type = rep("g", 9),
                lev = rep(1, 9),
                lags = 1,
                consec=dlist_ts$data_time$beepno)
```

and compute the predictability of each node similarly to above:

```
p_obj2 <- predict(var_obj, dlist_ts$data_mood,
                  errorCon = c("R2"))
```

Finally, we visualize the network structure together with the nodewise predictability measures, which results in

Fig. 3. Because we have only one predictability measure for each node, we can provide them in a vector via the `pie` argument:

```
set.seed(1)
qgraph(var_obj$wadj[,,1],
       edge.color = var_obj$edgecolor[,,1],
       labels = dlist_ts$labels,
       pie = p_obj2$errors[, 2],
       pieColor = rep("#90B4D4", 9),
       curveAll = TRUE,
       curveDefault = .6, cut = 0)
```

We see two groups of self-engaging mood variables in Fig. 3: (a) the positive mood variables *Cheerful*, *Enthusiastic* and *Satisfied* and (b) the negative mood variables *Irritated*, *Agitated*, *Restless* and *Suspicious*. *Worrying* seems to be influenced by both groups and *Relaxed* is rather disconnected and has a weak negative influence on group (b). These insights can be used to judge the effectiveness of possible interventions on these mood variables: for instance, if the goal is to change variables in group (a), one can do this by intervening on other variables in (a). In addition, we would expect an effect on *Worrying* when intervening on groups (a) and (b), however, the reverse is not true. *Relaxed*
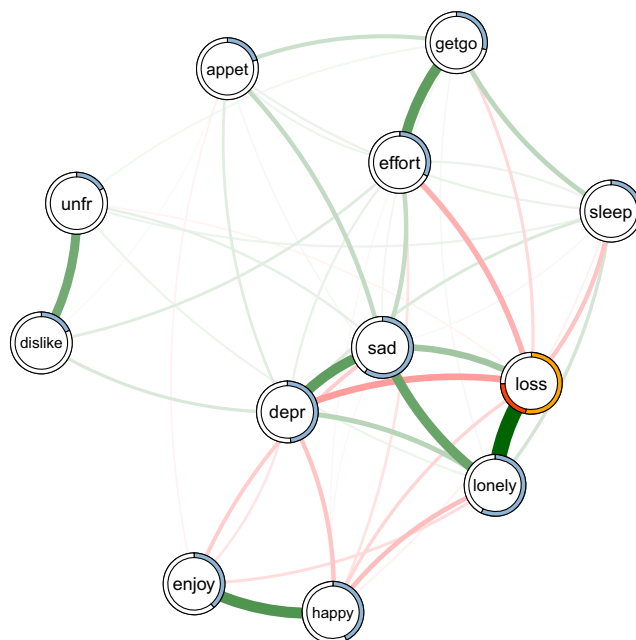

Fig. 2 Mixed graphical model estimated on the data from Fried et al. (2015). *Green edges* indicate positive relationships and *red edges* indicate negative relationships. The *blue ring* shows the proportion of explained variance (for continuous nodes). For the binary variable "loss", the orange part of the ring indicates the accuracy of the intercept model. The *red part of the ring* is the *additional* accuracy achieved by all remaining variables. The sum of both is the accuracy of the full model $\mathcal{A}$. The normalized accuracy $\mathcal{A}_{norm}$ is the ratio between the additional accuracy due to the remaining variables (*red*) and one minus the accuracy of the intercept model (*white + red*)

has a small influence on group (b), however, is itself not influenced by any of the variables in the network. Hence, in order to intervene on *Relaxed*, one has to search for additional variables influencing *Relaxed* that were not yet taken into account in the present network.

Discussion

In this paper, we introduced a method and easy-to-use software to compute nodewise predictability in network models and to visualize it in a typical network visualization. Predictability is an important concept that *complements* the network structure when interpreting network models: it gives a measure of how well a node can be predicted by all its neighboring nodes and is hence crucial information whenever one needs to judge the practical significance of a set of edges. An example is clinical practice, where it is important to make predictions of the outcome of interventions on an interpretable scale to optimally select treatments.

