Psychological Methods

Investigating the Performance of Exploratory Graph Analysis and Traditional Techniques to Identify the Number of Latent Factors: A Simulation and Tutorial

Hudson Golino, Dingjing Shi, Alexander P. Christensen, Luis Eduardo Garrido, Maria Dolores Nieto, Ritu Sadana, Jotheeswaran Amuthavalli Thiyagarajan, Agustin Martinez-Molina

Funded by:

- Fondo Nacional de Innovación y Desarrollo Científico y Tecnológico (FONDOCYT)
- National Institutes of Health, National Institute on Aging

Investigating the performance of Exploratory Graph analysis and traditional techniques to identify the number of latent factors: a simulation and tutorial.

Hudson Golino[1], Dingjing Shi[1], Alexander P. Christensen[2], Luis Eduardo Garrido[3], Maria Dolores Nieto[4], Ritu Sadana[5], Jotheeswaran Amuthavalli Thiyagarajan[5], & Agustín Martínez-Molina[6]

[1] Department of Psychology, University of Virginia

[2] Department of Psychology, University of North Carolina at Greensboro

[3] Department of Psychology, Pontificia Universidad Catolica Madre y Maestra

[4] Department of Psychology, Universidad Autónoma de Madrid

[5] Department of Ageing and Life Course, World Health Organization

[6] Universidad Autónoma de Madrid

Author note

Correspondence concerning this article should be addressed to Hudson Golino, 485 McCormick Road, Gilmer Hall, Room 102, Charlottesville, VA 22903. E-mail: hfg9s@virginia.edu

Abstract

Exploratory graph analysis (EGA) is a new technique that was recently proposed within the framework of network psychometrics to estimate the number of factors underlying multivariate data. Unlike other methods, EGA produces a visual guide—network plot—that not only indicates the number of dimensions to retain, but also which items cluster together and their level of association. Although previous studies have found EGA to be superior to traditional methods, they are limited in the conditions considered. These issues are here addressed through an extensive simulation study that incorporates a wide range of plausible structures that may be found in practice, including continuous and dichotomous data, and unidimensional and multidimensional structures. Additionally, two new EGA techniques are presented, one that extends EGA to also deal with unidimensional structures, and the other based on the triangulated maximally filtered graph approach (EGAtmfg). Both EGA techniques are compared with five widely used factor analytic techniques. Overall, EGA and EGAtmfg are found to perform as well as the most accurate traditional method, parallel analysis, and to produce the best large-sample properties of all the methods evaluated. To facilitate the use and application of EGA, we present a straightforward R tutorial on how to apply and interpret EGA, using scores from a well-known psychological instrument: the Marlowe-Crowne Social Desirability Scale.

Keywords: exploratory graph analysis, number of factors, dimensionality, exploratory factor analysis, parallel analysis

Word count:

Investigating the performance of Exploratory Graph analysis and traditional techniques to identify the number of latent factors: a simulation and tutorial.

Investigating the number of latent factors or dimensions that underlie multivariate data is an important aspect in the construction and validation of instruments in psychology (Timmerman & Lorenzo-Seva, 2011). It is also one of the first steps in the analysis of psychological data, since it can play a crucial role in the implementation of further analyses and conclusions drawn from the data (Lubbe, 2019). Determining the number of factors is also relevant in the construction of psychological theories, since some areas (e.g., personality and intelligence) rely heavily on the identification of latent structures to understand the organization of human traits (Garcia-Garzon, Abad, & Garrido, 2019b).

Since the 1960's, several techniques were developed to estimate the number of underlying dimensions in psychological data, such as parallel analysis (PA; Horn, 1965), the K1 rule (Kaiser, 1960), and the scree test (Cattell, 1966). Simulation studies, however, have consistently shown that each technique has its own limitations (e.g. see Garrido, Abad, & Ponsoda, 2013; Lubbe, 2019), indicating a need for new dimensionality assessment methods that can provide more accurate estimates. Furthermore, the factor-analytic techniques also present challenges beyond the estimation of the number of dimensions such as the rotation of the loadings matrix and the subjective interpretation of the factor loadings (Sass & Schmitt, 2010).

Recently, Golino and Epskamp (2017) proposed an alternative approach, *Exploratory Graph Analysis* (EGA), to identify the dimensions of psychological constructs from the network psychometrics perspective. *Network psychometrics* is a recent addition to the field of quantitative psychology, which applies the network modeling framework to study psychological constructs

(Epskamp, Rhemtulla, & Borsboom, 2017). The network psychometric perspective is provided by the Gaussian graphical model (GGM: Lauritzen, 1996), which estimates the joint distribution of random variables (i.e., nodes in the network) by modeling the inverse of the variance-covariance matrix (Epskamp et al., 2017). Nodes (e.g., test items) are connected by edges or links, which indicate the strength of the association between the variables (Epskamp & Fried, 2018). Edges are typically partial correlation coefficients (Epskamp & Fried, 2018). Absent edges represent zero partial correlations (conditionally independent variables) while non-absent edges represent the remaining association between two variables after controlling for all other variables (Epskamp & Fried, 2018; Epskamp et al., 2017). Importantly, absent edges in the model will only correspond to conditional independence if the data is multivariate normal. EGA combines the GGM model with a clustering algorithm for weighted networks (walktrap; Pons & Latapy, 2006) to assess the dimensionality of the items in psychological constructs. Preliminary investigations of EGA via simulation studies have shown that it's a promising alternative technique to assess the dimensionality of constructs (Golino & Epskamp, 2017).

Despite the promising initial evidence, the original EGA technique (Golino & Epskamp, 2017) is not expected to work well with unidimensional structures, because of limitations related to the walktrap algorithm (Pons & Latapy, 2006). Specifically, the modularity measure (used to quantify the quality of dimensions in the algorithm) penalizes network structures that have only one dimension (Newman, 2004). As a consequence, the original EGA algorithm would almost always identify more than one factor, even if the data is generated from a unidimensional structure. To overcome this limitation, the current paper will present a new EGA algorithm that leverages the walktrap's tendency to find multiple clusters in weighted networks. This new EGA algorithm is expected to work well in both unidimensional and multidimensional structures (i.e., when the underlying dimensionality is comprised of one or more factors). An in-depth analysis, however, is necessary to check the suitability of this new EGA algorithm to estimate the number of simulated factors across different conditions and compared to traditional factor-analytic techniques.

Present Research

The aims of the current paper is threefold. First, it aims to systematically investigate, via a Monte-Carlo simulation study, the performance of the new EGA algorithm in recovering the number of simulated factors under different conditions. Previous studies have shown that the interfactor correlations, number of items per factor, and sample size each have an impact on the original EGA's performance (Golino & Epskamp, 2017), but little is known about the impact of factor loadings in the accuracy of EGA. It is well established in the literature that factor loadings are one of the most important elements that affect the accuracy of traditional dimensionality assessment methods (Garrido et al., 2013). Skewness has also not been considered in previous simulations involving EGA, which has only used unskewed dichotomous data (Golino & Epskamp, 2017). To better resemble practical settings in psychological data, we examined continuous (i.e., multivariate normal) and dichotomous data with skew.

Second, this study also investigates an alternative network estimation method for EGA, the *Triangulated Maximally Filtered Graph* approach (TMFG; Massara, Di Matteo, & Aste, 2016), hereafter named EGAtmfg. By replacing the GGM model with the TMFG algorithm, the EGAtmfg method can potentially overcome some of the limitations of the former method. One of the advantages of the TMFG is that it is not restricted to multivariate normal distributions and partial correlation measures (i.e., any association measure can be used), and it can potentially make stable comparisons across sample sizes (Christensen, Kenett, Aste, Silvia, & Kwapil, 2018). We investigated the performance of the EGAtmfg method in this study, and compared it to the new EGA algorithm, which uses the GGM model. We discuss the performance of both approaches and suggest practical recommendations for them. Also, while preliminary studies have compared traditional factor analytic methods with EGA (Golino & Epskamp, 2017; Golino & Demetriou, 2017), there is a need to compare the performance of EGA with different types of parallel analysis as well as techniques based on the scree test (Cattell, 1966), which are among the most widely known methods historically applied in psychology.

Lastly, this article provides a tutorial on how to implement the EGA techniques using R. With this tutorial, researchers from different fields interested in estimating the dimensionality of their tests, questionnaires, and other types of instruments can readily apply EGA. EGA may be especially relevant for those working on the area of aging research, that needs to use dimensionality assessment/reduction techniques to investigate the structure of multiple scales, questionnaires and tests.[1]

The tutorial uses data from the Virginia Cognitive Aging Project (VCAP; Salthouse, 2018) and verifies the dimensionality of the Social Desirability Scale (SDS; Crowne & Marlowe, 1960). A key part of our tutorial will showcase the new EGA algorithm by demonstrating how it can be used to first estimate dimensionality and then verify the unidimensionality of the dimensions in the SDS.

[1] The current paper is part of an international effort to develop new techniques, methods and metrics for healthy aging launched in 2017 by the World Health Organization (International Consortium on Metrics and Evidence for Healthy Ageing).

Exploratory Graph Analysis

Golino and Epskamp (2017) proposed EGA as a new method to estimate the number of latent variables underlying multivariate data using undirected network models (Lauritzen, 1996). The original EGA technique proposed by Golino and Epskamp (2017) starts by estimating a network using the GGM model (Lauritzen, 1996) and then applies a clustering algorithm for weighted networks. In the next paragraphs, the connection between GGM and factor models will be made. We explain the walktrap algorithm in more extensive more detail in **Appendix A**.

Equating the GGM with Factor Models

Consider a set of random variables **y** that are normally distributed with a mean of zero and variance-covariance matrix **Σ**. Let **K** (kappa) be the inverse of **Σ**, also known as the precision matrix:

$$\mathbf{K} = \mathbf{\Sigma}^{-\mathbf{1}} \quad (1)$$

Each element k_{ij} can be standardized to yield the partial correlation between two variables y_i and y_j, given all other variables in $\mathbf{y}, \mathbf{y}_{-\mathbf{c}(\mathbf{i},\mathbf{j})}$ (Epskamp, Waldorp, Mõttus, & Borsboom, 2018):

$$Cor(Y_i, Y_j | \mathbf{y}_{-(i,j)}) = -\frac{k_{ij}}{\sqrt{k_{ii}}\sqrt{k_{jj}}}. \quad (2)$$

Epskamp et al. (2018) points out that modeling **K** in a way that every nonzero element is treated as a freely estimated parameter generates a sparse model for **Σ**. The sparse model of the variance-covariance matrix is the GGM (Epskamp et al., 2018). The level of sparsity of the GGM can be set using different methods. The most common approach in network psychometrics is to apply a variant of the *least absolute shrinkage and selection operator* (LASSO; Tibshirani, 1996) termed *graphical LASSO* (GLASSO; Friedman, Hastie, & Tibshirani, 2008). The GLASSO is a regularization technique that is very fast to estimate both the model structure and the parameters of a sparse GGM (Epskamp et al., 2018). It has a tuning parameter (γ), that can be chosen in a way to minimize the extended Bayesian information criterion (EBIC; Chen & Chen, 2008), which is used to estimate optimal model fit and has been shown to accurately retrieve the true network structure in simulation studies (Epskamp & Fried, 2018; Foygel & Drton, 2010).

Now, we'll connect the GGM with factor models, and show how network psychometrics can be used to discover underlying latent structures in multivariate data. Let $\mathbf{y}$ represent a centered, normally distributed variable and $\boldsymbol{\eta}$ represent a set of latent variables. A general model connecting $\mathbf{y}$ and $\boldsymbol{\eta}$ is given by:

$$\mathbf{y} = \boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\varepsilon}, \quad (3)$$

where $\boldsymbol{\Lambda}$ is a factor loading matrix leading to the factor analysis model:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Psi}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Theta}, \quad (4)$$

where $\boldsymbol{\Psi}$ is $Var(\boldsymbol{\eta})$ and $\boldsymbol{\Theta}$ is $Var(\boldsymbol{\varepsilon})$. Assuming a simple structure, $\boldsymbol{\Lambda}$ can be reordered to be block-diagonal (each item can load only in one factor), and assuming local independence, $\boldsymbol{\Theta}$ is a diagonal matrix indicating that after conditioning on all latent factors the variables are independent (Epskamp et al., 2018).

Golino and Epskamp (2017) showed a decomposition (using the Woodbury matrix identity; Woodbury, 1950) leads to two important properties connecting GGM and factor model to orthogonal factors, with the resulting GGM being composed of unconnected clusters, while for oblique factors, the resulting GGM is composed of weighted clusters that are connected for each factor. These two characteristics can be explained as follows. Let the inverse of the variance-covariance matrix be the precision matrix $\mathbf{K}$, as shown in equation (1), therefore (following Woodbury, 1950):

$$\mathbf{K} = (\boldsymbol{\Lambda}\boldsymbol{\Psi}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Theta})^{-1} = \boldsymbol{\Theta}^{-1} - \boldsymbol{\Theta}^{-1}\boldsymbol{\Lambda}(\boldsymbol{\Psi}^{-1} + \boldsymbol{\Lambda}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^{\top}\boldsymbol{\Theta}^{-1}. \quad (5)$$

If $\mathbf{X} = (\boldsymbol{\Psi}^{-1} + \boldsymbol{\Lambda}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{\Lambda})$, and knowing that $\boldsymbol{\Lambda}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{\Lambda}$ is diagonal, then $\mathbf{K}$ is a block matrix in which every block is the inner product of factor loadings and residual variances, with diagonal blocks scaled by diagonal elements of $\mathbf{X}$ and off-diagonal blocks scaled by the off-diagonal elements of $\mathbf{X}$. As Golino and Epskamp (2017) argue, constraining the diagonal values of $\mathbf{X}$ to one will not lead to information loss. Furthermore, the absolute off-diagonal elements of $\mathbf{X}$ will be smaller than one. Considering the formation of $\mathbf{X}$, its off-diagonal values will equal zero if the latent factors are orthogonal (Golino & Epskamp, 2017).

In sum, network modeling and factor modeling are closely connected (Epskamp et al., 2018), and the use of network psychometrics for dimensionality assessment is a direct consequence of the two properties pointed to earlier. If the resulting GGM of orthogonal factors is a network with unconnected clusters (often referred to as communities) and the resulting GGM of oblique factors is a set of connected weighted clusters for each factor, then a community detection algorithm for weighted networks (which detects these clusters) can be applied to transform a network psychometric model into a dimensionality assessment technique.

Walktrap Community Detection

Golino and Epskamp (2017) proposed the use of the Walktrap algorithm (Pons & Latapy, 2006) to detect the number of dimensions (i.e., communities) in a network. The algorithm uses “random walks” or a stochastic number of steps from one node, across an edge, to another. The number of steps the random walks take can be adjusted but for current estimation purposes, EGA always applies the default number of four. The choice of using four steps comes from previous simulation studies that have shown that the Walktrap algorithm outperforms other community detection algorithms for weighted networks using four steps (Gates, Henry, Steinley, & Fair, 2016; Yang, Algesheimer, & Tessone, 2016).

A limitation of the Walktrap algorithm as an automated way to identify clusters in networks is that it penalizes unidimensional structures, since this algorithm decides the best partitioning of the clusters using the modularity index (Newman, 2004). Therefore, EGA it is not expected to work well with unidimensional structures. An overview of the Walktrap algorithm and why the modularity index penalizes unidimensional structures can be found in **Appendix A**. A new EGA algorithm that takes advantage of this characteristic and that could potentially be used in both unidimensional and multidimensional structures will be presented in a later section.

EGA Performance

Golino and Epskamp (2017) studied the accuracy in estimating the number of dimensions of EGA along with six traditional techniques: very simple structure (VSS; Revelle & Rocklin, 1979), minimum average partial (MAP; Velicer, 1976), Bayesian information criterion (BIC), EBIC, K1, and PA with generalized weighted least squares extraction and random data generation from a multivariate normal distribution. The authors simulated 32,000 data sets to fit known factor structures, systematically manipulating four variables: number of factors (2 and 4), number of items (5 and 10), sample size (100, 500, 1000 and 5000), and correlation between factors (0, .20, .50 and .70). The results of Golino and Epskamp (2017) showed that the accuracies of the different techniques, in ascending order, were: 39% for VSS, 50% for MAP, 81% for K1, 81% for BIC, 82% for EBIC, 89% for PA, and 93% for EGA. EGA was especially superior to the traditional techniques in the cases of larger structures (4 factors) and very high factor correlations (.70), achieving an accuracy of 71% which was much higher than the next best method (PA = 40%). Golino and Epskamp (2017) ascertained that EGA was the most robust method because its accuracy was less affected by the manipulated variables than those of the other methods.

