Structural Equations with Latent Variables

Computational Solutions for Modern Data Problems

Erik-Jan van Kesteren

Structural Equations with Latent Variables: Computational
Solutions for Modern Data Problems
Proefschrift Universiteit Utrecht, Utrecht - met
samenvatting in het Nederlands

ISBN: 978-90-3937-358-3

9 789039 373583

Structural Equations with Latent Variables: Computational Solutions for Modern Data Problems

Structurele Vergelijkingen met Latente Variabelen: Computationele Oplossingen voor Moderne Dataproblemen (met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht
op gezag van de
rector magnificus, prof. dr. H.R.B.M. Kummeling,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op

vrijdag 29 januari 2021 des middags te 2.30 uur

door

Erik-Jan van Kesteren

geboren op 27 oktober 1992
te Dordrecht

Promotoren:

Prof. dr. I.G. Klugkist
Prof. dr. M.J.C. Eijkemans

Copromotor:

Dr. D.L. Oberski

Dit proefschrift werd mogelijk gemaakt met financiële steun van de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) onderzoekstalent beurs nummer 406.17.057

Beoordelingscommissie

Prof. dr. M.A.G. van Aken	Universiteit Utrecht
Prof. dr. P.G.M. van der Heijden	Universiteit Utrecht
Prof. dr. ir. R.C.H. Vermeulen	Universiteit Utrecht
Prof. dr. C. Finkenauer	Universiteit Utrecht
Prof. dr. E.J. Wagenmakers	Universiteit van Amsterdam
Prof. dr. M. van de Wiel	VU medisch centrum

Contents

Chapter 1
Introduction

The first time I encountered structural equation modeling (SEM) was in 2013, in a statistics class during my undergraduate studies in social science. The class had never seen SEM before, but in a matter of a few weeks we were dealing with rather complex statistical topics such as parameter constraints, model comparison, fit statistics, measurement error, and scale construction. I distinctly remember that we found the graphical approach to creating models very intuitive, and at the end of this course we were capable of testing complex hypotheses using different types of data. In hindsight, it is surprising how far we got in how little time. SEM has the rare combination of generality and simplicity, of flexibility and convenience.

This combination has made SEM a widely popular method for data analysis in the social and behavioural sciences. It is especially useful in research where the constructs of interest cannot be measured directly, or where the measurement instruments are fallible. An example of a traditional research situation for SEM is in social science, where a researcher may want to know how the welfare of Europeans (construct 1) affects their trust in institutions (construct 2). Each construct may be measured by three (or more) questions to which the answer is given on a scale from *completely disagree* to *completely agree*, using a questionnaire such as the European Social Survey (Norwegian Centre for Research Data, 2018). The researcher can create a figure which closely matches their underlying theoretical model (Figure 1.1), and SEM is the glue that binds this figure to the data: each of the arrows in the figure represents a parameter indicating the relation between one variable and another. The parameter of interest for our researcher would be the arrow from welfare to trust, and SEM can estimate this parameter and its standard error, allowing for statistical inference about the research question.

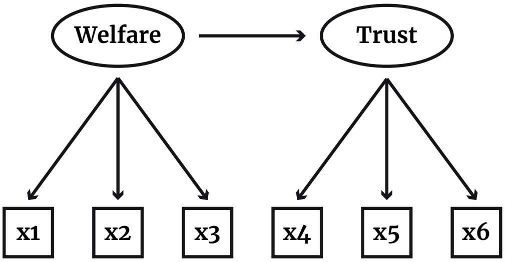

Figure 1.1 Graphical illustration of a structural equation model for the relation between welfare and trust, where each construct is measured through three survey questions. Residual variances not shown for clarity.

SEM encompasses many analysis techniques, including factor analysis (Brown, 2006), path modeling and mediation analysis (MacKinnon, 2008), multigroup models (Sörbom, 1974), multitrait-multimethod models (Campbell & Fiske, 1959; Kenny, 1976), and longitudinal latent variable models (Asparouhov, Hamaker, & Muthén, 2018; Bollen & Curran, 2006). However, despite its convenience and flexibility, SEM is reaching its limits in the modern data landscape. Classic survey and experimental research is being supplemented (and sometimes even supplanted) by research using measurements from register data, wearable sensors, images, internet databases, genetic sequencing, advanced brain imaging techniques, and more. These instruments can measure thousands of variables and / or millions of samples at a time. As a result, the pipelines for data processing can involve many steps, and the models used to process these types of data may have thousands of parameters – a size almost unheard of for SEM. But the problems of fallible measurement do not disappear in this modern data landscape, and many research questions using this data still involve causal relations between latent constructs. Thus, the analyses made possible by SEM can greatly benefit research using these new instruments.

The goal of this dissertation is to make SEM analyses available to a wider range of these modern datasets. To this end, I develop several solutions to problems encountered in the application of SEM to such data. An example of a problem is that of dimensionality: in traditional implementations, SEM becomes impractical when the number of variables becomes large. This holds for data which is naturally high-dimensional, but can also occur when researchers want to include the effect of many transformations (or basis functions) of the variables in the data. In the coming chapters, I extend traditional SEM by borrowing computational techniques from machine learning (Chapter 2), and I show how to create statistical models in situations constrained by high dimensionality (Chapter 3), biological structure (Chapter 4), privacy concerns

(Chapter 5) and algorithmic fairness (Chapter 6).

The following sections provide theoretical background for the solutions presented in this dissertation. There are three relatively independent sections – each can be skipped by readers familiar with their contents. First, the SEM approach and its notation are succinctly introduced in Section 1.1. Second, Section 1.2 provides an accessible tutorial on several computational tools for optimization, as they are used in many procedures presented in this dissertation. Third, Section 1.3 makes explicit some of the tacit views on model specification, generalizability, and regularization at the basis of this dissertation. Finally, Section 1.4 provides an outline for the remainder of the dissertation.