The analyses shown in the present paper can be extended to networks that are changing over time, which allows to investigate how edge-parameters and nodewise predictability change over time. The time-varying parameters can then be modeled by a second model, which could include variables from inside and outside the time-varying network. With this modeling approach, it would be possible to gather
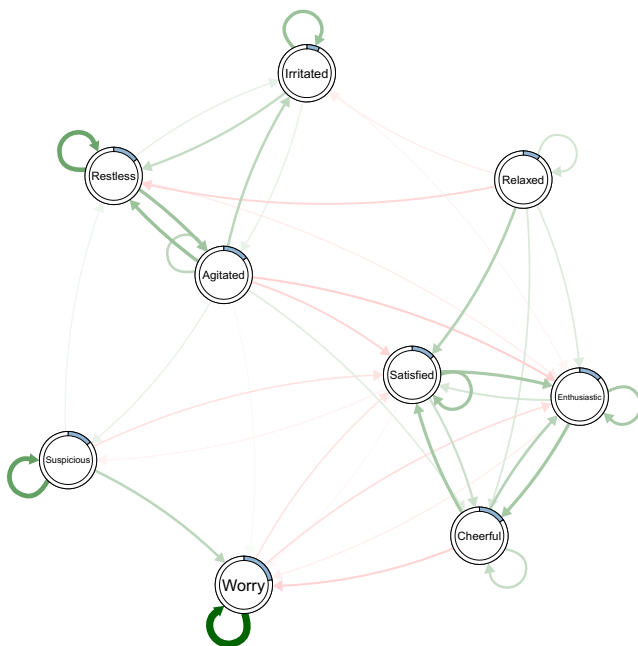

Fig. 3 Visualization of VAR network of the mood variables in Wichers et al. (2016). *Green edges* indicate positive relationships, *red edges* indicate negative relationships. The self-loops refer to the effect of the variable on itself over one time lag. The *blue rings* around the nodes indicate the proportion of explained variance in that node by all other nodes

evidence for the event of one (or several) variables causing the system to transition into another state, which is possibly reflected by a different network structure and nodewise predictability. For details about how to fit time-varying network models and time-varying predictability measures, see (Haslbeck & Waldorp, 2015a).

It is important to be clear about the limitations of interpreting nodewise predictability. First, we can only interpret the predictability of a node as the influence of its neighboring nodes if the network model is an appropriate model. A network model can be inappropriate for a number of reasons:

1. Two or more variables in the network models are caused by a variable that is not included in the network. This results in estimated edges between these variables in the network, even though they are only related via an unobserved common cause. In this situation, we cannot interpret predictability as influence by neighboring nodes because we know that the nodes are not influencing each other but are caused by a third variable outside the network.

2. In some situations, variables are logically dependent, for instance *age* and *age of diagnosis* are always related, because one cannot be diagnosed before being born. Clearly, in this situation the relation between the variables must be interpreted differently.

3. If two or more variables measure the same underlying construct (e.g., five questions about sad mood). In this situation, the edge-parameters indicate how similar the variables are and do not reflect mutual causal influence. Consequently, we would not interpret the predictability of these variables as the degree of determination by neighboring nodes. See Fried and Cramer (2016) for a discussion of this problem. Solutions could be to determine the topological overlap (Zhang et al., 2005) and choose only one variable in case of large overlap or to incorporate measurement models into the network model (Epskamp, Rhemtulla, & Borsboom, 2016).

Second, if we interpret the predictability of node A as a measure of how much it is determined by its neighbors, we assumed that all edges are directed towards node A. However, the direction of edges is generally unknown when the model is estimated from cross-sectional data. Estimates about the direction of edges can be made using causal search algorithms like the PC algorithm (Spirtes, Glymour, & Scheines 2000) or by using substantive theory. This means that the predictability of a node is an *upper bound* and in practice often lower because some edges might be bi-directional or point away from the node at hand. While this is a major limitation, note that the direction of edges is unknown for any model estimated on cross-sectional data. In models with lagged predictors, like the VAR model, this

problem does not exist because we use the direction of time to determine the direction of edges.