The higher accuracy of EGA, when compared to traditional factor analytic methods, might be explained by the network psychometrics approach focus on the unique variance between pairs of variables rather than the variance shared across all variables. When a dataset is simulated following a traditional factor model, the dimensionality structure becomes clearer when a network of regularized partial correlations is estimated. Figure 1 shows two simulated five-factor model (population correlations). One with loadings of .70, inter-factor correlations of .70, and eight items per factor, and the other with loadings of .70, orthogonal factors and eight items per factor. In this figure, the population correlation matrix is plotted as a network with a two-dimensional layout computed using the Fruchterman-Reingold algorithm (Fruchterman & Reingold, 1991).

In this layout, nodes with stronger edges (e.g. high correlations) are placed closer than nodes with weak edges (e.g. low correlations). The two-dimensional layout helps to visually inspect groupings of variables, since variables with higher correlations are plotted together. The colors of the nodes represent the factors. On the left side of the figure, the population correlation matrix is shown; on the right side the estimated EGA structure is shown. The high correlation structure is shown in the top of the figure, and the orthogonal structure in the bottom. Estimating a network using regularized partial correlations results in a clearer structure with five groups of variables for the high correlation structure. Also, the strength of the regularized partial correlations is stronger within clusters than between clusters for the high correlation structure (top), making the true simulated five-factor structure easier to depict, even if the true correlation between factors is high.

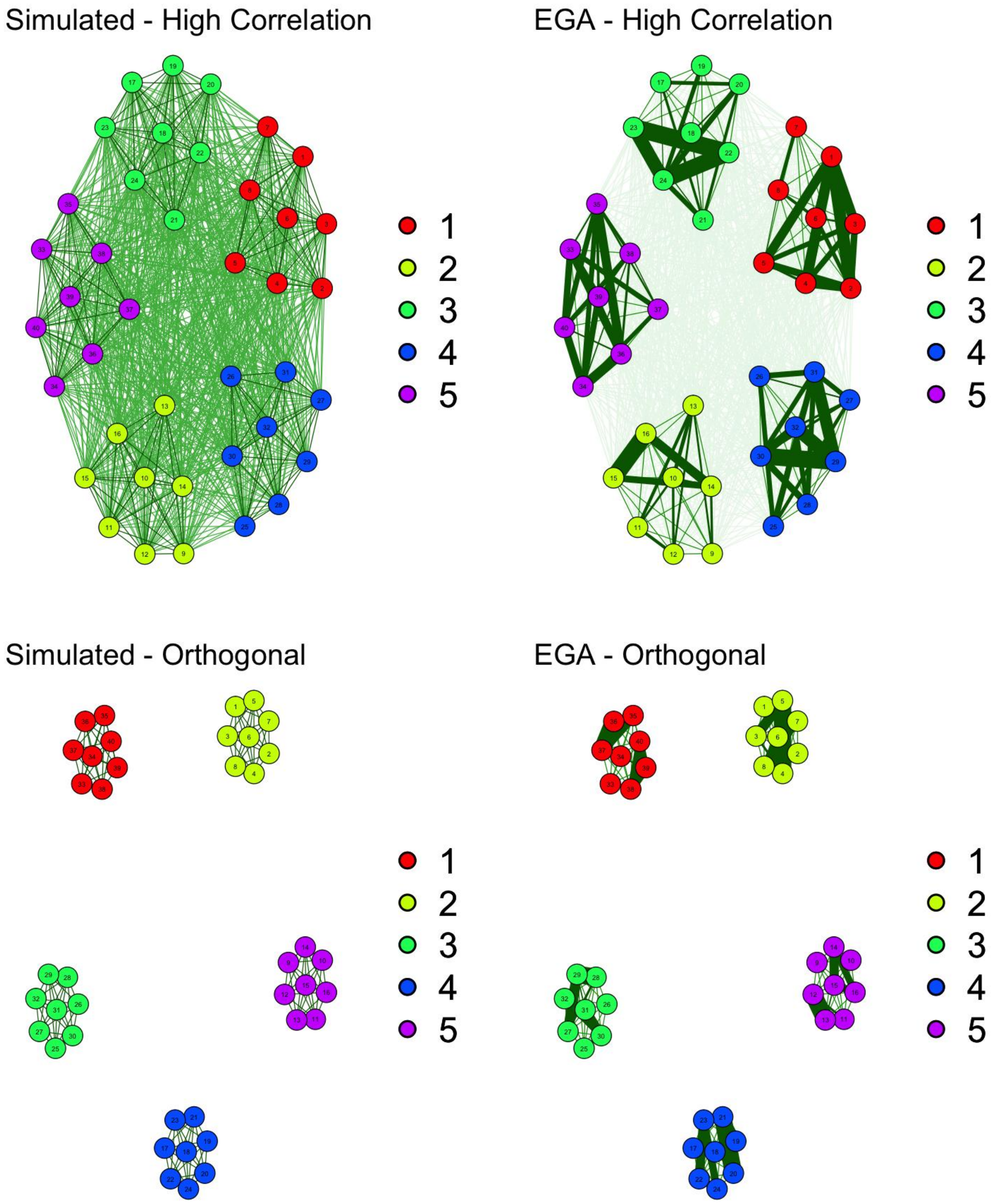
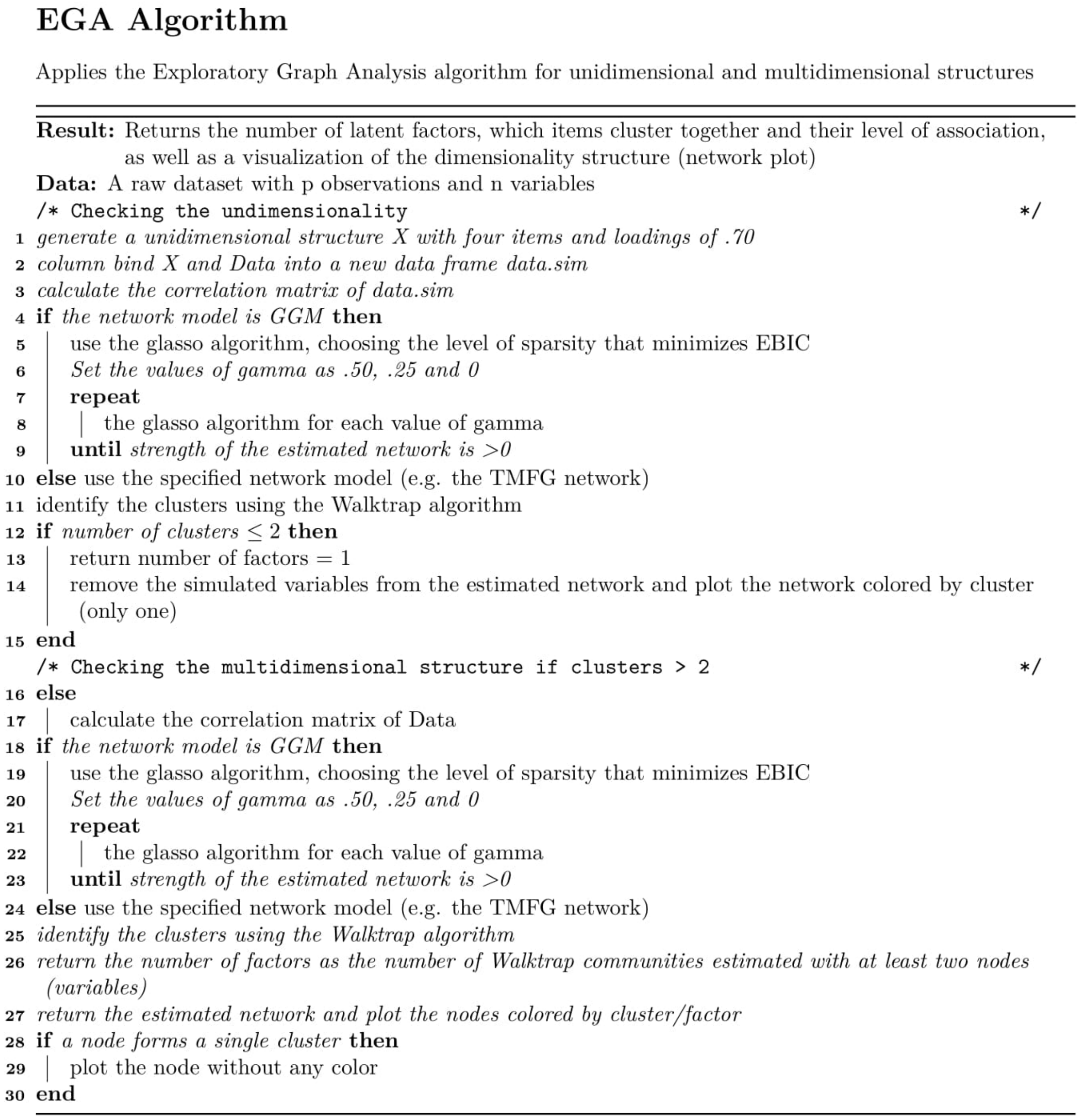


Figure 1. Simulated five factor model with loadings of .70 and 5,000 observations with interfactor correlation of .70 (top) and zero (bottom). The left side shows the population correlation matrix plotted as a network of zero-order correlations, while the left side shows the

EGA estimation of the population correlation matrix. Nodes represent variables, edges represent correlations, and the node colors indicates the simulated factors.

A New EGA Algorithm for Unidimensional and Multidimensional Structures

Considering the limitation of unidimensionality detection in the walktrap algorithm, the original EGA technique is not expected to work with single factor structures. To use EGA as a dimensionality assessment technique for both unidimensional and multidimensional structures, a new EGA algorithm is necessary. In the current paper, we propose such an algorithm that remedies this limitation of the walktrap algorithm. Figure 2 shows a description of the new EGA algorithm.



EGA Algorithm

Applies the Exploratory Graph Analysis algorithm for unidimensional and multidimensional structures

```
Result: Returns the number of latent factors, which items cluster together and their level of association,
        as well as a visualization of the dimensionality structure (network plot)
Data: A raw dataset with p observations and n variables
/* Checking the undimensionality                                                                  */
generate a unidimensional structure X with four items and loadings of .70
column bind X and Data into a new data frame data.sim
calculate the correlation matrix of data.sim
if the network model is GGM then
    use the glasso algorithm, choosing the level of sparsity that minimizes EBIC
    Set the values of gamma as .50, .25 and 0
    repeat
        the glasso algorithm for each value of gamma
    until strength of the estimated network is >0
else use the specified network model (e.g. the TMFG network)
identify the clusters using the Walktrap algorithm
if number of clusters ≤ 2 then
    return number of factors = 1
    remove the simulated variables from the estimated network and plot the network colored by cluster
      (only one)
end
/* Checking the multidimensional structure if clusters > 2                                        */
else
    calculate the correlation matrix of Data
if the network model is GGM then
    use the glasso algorithm, choosing the level of sparsity that minimizes EBIC
    Set the values of gamma as .50, .25 and 0
    repeat
        the glasso algorithm for each value of gamma
    until strength of the estimated network is >0
else use the specified network model (e.g. the TMFG network)
identify the clusters using the Walktrap algorithm
return the number of factors as the number of Walktrap communities estimated with at least two nodes
  (variables)
return the estimated network and plot the nodes colored by cluster/factor
if a node forms a single cluster then
    plot the node without any color
end
```

Figure 2. New EGA algorithm for unidimensional and multidimensional structures

The algorithm starts by simulating an unidimensional structure with four variables and loadings of .70. Then, it binds the simulated data with the empirical (user-provided) data. The next step is the estimation of the GGM (if the network model is set to be a GGM). The correlation matrix is computed using the `cor_auto` function of the `qgraph` package (Epskamp, Cramer, Waldorp, Schmittmann, & Borsboom, 2012). The EBICglasso function (from qgraph) is then used to estimate the GGM. The EBICglasso function will search for the optimal level of sparsity (using λ parameter in the glasso algorithm) in a network by choosing a value of λ that minimizes the extended Bayesian information criteria (EBIC; Chen & Chen, 2008). Following Foygel and Drton (2010), 100 values of λ are chosen. These values are logarithmically evenly spaced between λ_{Max} (the smallest value which will result in a completely empty network—that is, no edges between the nodes) and $\lambda_{Max}/100$. The ratio of the lowest λ value compared to λ_{Max} is set to 0.1. A hyperparameter (γ; gamma) of EBICglasso controls the severity of the model selection. EBIC is computed for values of gamma larger than zero. However, when gamma is zero, BIC is computed instead (for more details, see Chen & Chen, 2008).

In the implementation of the EGA algorithm, the gamma hyperparameter of the EBICglasso function is set to 0.5. If the resulting network has a node with the strength of zero (i.e., disconnected from the rest of the network), then gamma is set to 0.25. The process repeats until all nodes are connected in the resulting network or if the gamma parameter is zero. In this last case, the EBIC is equal to the regular BIC.

In the next step, the walktrap algorithm is used. If the number of estimated clusters in the network is equal to or lower than 2, then the empirical data is unidimensional. This is one of the most important parts of the new EGA algorithm. Since the walktrap algorithm will penalize networks with only one cluster, by adding a simulated dataset with a known unidimensional structure, the walktrap algorithm will estimate at least two clusters: one comprised by the simulated data, and the other by the empirical or user-provided dataset. In this case, the estimated number of factors/clusters in the empirical data is one, since the other cluster is composed by the simulated data. If the number of clusters is greater than two, then the new EGA algorithm will reestimate the network, and apply the walktrap algorithm as described above. The final clustering solution is defined by all clusters with at least two variables (or nodes/items). The resulting network plot will show the estimated network and the nodes are colored by cluster/factor. If one variable (or node) is estimated as belonging to a single cluster, this variable won't be colored in the plot. This strategy helps the user identify if there are any variables that do not pertain to any cluster in the network.

Another difference from the original EGA method is related to the `gamma` parameter of the `EBICglasso` function. Originally, Golino and Epskamp (2017) used the default of 0.5. This modification, together with the removal of clusters with single nodes, makes the result of EGA more likely to be stable, in the sense that it will generate less extreme results with the number of clusters approaching the number of variables.

EGA with TMFG estimation

More recently, a new approach to estimate psychometric networks, the TMFG, entered the field (Christensen et al., 2018). The TMFG method applies a structural constraint on the network, which restrains the network to retain a certain number of edges ($3n$-6, where n is the number of nodes; Massara et al., 2016). The network is composed of 3- and 4-node cliques (i.e., sets of connected nodes; a triangle and tetrahedron, respectively). The TMFG method constructs a network using zero-order correlations and the resulting network can be associated with the inverse covariance matrix (yielding a GGM; Barfuss, Massara, Di Matteo, & Aste, 2016). Notably, the TMFG can use any association measure and thus does not assume the data is multivariate normal.

Construction begins by forming a tetrahedron (Figure 3) of the four nodes that have the highest sum of correlations that are greater than the average correlation in the correlation matrix, which is defined as:

$$\bar{c} = \frac{\sum_i \sum_j c_{ij}}{n}, \quad (6)$$

$$w_i = \sum_j \left\{ \begin{matrix} c_{ij} > \bar{c} = c_{ij} \\ c_{ij} \leq \bar{c} = 0 \end{matrix} \right. , \quad (7)$$

where c_{ij} is the correlation between node i and node j, $\bar{c}$ is the average correlation of the correlation matrix (6), and w_i is the sum of the correlations greater than the average correlation for node i (7).