1.1 Structural equation models

Structural equation models are linear models for multivariate data (Bollen, 1989; Jöreskog, 1969). The term *structural equation model* is quite unfortunate: it does not describe well what the framework actually does. Very similar techniques are used in many fields under different names (e.g., Partial Least Squares, Gaussian Graphical Models, Bayesian networks). Perhaps a more informative term would be "Gaussian linear latent variable model", which makes explicit two of the most important design elements of SEM:

- **linearity** of the relations between the variables.
- **normality**, Gaussian distribution for the residuals of the latent variables and the residuals of the observed variables.

SEM is a combination of a linear measurement model (relating the latent variables to the observed variables) and a linear structural model (relating the latent variables to each other). One of the most common ways of representing SEM is the LISREL "all-y" form; below I combine the compact notations by Neudecker and Satorra (1991) and Oberski (2014):

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\varepsilon} \quad \text{(Measurement model)} \\ \boldsymbol{\eta} &= \boldsymbol{\alpha} + \boldsymbol{B}_0\boldsymbol{\eta} + \boldsymbol{\xi} \quad \text{(Structural model)} \end{aligned} \tag{1.1}$$

where $\boldsymbol{x}$ represents a vector of observable variables of length P, $\boldsymbol{\eta}$ represents the M latent variables, and $\boldsymbol{\varepsilon}$ and $\boldsymbol{\xi}$ are random vectors such that $\boldsymbol{\varepsilon}$ is uncorrelated with $\boldsymbol{\xi}$. The parameters of the model are encapsulated in four matrices: $\boldsymbol{\Lambda} \in \mathbb{R}^{P\times M}$ contains the factor loadings, $\boldsymbol{\Psi} \in \mathbb{R}^{M\times M}$ contains the covariance matrix of $\boldsymbol{\xi}$, $\boldsymbol{B}_0 \in \mathbb{R}^{M\times M}$ contains the regression parameters of the structural model, and $\boldsymbol{\Theta} \in \mathbb{R}^{P\times P}$ contains the covariance matrix of $\boldsymbol{\varepsilon}$.

The main advantage of the linearity assumption in combination with the normality of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\xi}$ is that the latent variables can be marginalized out easily, and the likelihood takes a very convenient form:

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta})) \tag{1.2}$$

The SEM likelihood is determined by the P-dimensional *model-implied* covariance matrix $\boldsymbol{\Sigma}$ and mean vector $\boldsymbol{\mu}$, both functions of the free parameters $\boldsymbol{\theta}$. $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ and $\boldsymbol{\mu}(\boldsymbol{\theta})$ have the following form:

$$\begin{aligned} \boldsymbol{\Sigma}(\boldsymbol{\theta}) &= \boldsymbol{\Lambda}\boldsymbol{B}^{-1}\boldsymbol{\Psi}\boldsymbol{B}^{-T}\boldsymbol{\Lambda}^T + \boldsymbol{\Theta} \\ \boldsymbol{\mu}(\boldsymbol{\theta}) &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{B}^{-1}\boldsymbol{\alpha} \end{aligned} \tag{1.3}$$

where $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{B}_0$ is assumed to be non-singular – that is, the structural path model relating the M latent variables to each other is assumed to be identified.

Structural parameters are an important part of SEM because they represent causal relations between constructs: the $\boldsymbol{B}_0$ matrix encodes an entire *directed graph* of relationships. Specifically, causality is an *assumption* of these parameters (Bollen, 1989, p. 4):

> The term "structural" stands for the assumption that the parameters are not just descriptive measures of association but rather that they reveal an invariant "causal" relation.

SEM can be considered an applied linear Gaussian form of the broader class of causal models as popularized by Pearl (2009). In causal modeling, the causal graph representing the researcher's theoretical model implies a certain independence structure on the observed variables, through factorization of the joint probability $p(\boldsymbol{x})$. SEM does the same via the implied covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, which considers only linear dependence.

1.2 Computation and optimization

Many of the chapters in this dissertation rely heavily on computational parameter estimation techniques to expand the area of application for SEM and related models. The goal of this section is to provide a tutorial on a few basic versions of these methods. As a running example, I use a linear regression model, and throughout the tutorial I show `R` code for each algorithm.

For simple linear regression, the likelihood is as follows:

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{X\beta}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2\right] \tag{1.4}$$

where $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ is a vector of outcome values, $\boldsymbol{X} \in \mathbb{R}^{N\times P}$ is a design matrix, and $\boldsymbol{\beta}$ is a P-vector of regression parameters. The log-likelihood, denoted ℓ, is usually simpler to work with:

$$\ell = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 \tag{1.5}$$

In maximum likelihood (ML) estimation, the objective is to find the parameters $\boldsymbol{\beta}$ that maximize the (log-)likelihood according to the following intuition:

we want to find the values of the parameters for which the data is most likely, given the model structure. Equation 1.5 shows that the value for $\boldsymbol{\beta}$ which minimizes the residual sum of squares $\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2$ (RSS) is also the value which maximizes the log-likelihood. This makes the ML estimates equivalent to the "ordinary least squares" (OLS) estimates in the framework of linear regression.

In this section, I show how to obtain the ML/OLS estimates for $\boldsymbol{\beta}$ using several different optimization methods. I will use a running data example, where $N = 100$ and $P = 10$, artificially generated as follows:

```
1   set.seed(45)
2   N <- 100
3   P <- 10
4   S <- toeplitz(1/(1:P)^0.707)
5   X <- matrix(rnorm(N*P), N) %*% chol(S)
6   b <- matrix(rnorm(P))
7   y <- X %*% b + rnorm(N, sd = sqrt(crossprod(b, S %*% b)))
```

In this code, $\boldsymbol{X}$ is sampled from a multivariate normal distribution with a mean of $\mathbf{0}$ and a Toeplitz-structured covariance matrix with smaller values further from the diagonal. Line 7 sets the true proportion of variance explained in $\boldsymbol{y}$ to 0.5.

1.2.1 Gradient descent

One of the simplest ways to optimize a convex function such as the least squares objective is through gradient descent (Cauchy, 1847). The idea is to take steps of size s in the direction of the negative gradient – the vector of partial derivatives – until it disappears. For convex functions, when the gradient is $\mathbf{0}$, a global minimum is obtained.

