Chapter 40

Machine-Learning Approaches to Structural Equation Modeling

Andreas M. Brandmaier & Ross C. Jacobucci

This is a draft of Chapter 40 to be included in the book entitled Handbook of Structural Equation Modeling, edited by Rick H. Hoyle with Guilford.

In social and behavioral research, theory construction is often thought to proceed in three stages: theory generation, theory development, and theory appraisal (Haig, 2014). Exploratory factor analysis (EFA) is often used as a means to develop theory, whereas structural equation modeling (SEM) more broadly, and confirmatory factor analysis (CFA) more specifically, are typically used for theory appraisal. Though theory generation and theory appraisal represent the ends of the exploratory-confirmatory research continuum, respectively, it is this middle area, theory development, that is often neglected with respect to statistical method development. In this chapter, we discuss two methods developed specifically for this purpose: regularized SEM and SEM trees. Both methods extend structural equation models to incorporate algorithms that fall under the umbrella of machine learning. They each make theory development structured, efficient, and reproducible. Both approaches are also well suited to handle larger numbers of variables and allow for the development and assessment of complex theories.

Machine learning brings about a paradigm shift from explanation to prediction (Breiman, 2001; Shmueli, 2010; Yarkoni & Westfall, 2017) by using flexible forms of regression that allow more freedom to discover patterns in data (Berk, 2008). In machine learning, we abandon the idea of being able to specify a true model and instead focus on models that predict well. Integrating machine learning with SEM can be seen as a blending of explanation and prediction, as explanation and prediction are not competing goals of scientific research, but each can be used as supplements to bolster the support of the other (Shmueli, 2010). We describe this new area of research as a fusion of explanation and prediction due to SEM being difficult to justify in a non-explanatory domain (e.g., Rigdon, 2016), while also not lending itself to prediction in a strict sense (generating observation-level expected values; Shmueli, Ray, Estrada, & Chatla, 2016). Rather, our aim is to improve our ability to model an outcome, either in the form of a single

variable or a multivariate model (e.g., CFA model), while retaining the theoretical aims that informed data collection and model specification. Machine learning algorithms can be used not only in concert with SEM, but also to improve explanatory models by providing a reference model with which a purely theory-driven model must compete. If a purely theory-based model fits much worse than one created with machine learning, it suggests the theory may be incomplete. Ideally, we can use information obtained from a predictive model to inform our process of theory building, which mimics the idea of the Inductive-Hypothetico-Deductive spiral (Cattell, 1966; Tellegen & Waller, 2008).

Although atheoretical research (i.e., exploration) is frequently described in a negative light and often equated with fishing for results, we see an explicit role for this type of research as a complement for theory-heavy SEM. As discussed earlier, the stage of theory development often involves testing a number of variables to assess confounding or heterogeneity, along with other characteristics of the data that present possible ways one might expand or amend theory. The chief concern with an atheoretical or exploratory model is that of overfitting (or equivalently, a lack of generalizability). However, machine learning research has long recognized the propensity for the highly flexible algorithms to overfit the sample data; thus, a number of measures have been developed to both assess and prevent overfitting (e.g., cross-validation). Both regularized SEM and SEM trees incorporate multiple mechanisms to prevent generating results that are unlikely to generalize.

The structure of this chapter is as follows. Prior to providing background on both regularized SEM and SEM trees, we first describe the empirical example that we will use throughout the chapter to illustrate the two methods. This is followed by detailing the results of applying both methods to the empirical example. We conclude with a more general discussion,

while highlighting other related methods.

LONGITUDINAL CHANGE IN EPISODIC MEMORY

To demonstrate both approaches, we build on work examining the interplay of physical health and memory across older age (Nelson, Jacobucci, Grimm, & Zelinski, 2020) with data from the Health and Retirement Study (HRS). In this study, bivariate latent change score models were used to examine nonlinear change within each construct, as well as the coupling between physical health and memory. Further, education, race, ethnicity, and gender were included as covariates.

We extend this work by including a larger set of covariates, spanning years of education, gender, medical service utilization, and difficulty in daily activities (see Supplementary Table 40.1 for a full description) from either waves 2 or 3 (some variables were only assessed in wave 3). We use these covariates to predict longitudinal change in episodic memory, as measured by the total word recall sum score, which was assessed at each wave. The Total Recall score (range: 0–20) reflects the number of words respondents correctly recalled from a list of 10 words, both immediately and after a delay. The data come from the RAND HRS Longitudinal File, Version O (Chien et al., 2015), which is part of the public release data files, and includes every individual who has ever completed an HRS Core Interview. We analyzed data from Waves 4 through 11, which were collected at two-year intervals. We only included participants' study waves across ages 50 – 87.

[-------- PLACE FIGURE 40.1 ABOUT HERE -------]

As a template model, we specified a quadratic latent growth curve model with definition variables on the factor loadings of the linear and quadratic slope in order to model age-based trajectories of memory decline. The definition variables of the linear slope were participants'

chronological age centered at age 66 (the mean age at Wave 4). The definition variables of the quadratic slope were the squared linear loadings, respectively, rescaled by $1/100$ for numerical stability. This model has a total of 10 parameters: the mean and variance of the intercept, the mean and variance of the linear slope, the mean and variance of the quadratic slope, the covariances among intercept, linear slope, and quadratic slope, and a residual error variance. Figure 40.1 shows the path diagram created in Ωnyx (Oertzen, Brandmaier, & Tsang, 2015). First, we fit the model with the complete sample in `OpenMx` (Neale et al., 2016). The maximum likelihood estimates obtained are provided in Table 40.1.

[------- PLACE TABLE 40.1 ABOUT HERE ----------]

In Table 40.1, we focus on the mean estimates for each latent variable. From this we can infer that the average participant achieved a Total Recall score of 10.6 at age 66 and that this expected score declined for each subsequent year of age (negative $slope$ coefficient), with the expected trajectory demonstrating accelerated decline in later years (negative $quad$ coefficient).

With the template model, we incorporate the covariates in two different ways based on the functionality of both regularized SEM and SEM trees. For regularized SEM, the covariates are included directly in the SEM model as linear predictors of each latent variable ($icept, slope, quad$). In contrast, SEM trees uses the set of covariates to identify heterogeneity by creating subsamples that evidence varying forms of change in the template model. This can be thought of as a multi-group model whose grouping structure is learned from the data. More details on the application of each method is provided below in the results.

REGULARIZED SEM

Regularized SEM (Jacobucci, Grimm, & McArdle, 2016; also termed penalized likelihood method for SEM, Huang, Chen, & Weng, 2017) alters the traditional fit functions used

in SEM to directly incorporate regularization (penalization) of key parameters. In classic regularized regression, coefficients are estimated with a penalized fit function that not only minimizes the sum of squared residuals but also attempts to pull coefficients towards zero to reduce the generalization error. Regularized SEM is a general method for regularization that can utilize a number of specific regularization methods such as ridge, lasso, or elastic net across a variety of models. To demonstrate this technique, we can expand on our example model and directly penalize each of the regression paths from the covariates ($c1, \ldots, c13$) to each of the latent variables ($icept$, $slope$, and $quad$). Building off of the commonly used maximum likelihood fit function, the new lasso regularization fit function is

$$F_{lasso} = F_{ML} + \lambda \cdot \left\| \begin{matrix} c1 \rightarrow icept \\ \vdots \\ c13 \rightarrow icept \\ c1 \rightarrow slope \\ \vdots \\ c13 \rightarrow slope \\ c1 \rightarrow quad \\ \vdots \\ c13 \rightarrow quad \end{matrix} \right\|_1 . \quad (40.1)$$

The key quantity in Equation 40.1 is the penalty, λ, which controls the degree of regularization. When $\lambda = 0$, maximum likelihood estimation is used. When $\lambda > 0$, some degree of penalization takes place, which has the effect of shrinking the coefficients towards zero. At a sufficiently large value of λ, some parameters will be set to zero, thus performing variable/path selection. As the optimal value of λ is not known ahead of time, it is common to test 10–50 different values, ranging from zero to the smallest value of λ that results in all penalized estimates being set to zero. Note that λ does not have an inherent scale and depends on the scale of the fit function and parameter estimates.

For the lasso, the ℓ_1 norm is used, which can be computed by taking the sum of absolute

values of the vector of covariate paths. Using the absolute value distinguishes lasso regularization from other forms and accounts for why the lasso performs variable/path selection. Beyond the lasso, there are a number of additional types of regularization. Ridge regularization was originally developed to deal with ill-posed problems in regression by adding small positive quantities to the diagonal of $X'X$ with X being the design matrix (Hoerl & Kennard, 1970). More generally, ridge regularization uses the ℓ_2 norm and does not perform variable selection. Instead, it is often used when collinearity exists among predictors (ridge regression). Even when moderate levels of correlation exist among the predictors, supplementing the lasso with ridge penalties has been found to be beneficial (Zou & Hastie, 2005). This is referred to as the elastic net, which introduces a mixing parameter that determines the relative contribution of the lasso and ridge penalties.