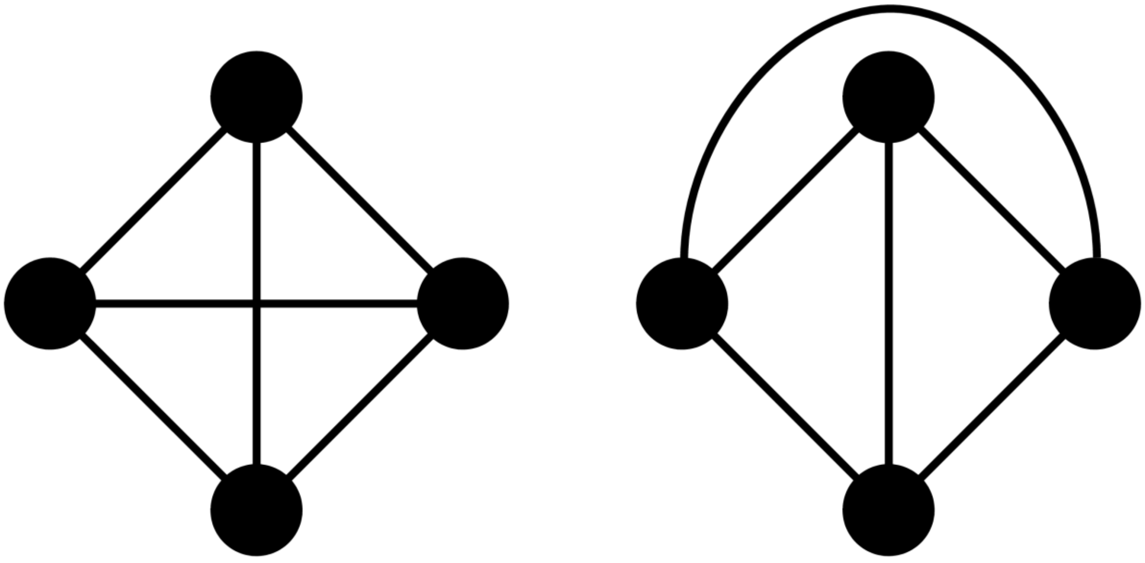


Figure 3. A depiction of a network tetrahedron (left) and a tetrahedron drawn so that no edges are crossing (right)

Next, the algorithm iteratively identifies the node that maximizes its sum of correlations to a connected set of three nodes (triangles) already included in the network and then adds that node to the network. In equation (8), this is mathematically defined as the maximum gain of the score function (S; e.g., sum of correlations) for each node (v) with each node in a set of triangles (t_1, t_2, t_3) in the network (Figure 4):

$$\mathbf{MaxGain} = \left(\max_{v \in v_1 \dots v_k} S(v, t_1), \max_{v \in v_1 \dots v_k} S(v, t_2), \dots, \max_{v \in v_1 \dots v_k} S(v, t_3) \right), \quad (8)$$

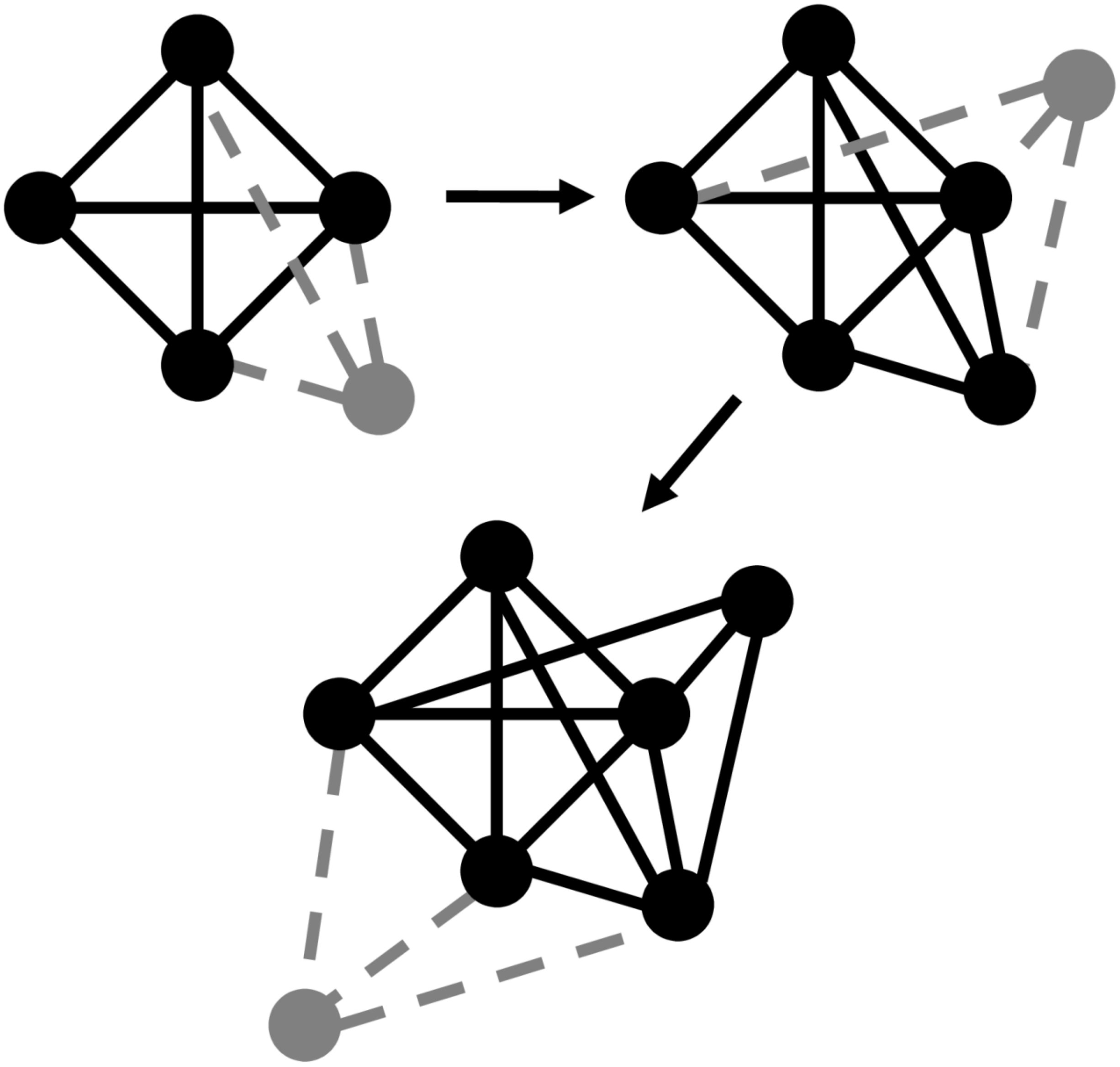


Figure 4. A depiction of how TMFG constructs a network. Starting with the tetrahedron, the node with the largest sum to three other nodes in the network is added (top left). This process continues until all nodes are included in the network.

The process is completed once every node is connected in the network. In this process, the network automatically generates what's called a *planar* network. A planar network is a network that *could* be drawn on a sphere with no edges crossing (Figure 3; often, however, the networks are depicted with edges crossing; Tumminello, Aste, Di Matteo, & Mantegna, 2005).

An intriguing property of planar networks is that they form a "nested hierarchy" within the overall network (Song, Di Matteo, & Aste, 2011). This simply means that sub-networks are nested within larger sub-networks of the overall network. The constituent elements of these sub-networks are 3-node cliques (i.e., triangles), which form an emergent hierarchy in the overall network (Song, Di Matteo, & Aste, 2012). Research that compared a novel algorithm, which exploited this hierarchical structure, to several traditional methods of hierarchical clustering (e.g., complete linkage and k-mediods) found that the novel algorithm outperformed the traditional methods, retrieving more information with fewer clusters (Song et al., 2012). Similar to EGA, EGAtmfg first constructs the network (using the TMFG method) and the walktrap algorithm is applied.

Factor Analytic Techniques

Eigenvalue-Based Methods

The eigenvalue-greater-than-one rule, also known as Kaiser's rule or K1, is perhaps the most well-known method for identifying the number of factors to retain. K1 indicates that only factors with eigenvalues above one should be retained. The rationale of this rule is that a factor should explain at least as much variance as a variable is bestowed in the standard score space and that components with eigenvalues above one are ensured to have positive internal consistencies (Garrido et al., 2013; Kaiser, 1960). However, the proofs for this rule were developed for population statistics, and a large body of research has shown that it doesn't perform well with finite samples (Hayton, Allen, & Scarpello, 2004). Nevertheless, recent studies have shown that this rule is still applied in practice frequently (Izquierdo, Olea, & Abad, 2014).

Parallel analysis was originally proposed by Horn (1965) as a modification of the K1 rule (Kaiser, 1960) that took into account the sampling variability of the latent roots. The rationale behind this method is that the true dimensions should have sample eigenvalues that are larger than those obtained from random variables that are uncorrelated at the population level. Parallel analysis has been one of the most studied and accurate dimensionality assessment methods for continuous and categorical variables to date (Crawford et al., 2010; Garrido et al., 2013; Garrido, Abad, & Ponsoda, 2016; Ruscio & Roche, 2012; Timmerman & Lorenzo-Seva, 2011).

Although Horn (1965) based PA on the eigenvalues obtained from the full correlation matrix using principal component analysis (PApca), Humphreys and Ilgen (1969) suggested that a more precise estimate of the number of common factors could be obtained by computing the eigenvalues from a reduced correlation matrix with estimates of communalities in its diagonal using principal axis factoring (PApaf). As a communality estimate, they chose the squared multiple correlations between each variable and all the others. Even though these two variants of PA have not been compared frequently, Crawford et al. (2010) found that for continuous variables their overall accuracies were similar for structures of one, two, and four factors (60% for PApca and 65% for PApaf), with neither method being superior to the other across all the studied conditions. With categorical variables (two to five response options), however, Timmerman and Lorenzo-Seva (2011) found that PApca clearly outperformed PApaf for structures of one and three major factors (overall accuracies of 95% for PApca and 70% for PApaf).

Automated Scree Test Methods

The scree test optimal coordinate (OC) and acceleration factor (AF) methods (Raiche, Walls, Magis, Riopel, & Blais, 2013) constitute two non-graphical solutions to Cattell's scree test (Cattell, 1966). A detailed description of OC and AF can be found on **Appendix B**. In their validation study with continuous variables, Raiche et al. (2013) found that the percentage of correct dimensionality estimates of OC (49%) was comparable to that of PA (53%), and between moderately to considerably higher than those for AF (39%), the Cattell-Nelson-Gorsuch scree test (30%), the K1 rule (21%), and the standard error scree (9%), among other methods. Similarly, Ruscio and Roche (2012) showed that the OC (74%), PA (76%), and the Akaike Information Criterion (73%) had comparable accuracies that were notably higher than other methods including the BIC (60%), MAP (60%), the chi-square test of model fit (59%), the AF (46%), and K1 (9%).

Method

Design

In order to evaluate the performance of the different dimensionality methods, six relevant variables were systematically manipulated using Monte Carlo methods: the number of factors, factor loadings, variables per factor, factor correlations, number of response options, and sample size. For each of these, their levels were chosen to represent conditions that are encountered in empirical research and that could produce differential levels of accuracy for the dimensionality procedures.

Number of factors: structures of 1, 2, 3, and 4 factors were simulated. These number of factors conditions include the important test of unidimensionality (Beierl, 2018), as well as dimensions that are below, at, and above the median number of first-order latent variables of 3 that is generally found in psychological factor analytic research (Jackson, Gillaspy Jr, & Purc-Stephenson, 2009). Additionally, these levels are in line with typical simulation studies in the area of dimensionality (e.g., Auerswald & Moshagen, 2019; Garrido et al., 2016).

Factor loadings: factor loadings were simulated with the levels of .40, .55, .70, and .85. According to Comrey and Lee (2016), loadings of .40, .55, and .70 can be considered as poor, good, and excellent, respectively, thus representing a wide range of factor saturations. In addition, loadings of .85 were also simulated, which although not frequently encountered in psychological data, allow for the evaluation of the dimensionality methods under ideal conditions.

Variables per factor: the factors generated were composed of 3, 4, 8, and 12 indicators with salient loadings. Three items are the minimum required for factor identification (Anderson, 1958), 4 items per factor represents a slightly overidentified model, while factors composed of 8 and 12 items may be considered as moderately strong and highly overidentified, respectively (Velicer, 1976; Widaman, 1993). It should be noted that the condition of 12 variables per factor was simulated for unidimensional structures only.

Factor correlations: factor correlations were simulated with the levels of .00, .30, .50, and .70. This includes the orthogonal condition (.00), as well as medium (.30) and large (.50) correlation levels, according to Cohen (1988). Further, although factor correlations of .70 are very large, in some areas within psychology (e.g., intelligence), researchers sometimes have to distinguish between constructs that are this highly correlated (e.g., Kane, Hambrick, & Conway, 2005).

Number of response options: normal continuous and dichotomous types of data were generated. The level of association between the continuous variables was measured using Pearson's correlations, while tetrachoric correlations were used for the dichotomous variables.

Sample size: datasets with 500, 1,000, and 5,000 observations were simulated. Sample sizes of 500 and 1,000 can be considered as medium and large, respectively (Li, 2016), while a sample of 5,000 observations allows for the evaluation of the dimensionality methods in conditions that can approximate their population performance. Further, these sample sizes were selected by taking into account that tetrachoric correlations require large sample sizes to achieve acceptable sampling errors, especially when the item difficulties vary substantially (such as when the data are skewed; Timmerman & Lorenzo-Seva, 2011).

In order to generate more realistic factor structures, several steps were undertaken. First, the factor loading for each item was drawn randomly from a uniform distribution with values ranging from $\pm.10$ of the specified level manipulated (e.g., for the level of .40 the loadings were drawn from the range of .30 to .50). Second, as it is common in practice to find complex structures in which items present non-zero loadings on multiple factors, we generated cross-loadings consistent to those commonly found in real data. The cross-loadings were generated following the procedure described in (Meade, 2008) and (Garcia-Garzon, Abad, & Garrido, 2019a): cross-loadings were randomly drawn from a normal distribution, $N(0, .05)$, for all the items. Third, the magnitude of skewness for each item was randomly drawn with equal probability from a range of -2 to 2 in increments of $.50$, following (Garrido et al., 2013). A skewness level of zero corresponds to a symmetrical distribution, while ±1 can be categorized as a meaningful departure from normality (Meyers, Gamst, & Guarino, 2016) and ±2 as a high level of skewness (Muthén & Kaplan, 1992).

As the simulation design of the current study is not completely crossed (e.g., there are no factor correlations for unidimensional structures), it can be broken down into two parts: (a) the unidimensional conditions with a 4 × 4 × 2 × 3 (factor loadings × variables per factor × number of response options × sample size) design, for a total of 96 condition combinations; and (b) the multidimensional conditions with a 4 × 3 × 4 × 2 × 3 (factor loadings × variables per factor × factor correlations × number of response options × sample size) design, for a total of 288 condition combinations. For each of these 384 conditions combinations, 500 replicates were simulated.

Data Generation

For each simulated condition, 500 sample data matrices were generated according to the common factor model. A detailed description of the data simulation approach can be found on **Appendix C**. The resulting continuous variables were also dichotomized by applying a set of thresholds according to specific levels of skewness (Garrido et al., 2013). For each sample data matrix generated, the convergence of EGA with GLASSO estimation was verified (see the convergence rate on **Appendix D**). If the analysis did not generate a numeric estimation (i.e. number of factors), the sample data matrix was discarded and a new one was generated, until we obtained 500 sample data matrices per condition.

Data analysis

We used *R* (R Core Team, 2017) for all our analyses. The AF and OC techniques were computed using the *nFactors* package (Raiche, 2010), while PA with resampling was applied using the *fa.parallel* function contained in the *psych* package (Revelle, 2018). Both versions of

EGA were applied using the *EGAnet* package (Golino & Christensen, 2019). The figures were generated using the *ggplot2* (Wickham, 2016) and *ggpubr* package (Kassambara, 2017).[2]

In order to evaluate the performance of the dimensionality methods three complementary criteria were used: the percentage of correct number of factors (PC), the mean bias error (MBE), and the mean absolute error (MAE). The first criteria (PC) is calculated as the sum of the estimated number of factors that are equal to the simulated number of factors divided by the number of sample data matrices simulated (i.e. the percentage of correct estimates). The second criteria (MBE) is the sum of the estimated number of factors minus the simulated number of factors, divided by the total number of sample data matrices simulated. The third criteria (MAE) is similar to MBE, but uses the absolute value of the difference between the estimated and the simulated number of factors.