The gradient ∇ of the RSS with respect to $\boldsymbol{\beta}$ is

$$\nabla(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) = -2\boldsymbol{X}^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) \tag{1.6}$$

(Petersen & Pedersen, 2012, eq. 84). The gradient of the RSS is equal to the *score function* in a maximum likelihood context (up to a multiplicative constant). Using the classic gradient descent algorithm, we initialize the estimates $\hat{\boldsymbol{\beta}}$ and then take steps of size s in the direction of the negative gradient:

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \hat{\boldsymbol{\beta}}^{(i)} - s \cdot \nabla(\hat{\boldsymbol{\beta}}^{(i)}) \tag{1.7}$$

In code, iteratively updating the estimates of $\boldsymbol{\beta}$ is shown below, where the gradient from Equation 1.6 is computed on line 6 and the update from Equation 1.7 happens on line 7.

```
s     <- .001         # step-size
maxit <- 1e6          # maximum iterations
tol   <- 1e-8         # convergence tolerance
bhat  <- matrix(0, P) # initial values for beta
for (i in 1:maxit) {
  grad <- -2 * crossprod(X, y - X %*% bhat)
  bhat <- bhat - s * grad
  if (all(abs(grad) < tol)) break
}
```

A visual display of this algorithm for the synthetic dataset is shown in Figure 1.2. Panel A shows a steady improvement in the RSS value over the iterations. Panel B shows the paths taken by the 10 parameters as the algorithm progresses, starting at the initial values of 0 and ending at the OLS/ML estimates. Panel C shows how the estimates for the first two parameters move across the loss surface from their starting value to their final estimates.

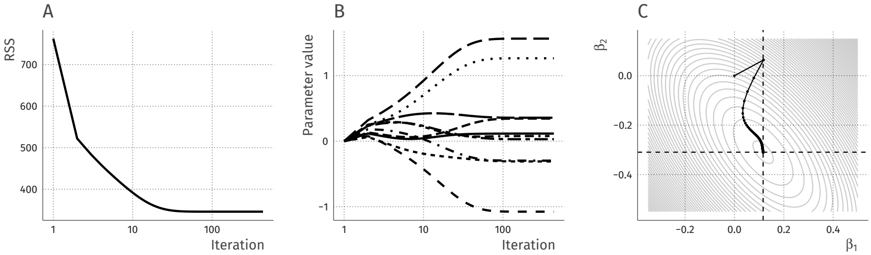

Figure 1.2 Visualization of 10-parameter gradient descent for linear regression. Panel A shows the residual sum of squares over the iterations (in log scale) as it approaches its minimum, panel B shows the parameter paths over iterations (in log scale), and panel C visualizes the paths of the first two parameters on the loss surface (contours, conditional on the final estimates of the remaining 8 parameters) as they approach the minimum.

The gradient method is a very flexible technique for optimizing functions of many different types. For convex functions, it will find the global minimum, but with small additions such as multiple starts, it is capable of optimizing more complicated functions too. However, a nontrivial problem for the gradient method is the choice of s. If s is too large, the method will "overshoot" the minimum; if s is too small, the steps may be too small and the algorithm may not converge in reasonable time at all.

1.2.2 Newton-Raphson optimization

Second-order methods add information about the curvature of the objective for each parameter to replace the predefined step-size. Specifically, s in Equation

1.7 is replaced by the inverse of the Hessian matrix $\mathcal{H}$, the matrix of second derivatives:

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \hat{\boldsymbol{\beta}}^{(i)} - \mathcal{H}^{-1}(\hat{\boldsymbol{\beta}}^{(i)})\nabla(\hat{\boldsymbol{\beta}}^{(i)}) \tag{1.8}$$

One intuition about this is the following: if the objective for a parameter value is flat, the steps in that direction should be large because otherwise the objective will not change much; if the objective is very curved, small steps should be taken to avoid overshooting a minimum. For our linear regression case, the matrix of second derivatives is the following:

$$\mathcal{H}(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}\nabla(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}} - 2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = 2\boldsymbol{X}^T\boldsymbol{X} \tag{1.9}$$

(Petersen & Pedersen, 2012, eq. 78). Plugging this into Equation 1.8 yields the Newton-Raphson algorithm for linear regression:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{(i+1)} &= \hat{\boldsymbol{\beta}}^{(i)} - (2\boldsymbol{X}^T\boldsymbol{X})^{-1}(-2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(i)}))\\ &= \hat{\boldsymbol{\beta}}^{(i)} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(i)})\end{aligned} \tag{1.10}$$

I could now show the code for this algorithm, but it is more insightful to further simplify it:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{(i+1)} &= \hat{\boldsymbol{\beta}}^{(i)} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(i)})\\ &= \hat{\boldsymbol{\beta}}^{(i)} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} - (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}^{(i)}\\ &= \hat{\boldsymbol{\beta}}^{(i)} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} - \hat{\boldsymbol{\beta}}^{(i)}\\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}\end{aligned} \tag{1.11}$$

Equation 1.11 shows that this algorithm does not depend on $\hat{\boldsymbol{\beta}}^{(i)}$ at all! In other words, no matter where you start, the Newton-Raphson method for our problem converges in a single iteration to the least squares (ML) estimates. In general, this method always converges in a single iteration for quadratic functions. In code, this amounts to the following single line to obtain the estimates for $\boldsymbol{\beta}$:

```r
bhat <- solve(crossprod(X), crossprod(X, y))
```

This analytic solution is very useful, and it is the default way for estimating models of this kind. However, there are two problems with this method: (a) the Gram matrix ($\boldsymbol{X}^T\boldsymbol{X}$) needs to be invertible, which it is not when $P > N$, and (b) even if the Gram matrix is invertible, the computational complexity of this operation grows quickly with increasing P. In addition, for optimization problems more complex than linear regression, the Hessian may not be available at all, if the likelihood is not twice continuously differentiable. In many modern data problems, these issues become pertinent: researchers are collecting more and more high-dimensional data (Chapter 3), partitioned data (Chapter 5), and data for which they may want to apply regularization (Section 1.3; Chapter 2). In this dissertation, I make use of variations on two main optimization algorithms to solve such issues: coordinate descent and adaptive first-order optimizers.