Finally, it is important to stress that a topic we did *not* cover here is to investigate how well node A can be predicted *by node B*. This is different from the problem studied in this paper, where the interest was in how well node A can be predicted *by all other nodes*. Unfortunately, there are no straightforward solutions for the former problem in the situation of correlated predictors, which is always the case in practice. For linear regression, there is work on decomposing explained variance (Grömping, 2012) and in the machine-learning literature there are methods to determine variable importance by replacing predictor variables by noise and investigate the drop in predictability (e.g., Breiman et al., 2001). It would certainly be interesting to try to extend these ideas to the general class of network models.

To sum up, if the network model is an appropriate model for the phenomena at hand, predictability is an easy-to-interpret measure of how strongly a given node is influenced by its neighbors in the network. This allows researchers to judge the practical relevance of edges connected to a node A on an absolute scale (0 = no influence on A at all, 1 = A fully determined) and thereby may help to predict intervention outcomes. In addition, the predictability of (parts of) the network is interesting on a theoretical level, as it indicates how self-determined the network is.

References

Albert, R., & Barabasi, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*(1), 47. http://journals.aps.org/rmp/abstract/10.1103/RevModPhys.74.47.

Beard, C., Millner, A., Forgeard, M., Fried, E., Hsu, K., Treadway, M., Leonard, C., Kertz, S., & Björgvinsson, T. (2016). Network analysis of depression and anxiety symptom relationships in a psychiatric sample. Psychological Medicine, 1–11.

Borsboom, D., & Cramer, A. O. (2013). Network analysis: An integrative approach to the structure of psychopathology. *Annual Review of Clinical Psychology*, *9*(1), 91–121. doi:10.1146/annurev-clinpsy-050212-185608.

Boschloo, L., van Borkulo, C. D., Rhemtulla, M., Keyes, K. M., Borsboom, D., & Schoevers, R. A. (2015). The network structure of symptoms of the diagnostic and statistical manual of mental disorders. *PloS One*, *10*(9), e0137–621.

Breiman, L. et al. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

Bringmann, L. F., Vissers, N., Wichers, M., Geschwind, N., Kuppens, P., Peeters, F., Borsboom, D., & Tuerlinckx, F. (2013). A network approach to psychopathology: New insights into clinical longitudinal data. *PloS One*, *8*(4), e60–188.

Byrne, B. M. (2013). Structural equation modeling with Mplus: Basic concepts, applications, and programming. Routledge.

Costantini, G., Epskamp, S., Borsboom, D., Perugini, M., Mõttus, R., Waldorp, L. J., & Cramer, A. O. (2015). State of the art personality research: A tutorial on network analysis of personality data in R. *Journal of Research in Personality*, *54*, 13–29.

Cramer, A. O., Sluis, S., Noordhof, A., Wichers, M., Geschwind, N., Aggen, S. H., Kendler, K. S., & Borsboom, D. (2012). Dimensions of normal personality as networks in search of equilibrium: You can't like parties if you don't like people. *European Journal of Personality*, *26*(4), 414–431.

Cramer, A. O. J., Waldorp, L. J., van der Maas, H. L. J., & Borsboom, D. (2010). Comorbidity: A network perspective. *Behavioral and Brain Sciences*, *33*, 137–193.

Epskamp, S., Cramer, A. O., Waldorp, L. J., Schmittmann, V. D., Borsboom, D., & et al. (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, *48*(4), 1–18.

Epskamp, S., Rhemtulla, M., & Borsboom, D. (2016). Generalized network psychometrics: Combining network and latent variable models. arXiv:160509288.

Foygel, R., & Drton, M. (2014). High-dimensional Ising model selection with Bayesian information criteria. arXiv:1403.3374.

Fried, E. I., & Cramer, A. O. J. (2016). Moving forward: Challenges and directions for psychopathological network theory and methodology. doi:10.17605/OSF.IO/BNEKP, osf.io/bnekp.

Fried, E. I., Bockting, C., Arjadi, R., Borsboom, D., Amshoff, M., Cramer, A. O., Epskamp, S., Tuerlinckx, F., Carr, D., & Stroebe, M. (2015). From loss to loneliness: The relationship between bereavement and depressive symptoms. *Journal of Abnormal Psychology*, *124*(2), 256.