[2] The paper was written following a reproducible approach, integrating text and code into two sets of files. The first set has all the code used in the simulation. The second set contains an *R Markdown* file integrating the manuscript text and code used for the statistical and graphical analysis presented in the results' section. The *papaja* package (Aust & Barth, 2018) was used to easily create a document following the APA guidelines. Two other methods that are available in R and that may be used by applied researchers are Velicer's MAP (Velicer, 1976) and the very simple structure (VSS; Revelle & Rocklin, 1979), with both being implemented in the *psych* package (Revelle, 2018). Since Golino and Epskamp (2017) already compared EGA with VSS and MAP, the current paper won't present and discuss these two methods. However, readers interested in comparing EGA and EGAtmfg with MAP and VSS can find a summary of the results in **Appendix E**.

The PC criterion varies from 0% (signaling complete inaccuracy) to 100% (indicating perfect accuracy). In the case of the MBE, 0 reflects a total lack of bias, while negative and positive values denote underfactoring and overfactoring, respectively. Regarding the MAE criterion, higher values signal larger departures from the population number of factors, while the value of 0 indicates perfect estimation accuracy.

Finally, analyses of variance (ANOVA) were conducted to investigate how the factor levels and their combinations impacted the accuracy of the dimensionality methods. The PC and MAE were set (separately) as the dependent variables and the manipulated variables constituted the independent factors. The partial eta squared (η^2) measure of effect size was used to assess the magnitude of the main effects and interactions, per technique. According to Cohen (1988), η^2 values of 0.01, 0.06, and 0.14 can be considered as small, medium, and large effect sizes, respectively. It is important to note that all the codes used in the current study is available at an *Open Science Framework* repository, for reproducibility purposes: https://osf.io/e9f2c/?view_only=3732b311ef304b1793ee92613dcb0fe7.

Results

Overall Performance

The overall performance of the dimensionality methods, as well as their performance across the levels of the independent variables, is presented in Table 1. According to the accuracy of the methods shown in the table, the methods can be classified into three groups: low (below 70%; AF and OC), moderate (70% and 80%; EGAtmfg and K1), and high accuracy (> 80%; PApaf, PApca and EGA). In terms of the PC criterion, the methods from best to worst were: EGA (M = 87.91%, SD = 32.60%), PApca (M = 83.01%, SD = 37.55%), PApaf (M = 81.88%, SD = 38.52%), K1 (M = 79.46%, SD = 40.40%), EGAtmfg (M = 74.61%, SD = 43.52%), OC (M = 66.36%, SD = 47.25%) and AF (M = 54.59%, SD = 49.79%).

PLEASE INSERT TABLE 1 ABOUT HERE

In terms of the MBE, EGA method showed the least overall bias, with a very small tendency to overfactor (0.02), followed by EGAtmfg (MBE = -0.12), PApaf (-0.25) and PApca (-0.29), which had a moderate tendency to underfactor. The rest of the methods had considerable larger MBEs, with OC (-0.61) and AF (-0.97) underfactoring, and K1 (0.33) overfactoring. Regarding the MAE, the two best methods were EGA (0.27) and PApca (0.30), followed by PApaf (0.32) and EGAtmfg (0.32). The remaining methods, K1 (0.46), OC (0.71) and AF (0.97), produced MAEs that were markedly worse.

Unidimensional Structures

Figure 5 shows the accuracy of the methods per sample size, factor loadings and number of variables for continuous (Figure 5A) and dichotomous (Figure 5B) data. In each plot, a dashed gray line represents an accuracy of 90%. Inspecting Figure 5 reveals several notable trends. First, while most methods presented an accuracy higher than 90% in the continuous data condition (Figure 5A), EGAtmfg fails considerably when the number of variables per factor is 12 (M = 26.20%). Second, K1 presents a low accuracy for sample size of 500, loadings of .40 and 12 variables per factor (M = 11.75%). Third, PApaf performs poorly when the factor loadings is .40 and the number of items is 3 or 4 (M = 0.35%), improving significantly for 3 or 4 variables per factor and loadings of .55 (M = 57.52%)

In the dichotomous data condition, the scenario is a slightly more nuanced for the percentage of correct dimensionality estimates. AF and PApca are the two most accurate methods

(99.78% and 99.27%, respectively), followed by OC (M = 94.57%) and EGA (M = 92.54%). The accuracy of K1 and OC decreases with an increase in the number of variables, for factor loadings of .40 and .55 and sample sizes of 500 and 1000. EGAtmfg once again presents a very low accuracy when the number of variables is 12 (M = 11.38%), although presenting a high accuracy for 3, 4 or 8 items (M = 97.29%). It is also notable that PApaf presents a much lower percentage of correct estimates for loadings of .40 (M = 40.87%) and .55 (M = 40.87%), especially when compared with EGA ($M_{LOAD=0.40}$ = 91.22%, $M_{LOAD=0.55}$ = 95.87%).

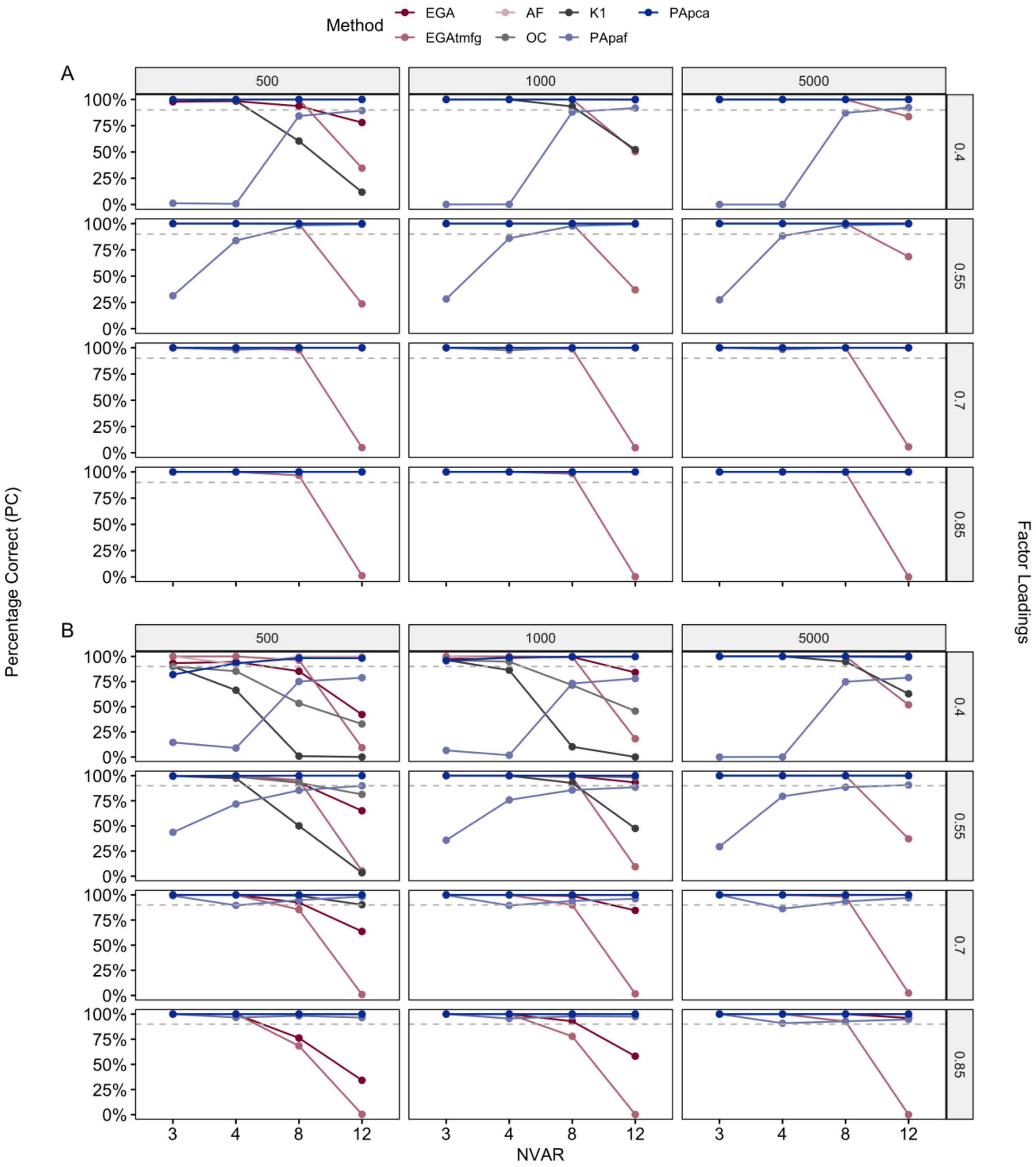


Figure 5. Accuracy per sample size, factor loadings and number of variables (NVAR) for unidimensional factors with continuous (A) and dichotomous (B) data.

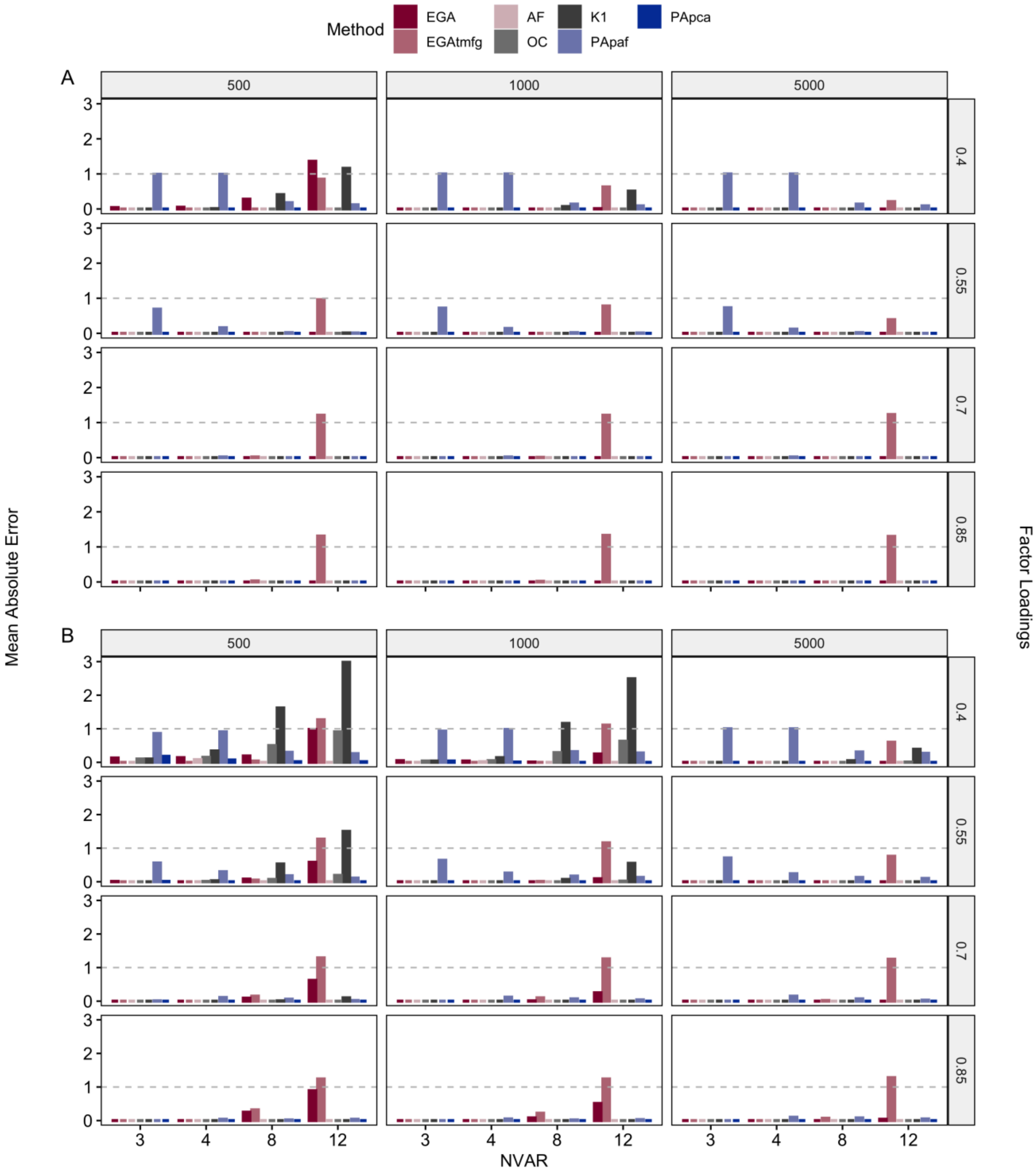


Figure 6. Mean Absolute Error (MAE) per sample size, factor loadings and number of variables (NVAR) for unidimensional factors with continuous (A) and dichotomous (B) data.

Figure 6 shows the absolute bias (MAE) for continuous (Figure 6A) and dichotomous data (Figure 6B). In the continuous data condition, PApca, OC and AF presented a MAE of zero, while EGA had a MAE 0.04, K1 0.05, K1 had 0.05, PApaf 0.20, and EGAtmfg 0.24.

Except for loadings of .40 and .55, EGAtmfg presented higher bias for conditions with 12 items, in general (MAE = 0.26). PApaf had higher MAE for loadings of .40 and three or four variables per factor (MAE = 1.00), and for loadings of .55 and 3 variables per factor (MAE = 0.71). Also, EGA, K1 and EGAtmfg presented an increased bias in the conditions with factor loadings of .40, 12 variables per factor and sample size of 500.

Bias increased in the dichotomous data conditions (Figure 6B). The order of MAE (from worst to best), however, remained the same: EGAtmfg (MAE = 0.24), PApaf (MAE = 0.20), K1 (MAE = 0.05 and EGA (MAE = 0.04). OC (MAE = 0), AF (MAE = 0) and PApca (MAE = 0) presented the lower bias.

Table 2 shows the effect sizes per condition simulated. K1 and PApaf were the methods that presented the highest effect sizes, in general. Both methods are very affected, in terms of accuracy and bias, by the variability in the number of variables, factor loadings and the interaction between factor loadings and number of variables. EGAtmfg is also very affected by the number of variables per factor, both in terms of accuracy and bias.

PLEASE INSERT TABLE 2 ABOUT HERE

Multidimensional structures

Figure 7 shows the accuracy of the methods per sample size, factor loadings, interfactor correlation and number of variables for continuous (Figure 7A) and dichotomous data (Figure

7B), for the five most accurate techniques (PApaf, EGA, EGAtmfg, K1 and PApca). In each plot, a dashed gray line represents an accuracy of 90%. For the continuous data condition, the order of the methods in terms of percentage of correct dimensionality estimates is: PApaf (M = 88.18%), EGA (M = 87.20%), K1 (M = 83.29%), PApca (M = 81.02%) and EGAtmfg (M = 76.33%).

The first notable trend in Figure 7 is the very high accuracy (above 90%) in the continuous data condition (Figure 7A) for loadings from .55 to .85 and interfactor correlation from zero to .50 for most methods, with the following exceptions. For loadings of .55, orthogonal factors and three variables per factor, the accuracy of PApaf is lower than 75%. The accuracy of K1 is also below 75% in conditions with eight items and samples of 500, as well as PApca in conditions with 3 or 4 items, samples of 500 and interfactor correlation of .50. EGAtmfg presents a PC lower than 75% irrespective of sample size when the interfactor correlation is .50 and three variables per factor.

It is important to note that the accuracy of K1 goes down with the increase in the number of variables per factor, in conditions with loadings of .40, sample sizes of 500 or 1000. The accuracy of EGA is almost always lower than 75% with loadings of .40 and sample size of 500. It is also notable that PApaf have very low PCs in conditions with loadings of .40 and 3 or 4 variables per factor.

In the conditions where the interfactor correlation is .70, factor loading is .40, and number of variables per factor is eight, PApaf presented a mean percentage of correct estimates of 92.13% and 99.87% for sample size of 1000 and 5000, while EGA presented an accuracy of 66.07% for sample size of 1000 and 98.06% for sample size of 5000. In the same conditions, EGAtmfg presented an accuracy of 69.67% and 92.73% for sample sizes of 1000 and 5000, while

PApca presented an accuracy of 48.60% and 100%, and K1 7.73% and 95.33% respectively for samples of 1000 and 5000.