1.2.3 Coordinate descent

The method of coordinate descent (e.g., Wright, 2015) is perhaps even simpler than the gradient descent method. It can be seen as a kind of meta-algorithm for multiparameter optimization. The idea is to pick one coordinate (parameter) at a time and optimize it using your method of choice, conditional on the current values of the other parameters. Then, you move on to the next component either randomly (randomized coordinate descent) or in order, cycling through the parameters over and over (cyclic coordinate descent). A famous application of coordinate descent in the field of statistics is for elastic net regularized regression, as implemented in `glmnet` (Friedman, Hastie, & Tibshirani, 2010).

In the context of linear regression, the closed-form univariate estimate $\hat{\beta}_p$ is $\text{cov}(\boldsymbol{x}_p, \boldsymbol{y})/\text{var}(\boldsymbol{x}_p)$ (e.g., Casella & Berger, 2002, p. 551). Using our centered design matrix $\boldsymbol{X}$, we can equivalently write $\hat{\beta}_p = \boldsymbol{x}_p^T \boldsymbol{y} / \boldsymbol{x}_p^T \boldsymbol{x}_p$. To compute estimates *conditional* on values for the other parameters we replace $\boldsymbol{y}$ by the residual with respect to the parameters excluding β_p: $\hat{\boldsymbol{\epsilon}}_{\text{-}p}$ (Friedman et al., 2010). This leads to the following algorithm:

$$\begin{aligned} \hat{\boldsymbol{\epsilon}}_{\text{-}p}^{(i)} &= \boldsymbol{y} - \boldsymbol{X}_{\text{-}p} \hat{\boldsymbol{\beta}}_{\text{-}p}^{(i)} \\ \hat{\beta}_p^{(i+1)} &= \boldsymbol{x}_p^T \hat{\boldsymbol{\epsilon}}_{\text{-}p}^{(i)} / \boldsymbol{x}_p^T \boldsymbol{x}_p \end{aligned} \tag{1.12}$$

A simple `R` implementation of the cyclic version of coordinate descent is shown below, where the algorithm is stopped when the parameter values do not change (crudely approximating a vanishing gradient). The update of the p^{th} parameter happens on line 8.

```
 1 maxit <- 1e6          # maximum iterations
 2 tol   <- 1e-8         # convergence tolerance
 3 bhat  <- matrix(0, P) # initial values for beta
 4 bold  <- bhat         # previous values for beta
 5 for (i in 1:maxit) {
 6   for (p in 1:P) {
 7     res      <- y - X[,-p] %*% bhat[-p,]
 8     bhat[p,] <- crossprod(X[,p], res) / crossprod(X[,p])
 9   }
10   if (all(abs(bold - bhat) < tol)) break
11   bold <- bhat
12 }
```

Figure 1.3 shows visually the progress of this algorithm for our example linear regression problem. The coordinate-wise updating procedure is clearly visible in panel C, where steps happen either horizontally or vertically, and in panel B which shows a step pattern rather than a smooth updating pattern as in Figure 1.2.

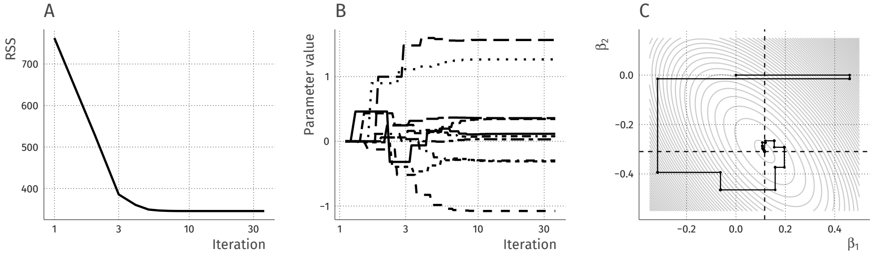

Figure 1.3 Visualization of 10-parameter coordinate descent for linear regression. Panel A shows the residual sum of squares over the iterations (in log scale) as it approaches its minimum, panel B shows the parameter paths over iterations (in log scale), and panel C visualizes the paths of the first two parameters on the loss surface (contours, conditional on the final estimates of the remaining 8 parameters) as they approach the minimum.

The advantage of coordinate descent is that the $P \times P$ Hessian matrix does not need to be directly inverted, and thus the computational load is relatively low for large P. In Chapter 3 I show a situation where P is so large that inverting the Hessian is not possible, but a good approximation to the parameter estimates is available in a random coordinate descent framework. In Chapter 5 I show a situation of vertically partitioned data where it is impossible to obtain the full Hessian, but blockwise cyclic coordinate descent helps with obtaining the ML estimates of generalized linear models.

1.2.4 Adaptive first-order optimizers

Another alternative for situations in which the Hessian is unavailable or computationally expensive is to approximate $\mathcal{H}^{-1}$ using *change* in the gradients. There are many members in this family of quasi-Newton optimization methods, such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Fletcher, 2013), which is used widely for optimization of statistical models (e.g., Rosseel, 2012, used in Chapter 4). Similar (but simpler) techniques are being applied in the field of deep learning, where parameters are numerous, Hessians unwieldy, but gradients easy to compute (Goodfellow, Bengio, & Courville, 2016). Here, the entire history of gradients is being used to dynamically adapt the step-size per parameter, according to the intuition that if the gradients do not change much, the step-size can be bigger, and if they change a lot, the step-size should be reduced.