While the lasso is likely the most common type of regularization applied to regression, network models, SEM, and others, it has a few well-known limitations – namely, inconsistency and high false positive rates (see Li & Jacobucci, in press, for a discussion). To overcome these issues, a number of alternative forms of regularization have been proposed. Three alternatives that have seen application in SEM are the adaptive lasso (Zou, 2006), minimax concave penalty (MCP; Zhang, 2010), and the smoothly clipped absolute deviation (SCAD; Wang, Li, & Tsai, 2007). Although the MCP and SCAD result in sparser solutions than the lasso, they can make model convergence difficult in large structural equation models. The adaptive lasso has been found to work well and is computationally equivalent to the lasso. The adaptive lasso scales each parameter by their least-squares (regression) or maximum likelihood (SEM) estimates, manifesting itself as $\lambda\|\beta_{ML}^{-1} \times \beta_{pen}\|_1$; thus, each parameter receives a unique penalty relative to its magnitude, resulting in lower false positive rates. An additional benefit is that this negates

complications with respect to scaling. For instance, when using the lasso, predictors need to be standardized, as the lasso places constraints on the size of each parameter. If parameters are of varying magnitude, this will result in unequal penalization solely due to the scale of the parameter. In the case of our example SEM, using the lasso would require setting the variance of each latent variable to one to ensure that larger penalties are not placed on the paths to the latent variable with larger variance.

Regularization represents a largely atheoretical approach to model modification and is somewhat similar to the use of modification indices, with a few important distinctions. First, modification indices denote the expected change in model fit by altering one parameter (typically by freeing a parameter set to zero; see Yuan & Liu, 2021), whereas regularization requires users to identify a set of potential parameters for removal. Therefore, not only does regularization represent a more global approach to model modification, it approaches this modification in the opposite way from modification indices (i.e., backwards instead of forwards). This contrast in approach mirrors theoretical differences between the approaches: regularization is most commonly used when uncertainty is identified prior to model fitting, whereas the use of modification indices most often occurs after a theoretical model is found to fit poorly. An important caveat to this distinction is that regularization can be used similarly to modification indices, such as adding regularization after a researcher realizes they do not have a sufficient sample size to trust estimates from maximum likelihood, or, when a model fits poorly, adding a set of residual covariances in order to improve model fit (see Li & Jacobucci, in press, for an example), among many other possible uses.

Regularization has been applied in the context of SEM in various ways. The first formulations were in the context of EFA (Hirose & Yamamoto, 2014), using regularization in the

place of rotation to achieve simple structure (Scharf & Nestler, 2019), with a more recent focus on performing variable/loading selection (e.g., Huang, Chen, & Weng, 2017) or identification of residual covariances (Li & Jacobucci, in press; Ober, Cheng, Jacobucci, & Whitney, 2020). In the context of CFA, either uncertainty as to the factor structure is acknowledged at the outset and regularization is used as a tool to identify a well-fitting model, or a theoretical model results in poor fit and regularization is used in a second step to modify the original model. Similar to recognizing uncertainty at the outset, regularization has been applied in additional SEM formulations to "learn" the optimal structure, including directionality discovery in individual dynamic models (Ye, Gates, Henry, & Luo, 2021), identifying mediators among a large set of potential variables (Serang, Jacobucci, Brimhall, & Grimm, 2017), and identifying differential item functioning (Liang & Jacobucci, 2020) or violations of measurement invariance (Bauer, Belzak, & Cole, 2020), among others. Finally, regularization can be used more explicitly as a fusion of theoretical and atheoretical modeling, where a subset of the model is specified based on theory, but less theoretical backing exists for other parts of the model. One example is when heterogeneity is hypothesized, but a researcher is unsure as to which of several potential grouping variables results in meaningfully different SEM parameters (Huang, 2018). Regularization has also been applied to multiple indicator, multiple cause models to identify which covariates are potential causal factors of latent variables in the model (Jacobucci, Brandmaier, & Kievit, 2019).

Regularization can be applied to SEM in a number of different software packages. Both `regsem` (Jacobucci, 2017) and `lslx` (Huang, 2020) are more general packages in R that can either be run using `lavaan` syntax (`regsem`) or custom SEM syntax (`lslx`). Both packages implement all forms of regularization discussed previously, along with the use of full

information maximum likelihood for data sets with missing data, and estimation for categorical variables (just `lslx`). For the empirical example here, we opted to write a custom fit function in `OpenMx`, as there are limitations with current software for the application of definition variables, and `OpenMx` is flexible enough to incorporate custom fit functions that included penalization. Finally, we note that the forms of regularization discussed here are frequentist in nature; a host of similar regularization forms can be used in Bayesian estimation approaches (e.g., Van Erp, Oberski, & Mulder, 2019).

Empirical Example

For selecting which of the covariates are important predictors of each of the three latent variables, we opted to use the adaptive lasso for regularization, as this has demonstrated the best performance in Li and Jacobucci (in press) for this size of model while having a modest runtime. We chose the Bayesian Information Criterion (BIC; Burnham & Anderson, 2004) to select a final model. The following analysis took approximately 30 minutes to run the 30 separate penalties in parallel on a computer cluster. The resulting BIC values across the 30 models with different penalizations are displayed in Figure 40.2.

[---------- PLACE FIGURE 40.2 ABOUT HERE ----------]

In Figure 40.2, we can see that the BIC improves (declines) rather slowly until relatively large penalties are used, and then increases precipitously. This happens due to some of the more important paths being set to zero, which can be visualized in Figure 40.3.

[---------- PLACE FIGURE 40.3 ABOUT HERE -------]

In Figure 40.3, we can see that the parameters move towards zero rather slowly, and then once $lambda$ moves beyond a value of around one, the majority of the small coefficients are set to zero, with only a few parameters not set to zero at λ values greater than around 100. Further,

we can see that the best fitting model, indicated by the dashed vertical line, has only a few non-zero coefficients. The full set of parameters from the best fitting model are displayed in Table 40.2.

[------ PLACE TABLE 40.2 ABOUT HERE -----]

When using regularization for variable/path selection, it is common practice to interpret non-zero coefficients as "important," as they survived the selection procedure. Note that the coefficients are unstandardized; that is, the absolute magnitude of each coefficient should not be compared across latent variables, as the variance differs to a large extent from the $icept$ to $slope$ and $quad$. However, we can interpret magnitude within each column, further taking into account the scale of each variable (i.e., comparing the estimates for the binary variables). For instance, those with higher education have a higher intercept. Note that the regularized part of the model is essentially a multiple, multivariate regression model with latent outcomes, so the usual caution with interpreting partial regression coefficients apply. For example, the model indicates that if someone has had a hospital stay in the past two years, they are expected to have a higher initial total recall score, which here is a suppression effect. Furthermore, both females and Hispanics have more negative initial slopes. Conversely, those with higher values of self-reported health have less decline in total recall scores during the later ages, which is in line with the findings of Nelson et al. (2020).

SEM TREES

Structural equation model trees (SEM trees) are another machine learning inspired tool for uncovering heterogeneity with respect to parameters of a SEM. In contrast to regularized SEM, SEM trees leverage a non-parametric approach to identify important covariates and thus allows for finding non-linear and interactive effects of covariates. In SEM trees, we select

predictors to recursively split the complete data into subgroups, such that these subgroups are maximally different from each other. This way, we grow a tree structure of predictors that divide a given data set recursively into homogeneous subsets that are each described by different sets of parameter estimates for the original SEM. Figure 40.4 illustrates the recursive splitting process. First, we fit a SEM (here, represented as a bivariate Gaussian distribution) to the complete sample (left panel). Then, we search all potential predictors (typically a set of variables not yet modelled in the SEM) and estimate whether splitting the sample into subgroups (according to that predictor) improves model fit. If we find such a split, we permanently split the data into two groups, keep separate models for each of the two group (middle panel), and search for the best split within the resulting groups (right panel).

[------- PLACE FIGURE 40.4 ABOUT HERE ------]

The generic algorithm for model-based partitioning proceeds in the following steps (Zeileis, Hothorn, & Hornik, 2008). First, fit the template SEM to the current data set. Then, for each predictor, assess the improvement of fit by splitting the data set according to that predictor. If the improvement in fit of the optimal predictor is better than a threshold, split the data permanently according to that predictor and start this procedure all over with each of the resulting subgroups.