In conditions with interfactor correlation of .70 and factor loadings of .55, PApca and K1 only presented percentage of correct dimensionality estimates above 90% with eight variables per factor and sample size of 1000 and 5000. EGA and EGAtmfg presented an accuracy higher than 90% irrespective of sample size with eight variables per factor, for a loading of .55 and interfactor correlation of .70. EGA (86.07%) and PApaf (99.59%), on the other side, presented high PCs for loadings varying from .55 to .85 and sample sizes of 1000 and 5000, irrespective of the number of variables per factor when the interfactor correlation is .70.

The accuracy for EGA and PApaf for factor loadings of .70, across all conditions, is 98.83% and 99.99%, respectively. For factor loadings of .85 is 100% for both EGA and PApaf. At the same time, EGAtmfg presented an accuracy of 82.12% for loadings of .70 and 85.54% for loadings of .85, while K1 presented an accuracy of 91.27% and 92.01%, and PApca of 84.99% and 87.78% for loadings of .70 and .85, respectively.

In the dichotomous data condition, the scenario is, again, more nuanced in terms of accuracy than in the continuous data condition (Figure 7B). EGA is the most accurate method (M = 81.47%), followed by PApaf (M = 78.74%), PApca (M = 70.23%), EGAtmfg (M = 69.38%) and K1 (M = 65.78%).

Figure 7B reveals two general tendencies. One is the increase of PC with the increase of number of variables per factor, sample size and factor loadings. The second one is the decrease in accuracy as the interfactor correlation increases from zero to .70. With loadings of .40, most techniques present accuracies lower than 90%, except in the following conditions. For a sample size of 1000, eight items per factor and orthogonal factors, EGA, PApca and EGAtmfg presented an accuracy greater than 90%. For a sample size of 5000 and orthogonal factors, EGA and PApca achieved an accuracy higher than 90% irrespective of the number of variables per factor, while PApaf increased the accuracy with the increase in the number of variables and K1 decreased the accuracy with the number of items going from 3 to 8. With an interfactor correlation of .30, PApca achieved an accuracy higher than 90% with eight items and a sample of 1000, and with a sample size of 5000, the accuracy was above 90% irrespective of the number of variables, while EGA achieved the same level of accuracy only with four or eight variables per factor. With an interfactor correlation of .50, EGA, EGAtmfg, PApaf and PApca presented accuracies above 90% with eight items and sample size of 5000. When the correlation was .70, only EGA presented an accuracy higher than 90%, with a sample size of 500 and eight variables per factor.

As the factor loadings increase, the accuracy of the methods also increase, even if the interfactor correlation is .70. EGA presented an accuracy of 23.92% for loadings of .40, 54.21% for loadings of .55, 89.32% for loadings of .70 and 99.22% for loadings of .85. PApaf presented a similar pattern, with PC of 44.64% for loadings of .40, 80.46% for loadings of .55, 94.59% for loadings of .70 and 98.99% for loadings of .85.

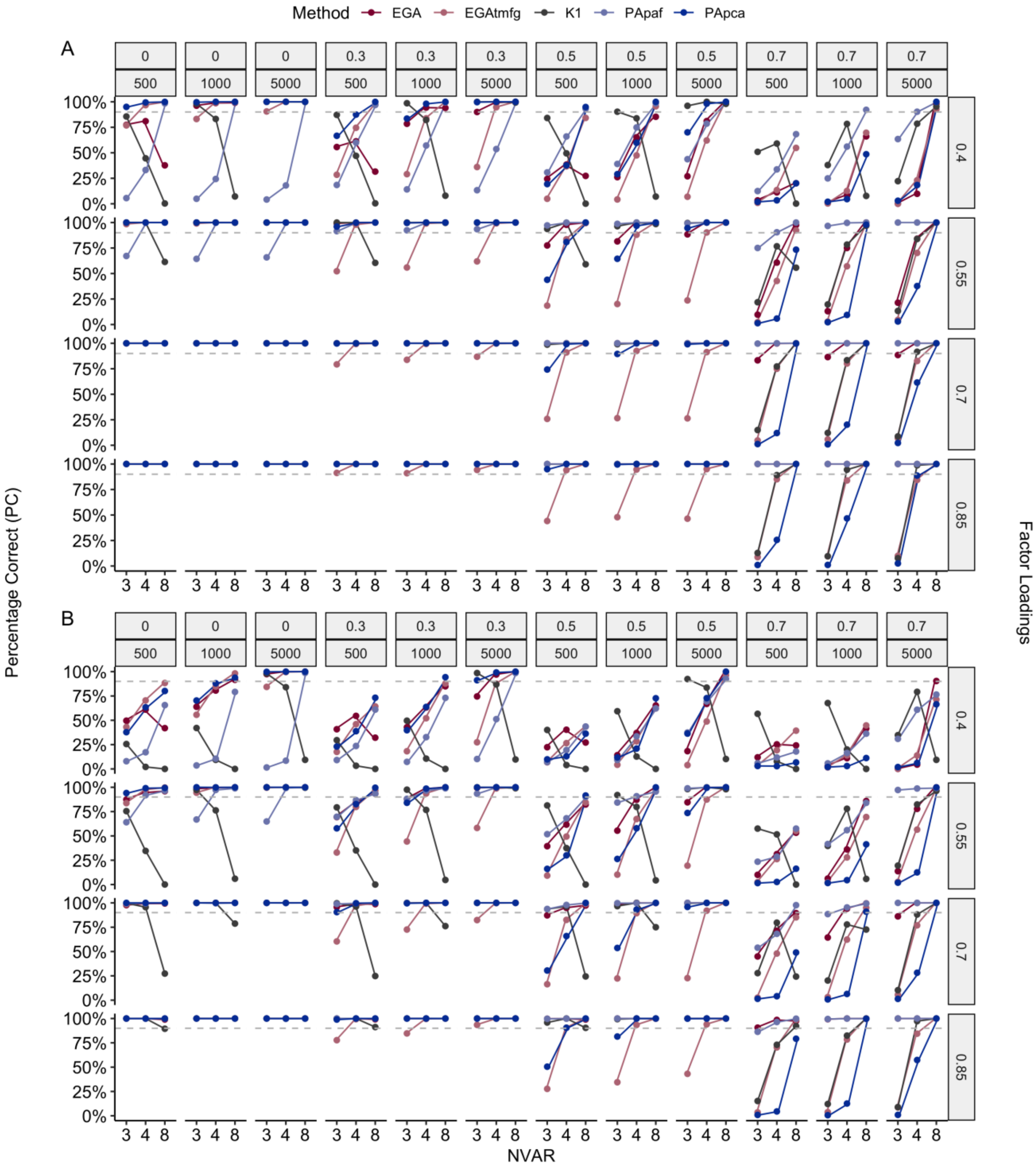


Figure 7. Accuracy per sample size, factor loadings and number of variables (NVAR) for multidimensional factors with continuous (A) and dichotomous (B) data.

Figure 8 shows the percentage of correct dimensionality estimates by interfactor correlation and factor loadings for EGA, PApaf and PApca in multidimensional structures with dichotomous data. It is interesting to note that EGA presents a higher accuracy than PApaf for factor loadings of .40, in conditions with interfactor correlations of zero and 0.30. At the same time, EGA is more accurate than PApca in conditions with interfactor correlations of .50 and .70.

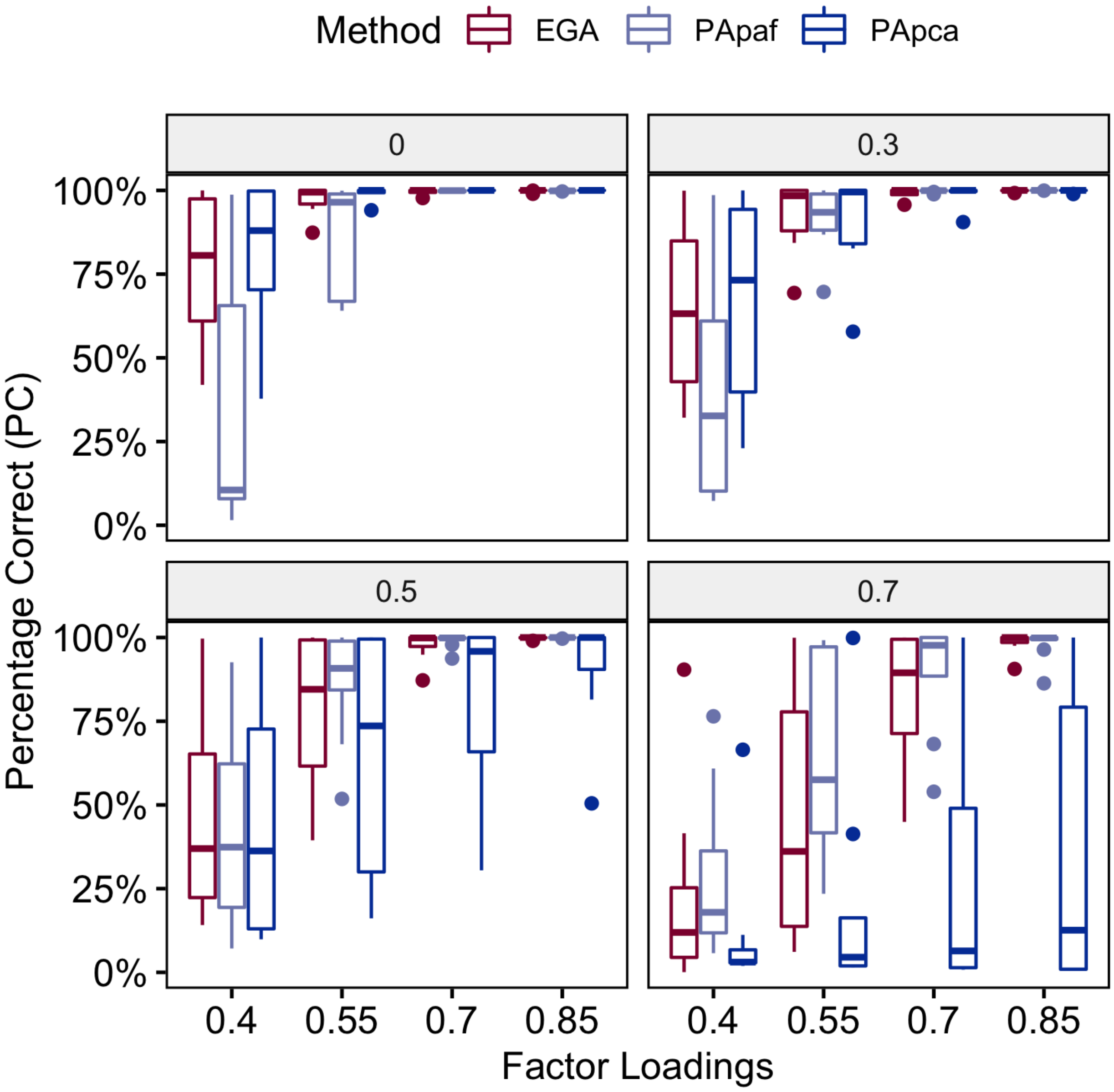


Figure 8. Boxplot comparing the percentage of correct estimates between EGA, PApaf and PApca in multidimensional structures with dichotomous data by interfactor correlation and factorloadings.

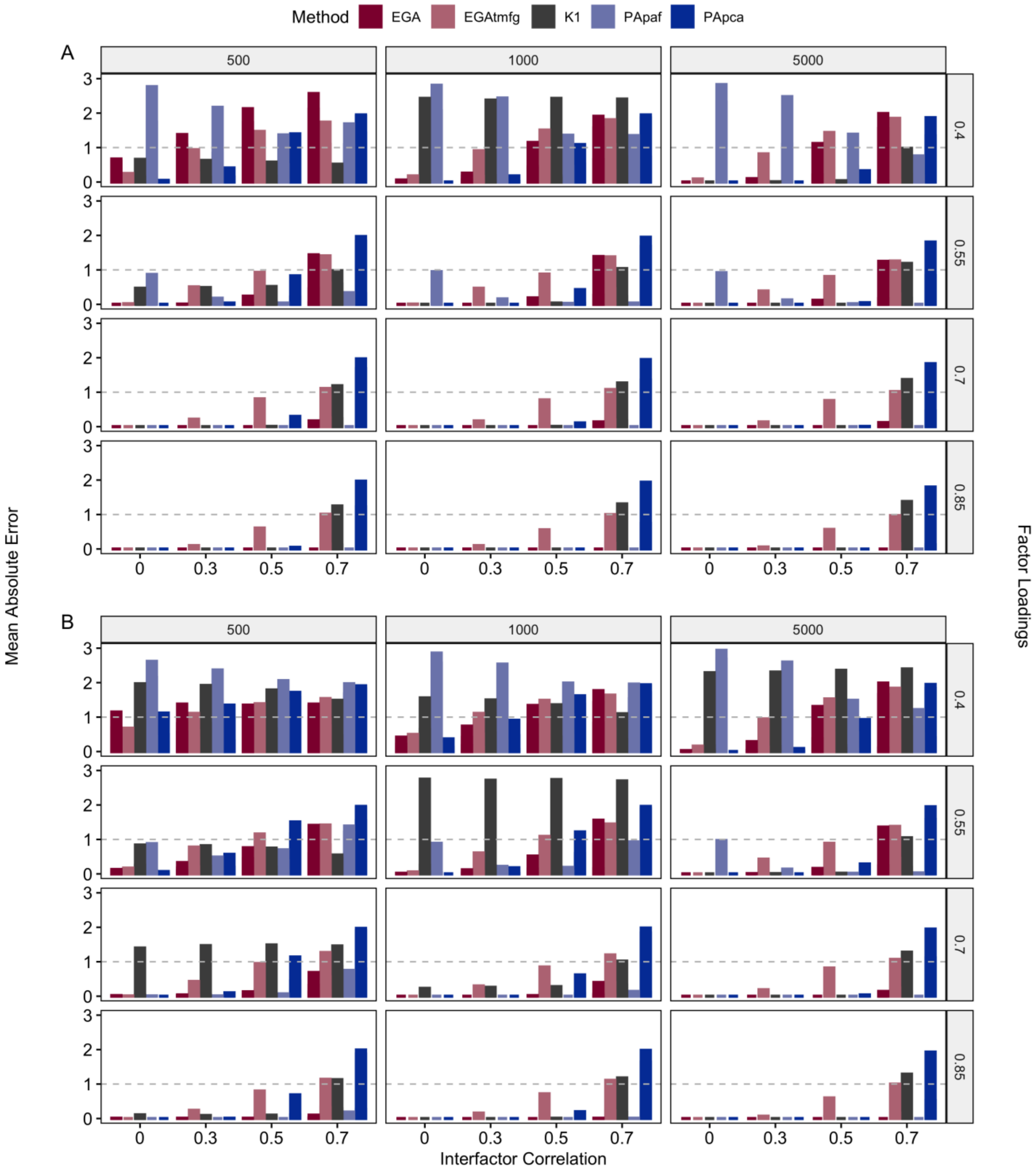


Figure 9. Mean Absolute Error (MAE) per sample size, factor loadings and interfactor correlation for unidimensional factors with continuous (A) and dichotomous (B) data.

Figure 9 shows the bias (MAE) for continuous (Figure 9A) and dichotomous data (Figure 9B). In the continuous data condition, PApaf presented the lowest bias (MAE = 0.28), followed by EGAtmfg (MAE = 0.29), K1 (MAE = 0.32), PApca (MAE = 0.33) and EGA (MAE = 0.45). The bias of the techniques increases with the increase of interfactor correlation, but decreases with higher sample sizes and higher factor loadings. Interestingly, while EGA presented a mean absolute error of 1.62 for loadings of .40, it shrank to 0.15 for loadings of .55 and to 0.01 for loadings of .70 or .85. PApaf had a similar pattern, presenting a mean absolute error of 1.01 for loadings of .40, 0.11 for loadings of .55 and 0 for loadings of .70 or .85. In contrast, PApca presented a mean absolute error of 0.50, 0.33 and 0.24 for loadings of .40, .55 and to .70 or .85, respectively.