A particularly popular algorithm in this field is Adam (Kingma & Ba, 2014), which contains two improvements to the default gradient descent framework. First, the step direction $\boldsymbol{m}$ in Adam is an exponentially decaying average of the history of gradient directions. This makes the optimization path smooth, avoiding sharp turns, similar to the effect of momentum on a physical ball rolling on a surface. Second, for the step-size Adam uses a base value divided by the square root of $\boldsymbol{v}$, an exponentially decaying average of the history of squared gradients. $\boldsymbol{v}$ can be interpreted as the variance, or uncertainty, around each element of the gradient. The more uncertain the direction, the smaller the step-size (for more information, see Appendix A.1).

A basic implementation of Adam is shown below. The main updates are on lines 11, 12, 13, and 16, and the remaining lines are used for bias-correction and convergence checking.

```
 1 s     <- 0.001          # base step-size
 2 gamma <- c(0.9, 0.999) # decay values
 3 maxit <- 1e6           # maximum iterations
 4 tol   <- 1e-8          # convergence tolerance
 5
 6 bhat  <- matrix(0, P)  # initial values for beta
 7 m     <- matrix(0, P)  # first moment
 8 v     <- matrix(0, P)  # second moment
 9
10 for (i in 1:maxit) {
11   grad <- -2 * crossprod(X, y - X %*% bhat)
12   m    <- gamma[1] * m + (1 - gamma[1]) * grad
13   v    <- gamma[2] * v + (1 - gamma[2]) * grad^2
14   mhat <- m / (1 - gamma[1])
15   vhat <- v / (1 - gamma[2])
16   bhat <- bhat - s * mhat / (sqrt(vhat) + 1e-8)
17   if (all(abs(grad) < tol)) break
18 }
```

Visualizing the optimization path of Adam (Figure 1.4, panel C) shows a smooth path, almost like a ball rolling across the RSS function surface. This behaviour is particularly useful in the case of non-smooth objectives, flat spots, and local minima: given enough momentum, the ball will continue rolling where default gradient descent may get stuck.

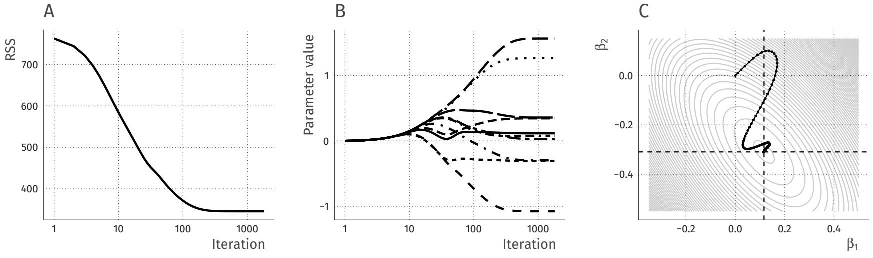

Figure 1.4 Visualization of 10-parameter Adam for linear regression. Panel A shows the residual sum of squares over the iterations (in log scale) as it approaches its minimum, panel B shows the parameter paths over iterations (in log scale), and panel C visualizes the paths of the first two parameters on the loss surface (contours, conditional on the final estimates of the remaining 8 parameters) as they approach the minimum.

Adam has several crucial advantages: it converges for a wide range of functions with a wide range of base step-sizes; it uses only the gradient, and needs limited memory and computation; it can be used as a form of stochastic gradient descent for data with many rows, taking in a small batch of data at a time. Its popularity in deep learning field stems from these advantages. I have used Adam to extend structural equations with latent variables in Chapter 2, where penalties and alternative fitting functions for SEM preclude the use of more traditional Newton-Raphson optimization.

1.3 Generalization and regularization

Why do we compute parameters of linear models? What do we want to learn from data? When is learning from data useful? One view is that learning from data means obtaining the ability to approximate new data from the data-generating process (DGP) under investigation. This view has several names, e.g., generalization, predictive fit, or predictive validity. From a practical point of view (my preferred point of view), models which generalize well are useful: they can be used for prediction or forecasting, and they also have the advantage that conclusions drawn from them are valid for new data from the same DGP (although this by no means guarantees "true"; see McElreath, 2020, p. 71 for a good historical example). These are strong arguments in favor of using generalization as a criterion in model specification and *model selection*: models which generalize well are preferred, even if they are not true models.

So how well does a particular model generalize? Unfortunately, the true level of generalization of a model can never really be known, because the true DGP is unknown in real-world research. However, generalization error or predictive fit can be *estimated* from the sample at hand. There are two historical schools to this, roughly aligning with the two cultures of statistical modeling (Breiman, 2001b): on the one hand there are information criteria (AIC, DIC, WAIC), based on asymptotics, likelihood, and degrees of freedom or the effective number of parameters; on the other hand there are computational methods, such as train-test splitting, cross-validation and leave-one-out error estimation. These two schools have the same goal: to estimate the generalization error, or how well the model predicts new data. In some cases, these schools are asymptotically equivalent (see, e.g, Stone, 1977; Browne & Cudeck, 1989; chapter 7 of Gelman et al., 2013).

In using generalization as a model selection criterion, there is a tradeoff: models which are too simplistic do not generalize well because they miss important parts of the data-generating process, and models which are too complicated do not generalize well because they tend to explain random noise, overfitting to the sample. This bias-variance tradeoff is fundamental to statistical learning: balancing fit to the sample and complexity of the model (G. James, Witten, Hastie, & Tibshirani, 2015). Overfitting in particular is a pervasive problem, and in the context of SEM this problem is compounded by the readily available metrics for improving model fit (modification indices; Bollen, 1989, p. 299), and by the established exploratory procedures around model modification. A popular term for overfitting in this context is “capitalization on chance” (MacCallum, Roznowski, & Necowitz, 1992).

There are several solutions to the problem of overfitting. An example is to make the model simpler: instead of a nonlinear spline model with many knots, a simple linear regression might generalize better. Another way is by *regularization*. Generally, regularization is anything which introduces bias in the parameters to improve generalization (Goodfellow et al., 2016), but often it amounts to shrinking the parameters towards 0 – a model with a parameter set to 0 is equivalent to a simpler model without this parameter. There are many examples of regularization procedures, including ridge estimation, early stopping, dropout training, Bayesian priors, LASSO penalties, multilevel modeling, data augmentation, and additive noise. All of these procedures prevent overfitting by adding information about the DGP to the analysis (see Table 1.1).