Fried, E. I., Epskamp, S., Nesse, R. M., Tuerlinckx, F., & Borsboom, D. (2016). What are 'good' depression symptoms? Comparing the centrality of DSM and non-DSM symptoms of depression in a network analysis. *Journal of Affective Disorders*, *189*, 314–320.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3–4), 601–620. doi:10.1089/106652700750050961.

Fruchterman, T. M., & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, *21*(11), 1129–1164.

Grömping, U. (2012). Estimators of relative importance in linear regression based on variance decomposition. The American Statistician.

Hamilton, J. D. (1994). *Time Series analysis*, 1st edn. Princeton: NJ.

Haslbeck, J. M. B., & Fried, E. I. (2017). How predictable are symptoms in psychopathological networks? A reanalysis of 18 published datasets. Psychological Medicine. doi:10.1017/S0033291717001258.

Haslbeck, J. M. B., & Waldorp, L. J. (2015a). mgm: Structure estimation for time-varying mixed graphical models in high-dimensional data. arXiv:151006871.

Haslbeck, J. M. B., & Waldorp, L. J. (2015b). Structure estimation for mixed graphical models in high-dimensional data. arXiv:151005677.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). Statistical learning with sparsity: The lasso and generalizations. CRC Press.

Huang, S., Li, J., Sun, L., Ye, J., Fleisher, A., Wu, T., Chen, K., & Reiman, E. (2010). Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, *50*(3), 935–949. doi:10.1016/j.neuroimage.2009.12.120, http://www.sciencedirect.com/science/article/pii/S1053811909014281.

Kossakowski, J. J., Epskamp, S., Kieffer, J. M., van Borkulo, C. D., Rhemtulla, M., & Borsboom, D. (2016). The application of a network approach to health-related quality of life (HRQoL): Introducing a new method for assessing HRQoL in healthy adults and cancer patients. *Quality of Life Research*, *25*(4), 781–792.

Kossakowski, J. J., Groot, P. C., Haslbeck, J. M. B., Borsboom, D., & Wichers, M. (2016). Data from 'critical slowing down as a personalized early warning signal for depression'. Journal of Open Psychology Data (submitted).

McNally, R. J., Robinaugh, D. J., Wu, G. W., Wang, L., Deserno, M. K., & Borsboom, D. (2015). Mental disorders as causal systems a network approach to posttraumatic stress disorder. *Clinical Psychological Science*, *3*(6), 836–849.

Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, *34*(3), 1436–1462. doi:10.1214/009053606000000281, http://projecteuclid.org/Dienst/getRecord?id=euclid.aos/1152540754/.

Newman, M. (2010). Networks: An Introduction. Oxford University Press.

Pe, M. L., Kircanski, K., Thompson, R. J., Bringmann, L. F., Tuerlinckx, F., Mestdagh, M., Mata, J., Jaeggi, S. M., Buschkuehl, M., Jonides, J., & et al. (2015). Emotion-network density in major depressive disorder. *Clinical Psychological Science*, *3*(2), 292–300.

Pfaff, B. (2008). *Analysis of integrated and cointegrated time series with R*, 2nd edn. New York: Springer-Verlag.

Shulgin, B., Stone, L., & Agur, Z. (1998). Pulse vaccination strategy in the SIR epidemic model. *Bulletin of Mathematical Biology*, *60*(6), 1123–1148.

Spirtes, P., Glymour, C. N., & Scheines, R. (2000). Causation, prediction, and search. MIT Press.

Wainwright, M. J., & Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, *1*(1–2), 1–305. doi:10.1561/2200000001

Wichers, M., Groot, P. C., Psychosystems, E., Group, E., et al. (2016). Critical slowing down as a personalized early warning signal for depression. *Psychotherapy and Psychosomatics*, *85*(2), 114–116.

Yang, E., Baker, Y., Ravikumar, P., Allen, G., & Liu, Z. (2014). Mixed Graphical Models via Exponential Families *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. http://jmlr.org/proceedings/papers/v33/yang14a.pdf (pp. 1042–1050).

Zhang, B., Horvath, S., et al. (2005). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, *4*(1), 1128.