Finally, in the dichotomous data condition, EGA presented the lowest bias (MAE = 0.27), followed by EGAtmfg (MAE = 0.38), PApaf (MAE = 0.44), PApca (MAE = 0.52) and K1 (MAE = 0.89). Similarly to the continuous variables, the bias of the techniques increases with the increase of interfactor correlation, but decreases with higher sample sizes and higher factor loadings.

Table 3 shows the effect size for the five most accurate methods (a heatmap version of Table 3 is available in **Appendix F**). It is interesting to note that EGA presents a high effect size for factor loading, both in terms of accuracy and bias. EGAtmfg presents a high effect size for the number of variables and interfactor correlation, while PApaf is more affected by factor loadings. PApca presents a high effect size for interfactor correlation and factor loadings. As with the unidimensional structures, K1 presented a higher number of moderate and high effect sizes.

PLEASE INSERT TABLE 3 ABOUT HERE

In sum, the results revealed that AF and OC presented high accuracy only in the unidimensional conditions, K1 and EGAtmfg presented a moderately good accuracy in both unidimensional and multidimensional structures, and EGA, PApaf, PApca presented higher accuracies in general. The most accurate technique was EGA, with a mean accuracy of 88% accross conditions, followed by PApca (83%) and PApaf (82%).

How to use EGA in R

In order to demonstrate how to implement the new EGA algorithm in R, a brief example will be presented. We will use a dataset that included 2247 people that participated in the Virginia Cognitive Aging Project (VCAP; Salthouse, 2018), who completed the 33-item Social Desirability Scale (SDS; Crowne & Marlowe, 1960) during the first measurement occasion (between 2001 and 2017). The participants' (64.8% women) age ranged from 18 to 97 years old (*M* = 50.72, *SD* = 18.73) and had an average of 15.65 years of education.

To start, the *EGAnet* package can be downloaded and installed from CRAN:

```r
# Install 'EGAnet' package
install.packages("EGAnet")
```

The *EGAnet* package was developed as a simple and easy way to implement the exploratory graph analysis technique. The package has several functions but we will focus on the new `EGA` algorithm in this tutorial. This function simultaneously integrates the algorithm to determine unidimensional and multidimensional structures. The number of dimensions is given by the GLASSO with the lambda parameter set via *EBIC* or using the TMFG method. The number of underlying dimensions (or factors) is detected using the walktrap algorithm.

Arguments of the EGA Function

The new `EGA` function has several arguments: `data`, `model`, `plot.EGA`, `n`, `steps`, `nvar`, `nfact`, `load`, and `...`. The first argument, `data`, is the input of variables, which can be in the form of raw data or an already computed correlation matrix. If the data is a correlation matrix, then the sample size needs to be specified using the `n` argument. The second argument specifies the network estimation `model` to use (either `"glasso"` or `"TMFG"`) and defaults to `"glasso"`. The `plot.EGA` argument determines whether to plot the EGA results (defaults to `TRUE`). Next, the `steps` argument is the number of steps to be used in the walktrap algorithm. This argument defaults to `4`, which is recommended.

```r
# EGA arguments
EGA(data, model = c("glasso", "TMFG"), plot.EGA = TRUE,
    n, steps = 4, nvar = 4, nfact = 1, load = .70, ...)
```

The next three arguments: `nvar`, `nfact`, and `load` are parameters used to simulate data for detecting unidimensionality. `nvar` sets the number of variables (defaults to `4`), `nfact` sets the number of factors (defaults to `1`), and `load` sets the item loadings on each factor (defaults to `.70`). We recommend using the default values when estimating multidimensional structures but adjusting the `nvar` value for unidimensional structures. Our tutorial will provide recommendations for how to do so. Finally, the `...` argument is used to pass additional network estimation arguments into `glasso` or `TMFG` functions. Links to these functions are provided in the `EGA` function's documentation.

Tutorial

The first step is to load the *EGAnet* package. Then, the dataset should be imported into R. In this case, the SDS dataset composed of dichotomous (TRUE/FALSE) variables is saved as a .csv file in the local directory, so the function to import the dataset into R is the `read.csv` function. An object named sds can be created to store the data and, as a last step, the EGA function is used. It is important to note that before importing the dataset the reversed items had been recoded so that all the items have the same direction.

```
# Load 'EGAnet' package
library("EGAnet")
# Read in data
sds <- read.csv("./Datasets/SDS.csv")
# Estimate EGA network
ega.sds <- EGA(data = sds, model = "glasso", plot.EGA = TRUE)
```

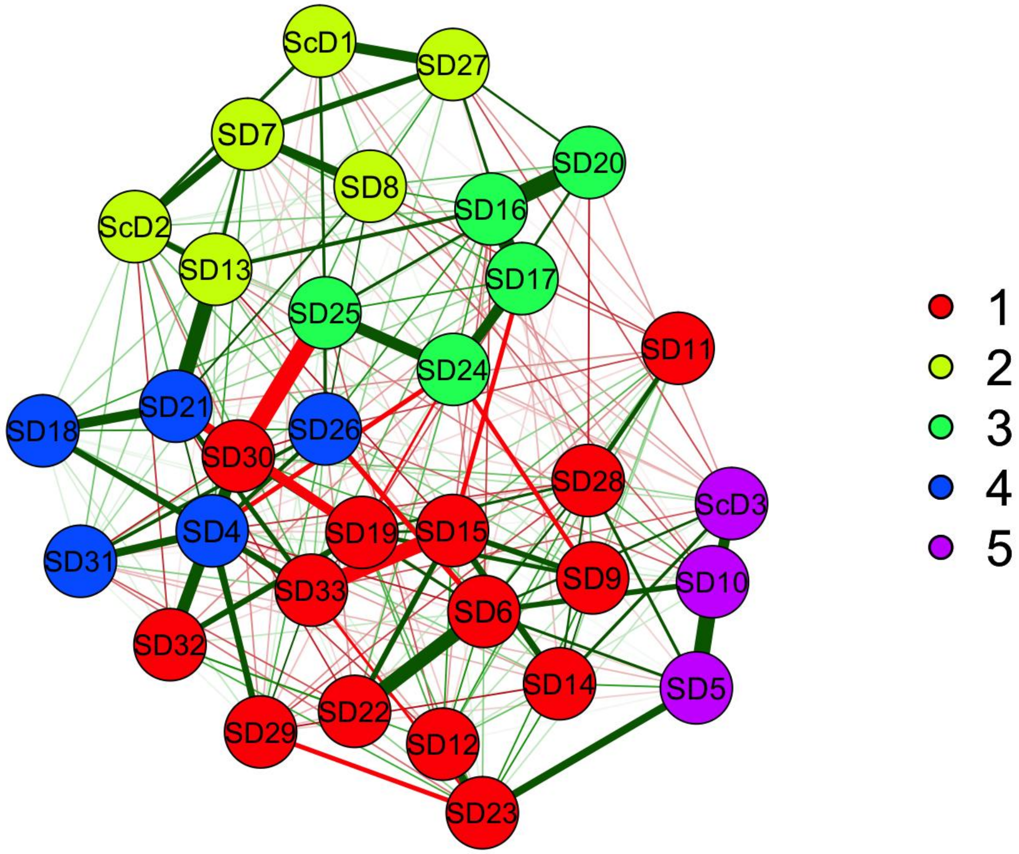


Figure 10. EGA dimesional structure of the Social Desirability Scale.

The results in Figure 10 show five dimensions for the SDS, which can be interpreted as follows. The first dimension (red nodes) reflects behaviors and attitudes that are egoist, insouciant, a little bit manipulative and resentful, with items such as item 19: *I sometimes try to get even rather than forgive and forget*. The second reflects behaviors and attitudes of a cautious and well-mannered people, with items similar to item 27: *I never make a long trip without checking the safety of my car*. The third factor, in turn, indicates a trait of integrity and credibility, with items such as: *I would never think of letting someone else be punished for my wrongdoings* (item 24). The fourth factor indicates a trait of sympathy, generally exhibited by people that are easy to get along with, with items as item 4: *I have never intensely disliked anyone* (item 4). Finally, the fifth factor reflects a low self-esteem trait with items such as item 5: *On occasion I have had doubts about my ability to succeed in life* (item 5).

The results above differs from the most common dimensionality structure of the SDS scale, proposed by Millham (1974), that suggested two constructs of social desirability: one involving self-denial of undesirable characteristics (**denial**) and another involving a tendency to attribute socially desirable characteristics (**attribution**; Ventimiglia & MacDonald, 2012).

To check which structure presents a better fit to the data, the `CFA` function from the `EGAnet` package can be used. This function takes the object generated by the `EGA` function, and fits the corresponding confirmatory factor model using `lavaan` (Rosseel, 2012). The `CFA` function can be used as follows

```
# Fit a confirmatory factor model using an EGA object:
cfa.ega.sds <- CFA(ega.obj = ega.sds, data = sds, estimator = "WLSMV",
                   plot.CFA = FALSE)
# Fit an alternative confirmatory factor model using lavaan,
# but following the approach implemented in the EGA code.

# The first step is to duplicate an EGA object
ega.sds.theory <- ega.sds
# And change the column names of the dim.variables component of the EGA
object
ega.sds.theory$dim.variables[,1] <- colnames(sds)
```

```r
# Select the items that are part of Factor 1
ega.sds.theory$dim.variables[c(3,5, 6, 9, 10, 11, 12, 14, 15, 19, 22, 23,
                               28, 30, 32),2] <- rep(1, 15)
# Select the items that are part of Factor 2:
ega.sds.theory$dim.variables[c(1, 2, 4, 7, 8, 13, 16, 17, 18, 20, 21, 24,
                               25, 26, 27, 29, 31, 33),2] <- rep(2, 18)
# Fit the CFA model:
cfa.sds.theory <- CFA(ega.obj = ega.sds.theory, estimator = 'WLSMV',
                      plot.CFA = FALSE, data = sds)
```

The fit of the CFA model can be inspected using `cfa.ega.sds$fit.measures`, and a plot can be called using the `plot(cfa.ega.sds)`. The five-factor structure estimated using EGA presented the highest CFI (0.97) and the lowest RMSEA (0.03) compared to the theoretical two-factor (*attribution-denial*) model: CFI = 0.95, RMSEA = 0.03).

To determine whether the SDS dimensions described above are unidimensional, we can apply `EGA` and adjust the `nvar` argument for data generation. The default value of `4` was used in the simulation to keep the argument consistent across the conditions. We recommend, however, to adjust this value when testing whether data is unidimensional. We recommend setting `nvar` to the number of variables that are in the dimension being tested. Factor one, for example, had 14 items (Figure 5), so `nvar` should be set to `14`. Factor two had 6 items, factor three and four had 5 items and factor five had 3 items, so `nvar` should be set to `6`, `5`, `5`, and `3`, respectively. We also computed parallel analysis with PAF and PCA using tetrachoric correlations and data generation via resampling from the `psych` package (Revelle, 2018). To demonstrate how to implement this procedure, the following code can be applied:

```r
# Load 'psych' package
library("psych")
# Initialize result vectors
# EGA
ega.res <- vector("numeric", length = max(ega.sds$wc))
# PApaf
papaf.res <- vector("numeric", length = max(ega.sds$wc))
# PApca
```

```
papca.res <- vector("numeric", length = max(ega.sds$wc))
# Run 'for' loop to determine dimensions
for(i in 1:max(ega.sds$wc))
{
  # Identify target items
  target <- which(ega.sds$wc == i)
  # Estimate dimensions
  # EGA
  ega.res[i] <- max(EGA(sds[,target], model = "glasso",
                        plot.EGA = FALSE, nvar = length(target))$wc)
  cap <- capture.output(pa <- fa.parallel(sds[,target], sim = FALSE,
                                          cor = "poly", plot = FALSE))
  # PApaf
  papaf.res[i] <- pa$nfact
  # PApca
  papca.res[i] <- pa$ncomp
}
# Combine and name results
res <- rbind(ega.res, papaf.res, papca.res)
row.names(res) <- c("EGA", "PApaf", "PApca")
colnames(res) <- paste("Factor",1:5)
# Return results
res
```

As the results show in Table 4, EGA and PApca estimated unidimensional structures for all 5 factors, while PApaf only estimated one factor as unidimensional.

These results are consistent with our simulation findings, suggesting that EGA and PApca are effective, while PApaf is inaccurate at estimating unidimensionality in dichotomous data. This tutorial demonstrates how EGA can first be used to detect the number of dimensions in a multidimensional construct. Then, it shows how EGA can be applied to the dimensions identified in a construct to verify that each dimension is indeed unidimensional. For applied researchers, the steps demonstrated in this tutorial are particularly useful for applying EGA to their own dimensional assessments. This has particular implications for scale development and psychometric assessment practices. EGA appears to be robust for both multidimensional and unidimensional assessments, whereas traditional methods such as PApaf and PApca would be necessary to estimate multidimensional and unidimensional structures, respectively. Thus, applied researchers can use EGA as a single, all-around dimension identification approach.

Discussion

The present study examined the dimensionality identification accuracy of two new *exploratory graph analysis* methods (one that can deal with both unidimensional and multidimensional structures, and the other that implements a new network estimation), as well as several traditional factor-analytic techniques, using an extensive Monte Carlo simulation. Aside from manipulating salient variables across ranges of plausible values that may be found in applied settings, all the structures that were generated had varying main factor loadings, cross-loadings, and skewness across items in order to enhance the ecological validity of the simulation. Additionally, previous studies comparing EGA with traditional factor-analytic methods only included dichotomous variables in the simulation design. The current paper also included continuous data, expanding our knowledge about the suitability of EGA as a dimensionality assessment technique compared to traditional methods.

In addition to the Monte-Carlo simulation, a straightforward R tutorial on how to use and interpret EGA was provided, and the method was applied to an empirical dataset composed of scores from a well-known social desirability scale. This study extends previous research for EGA with GLASSO estimation by providing evidence of its accuracy across a broader set of conditions than previously considered, and is the first to examine the performance of EGA in unidimensional structures and the performance of EGA with the TMFG estimation, which emerges as an important complementary technique.

Method Performance

The results from the simulation study revealed that the methods could be classified into three groups: those with high accuracy only in the unidimensional conditions (AF and OC), those with a moderately good accuracy in both unidimensional and multidimensional structures (K1, EGAtmfg) and those with higher accuracies in general (EGA, PApaf, PApca). Of the high performing methods, none was the best across every condition and criteria, and all showed strengths and weaknesses.

Overall, the new EGA algorithm presented the highest accuracy to correctly estimate the number of simulated factors, and the lowest mean bias error. It is important to note that the new EGA algorithm can adequately deal with unidimensional structures, a condition that the original EGA method proposed by Golino and Epskamp (2017) could not handle. At the same time, the new EGA algorithm was implemented in a way that doesn't change the original EGA method if the data presents more than two factors. Both EGA and EGAtmfg performed similarly to the most accurate traditional technique, parallel analysis, in a number of conditions.

The new EGA algorithm (using the GGM model) was the most accurate method with medium (.55), and the second best with high (.70) and very high (.85) factor loadings, followed closely by PApaf. Also, of the five best methods, EGA and PApaf were the two most robust to the factor correlations, sustaining the smallest decreases in accuracy with higher factor correlations. The excellent performance of EGA in these conditions is in line with previous research (Golino & Epskamp, 2017). With low loadings (.40) combined with smaller samples (500), however, the performance of EGA was lower, but still presented rates of correct estimates that were in line with those of the other well performing methods. Recent developments in the area of network psychometrics seems to improve the estimation of the GGM model to deal with low sample sizes and large number of variables (Williams, 2018; Williams & Rast, 2019). Future studies should investigate how these new GGM estimation procedures can improve the accuracy of EGA, especially in conditions with low sample size, low factor loadings and moderate or high interfactor correlation.