Various implementations of this algorithm have been proposed. The implementations can be differentiated by how they assess the improvement in fit and how they determine the threshold for continuing to split. The classic SEM tree algorithm (Brandmaier, Oertzen, McArdle, & Lindenberger, 2013) uses a multi-group SEM approach. For each predictor, the sample at hand is split into subgroups and a multi-group model is fit to the subgroups. For non-dichotomous predictors, all possible dichotomizations become possible split variables. A

likelihood ratio test evaluates whether the improvement in fit is significantly different from zero. This straightforward approach is implemented in the R packages `semtree` (Brandmaier, Oertzen, McArdle, & Lindenberger, 2013) and `MplusTrees` (Serang et al., 2021), however, it suffers from several problems. First of all, the approach is computationally demanding. For every possible split of every variable, a multi-group model has to be estimated, typically using a gradient descent approach. Even though split evaluation can be massively parallelized, it makes growing SEM trees tedious, if not unfeasible, on desktop computers. Second, this naïve implementation of maximum-likelihood split selection suffers from variable selection bias (Strobl, Boulesteix, Zeileis, & Hothorn, 2007); that is, the chance level of selecting an uninformative predictor depends on the scale of measurement.

The `semtree` package offers a variant of this split selection principle addressing the variable selection bias by using a split sample approach (the so-called "fair" approach). First, it selects the best cut-point for each variable in one half of the data. In a second step, it selects among the best cut points using the second half of the data. Although this removes the selection bias, it comes at the cost of a drastic loss of statistical power because only half of the sample is used for selecting the best predictor, and because the split sample approach is based on a random split that induces instability of trees across multiple runs on the same data. Arnold, Voelkle, and Brandmaier (2021) implemented a likelihood-ratio test statistic (*maxLR*) that is not affected by variable selection bias, has no loss of statistical power but is still computationally expensive. Score-based test statistics based in the structural change framework (Zeileis & Hornik, 2007) offer a further way to select optimal splits while promising enormous gains in computational efficiency. Arnold, Voelkle, and Brandmaier (2021) proposed using score-based test statistics for SEM trees and forests; they provide an implementation of this approach in the `semtree` package.

Score-based SEM trees do not need to refit a SEM to evaluate a potential further split. Rather, they compute a statistic based on the score function, which is the first-order partial derivatives of the case-wise log-likelihood function with respect to the parameters. These case-wise derivatives, also called scores, quantify how well an individual is represented by the global model parameters. The smaller the score, the smaller the misfit of a given model parameter for a given case. For a vector of parameter estimates $\hat{\theta}$ and case y_i, the score function is:

$$S(\hat{\theta}; y_i) = \left[\frac{\partial \ln L(\theta; y_i)}{\partial \theta_1}|_{\theta=\hat{\theta}} \dots \frac{\partial \ln L(\theta; y_i)}{\partial \theta_q}|_{\theta=\hat{\theta}}\right]' \quad (40.2)$$

By definition, the scores sum up to zero if $\hat{\theta}$ is a valid maximum likelihood estimate. From this, we can construct a cumulative score process (CSP) function, which yields the cumulative sum over the first k rescaled and de-correlated scores:

$$CSP(\hat{\theta}; k) = \frac{1}{\sqrt{n}} I(\hat{\theta})^{-1/2} \sum_{i=1}^{k} S(\hat{\theta}; y_i) \quad (40.3)$$

with n being the sample size and $I(\hat{\theta})$ the estimated Fisher information matrix. Because the CSP is a cumulative sum over cases, its shape depends on how cases are sorted. Under the null hypothesis of no differences in true parameters across cases, the CSP converges in distribution to a Brownian bridge. This allows us to derive test statistics and p-values to test whether the sorting of participants' scores according to the observations in a given predictor leads to rejecting the null hypothesis of parameter constancy across cases. Different test statistics can be constructed (Merkle & Zeileis, 2013; Arnold, Voelkle, & Brandmaier, 2021); the choice primarily depends on the level of measurement of the predictors. The `semtree` package automatically chooses appropriate statistics depending on the specified level of measurement for each predictor.

To reiterate, instead of re-estimating every possible split to find the optimal predictor, we first compute the scores once. For each predictor, we order the scores with respect to its observed

values and derive the CSP, which we then test against a null hypothesis of parameter constancy across the predictor. Even in simple scenarios, the gain in runtime is tremendous. Furthermore, there is no loss in power and no variable selection bias (Arnold, Voelkle, & Brandmaier, 2021). However, score-based statistics as currently implemented in the `semtree` package are limited to complete data sets.

Figure 40.5 illustrates the cumulative score process on some simulated data. If the cases are sorted according to an uninformative predictor (right panel of the figure), the CSP fluctuates randomly around zero. When the scores are sorted according to an informative predictor that, in this example, separates the sample into two equally sized groups with a cut point between case 50 and 51 (left panel), the CSP shows an ordered structure. The maximum of the CSP is outside the critical value, thus we reject the null hypothesis of parameter constancy across these groups and split the sample.

[--------- PLACE FIGURE 40.5 ABOUT HERE ---------]

Empirical Example

To load the `semtree` package, install the latest stable version from the Comprehensive R Archive Network (CRAN) or the latest developer version from the github project site (https://github.com/brandmaier/semtree) and then type in R:

```
library(semtree)
```

Growing a SEM tree is done with the following R function:

```
semtree(model = model, data = data, control = semtree.control())
```

where `model` is either a `lavaan` or `OpenMx` model, `data` is an R data.frame including all manifest variables of `model`, and `ctrl` an optional object containing all meta-parameters governing how the tree is grown. The default control object is created with the constructor `semtree.control()`. The control object has fields that determine the type of statistic (`method`), the stopping criteria

such as the maximum tree depth (`max.depth`), the minimum number of cases in a leaf (`min.N`), or whether Heywood cases should be excluded when evaluating splits (`exclude.heywood`). Details about all options are provided in the package documentation.

The SEM tree on the HRS data modeling age-related decline in memory is shown in Figure 40.6.

[------- INSERT FIGURE 40.6 ABOUT HERE ----------]

The top-level split is *years of education*. The most informative split is the cut point of whether participants completed 12 or more years of education—that is, whether participants finished secondary education or not. This finding is in line with recent meta-analyses showing that the associations between education and aging-associated rates of cognitive declines as well as brain volume shrinkage are negligible (Lövdén, Fratiglioni, Glymour, Lindenberger, & Tucker-Drob, 2020; Nyberg et al., 2021). For those who have less than 12 years of education, the conditionally most informative split was whether they completed at least 9 years of education or not. For those with education of 12 or more years, gender was the conditionally most informative variable. In Figure 40.7, we see that model-predicted rates of changes do not differ much across the leafs of the pruned tree, whereas education and gender largely only predict intercept differences. The full tree has a depth of 7 and contained the covariates education (RAEDYRS), gender (RAGENDER), “ever drank alcohol” (R2DRINK), self-report of health (R2SHLT), and race (RARACEM).

[------- INSERT FIGURE 40.7 ABOUT HERE ----------]

SEM Forests

After the first split, all splits in SEM trees are to be read as conditional effects. That is, the selection of a split further down the tree depends on all splits selected further up the tree. Slight

changes to the original data set (e.g., sampling a few more cases or removing a few outliers) may sometimes change one of the top splits and, thus, leads to entirely different tree structure (are unstable; see Philip, Rusch, Hornik, & Strobl, 2018). Often, there are numerous structurally different trees that predict equally well but use quite different covariate combinations and, thus, would have quite different implications for theory building. To leverage the well-recognized problem of tree instability to robustly search through a variety of conditional effects, Brandmaier, Prindle, McArdle, and Lindenberger (2016) developed SEM forests as an ensemble approach to non-parametrically estimate the importance of covariates for a given SEM. SEM forests are based on random forests (Breiman, 2001). In random forests, a large set of trees is built on random samples of cases and predictors. This is akin to the idea of the wisdom of the crowds. The average prediction of an ensemble is often better than individual predictions if there is no bias and the prediction errors are uncorrelated (Bauer & Kohavi, 1999; Breiman, 2001). In such ensembles, predictions are made by averaging the predictions of the ensemble members. Inducing variability by resampling predictors for every potential split is key because it decorrelates the model errors and improves the ensemble predictions. Specifically, Breiman (2001) showed that the upper bound of the generalization error of an ensemble depends on the correlation between the trees. Thus, the central parameter for forests is the number of predictors that are randomly sampled at each split evaluation. Smaller values allow for more diversity because it is less likely that the tree structure will be dominated from the same strong predictors. Typically, this value is heuristically set to the square-root of the number of predictors.