Table 1.1 Information encoded in various regularization strategies.

Regularization method	Encoded information
Ridge estimation	The parameters are closer to 0 than my sample suggests, and they are less correlated than my sample suggests.
Early stopping	The parameters are closer to the initial values than my sample suggests.
Dropout	The parameters have less correlation than my sample suggests.
Bayesian priors	This particular distribution encodes what I know about these parameters.
LASSO penalty	Bet on sparsity: some / most parameters are 0.
Multilevel modeling	Observations within groups are similar, so their parameters are too.
Data augmentation	Certain transformations of the data do not change what they represent, so the chosen parameters should be robust to them.
Additive noise	The parameters should be robust to a specific type of noise in the observations.

Many regularization methods can be rephrased in terms of each other. For example, early stopping and dropout are used in neural networks, as they are computationally cheap alternatives to applying explicit penalties (Goodfellow et al., 2016, p. 247) or structured priors on the parameters (Nalisnick, Hernandez-Lobato, & Smyth, 2019; Wager, Wang, & Liang, 2013). I would even argue that all these available methods are ad-hoc ways of implementing certain Bayesian priors. In this view, regularization is an inherent part of model development. Of course, explicitly specifying Bayesian priors is powerful, but it is also very difficult, and Bayesian estimation can be computationally expensive (van Erp, Oberski, & Mulder, 2019). Choosing an alternative regularization method is then not only more convenient, but sometimes even necessary to make analyses tractable.

In conclusion, I believe that all of these available regularization methods are useful in creating solutions for modern data problems. However, only a few of these regularization options are currently available for SEM (Jacobucci, Grimm, & McArdle, 2016; Merkle & Rosseel, 2015). Therefore, in the first chapter I create a SEM framework in which all of these options are readily available, and I show how these can contribute to existing SEM procedures. Furthermore, in the remainder of this dissertation I add a wide range of prior knowledge to existing SEM models, either using existing solutions or by developing new solutions: the brain is largely symmetric (Chapter 4), indicators of health in the United States exhibit racial bias (Chapter 6), childhood trauma will only affect a few locations in the genome relevant to stress reactivity (Chapter 3).

1.4 Outline

In this introduction, I have provided background on three main components that underlie the following chapters: structural equation modeling, computation and optimization, and regularization. Not all the chapters focus on all three components, but they all extensively draw upon the background given here. For example, although Chapter 4 uses existing solutions for optimization, the accompanying software package contains enhancements to stabilize the estimation procedure. Similarly, Chapter 6 does not explicitly mention regularization, but the fairness procedure used in this chapter can be seen as biasing the estimates towards the idealized “fair world” data-generating process (Nabi & Shpitser, 2018). Furthermore, Chapter 5 does not mention SEM at all, yet its contribution can be easily translated to the SEM framework. It is in the three components of this introduction – SEM, computation, and regularization – that this dissertation presents solutions to modern data problems.

The remainder of this thesis is organized as follows. In **Chapter 2** I introduce a new method of specifying and estimating structural equation models. This method is based on existing techniques from deep learning and neural networks. Using this technique, many adjustments such as regularization and penalization – often used in the analysis of modern data – are at once available to SEM. I show the advantages of this novel method in three compelling examples of useful, novel extensions to classical structural equation models.

In **Chapter 3** I develop an algorithm to perform mediation analysis (a special case of SEM) on high-dimensional, epigenetic sequencing data. The problem with this data is the large number of measurements per sample, up to hundreds of thousands of values. The algorithm is an alternative for the classic SEM estimation procedure, which cannot handle such high-dimensional situations. I make use of the widespread availability of computation to approximate “regular” mediation analysis, and I show that this new method improves upon existing high-dimensional mediation methods in various situations.

In **Chapter 4** I develop an extension to exploratory factor analysis (EFA; another special case of SEM) for use in brain imaging data. Procedures such as factor analysis are often used in this field because it reduces the amount of data with the smallest possible loss of information – an important step for further reseach in for example brain development. The extension I present makes use of the specific prior knowledge that the brain is largely symmetric, which has not been attempted before in the context of EFA. With various examples of structural and functional brain imaging data I show the flexibility of the extension, and I show that this method is an improvement to standard EFA.

In **Chapter 5** I present a solution for the problem of data analysis in the context of vertically partitioned data, meaning data where the features are stored at different locations. Privacy-sensitive medical features are an example of such data. The solution enables two parties to collaborate in estimating generalized linear models – including standard errors – by sharing only their linear prediction of the outcome variable. Using several applied examples I present an implementation of this solution, which includes encryption to safely distribute the computations.

Finally, in **Chapter 6** I propose a structural equation model to tackle a different modern data problem: algorithmic fairness. This chapter is based on a situation where medical predictions based on register data lead to a racially biased treatment of white patients over black patients. I show in a real-world dataset that this problem does not occur when using a classical latent variable model for prediction, in combination with existing techniques for fair inference.

Chapter 2

Flexible Extensions to Structural Equation Models using Computation Graphs

Van Kesteren, E. J., & Oberski, D. L. (2019, in preparation) Structural equation models as computation graphs. arXiv:1905.04492

Structural equation modeling (SEM) is being applied to ever more complex data types and questions, often requiring extensions such as regularization or novel fitting functions. To extend SEM, researchers currently need to completely reformulate SEM and its optimization algorithm – a challenging and time-consuming task. In this paper, we introduce the *computation graph* for SEM, and show that this approach can extend SEM without the need for bespoke software development. We show that both existing and novel SEM improvements follow naturally. To demonstrate, we introduce three SEM extensions: least absolute deviation estimation, Bayesian LASSO optimization, and sparse high-dimensional mediation analysis. We provide an implementation of SEM in PyTorch – popular software in the machine learning community – to accelerate development of structural equation models adequate for modern-day data and research questions.