EGA with TMFG provided correct dimensionality estimates just below that of the other high performing methods, but its most notable characteristic was that its estimates, along those of the new EGA and PApaf, were the closest to the population values. In comparison to the other good performing methods, EGAtmfg was at its best in the unidimensional structures for fewer variables per factor, and in the multidimensional conditions it was best for structures weaker factor correlations (≤ .50), and eight variables per factor. In contrast, the biggest limitations of EGAtmfg came from structures that were composed of many variables per factor, and with highly correlated factors. It is likely that these conditions create problems for EGAtmfg due to the way it constructs the network, through the formation of tetrahedrons (groups of four nodes), which severely limits (or enforces) cross-dimension connections. Future simulations should examine a new method that constructs the network in a similar way as the TMFG but eliminates its artificial structural constraint (i.e., 3- and 4-node cliques; Massara & Aste, 2019).

In terms of the two PA methods, they generally performed well, thus extending the vast literature supporting the accuracy of this procedure (e.g., Garrido et al., 2013, 2016; Timmerman & Lorenzo-Seva, 2011). Comparing both parallel analysis methods, it’s interesting to point that while PApca was more accurate in the unidimensional conditions, PApaf was more robust in the multidimensional conditions, especially with higher interfactor correlations. These two methods complemented each other, with one being stronger where the other was weaker, and vice versa (e.g., for factor loadings, variables per factor, and factor correlations). In the case of PApca, the method showed a clear bias in the condition of multiple factors, few variables per factor (3 or 4) combined with moderate (.50) or very high factor correlations (.70). In these cases the method will generally produce a one-factor estimate regardless of the actual dimensionality of the data. The reason for this is simple: the population eigenvalues after that corresponding to the first factor will be lower than one, and thus, asymptotically PApca is not able to retain them. In terms of PApaf, it produced comparatively poorest performance with low factor loadings (.40).

It is important to note that PApca, which is generally a well performing dimensionality method, is biased at the population level for models with high factor correlations. The null model used to compute the reference eigenvalues only constitutes a strictly adequate reference for the first observed eigenvalue (Braeken & Van Assen, 2017). The values of subsequent eigenvalues for the data under consideration are conditional upon the structure in the data captured by previous eigenvalues. Particularly, when factors are highly correlated and the number of variables is small, the first eigenvalue will be very large, whereas succeeding eigenvalues will be necessarily notably smaller (as the sum of the eigenvalues is always constrained to be equal to the total variance). This situation will give rise to scenarios where the eigenvalues from major factors after the first will be lower than the reference eigenvalues at the population level, thus limiting the accuracy of the method for these conditions. EGA, in contrast, performs considerably more accurately in these conditions.

It is also interesting to note that the automated scree methods presented a very high accuracy in the unidimensional conditions, but moderately low accuracies in the multidimensional conditions. Their percentage of correct estimates was between 20% and 30% below to that of the EGA and PA methods. The AF method was one of the most accurate methods for orthogonal structures and for single factors (unidimensional structures), but its accuracy shrinks as the interfactor correlation increases. In the case of K1, the method tended to overestimate the population dimensionality by very large amounts, as has been widely documented in the literature (Costello & Osborne, 2005). Surprisingly, the accuracy of K1 in the current simulation was not bad. This can be explained by the use of three and four variables per factor in the simulation design, a condition in which K1 presents higher accuracies. However, the results of the present study show very clear that the K1 technique should be avoided in situations where the number of variables per factor is relatively high, and the factor loadings are small or moderate. A similar pattern was identified for MAP and VSS (see **Appendix D**). MAP presented a moderately low accuracy for 2 (52.5%), 3 (47.4%) and 4 (44.4%) factors, while VSS presented very low accuracies (14.7%, 7.3% and 5.9%, respectively). However, MAP presented a very high accuracy for unidimensional structures (99.7%), and VSS followed in the same direction (91%).

The current paper presents limitations that should be addressed in future studies. A question that remains open regards the accuracy of the EGA techniques compared to PApaf and PApca when the simulated data has a complex structure where items have large loadings on more than one factor. Also, little is known about the accuracy of EGA in the presence of population error. Lim and Jahng (2019), for example, investigated several variants of parallel analysis, and discovered that the majority of the PA methods presented much lower accuracies in the presence of population error. Both the issue of complex factor structures and population error should be addressed in future studies comparing EGA and PA techniques.

EGA in Practice

Which EGA method should be used with empirical data? In this section we will provide some practical recommendations to guide researchers in the implementation of EGA and EGAtmfg. On one hand, it is useful to always compute both EGA and EGAtmfg and see if their estimates agree. In our simulation, 58.0% of the cases where EGA erred it did so by overfactoring, while in 85.6% of the cases that EGAtmfg erred it was due to underfactoring. Thus, when the methods agree it is likely because they have found the optimal solution. For example, in this study EGA and EGAtmfg provided the same estimate for 78% of the datasets, and for these, their accuracy was nearly perfect (PC = 91.85%, MAE = 0.10). Therefore, if both EGA and EGAtmfg produce the same dimensionality estimate researchers can have increased confidence that the solution suggested is optimal, or if not, very close to it. On the other hand, when the two methods disagreed in the present study the accuracy of EGA (PC = 73.73%, MAE = .82) decreases and EGAtmfg (PC = 12.94%, MAE = 1.07) significantly decreases. In these instances when EGA and EGAtmfg provide different estimates in practice, researchers can look at the line plots presented in Figures 5 and 7 to see the method that is likely to perform better in the conditions that they think most apply to their data. Additionally, in these cases where EGA and EGAtmfg disagree, it is important to more strongly consider potential alternative solutions (with less or more dimensions, respectively) to those suggested by the methods. In particular, to help the researchers decide which dimensionality estimate is better, a fit index was recently developed specifically for EGA (Golino et al., 2019) and could be used to check which dimensionality structure (i.e., estimated using EGA or EGAtmfg) fits the the data better. Lastly, researchers could also use PApaf to check if the number of factors matches the number of factors estimated using the EGA techniques (Garcia-Garzon et al., 2019b).

Conclusion

This paper describes the EGA method and shows, through an extension simulation, that it performs as well as the best factor-analytic techniques. On top of excellent performance, EGA possess several advantages over traditional methods. First, with EGA, researchers do not need to decipher a factor loading matrix but instead can immediately interpret which items belong to which factor with the color-coded network plot. Second, EGA does not require the researcher to make any decisions about the type of rotation to use for the factor structure. There are an enormous number of factor rotations for researchers to chose from, which can make it difficult for researchers to know whether they are using the appropriate rotation method. Third, EGA is a single step approach and does not require additional steps to verify factors, while with traditional methods, the number of dimensions are estimated first and then are followed by exploratory factor analysis with the specified number of dimensions. These last two advantages ultimately reduce the number of researcher degrees of freedom and eliminate most of the potential for bias and errors. In sum, we show that EGA is a promising method for accurate dimensionality estimation.

References

Anderson, H., T.W. & Rubin. (1958). Statistical inference in factor analysis. In *Proceedings of the 3rd berkeley symposium on mathematics, statistics, and probability* (Vol. 5, pp. 111–150).

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological Methods*, *24*, 468–491. https://doi.org/10.1037/met0000200

Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown*. Retrieved from https://github.com/crsh/papaja

Barfuss, W., Massara, G. P., Di Matteo, T., & Aste, T. (2016). Parsimonious modeling with information filtering networks. *Physical Review E*, *94*(6), 062306.

Beierl, B., E. T. (2018). Is that measure really one-dimensional? Nuisance parameters can mask severe model misspecification when assessing factorial validity. *Methodology*, *14*(4), 188–196.

Braeken, J., & Van Assen, M. A. (2017). An empirical kaiser criterion. *Psychological Methods*, *22*(3), 450.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, *1*(2), 245–276.

Chen, J., & Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, *95*(3), 759–771.

Christensen, A. P., Kenett, Y. N., Aste, T., Silvia, P. J., & Kwapil, T. R. (2018). Network structure of the wisconsin schizotypy scales–Short forms: Examining psychometric network filtering approaches. *Behavior Research Methods*, *50*(6), 2531–2550. https://doi.org/doi:10.3758/s13428-018-1032-9

Cohen, J. (1988). Statistical power analysis for the behavioral sciences. 2nd. Hillsdale, NJ: Erlbaum.

Comrey, A. L., & Lee, H. B. (2016). *A first course in factor analysis*. New York: Routledge.

Costello, A. B., & Osborne, J. W. (2005). Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research & Evaluation*, *10*(7), 1–9.

Crawford, A. V., Green, S. B., Levy, R., Lo, W.-J., Scott, L., Svetina, D., & Thompson, M. S. (2010). Evaluation of parallel analysis methods for determining the number of factors. *Educational and Psychological Measurement*, *70*(6), 885–901.

Crowne, D., & Marlowe, D. (1960). A new scale of social desirability independent of psychopathology. *Journal of Consulting Psychology*, *24*(4), 349.

Epskamp, S., & Fried, E. (2018). A tutorial on regularized partial correlation networks. *Psychological Methods*, *23*(4), 617–634. https://doi.org/10.1037/met0000167

Epskamp, S., Cramer, A. O. J., Waldorp, L. J., Schmittmann, V. D., & Borsboom, D. (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, *48*(4), 1–18. Retrieved from http://www.jstatsoft.org/v48/i04/

Epskamp, S., Rhemtulla, M., & Borsboom, D. (2017). Generalized network pschometrics: Combining network and latent variable models. *Psychometrika*, *82*(4), 904–927.

Epskamp, S., Waldorp, L. J., Mõttus, R., & Borsboom, D. (2018). The gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, *53*(4), 453–480. https://doi.org/10.1080/00273171.2018.1454823

Foygel, R., & Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. In *Proceedings of the 23rd international conference on neural information processing systems - volume 1* (Vol. 1, pp. 604–612). Vancouver, Canada.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441.

Fruchterman, T. M. J., & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, *21*, 1129–1164. https://doi.org/10.1002/spe.4380211102

Garcia-Garzon, E., Abad, F. J., & Garrido, L. E. (2019a). Improving bi-factor exploratory modelling: Empirical target rotation based on loading differences. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, *15*(2), 45–55. https://doi.org/10.1027/1614-2241/a000163

Garcia-Garzon, E., Abad, F. J., & Garrido, L. E. (2019b). Searching for g: A new evaluation of spm-ls dimensionality. *Journal of Intelligence*, *7*(3), 14. https://doi.org/10.3390/jintelligence7030014

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2013). A new look at horn's parallel analysis with ordinal variables. *Psychological Methods*, *18*(4), 454–74. https://doi.org/10.1037/a0030005

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2016). Are fit indices really fit to estimate the number of factors with categorical variables? Some cautionary findings via monte carlo simulation. *Psychological Methods*, *21*(1), 93–111. https://doi.org/http://dx.doi.org/10.1037/met0000064

Gates, K. M., Henry, T., Steinley, D., & Fair, D. A. (2016). A monte carlo evaluation of weighted community detection algorithms. *Frontiers in Neuroinformatics*, *10*, 45. https://doi.org/10.3389/fninf.2016.00045

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS One*, *12*(6), e0174035.

Golino, H., & Christensen, A. P. (2019). *EGAnet: Exploratory graph analysis: A framework for estimating the number of dimensions in multivariate data using network psychometrics*. Retrieved from https://CRAN.R-project.org/package=EGAnet

Golino, H., & Demetriou, A. (2017). Estimating the dimensionality of intelligence like data using exploratory graph analysis. *Intelligence*, *62*, 54–70.

Golino, H., Moulder, R., Shi, D., Christensen, A., Neito, M., Nesselroade, J. R., & Boker, S. (2019). Entropy fit index: A new fit measure for assessing the structure and dimensionality of multiple latent variables. *PsyArXiv*. https://doi.org/10.31234/osf.io/mtka2

Hayton, J. C., Allen, D. G., & Scarpello, V. (2004). Factor retention decisions in exploratory factor analysis: A tutorial on parallel analysis. *Organizational Research Methods, 7*(2), 191–205.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, *30*(2), 179–185.

Humphreys, L. G., & Ilgen, D. R. (1969). Note on a criterion for the number of common factors. *Educational and Psychological Measurement*, *29*(3), 571–578.

Izquierdo, I., Olea, J., & Abad, F. J. (2014). Exploratory factor analysis in validation studies: Uses and recommendations. *Psicothema*, *26*(3), 395–400.

Jackson, D. L., Gillaspy Jr, J. A., & Purc-Stephenson, R. (2009). Reporting practices in confirmatory factor analysis: An overview and some recommendations. *Psychological Ethods*, *14*(1), 6–23.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*(1), 141–151.

Kane, M. J., Hambrick, D. Z., & Conway, A. R. (2005). Working memory capacity and fluid intelligence are strongly related constructs: Comment on ackerman, beier, and boyle (2005). *Psychological Bulletin*, *131*, 66–77. https://doi.org/http://dx.doi.org/10.1037/0033-2909.131.1.66

Kassambara, A. (2017). *Ggpubr: 'Ggplot2' based publication ready plots*. Retrieved from https://CRAN.R-project.org/package=ggpubr

Lauritzen, S. L. (1996). *Graphical models* (Vol. 17). Oxford: Clarendon Press.

Li, C.-H. (2016). Confirmatory factor analysis with ordinal data: Comparing robust maximum likelihood and diagonally weighted least squares. *Behavior Research Methods*, *48*(3), 936–949.

Lim, S., & Jahng, S. (2019). Determining the number of factors using parallel analysis and its recent variants. *Psychological Methods*, *24*(4), 452–467.

Lubbe, D. (2019). Parallel analysis with categorical variables: Impact of category probability proportions on dimensionality assessment accuracy. *Psychological Methods*, *24*(3), 339–351. https://doi.org/http://dx.doi.org/10.1037/met0000171

Massara, G. P., & Aste, T. (2019). Learning clique forests. *arXiv*. Retrieved from https://arxiv.org/abs/1905.02266

Massara, G. P., Di Matteo, T., & Aste, T. (2016). Network filtering for big data: Triangulated maximally filtered graph. *Journal of Complex Networks*, *5*(2), 161–178.

Meade, A. W. (2008). Power of afi's to detect cfa model misfit. In *Paper presented at the 23th annual conference of the society for industrial and organizational psychology*. San Francisco, CA. Retrieved from pdfs.semanticscholar.org/a23c/45ca18db70125a9a0ad983926513d40fa32b.pdf

Meyers, L. S., Gamst, G., & Guarino, A. J. (2016). *Applied multivariate research: Design and interpretation*. Thousand Oaks: SAGE Publications.

Millham, J. (1974). Two components of need for approval score and their relationship to cheating following success and failure. *Journal of Research in Personality*, *8*(4), 378–392.

Muthén, B., & Kaplan, D. (1992). A comparison of some methodologies for the factor analysis of non-normal likert variables: A note on the size of the model. *British Journal of Mathematical and Statistical Psychology*, *45*(1), 19–30.

Newman, M. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*. https://doi.org/10.1103/PhysRevE.69.066133

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, *10*(2), 191–218.

R Core Team. (2017). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Raiche, G. (2010). *An r package for parallel analysis and non graphical solutions to the cattell scree test*. Retrieved from http://CRAN.R-project.org/package=nFactors

Raiche, G., Walls, T. A., Magis, D., Riopel, M., & Blais, J.-G. (2013). Non-graphical solutions for cattell's scree test. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, *9*(1), 23–29. https://doi.org/10.1027/1614-2241/a000051

Revelle, W. (2018). *Psych: Procedures for psychological, psychometric, and personality research*. Evanston, Illinois: Northwestern University. Retrieved from https://CRAN.R-project.org/package=psych

Revelle, W., & Rocklin, T. (1979). Very simple structure: An alternative procedure for estimating the optimal number of interpretable factors. *Multivariate Behavioral Research*, *14*(4), 403–414.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. Retrieved from http://www.jstatsoft.org/v48/i02/

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. *Psychological Assessment*, *24*(2), 282–292. https://doi.org/10.1037/a0025697

Salthouse, T. (2018). The virginia cognitive aging project. Retrieved from http://www.mentalaging.com

Sass, D. A., & Schmitt, T. A. (2010). A comparative investigation of rotation criteria within exploratory factor analysis. *Multivariate Behavioral Research*, *45*(1), 73–103.

Song, W.-M., Di Matteo, T., & Aste, T. (2011). Nested hierarchies in planar graphs. *Discrete Applied Mathematics*, *159*(17), 2135–2146.