In the `semtree` package, the parameters governing the growth of a tree can be given in a list of type `semforest.control`. The following code creates a control object that requests a forest with 1,000 trees, then uses subsampling as a resampling method, with a random draw of

two predictors (`mtry`) to be evaluated at each level of the tree:

```
ctrl <- semforest.control(
  num.trees = 1000,
  sampling = "subsample",
  mtry = 2
)
```

The command to grow a SEM forest is similar to growing a single tree, where `model` is either a `lavaan` or `OpenMx` model, data is an R `data.frame` including all manifest variables of `model` and `ctrl` is an optional object with all meta-parameters governing how the forest is grown:

```
forest <- semforest(model = model, data = data, control = ctrl)
```

Are SEM forests always favored over single trees? With forests, we typically increase the predictive accuracy of our model. In fact, random forests are among the best predictors across a variety of tasks (Fernández-Delgado, Cernadas, Barro, & Amorim, 2014). If prediction itself is a goal, we can expect random forests to be among the best classifiers. But what if we want to employ SEM forests for theory building? Then, we need a way to inspect the contributions of the predictors to predicting differences in the parameters of the template SEM. These contributions can be estimated using a permutation-based approach. The idea is that we randomly permute a given predictor and assess how much worse the forest fits the scrambled data set. If the loss in fit is substantial, we conclude that the variable must have made an important contribution to the prediction. Permutation variable importance can be computed from a given forest with the following command:

```
vim <- varimp(forest)
```

The result can either be printed using the `print`-command or plotted using the `plot`-command.

This method of computing permutation-based variable importance suffers from a bias that has long gone unnoticed. Strobl, Boulesteix, Zeileis, and Hothorn (2007) pointed out that randomly permuting a predictor also breaks the associations with the remaining predictors and, thus, the estimated effect size of variable importance confounds the association of a given predictor with both the multivariate outcomes as well as with the other predictors. They proposed a new approach for computing conditional variable importance that corrects for this bias of marginal variable importance. This scheme permutes the variables with respect to a factorization of the joint distribution, such that the associations within predictors are kept and only the association of a given predictor and the outcome is shuffled. In `semtree`, both permutation schemes are implemented.

[------- INSERT FIGURE 40.8 ABOUT HERE ----------]

We computed variable importance on the quadratic growth curve model. Figure 40.8 shows a bar plot with the estimated permutation variable importance values. The forest analysis confirms that years of education is the most influential predictor for the quadratic change score model, followed by gender, race, Hispanic, and self-reported health (similar to the results from regularized SEM).

DISCUSSION

Incorporating ideas and algorithms from machine learning into SEM begs the question: When is SEM no longer SEM? For instance, recent research has shown how item response models can be run in neural network software (Urban & Bauer, 2021), while van Kesteren and Oberski (2019) discuss how optimization and parallelization approaches that are typically used to estimate parameters in deep learning models can profitably be employed to estimate SEM with various objective functions, such as regularized SEM. The same can be said for machine

learning: though broad definitions can be offered for what machine learning is, there are often contradictions in what is considered to fall under the umbrella of machine learning. For instance, applying the adaptive lasso to the regression paths results in a less complex model than if it was estimated with maximum likelihood. Highlighting this fuzzy labeling of methods returns to the original point made in the introduction: the key to conducting research in this area is the intention with respect to theory generation, theory development, or theory appraisal. Although some methods are more appropriate for types of research, the most important factor is in how a method is applied.

Regularized SEM and SEM trees are attractive methods to investigate the influence of yet unmodeled variables onto the multivariate relationships of a set of variables in a SEM. To make this more concrete, we can revisit our empirical example. Although latent growth curves allow for observation-specific change trajectories, they can often leave us with unanswered questions as to why a set of individuals evidence varying change (i.e., heterogeneity in the latent change variable). But also, we may be interested whether there is heterogeneity in the variability, that is, asking ourselves whether one group of individuals is more variable than another, or whether the model fits equally well across individuals (i.e., heterogeneity in the residuals). The choice between the two proposed approaches to answer such questions should be guided by deciding what aspects of theory development are more heavily guided by theory or by data. Common to both methods is that we find ourselves in a starting position with a theory in the form of a structural equation model and a set of additional covariates, of which we are not sure which are important for our theory. When we can assume that the effects are linear, regularized SEM provides a powerful approach to theory development that can operate under conditions in which SEM is typically thought to perform less well (e.g., small samples, large number of variables,

multicollinearity). As in other linear regression frameworks, non-linearities and interactions can be explicitly modeled by transforming the covariates. SEM trees, on the other hand, automatically search for possible non-linearities and interactions by the non-parametric nature of their model search, easily handling all levels of measurement in a single framework. SEM forests aggregate over many possible SEM trees and provide a global measure of importance for explaining heterogeneity in a model. Though SEM forests do not provide a straightforward way to indicate how the initial theory can be developed further, SEM trees and regularization directly yield a model that can be tested in an independent sample to assess generalization of the findings.

A further difference between the methods is not only how they treat the covariates but also where they search for differences in the model. In regularized SEM, researchers specify exactly what parameters should be regularized. They could either impose simple structure in the loadings of a latent variable or, as we have seen above, in a path selection problem. SEM trees, by default, searches for differences in the model-implied covariance matrix—that is, searching for potential heterogeneity in all model parameters. If researchers wish for a more theory-guided exploration, they can restrict the SEM tree algorithm to search only for covariates that predict heterogeneity in a single parameter or a few selected parameters of interest.

Practically relevant to a comparison of methods is sample size. The non-parametric nature of the model search in SEM trees requires a substantially larger sample size than what is needed to run regularized models. Specifically, small to moderate samples can be run with regularized SEM, with regularized SEM often resulting in lower estimation error than maximum likelihood in small sample scenarios. What is considered large enough depends on the complexity of the model and the effect sizes, making rules-of-thumb difficult. However, it is useful to keep in mind

that SEM trees work by splitting the sample. To afford at least a first split, the minimum sample size for stable estimation of a given model must thus be doubled (for symmetric covariates). If we are interested in detecting first-level interactions, we should at least assume four times the minimum sample size (and so forth). In practice, then, SEM trees easily require sample sizes of several hundred—if not thousands—observations if developing their full strength is desired.

Related and other machine-learning inspired approaches to SEM

The umbrella of machine learning is quite wide. Instead of attempting to detail all of the ways in which machine learning has been integrated with SEM, we highlight what we view as some of the most promising related approaches. First, Karch, Brandmaier, and Voelkle (2020) proposed to use Gaussian processes (GP; Schulz, Speekenbrink, & Krause, 2018) as a generalized modeling technique that enables us to express a variety of models, such as SEM, state-space-models, and others in a single model family. A Gaussian process is a stochastic process for which any finite subset (that is, random vector) is distributed according to a Gaussian distribution. GP are defined by a mean function (the generalization of the model-implied vector of observed means) and a kernel function (the continuous-time generalization of the model-implied covariance matrix). Karch et al. (2020) demonstrate how classic linear SEM, such as the latent growth curve model, can be expressed as mean and kernel functions, showing how GP modeling allows one to model change over time using non-parametric, smooth functions (the squared-exponential function). Importantly, a GP can be constructed by combining several kernel functions and, thus, it is possible to build kernels that combine both parametric and non-parametric aspects; for example, a linear latent growth curve models with a smooth squared-exponential residual covariance matrix. Importantly, this model class allows us to perform inferential statistics as well as exploratory approaches (learning the kernel from data). To

perform variable selection in GP, regularization can be implemented by specifying corresponding priors over parameters (see Jacobucci & Grimm, 2018).

While both regularized SEM and SEM trees can be applied in myriad ways, a common two-stage process is involved: an initial SEM is tested, then further evaluated. Taking the misfit from the initial SEM mimics the process used in SEM trees, where the tree building process leverages the person-specific derivatives of the log-likelihood function (scores) to test for heterogeneity. Once the score test rejects a null hypothesis of no differences in the model parameters across persons (given the set of covariates), recursive splits of the sample are performed. As an alternative to this non-parametric way of addressing heterogeneity, Arnold, Oberski, Brandmaier, and Voelkle (2020) proposed a parametric approach to model heterogeneity based on score-based estimation: Individual Parameter Contribution (IPC) Regression. In IPC regression, the scores are regressed on to a set of covariates using a linear regression approach. Arnold et al. (2020) suggest that with many covariates, this regression is best estimated using lasso regularization. The R package `ipcr` provides functions to estimate IPC regression with `OpenMx` and `lavaan` models (see Arnold, Brandmaier, & Voelkle, 2021 for a tutorial). IPC regression serves as a computationally fast tool to explore heterogeneity, that is, to investigate whether parameters differ across observations as a linear function of covariates without the need to refit the model of interest.

The aim of applying SEM trees to model heterogeneity with respect to a template model is similar to how mixture models are often applied in social and behavioral research. With SEM trees we hypothesize that heterogeneity is due to observed covariates, whereas mixture models make the assumption that heterogeneity is due to unobserved variables, thus the need to include latent classes. This distinction has been highlighted previously in Jacobucci, Grimm, and

McArdle (2017), which has led to an integration of both methods in Grimm, Jacobucci, Stegmann, and Serang (in press). Here, the template model is a growth mixture model, while trees are used to assess distinctions in class membership based on observed covariates. This method capitalizes on mixture models' increased flexibility to identify heterogeneity (as mixtures are not limited to just observed variables; see Steinley, Chapter 29, this volume); trees offer improvements in incorporating observed predictors of class membership relative to traditional linear formulations (e.g., Asparouhov & Muthén, 2014).