2.1 Introduction

Structural equation modeling (SEM) is a popular tool in the social and behavioural sciences, where it is being applied to ever more complex data types. For example, SEM extensions now perform variable selection in high-dimensional situations (Jacobucci, Brandmaier, & Kievit, 2018; van Kesteren & Oberski, 2019), modeling of intensive longitudinal data (Asparouhov et al., 2018; Voelkle & Oud, 2013), and analysis of intricate online survey experiments (Cernat & Oberski, 2019). In these situations, the SEM model often needs to be reformulated and traditional optimization approaches need to be extended to obtain parameter estimates – a challenging and time-consuming task. For

Author contributions: EJK & DLO developed the idea. EJK created the software, implemented the experiments, and wrote the manuscript. DLO provided extensive feedback on all components.

example, applying SEM to high-dimensional data necessitates parameter penalization, and special model types such as genomic SEM (Grotzinger et al., 2019) or network models (Epskamp, Rhemtulla, & Borsboom, 2017) can lead to alternative fitting functions. Additionally, even before the extension of SEM to novel data structures there have been several examples of the instability of the latent variable approach – such as Heywood cases (Kolenikov & Bollen, 2012) and convergence problems in multitrait-multimethod (MTMM) models (Revilla & Saris, 2013), which may benefit from regularization to obtain a stable result.

While the current growth of new types of structural equation models is exciting, developments in SEM are still far from caught up with the state-of-the-art in modern data analysis. In particular, the machine learning literature has exploded over the past decades to develop methods that deal with the complex nature of modern data, making great strides in difficult data analysis problems, including computer vision, natural language processing, and genomics (see Goodfellow et al., 2016, and the references therein for an overview). Each of these data sources holds great potential for questions traditionally addressed in SEM, in particular those found in the social, behavioral, ecological, or biomedical sciences. However, traditional implementations of SEM are difficult to integrate with the solutions pioneered in the field of machine learning.

In this paper, we propose allowing direct integration of SEM and methods from the field of deep learning, by specifying SEM as a *computation graph.* We demonstrate the utility of our approach by straightforwardly implementing three potentially useful extensions to SEM, of which two are novel:

1. We implement Least Absolute Deviation (LAD) estimation, which exhibits robustness to outliers in the residual covariance matrix (Siemsen & Bollen, 2007).

2. To deal with high-dimensional indicators, we create a novel Bayesian LASSO estimation procedure (Park & Casella, 2008), and we apply it to an existing dataset to obtain a sparse linear combination of audio recording features related to Parkinson's disease status at the latent variable level.

3. To analyze mediation models in which there are more potential mediators than rows, we develop a variant of sparse high-dimensional mediation analysis based on unweighted least squares (ULS). Using this method, we perform exploratory mediator selection in an epigenetic dataset (Schaid & Sinnwell, 2020; van Kesteren & Oberski, 2019; Zhang et al., 2016).

These extensions are intended to demonstrate the power and flexibility of the proposed approach. The main purpose of this paper is to make this approach available to the SEM community to facilitate rapid development of novel extensions to SEM that will be useful in modern-day applications. To this end, we also provide an open source software package, `tensorsem` (doi:10.5281/zenodo.3957287).

This paper is structured as follows. First, SEM will be framed as an optimization problem, and a brief overview will be given of the current methods of SEM parameter estimation. Then, we will introduce the concept of computation graphs, as used in the field of deep learning. Subsequently, we will develop the computation graph for SEM, after which we show how this can be used to extend SEM to novel situations. Lastly, we discuss the implications of this novel framework for SEM and we provide directions for future research. The methods introduced this paper are reproducible using the open-source software we have developed (github.com/vankesteren/tensorsem), combining the popular R package lavaan (R Core Team, 2018; Rosseel, 2012) and the PyTorch neural network software (Paszke et al., 2019). All the examples associated with this paper are reproducible using the code in the supplementary material.

2.2 Background

2.2.1 SEM as an optimization problem

SEM in its basic form (Bollen, 1989) is a framework to model the covariance matrix of a set of observed variables. Through separation of structural and measurement models, it enables a wide range of multivariate models with both observed and latent variables. SEM generalizes many common data analysis methods, such as linear regression, seemingly unrelated regression, errors-in-variables models, confirmatory and exploratory factor analysis (CFA / EFA), multiple indicators multiple causes (MIMIC) models, instrumental variable models, random effects models, and more.

Below, we reiterate how the parameter configuration of the SEM framework creates a model-implied covariance matrix. Then, we show how this matrix is the basis for objective functions representing the distance between the model-implied and the observed covariance matrix. Next, we show how such objective functions are used to estimate the parameters of interest in the maximum likelihood (ML) and generalized least squares (GLS) frameworks.

The most commonly used formulations of SEM are the LISREL notation (Jöreskog & Sörbom, 1993) used in software packages such as lavaan (Rosseel, 2012) and the Reticular Action Model (RAM) notation (McArdle & McDonald, 1984) used in software such as OpenMX (Neale et al., 2016). In this paper, we adopt a variant of the LISREL notation used in lavaan and Neudecker and Satorra (1991), also known as the "all-y" version:

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\varepsilon} \quad \text{(Measurement model)} \\ \boldsymbol{\eta} &= \boldsymbol{B}_0\boldsymbol{\eta} + \boldsymbol{\xi} \quad \text{(Structural model)} \end{aligned} \tag{2.1}$$

where $\boldsymbol{z}$ represents a vector of centered observable variables of length P, and $\boldsymbol{\eta}$, $\boldsymbol{\varepsilon}$, and $\boldsymbol{\xi}$ are random vectors such that $\boldsymbol{\varepsilon}$ is uncorrelated with $\boldsymbol{\xi}$ (Neudecker &