Song, W.-M., Di Matteo, T., & Aste, T. (2012). Hierarchical information clustering by means of topologically embedded graphs. *PLoS One*, *7*(3), e31929.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

Timmerman, M. E., & Lorenzo-Seva, U. (2011). Dimensionality assessment of ordered polytomous items with parallel analysis. *Psychological Methods*, *16*(2), 209–220.

Tumminello, M., Aste, T., Di Matteo, T., & Mantegna, R. N. (2005). A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(30), 10421–10426. https://doi.org/10.1073/pnas.0500298102

Velicer, W. F. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika*, *41*(3), 321–327.

Ventimiglia, M., & MacDonald, D. A. (2012). An examination of the factorial dimensionality of the marlowe crowne social desirability scale. *Personality and Individual Differences*, *52*(4), 487–491.

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from http://ggplot2.org

Widaman, K. F. (1993). Common factor analysis versus principal component analysis: Differential bias in representing model parameters? *Multivariate Behavioral Research*, *28*(3), 263–311.

Williams, D. R. (2018). Bayesian inference for gaussian graphical models: Structure learning, explanation, and prediction. *PsyArXiv*. https://doi.org/10.31234/osf.io/x8dpr

Williams, D. R., & Rast, P. (2019). Back to the basics: Rethinking partial correlation network methodology. *British Journal of Mathematical and Statistical Psychology [Epub Ahead of Print]*. https://doi.org/10.1111/bmsp.12173

Woodbury, M. (1950). *Inverting modified matrices* (Vol. 42, pp. 99–117). Statistical Research Group, Memo. Rep. no. 42, Princeton University, Princeton, N. J.

Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, *6*, 30750.

Appendix A

Walktrap Community Detection

To define the random walk, let **A** be a square matrix of edge weights (e.g., partial correlations) in the network, where $\mathbf{A}_{ij}$ is the strength of the (partial) correlation between node i and j and a node's strength is the sum of node i's connections to its neighbors $NS = \sum_j A_{ij}$. The steps move from one node to another randomly and uniformly using a transition probability, $\mathbf{P}_{ij} = \frac{\mathbf{A}_{ij}}{NS(i)}$, which forms the transition matrix, **P**.

To determine the communities that the nodes belong to, the transition matrix is used to compute a distance metric, r, which measures the structural similarity between nodes (1). This structural similarity is defined as (Pons & Latapy, 2006):

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(\mathbf{P}_{ik} - \mathbf{P}_{jk})^2}{NS(k)}} \quad \text{(A1)}$$

This distance can be generalized to the distance between nodes and communities by beginning the random walk at a random node in a community, C. This can be defined as:

$$\mathbf{P}_{C_j} = \frac{1}{|C|} \sum_{i \in C} \mathbf{P}_{ij} \,. \quad \text{(A2)}$$

Finally, this can be further generalized to the distance between two communities:

$$r_{C_1 C_2} = \sqrt{\sum_{k=1}^{n} \frac{(\mathbf{P}_{C_1 k} - \mathbf{P}_{C_2 k})^2}{NS(k)}}, \quad \text{(A3)}$$

where this definition is consistent with the distance between nodes in the network (Eq. A1).

Algorithm

The algorithm begins by having each node as a cluster (i.e., n clusters). The distances, r, are computed between all adjacent nodes, and the algorithm then begins to iteratively choose two clusters. These two clusters chosen are then merged into a new cluster, updating the distances between the node(s) and cluster(s) with each merge (in each $k = n - 1$ steps).

Clusters are only merged if they are adjacent to one another (i.e., an edge between them). The merging method is based on Ward's agglomerative clustering approach (Ward, 1963) that depends on the estimation of the squared distances between each node and its community (σ_k), for each k steps of the algorithm. Since computing σ_k is computationally expensive, Pons and Latapy (2006) adopted an efficient approximation that only depends on the nodes and the communities rather than the k steps. The approximation seeks to minimize the variation of σ that would be induced if two clusters (C_1 and C_2) are merged into a new cluster (C_3):

$$\varDelta\sigma(C_1, C_2) = \frac{1}{n}\left(\sum_{i\in C_3} r_{iC_3}^2 - \sum_{i\in C_1} r_{iC_1}^2 - \sum_{i\in C_2} r_{iC_2}^2\right). \quad \text{(A4)}$$

Since Ward's approximation adopted by Pons and Latapy (2006) only merges adjacent clusters, the total number of times $\varDelta\sigma$ is updated is not very large, and the resulting values can be stored in a balanced tree. A sequence of P_k partitions into clusters ($1 \leq k \leq n$, being n the total number of nodes) is obtained. The best number of clusters is defined as the partition that maximizes modularity.

Modularity is a measure that was proposed by Newman (2004) to identify meaningful clusters in networks and is calculated as follows. Let j and k be two clusters in a network with m and n nodes. If the number of edges between clusters is p, then one-half of fraction of the edges linking j and k is $e_{jk} = \frac{1}{2}p$, so that the total fraction of edges between the two clusters is $e_{jk} + e_{jk}$ (Newman, 2004). On the other hand, e_{jj} represents the fraction of edges that fall within cluster j, whose sum equals one: $\sum_j e_{jj} = 1$. Newman (2004) points out that a division of networks into clusters is meaningful if the value of the sums of e_{jj} and e_{ii} is maximized. However, in cases where only one cluster is presented, the maximal value will be one, which is also the value of $\sum_j e_{jj}$. Therefore, for networks composed by only one cluster this index is not informative. A solution Newman (2004) proposed was to calculate an index that takes $\sum_j e_{jj}$ and subtract from it the value that it would take if edges were placed at random. For a given cluster j, the modularity is calculated as:

$$Q = \sum_j (e_{jj} - a_j^2)\,, \quad \text{(A5)}$$

where a_j is given by $\sum_i e_{ij}$. Therefore, the modularity index penalizes network structures with only one cluster, since in this condition the value of Q would be zero (Newman, 2004).

Appendix B

For *p* number of variables, the OC procedure aims to identify the actual factors by computing *p–2* two-point regression models, and verifying if the eigenvalue in question is greater than the one estimated by these models. The last positive verification, starting from the second eigenvalue, and continuing without interruption, is used to determine the number of factors to retain. The predicted eigenvalue $\hat{\lambda}_i$, known as the optimal coordinate, is estimated through the linear regression model using only the last eigenvalue and the $(i+1)^{th}$ eigenvalue so that

$$\hat{\lambda}_i = a_{(i+1)} + b_{(i+1)}(i) \quad \text{(B1)}$$

with

$$b_{(i+1)} = (\lambda_p - \lambda_{(i+1)})/(p - i - 1) \quad \text{(B2)}$$

and

$$a_{(i+1)} = \lambda_{(i+1)} - b_{(i+1)}(i+1). \quad \text{(B3)}$$

On the other hand, the AF method searches for the point in the eigenvalue plot where the slope of the curve changes abruptly. In order to achieve this, the AF evaluates an approximation to the second derivative of the OC equation,

$$\hat{\lambda}_i = a_{(i+1)} + b_{(i+1)}(i), \quad \text{(B4)}$$

at each of the i eigenvalues (from 2 to *p* - 1) using the function

$$f'(i) = f(i+1) - 2f(i) - f(i-1). \quad \text{(B5)}$$

Additionally, Raiche, Walls, Magis, Riopel, and Blais (2013) complement the OC and AF methods with the K1 rule or PApca, such that no eigenvalues are retained that are below one (K1) or below the eigenvalue obtained from independent variates (PApca).

Appendix C

Data simulation approach

First, the reproduced population correlation matrix (with communalities in the diagonal) was computed:

$$\mathbf{R_R} = \mathbf{\Lambda\Phi\Lambda}', \quad \text{(C1)}$$

where $\mathbf{R_R}$ is the reproduced population correlation matrix, *lambda* ($\mathbf{\Lambda}$) is the measurement model (i.e. a $k \times r$ factor loading matrix for k variables and r factors) and *phi* ($\mathbf{\Phi}$) is the structure matrix of the latent variables (i.e. a $r \times r$ matrix of correlations among factors). The population correlation matrix $\mathbf{R_P}$ was then obtained by inserting unities in the diagonal of $\mathbf{R_R}$, thereby raising the matrix to full rank. The next step was performing a Cholesky decomposition of $\mathbf{R_P}$, such that:

$$\mathbf{R_P} = \mathbf{U'U}. \quad \text{(C2)}$$

If either $\mathbf{R_P}$ was not positive definite (i.e., at least one eigenvalue was ≤ 0) or an item's communality was greater than 0.90, the $\mathbf{\Lambda}$ matrix was replaced and a new $\mathbf{R_P}$ matrix was computed following the same procedure. Subsequently, the sample data matrix of continuous variables was computed as:

$$\mathbf{X} = \mathbf{ZU}, \quad \text{(C3)}$$

where $\mathbf{Z}$ is a matrix of random standard normal deviates with rows equal to the sample size and columns equal to the number of variables.

Appendix D

Overall, the convergence rates (CRs) of the EGA analysis are high across most conditions. Those with lower CRs are small factor loading conditions (i.e., loadings = 0.4) associated with small to medium sample size (i.e., N=500 or 1000). This is expected as the results are consistent with the performance of EGA, where EGA works best with medium to high factor loadings or small loadings with large sample size. We think the reason for the nonconvergence could be related to the GLASSO regularization procedure. This pattern is consistent for both unidimensional and multidimensional conditions.

Among the small loading and small sample conditions, in multidimensional conditions, the number of factors affects the CRs. The more the factors, the lower the CRs tend to be. Furthermore, consistent with the performance of EGA, CRs for medium to high factor loading conditions (i.e., loadings = 0.55, 0.7 or 0.85) are very high, with occasionally a few non-converged conditions when loadings = 0.5 and sample size is small. All unidimensional cases with medium to high loadings have 100% CRs. In sum, the CR was 97% for the multidimensional and 99.59% for the unidimensional structures.

Appendix E

Table E1. *Mean accuracy (PC) for EGA, EGAtmfg, VSS and MAP*

Method	NFAC	Mean	SD
EGA	1	0.96	0.20
EGA	2	0.82	0.39
EGA	3	0.84	0.37
EGA	4	0.79	0.41
EGAtmfg	1	0.79	0.41
EGAtmfg	2	0.70	0.46
EGAtmfg	3	0.74	0.44
EGAtmfg	4	0.64	0.48
VSS	1	0.92	0.28
VSS	2	0.15	0.35
VSS	3	0.07	0.26
VSS	4	0.06	0.24
MAP	1	1.00	0.05
MAP	2	0.52	0.50
MAP	3	0.47	0.50
MAP	4	0.44	0.50

Table E2. *Mean Bias Error (MBE) for EGA, EGAtmfg, VSS and MAP*

Method	NFAC	Mean	SD
EGA	1	0.07	0.20
EGA	2	-0.09	0.39
EGA	3	-0.13	0.37
EGA	4	-0.20	0.41
EGAtmfg	1	0.27	0.41
EGAtmfg	2	-0.24	0.46
EGAtmfg	3	-0.28	0.44
EGAtmfg	4	-0.43	0.48
VSS	1	0.19	0.28
VSS	2	1.41	0.35
VSS	3	1.43	0.26
VSS	4	0.99	0.24
MAP	1	0.00	0.05
MAP	2	-0.47	0.50
MAP	3	-1.00	0.50
MAP	4	-1.59	0.50

Table E3. *Mean Absolute Error (MAE) for EGA, EGAtmfg, VSS and MAP*

Method	NFAC	Mean	SD
EGA	1	0.07	0.20
EGA	2	0.20	0.39
EGA	3	0.25	0.37
EGA	4	0.35	0.41
EGAtmfg	1	0.27	0.41
EGAtmfg	2	0.31	0.46
EGAtmfg	3	0.35	0.44
EGAtmfg	4	0.48	0.48
VSS	1	0.19	0.28
VSS	2	2.01	0.35
VSS	3	2.92	0.26
VSS	4	3.45	0.24
MAP	1	0.00	0.05
MAP	2	0.48	0.50
MAP	3	1.01	0.50
MAP	4	1.59	0.50

Appendix F

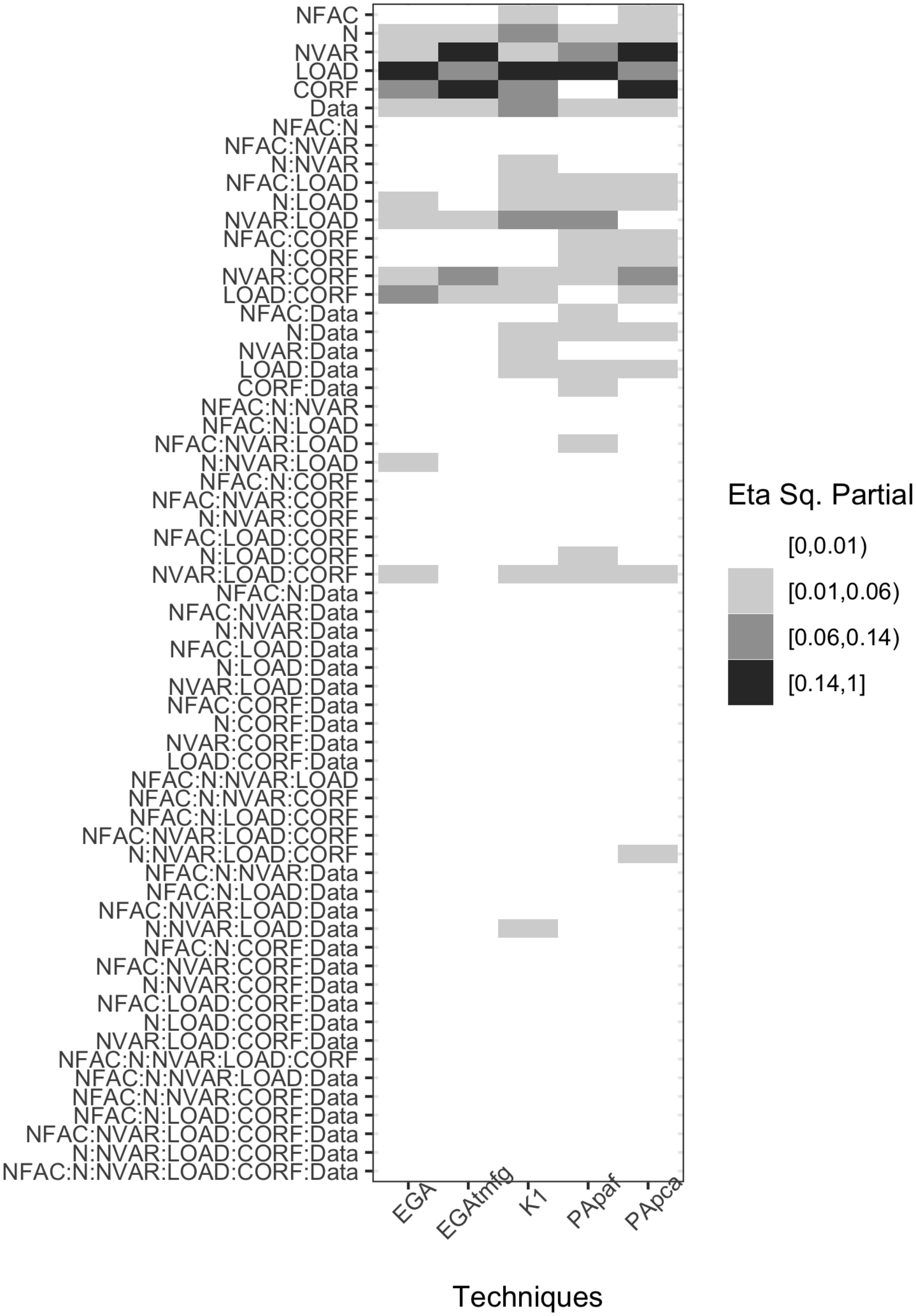


Figure F1. Effect Size - Multidimensional Structures

References

Newman, M. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*. https://doi.org/10.1103/PhysRevE.69.066133

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, *10*(2), 191–218.

Raiche, G., Walls, T. A., Magis, D., Riopel, M., & Blais, J.-G. (2013). Non-graphical solutions for cattell’s scree test. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, *9*(1), 23–29. https://doi.org/10.1027/1614-2241/a000051

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244. https://doi.org/10.2307/2282967