Finally, acknowledging uncertainty with respect to parts of an SEM, and thus *searching* or *exploring* the addition or subtraction of paths or variables ultimately describes specification search in SEM (i.e., Long, 1983). Specification search can be seen as consisting of a variegated set of techniques, including specifying a sequence of manually fitted models with the help of heuristics (such as model modification indices). Alternatively, there are also more global, automatic specification search procedures, which can at least be traced back to the idea of discovering causal structure from data by Glymour, Scheines, Spirtes, & Kelly, 1987. Because the number of possible SEMs exponentially grows with the number of paths in the specification search, one usually resorts to heuristics to find an optimal SEM (for an overview, see Marcoulides & Ing, 2012).

Limitations

The criticisms that have followed modification indices (MacCallum, Roznowski, & Necowitz, 1992) can easily be made against the use of both regularized SEM and SEM trees. While methods such as bootstrapping or cross-validation have been paired with regularization and trees to minimize the propensity to capitalize on chance, we view the main threat to both reproducibility and generalizability as a failure to disclose or report the full set of analyses

conducted. This is a problem that has plagued SEM for some time, perhaps best exemplified by failing to report residual covariances added to the model (e.g., Hermida, 2015). An additional factor unique to machine learning algorithms is the dependence on meta-parameters, which are sometimes chosen heuristically (such as the number of trees in a forest) or based on cross-validation (such as the λ term in regularized SEM). To make analyses as transparent and reusable as possible, we recommend paying attention to reproducibility from the outset. There are different kinds of threats to reproducibility, such as changes in the default settings of dependent R package as they are being developed (Epskamp, 2019) or even in the programming environment itself (Peikert & Brandmaier, 2021). Therefore, we encourage readers to use workflows with containerized environments that guarantee code is executed independent of platform and locally divergent (or even conflicting) versions of R and installed packages, such as the one proposed by Peikert and Brandmaier (2021), which is provided by the R package `repro`. At minimum, we recommend making code publicly available so the full set of settings that were used can be examined, which can be difficult to fully report in a manuscript. The code to reproduce our results is available from https://osf.io/3uyjt/.

Conclusion

Regularization, trees, and forests in combination with SEM may inform researchers about variables that provide additional information about the phenomena or processes about which they are theorizing. In contrast to applying heuristics, such as manual model modification, these methods provide structured approaches to model modification or expansion, making both approaches suitable for theory generation or development. Although these general methods are still in the early stages of development, there have been various successful applications of both approaches; we look forward to many more.

References

Arnold, M., Brandmaier, A. M., & Voelkle, M. C. (2021). Predicting Differences in Model Parameters with Individual Parameter Contribution Regression Using the R Package Ipcr. *Psych*, 3(3), 360–385.

Arnold, M., Oberski, D. L., Brandmaier, A. M., & Voelkle, M. C. (2020). Identifying heterogeneity in dynamic panel models with individual parameter contribution regression. *Structural Equation Modeling: A Multidisciplinary Journal*, *27*(4), 613–628.

Arnold, M., Voelkle, M. C., & Brandmaier, A. M. (2021). Score-guided structural equation model trees. *Frontiers in Psychology*, *11*, 3913.

Asparouhov, T., & Muthén, B. (2014). Auxiliary variables in mixture modeling: Three-step approaches using m plus. *Structural Equation Modeling: A Multidisciplinary Journal*, *21*(3), 329–341.

Bauer, D. J., Belzak, W. C., & Cole, V. T. (2020). Simplifying the assessment of measurement invariance over multiple background variables: Using regularized moderated nonlinear factor analysis to detect differential item functioning. *Structural Equation Modeling: A Multidisciplinary Journal*, *27*(1), 43–55.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*(1), 105–139.

Berk, R. A. (2008). *Statistical learning from a regression perspective (Springer Texts in Statistics, Vol. 14)*. Springer.

Brandmaier, A. M., Oertzen, T. von, McArdle, J. J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, *18*(1), 71–86.

Brandmaier, A. M., Prindle, J. J., McArdle, J. J., & Lindenberger, U. (2016). Theory-guided

exploration with structural equation model forests. *Psychological Methods*, *21*(4), 566–582.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research*, *33*(2), 261–304.

Cattell, R. (1966). The meaning and strategic use of factor analysis. In RB cattell (ed.), Handbook of multivariate experimental psychology. Chicago: Rand McNally.

Chien, S., Campbell, N., Chan, C., Hayden, O., Hurd, M., Main, R., & Moldoff, M. (2015). RAND HRS data documentation (version o). Santa monica, CA: RAND center for the study of aging.

Epskamp, S. (2019). Reproducibility and replicability in a fast-paced methodological world. *Advances in Methods and Practices in Psychological Science*, *2*(2), 145–155.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, *15*(1), 3133–3181.

Glymour, C., Scheines, R., Spirtes, P., & Kelly, K. (1987). *Discovering causal structure: Artificial intelligence, philosophy of science, and statistical modeling.* San Diego, CA: Academic Press.

Grimm, K. J., Jacobucci, R., Stegmann, G., & Serang, S. (in press). Explorations of individual change processes and their determinants: A novel approach and remaining challenges. *Multivariate Behavioral Research*.

Haig, B. D. (2014). *Investigating the psychological world: Scientific method in the behavioral sciences*. Cambridge, Massachusetts: MIT press.

Hermida, R. (2015). The problem of allowing correlated errors in structural equation modeling: Concerns and considerations. *Computational Methods in Social Sciences*, *3*(1), 5–17.

Hirose, K., & Yamamoto, M. (2014). Estimation of an oblique structure via penalized likelihood factor analysis. *Computational Statistics & Data Analysis*, *79*, 120–132.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Huang, P.-H. (2018). A penalized likelihood method for multi-group structural equation modelling. *British Journal of Mathematical and Statistical Psychology*, *71*(3), 499–522.

Huang, P.-H. (2020). Lslx: Semi-confirmatory structural equation modeling via penalized likelihood. *Journal of Statistical Software*, *93*(7), 1–37.

Huang, P.-H., Chen, H., & Weng, L.-J. (2017). A penalized likelihood method for structural equation modeling. *Psychometrika*, *82*(2), 329–354.

Jacobucci, R. (2017). Regsem: Regularized structural equation modeling. *arXiv Preprint arXiv:1703.08489*.

Jacobucci, R., Brandmaier, A. M., & Kievit, R. A. (2019). A practical guide to variable selection in structural equation modeling by using regularized multiple-indicators, multiple-causes models. *Advances in Methods and Practices in Psychological Science*, *2*(1), 55–76.

Jacobucci, R., & Grimm, K. J. (2018). Comparison of frequentist and bayesian regularization in structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(4), 639–649.

Jacobucci, R., Grimm, K. J., & McArdle, J. J. (2016). Regularized structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(4), 555–566.

Jacobucci, R., Grimm, K. J., & McArdle, J. J. (2017). A comparison of methods for uncovering

sample heterogeneity: Structural equation model trees and finite mixture models. *Structural Equation Modeling: A Multidisciplinary Journal*, *24*(2), 270–282.

Karch, J. D., Brandmaier, A. M., & Voelkle, M. C. (2020). Gaussian process panel modeling – machine learning inspired analysis of longitudinal panel data. *Frontiers in Psychology*, *11*, 351.

Kesteren, E.-J. van, & Oberski, D. L. (2019). Structural equation models as computation graphs. *arXiv Preprint arXiv:1905.04492*.

Li, X., & Jacobucci, R. (in press). Regularized structural equation modeling with stability selection. *Psychological Methods*.

Liang, X., & Jacobucci, R. (2020). Regularized structural equation modeling to detect measurement bias: Evaluation of lasso, adaptive lasso, and elastic net. *Structural Equation Modeling: A Multidisciplinary Journal*, *27*(5), 722–734.

Long, J. S. (1983). *Covariance structure models: An introduction to LISREL.* Newbury Park, CA: Sage Publications.

Lövdén, M., Fratiglioni, L., Glymour, M. M., Lindenberger, U., & Tucker-Drob, E. M. (2020). Education and cognitive functioning across the life span. *Psychological Science in the Public Interest*, *21*(1), 6–41.

MacCallum, R. C., Roznowski, M., & Necowitz, L. B. (1992). Model modifications in covariance structure analysis: The problem of capitalization on chance. *Psychological Bulletin*, *111*(3), 490–504.

Marcoulides, G. A., & Ing, M. (2012). Automated structural equation modeling strategies. In R. Hoyle (Ed.), Handbook of structural equation modeling (pp. 690–704). New York, NY:.Guilford.

Merkle, E. C., & Zeileis, A. (2013). Tests of measurement invariance without subgroups: A generalization of classical methods. *Psychometrika,* 78(1), 59–82.

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., … Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, *81*(2), 535–549.

Nelson, N. A., Jacobucci, R., Grimm, K. J., & Zelinski, E. M. (2020). The bidirectional relationship between physical health and memory. *Psychology and Aging*, *35*(8), 1140–1153.

Nyberg, L., Magnussen, F., Lundquist, A., Baaré, W., Bartrés-Faz, D., Bertram, L., … Fjell, A. M. (2021). Educational attainment does not influence brain aging. *Proceedings of the National Academy of Sciences*, *118*(18).

Ober, T. M., Cheng, Y., Jacobucci, R., & Whitney, B. M. (2020). Examining the factor structure of the big five inventory-2 personality domains with an adolescent sample. *Psychological Assessment*, *33*(1), 14–28, .

Oertzen, T. von, Brandmaier, A. M., & Tsang, S. (2015). Structural equation modeling with Ωnyx. *Structural Equation Modeling: A Multidisciplinary Journal*, *22*(1), 148–161.

Peikert, A., & Brandmaier, A. M. (2021). A reproducible data analysis workflow with R markdown, git, make, and docker. *Quantitative and Computational Methods in Behavioral Sciences*, *1, Article e3763*.

Philipp, M., Rusch, T., Hornik, K., & Strobl, C. (2018). Measuring the stability of results from supervised statistical learning. *Journal of Computational and Graphical Statistics*, *27*(4), 685-700.

Rigdon, E. E. (2016). Choosing PLS path modeling as analytical method in european

management research: A realist perspective. *European Management Journal*, *34*(6), 598–605.

Scharf, F., & Nestler, S. (2019). Should regularization replace simple structure rotation in exploratory factor analysis? *Structural Equation Modeling: A Multidisciplinary Journal*, *26*(4), 576–590.

Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, *85*, 1–16.

Serang, S., Jacobucci, R., Brimhall, K. C., & Grimm, K. J. (2017). Exploratory mediation analysis via regularization. *Structural Equation Modeling: A Multidisciplinary Journal*, *24*(5), 733–744.

Serang, S., Jacobucci, R., Stegmann, G., Brandmaier, A. M., Culianos, D., & Grimm, K. J. (2021). Mplus trees: Structural equation model trees using mplus. *Structural Equation Modeling: A Multidisciplinary Journal*, *28*(1), 127–137.

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289–310.

Shmueli, G., Ray, S., Estrada, J. M. V., & Chatla, S. B. (2016). The elephant in the room: Predictive performance of PLS models. *Journal of Business Research*, *69*(10), 4552–4564.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25.

Tellegen, A., & Waller, N. G. (2008). Exploring personality through test construction: Development of the multidimensional personality questionnaire. *The SAGE Handbook of Personality Theory and Assessment*, *2*, 261–292.

Urban, C. J., & Bauer, D. J. (2021). A deep learning algorithm for high-dimensional exploratory item factor analysis. *Psychometrika*, *86*(1), 1–29.

Van Erp, S., Oberski, D. L., & Mulder, J. (2019). Shrinkage priors for bayesian penalized regression. *Journal of Mathematical Psychology*, *89*, 31–50.

Wang, H., Li, R., & Tsai, C.-L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, *94*(3), 553–568.

Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122.

Ye, A., Gates, K. M., Henry, T. R., & Luo, L. (2021). Path and directionality discovery in individual dynamic models: A regularized unified structural equation modeling approach for hybrid vector autoregression. *Psychometrika, 86, 404–441*.

Yuan, K. H., & Liu, F. (2021). Which method is more reliable in performing model modification: lasso regularization or lagrange multiplier test?. *Structural Equation Modeling: A Multidisciplinary Journal*, *28*(1), 69-81.

Zeileis, A., & Hornik, K. (2007). Generalized M-fluctuation tests for parameter instability. *Statistica Neerlandica*, *61*(4), 488–508.

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, *17*(2), 492–514.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, *38*(2), 894–942.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of*

the Royal Statistical Society: Series B (Statistical Methodology), *67*(2), 301–320.

Table 40.1: *Estimates from a quadratic latent change score model*

	Estimate	Std.Error
residual_var	5.16	0.03
icept_var	5.58	0.11
icept_slope_cov	-0.03	0.01
slope_var	0.01	0.00
icept_quad_cov	-0.62	0.05
slope_quad_cov	0.01	0.00
quad_var	0.04	0.04
icept_mu	10.60	0.03
slope_mu	-0.14	0.00
quad_mu	-0.31	0.02

Note. Parameter names correspond to the labels in the path diagram; var = variance, mu = mean, icept = intercept, quad = quadratic

Table 40.2: *Parameter estimates from the best fitting adaptive lasso model.*

Predictor	Variable	icept	slope	quad
Years of education	RAEDYRS	0.319		
Gender	RAGENDER	-0.001	-1.196	
Self-report of health	R2SHLT	-0.01		0.314
Hospital stay, previous two years	R2HOSP	1.259		
Race (Hispanic)	RAHISPAN		-0.596	0.016
Race (Black)	RARACEM			-0.074
Race (Other)	RARACEM	-0.134		0.006
Smoked ever	R2SMOKEV	-0.003		
Previous cholesterol	R3CHOLST	0.011		
Previous flu shot	R3FLUSHT	-0.074		
Ever drank alcohol	R2DRINK	-0.002	0.258	
Some differences in getting up from a chair	R2CHAIRA		-0.019	0.023
Back problems	R2BACK			
Some differences in jogging one mile	R2JOGA		0.071	

Note. Parameters estimated to be zero are not shown.

Supplementary Table 1: List of covariates.

Name	*Description*	*Levels*	*VR*
RAEDYRS	*Years of education*	*18*	*0.65*
RAGENDER	*Gender*	*2*	*0.41*
R2SHLT	*Self-report of health*	*5*	*0.67*
R2HOSP	*Hospital stay, previous two years*	*2*	*0.13*
RARACEM	*Race*	*3*	*0.17*
RAHISPAN	*Hispanic*	*2*	*0.09*
R2SMOKEV	*Smoked ever*	*2*	*0.43*
R3CHOLST	*Previous cholesterol*	*2*	*0.28*
R3FLUSHT	*Previous flu shot*	*2*	*0.38*
R2DRINK	*Ever drank alcohol*	*2*	*0.42*
R2CHAIRA	*Some differences in getting up from a chair*	*2*	*0.31*
R2BACK	*Back problems*	*2*	*0.28*
R2JOGA	*Some differences in jogging one mile*	*2*	*0.32*

Note. Levels = number of factor levels; VR = variation ratio, defined as the proportion of cases which are not in the mode category. Variables starting with R2 were assessed at the second wave, variables starting with R3 were assessed at the third wave.

Figure 40.1. Path diagram of a quadratic growth curve model over eight waves of data acquisition (R4–R11). The factor loadings of the linear slope ($slope$) and the quadratic slope ($quad$) are person-specific loadings, so-called *definition variables*, such that we model changes in the number of *words recalled* over age. In the diagram, loadings are shown for a person aged 62 at the first measurement occasion (R1). The model is set up such that the intercept ($icept$) is anchored at the mean sample age at Wave 4 (66 years); hence, factor loadings of the linear and quadratic slope are 0 at the third measurement occasion for this person.

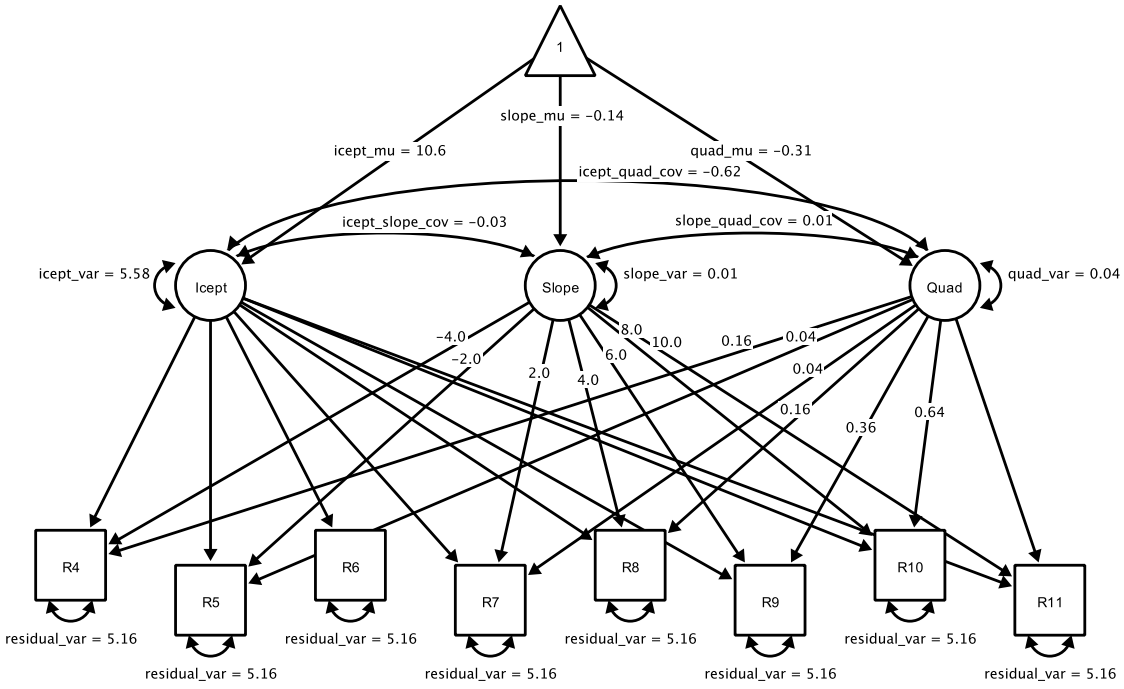

Figure 40.2. BIC values across the thirty 30 penalties. Note that one BIC value was removed, as this model failed to converge. Smaller values indicate better fit. The dashed vertical line corresponds to the lowest BIC.

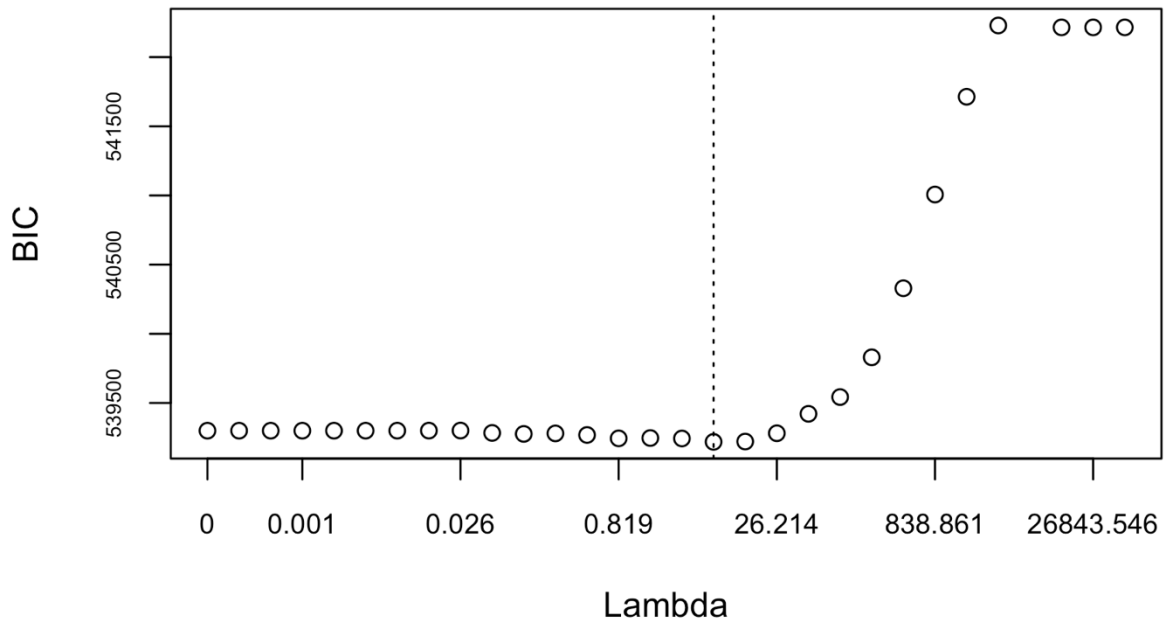

Figure 40.3. Parameter trajectory plot. Each line corresponds to a different penalized parameter, displayed across the vector of penalties (Lambda). The vertical dashed line corresponds to the lowest BIC value, thus the final selected model.

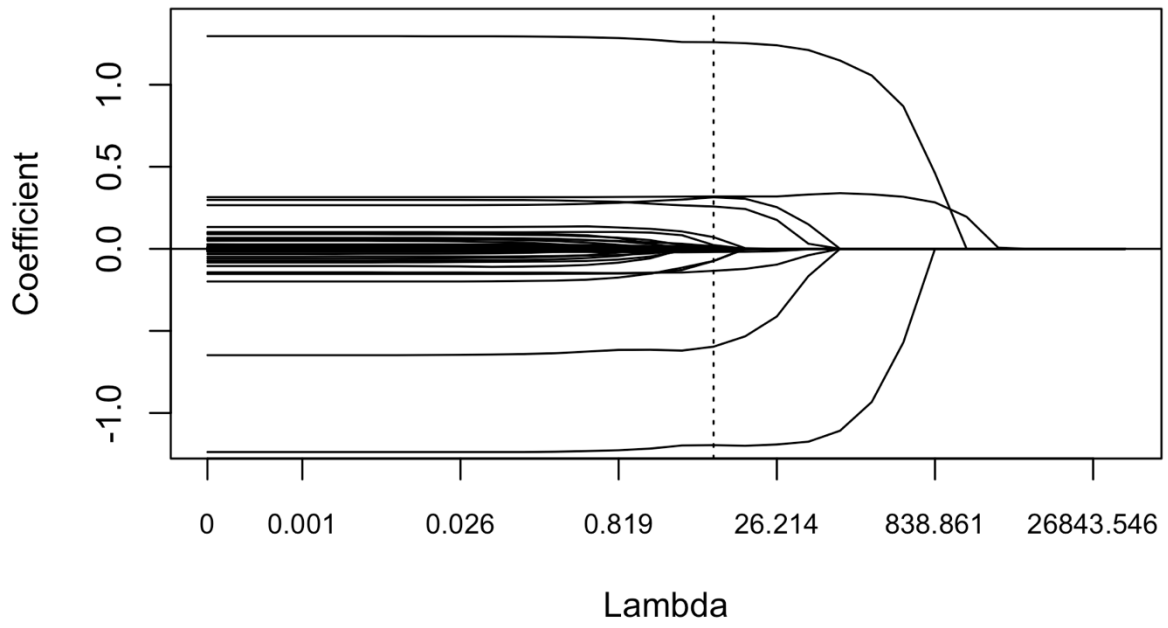

Figure 40.4. Illustration of the splitting process in SEM trees. A SEM represents a multivariate normal fit (illustrated as contours) to some data (illustrated as points). Left: a single Gaussian model describes the entire set. Center: A potential split into two subgroups may be a better description of the data. Right: Yet another split of the left group of the two-group model may promise an even better fit. Note that the locations of the points are exactly identical across all panels.

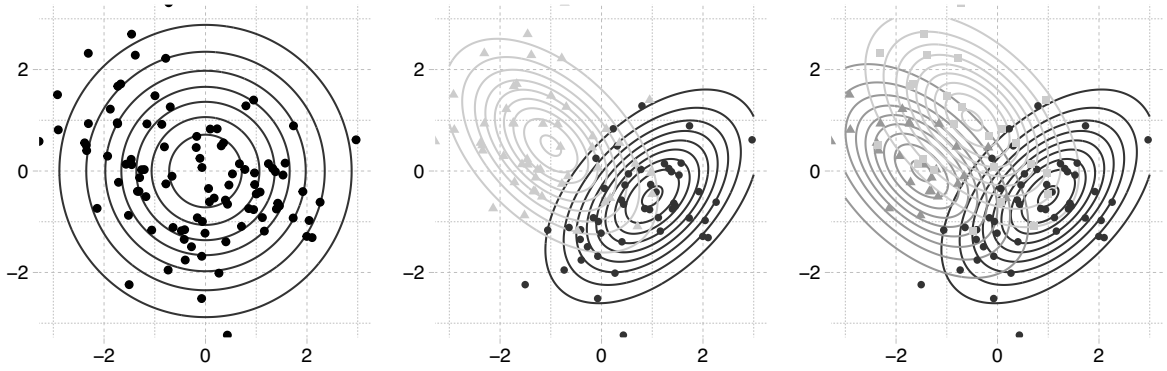

Figure 40.5. Cumulative score process on simulated data for an informative predictor (left) and an uninformative predictor (right). The grey area between dashed lines indicates the area between the critical values of the DM test statistic.

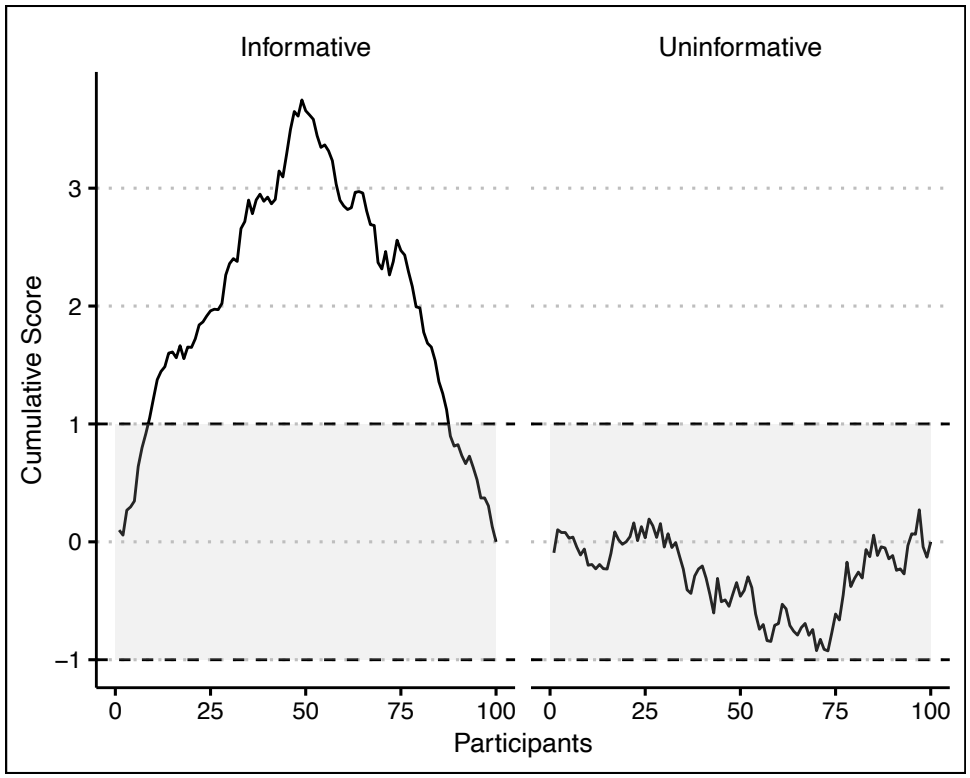

Figure 40.6. The first two levels of a SEM tree based on the RAND HRS longitudinal data for predicting differences in linear latent growth curve models of episodic memory. The upper tree shows SEM estimates according to the first split. The lower tree shows SEM estimates according to the first two levels of the tree. The primary split is according to years of education. Splits on the second level are with respect to education and gender.

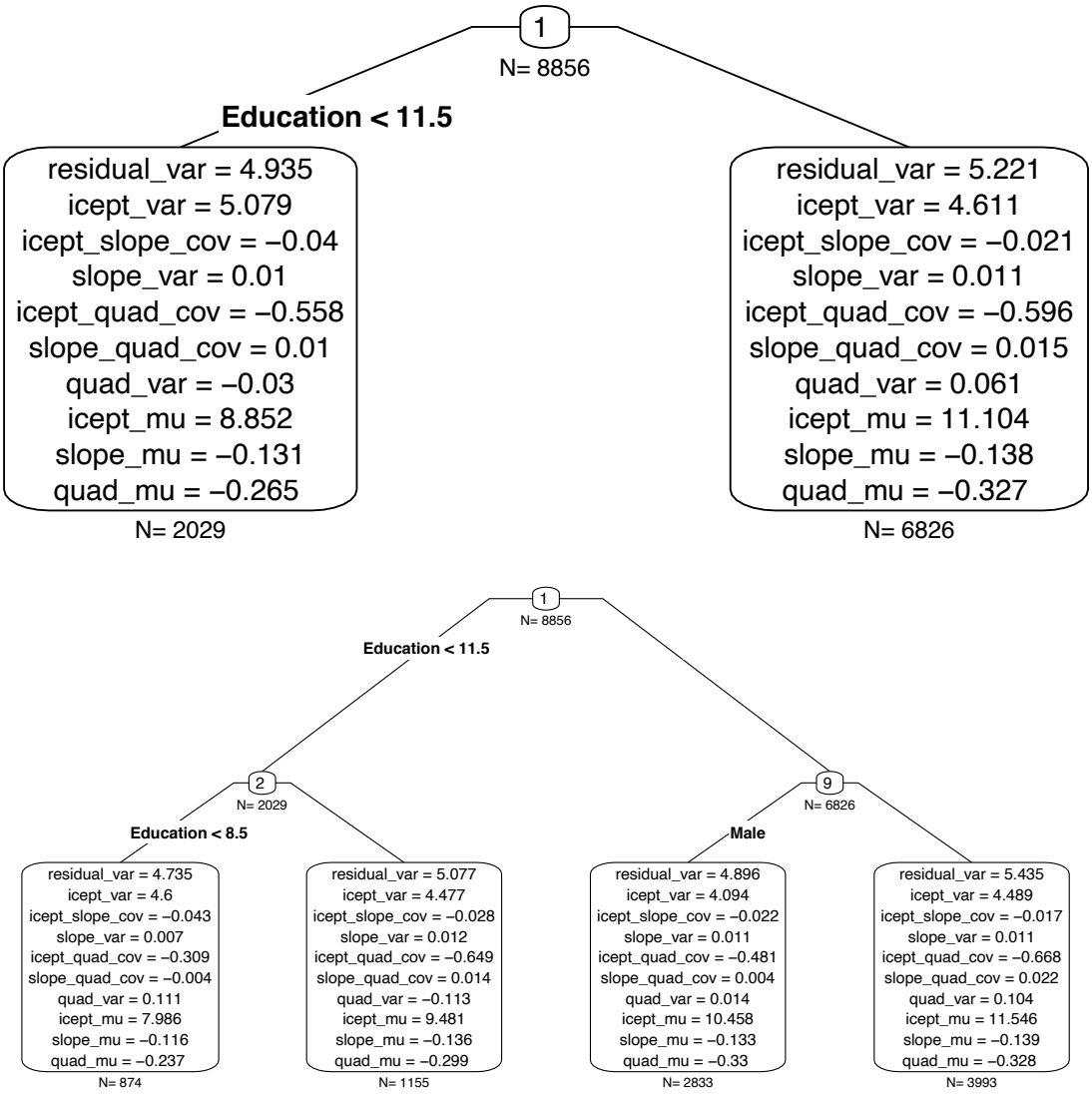

Figure 40.7. Predictions of mean trajectories corresponding to each of the four leaf nodes of a tree pruned back to the first two levels.

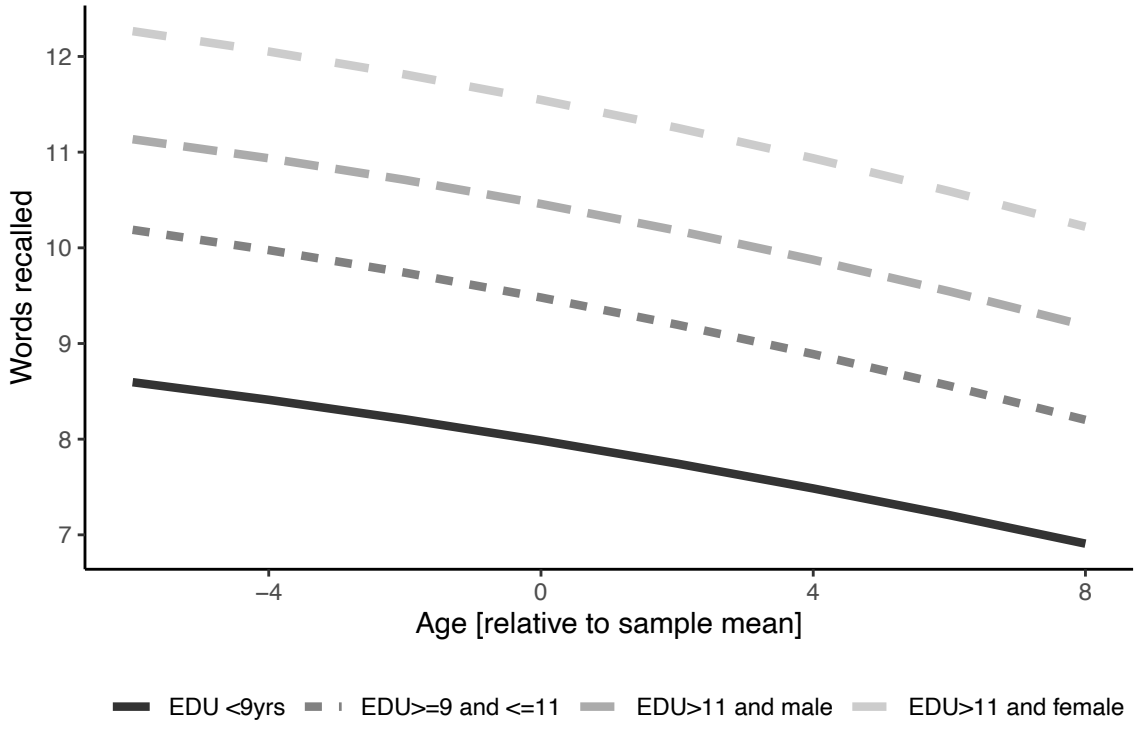

Figure 40.8. Permutation-based variable importance based on a SEM forest. Each bar depicts the estimated decrease in fit when randomly permuting a given predictor. The larger the decrease in fit, the higher the importance. The most important predictor is years of education (RAEDYRS), followed by gender (RAGEND), hispanic (RAHISP), race (RARACEM), and self-reported health (R2SHLT). See Table 40.2 for a description of all variable names.