Satorra, 1991). The parameters of the model are encapsulated in four matrices: $\boldsymbol{\Lambda}$ contains the factor loadings, $\boldsymbol{\Psi}$ contains the covariance matrix of $\boldsymbol{\xi}$, $\boldsymbol{B}_0$ contains the regression parameters of the structural model, and $\boldsymbol{\Theta}$ contains the covariance matrix of $\boldsymbol{\varepsilon}$. From these matrices, we construct the full parameter vector $\boldsymbol{\delta}$ as follows:

$$\boldsymbol{\delta} = \left[(\operatorname{vec}\boldsymbol{\Lambda})^T, (\operatorname{vech}\boldsymbol{\Theta})^T, (\operatorname{vech}\boldsymbol{\Psi})^T, (\operatorname{vec}\boldsymbol{B}_0)^T\right]^T \tag{2.2}$$

where the vec operator transforms a matrix into a vector by stacking the columns, and the vech operator does the same but eliminates the supradiagonal elements of the matrix. Specific models impose specific restrictions on this parameter vector. This leads to a subset of *free parameters* $\boldsymbol{\theta}$. $\boldsymbol{\delta}$ is identified through predefined restrictions: $\boldsymbol{\delta} = \boldsymbol{\delta}(\boldsymbol{\theta})$. The model-implied covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is a function of the free parameters, defined as follows (Bock & Bargmann, 1966; Jöreskog, 1966):

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{\Lambda}\boldsymbol{B}^{-1}\boldsymbol{\Psi}\boldsymbol{B}^{-T}\boldsymbol{\Lambda}^T + \boldsymbol{\Theta} \tag{2.3}$$

where $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{B}_0$ is assumed to be non-singular – that is, the structural path model $\boldsymbol{B}_0$ is assumed to be identified.

In order to estimate $\boldsymbol{\theta}$, a fitting function needs to be defined. All common SEM objectives are measures of the distance between the model-implied covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ and the observed covariance matrix $\boldsymbol{S}$: the model fits better if the model-implied covariance matrix more closely resembles the observed covariance matrix. The maximum-likelihood (ML) objective function F_{ML} is such a distance measure. Under the assumption that the observed covariance matrix follows a Wishart distribution or, equivalently, the observations follow a multivariate normal distribution, the maximum-likelihood fitting function is the following (Bollen, 1989; Jöreskog, 1967):

$$F_{\text{ML}}(\boldsymbol{\theta}) = \log|\boldsymbol{\Sigma}(\boldsymbol{\theta})| + \operatorname{tr}\left[\boldsymbol{S}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\right] \tag{2.4}$$

Note that the ML fit function is a special case of the generalized least squares (GLS) fitting function (Browne, 1974) which is defined as the following quadratic form:

$$F_{\text{GLS}}(\boldsymbol{\theta}) = (\boldsymbol{s} - \boldsymbol{\sigma}(\boldsymbol{\theta}))^T\boldsymbol{W}(\boldsymbol{s} - \boldsymbol{\sigma}(\boldsymbol{\theta})) \tag{2.5}$$

Where $\boldsymbol{s} = \operatorname{vech}\boldsymbol{S}$, and $\boldsymbol{\sigma}(\boldsymbol{\theta}) = \operatorname{vech}\boldsymbol{\Sigma}(\boldsymbol{\theta})$. Here, $F_{\text{GLS}} = F_{\text{ML}}$ when $\boldsymbol{W} = 2^{-1}\boldsymbol{D}^T(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}) \otimes \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}))\boldsymbol{D}$ (Neudecker & Satorra, 1991), where $\boldsymbol{D}$ is the duplication matrix and $\otimes$ indicates the Kronecker product. Other choices for $\boldsymbol{W}$ lead to other estimators, such as unweighted least squares (ULS) or diagonally weighted least squares (DWLS).

With this formulation, the gradient $\boldsymbol{g}(\boldsymbol{\theta})$ of F_{GLS} with respect to the parameters $\boldsymbol{\theta}$ and the Hessian $\boldsymbol{H}(\boldsymbol{\theta})$ – the matrix of second-order derivatives – were derived by Neudecker and Satorra (1991). These two quantities are the basis for standard errors, robust statistical tests for model fit (Satorra & Bentler, 1988), as well as fast and reliable Newton-type estimation algorithms (Lee & Jennrich, 1979). One such algorithm is the Newton-Raphson algorithm, where the parameter estimates at iteration $i+1$ are defined as the following function of the estimates at iteration i:

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \boldsymbol{H}^{-1}(\boldsymbol{\theta}^{(i)}) \cdot \boldsymbol{g}(\boldsymbol{\theta}^{(i)}) \tag{2.6}$$

Together, the objective function and the algorithm comprise an *estimator* – a way to compute parameter estimates using the data. Note that it is developed specifically for GLS estimation of SEM. With every extension to GLS, this work needs to be redone: a bespoke new estimator – objective, gradient, Hessian, and algorithm – needs to be derived and implemented.

2.2.2 Optimization problems as computation graphs

In this paper, we suggest implementing the SEM optimization problem as a computation graph, to leverage the advances of the deep learning field for extending the SEM framework. A computation graph is a graphical representation of the operations required to compute a loss or objective value $F(\boldsymbol{\theta})$ from (a vector of) parameters $\boldsymbol{\theta}$ (Abadi et al., 2016). The full computation is split into a series of differentiable smaller computational steps. Each of these steps is represented as a node, with directed edges representing the flow of computation towards the final result. Because the nodes are differentiable, computing gradients of the final (or any intermediate) result with respect to any of its inputs is automatic. Gradients are obtained by applying the chain rule of calculus starting at the node of interest and moving against the direction of the arrows in the graph. Thus, computation graphs are not only a convenient way of representing an objective function in a computer, but they also immediately provide the derivatives (and second derivatives), which are necessary to optimize functions or estimate standard errors. For example, consider the familiar ordinary least squares objective for linear regression:

$$F_{\text{LS}}(\boldsymbol{\beta}) = \sum_i (y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{2.7}$$

The computation graph of this objective function can be constructed as in Figure 2.1. This figure represents the objective by ”unwrapping” the equation from the inside outward into separate matrix operations: first, there is a matrix-vector multiplication of the design matrix $\boldsymbol{X}$ with the parameter vector $\boldsymbol{\beta}$. Then, the resulting $n \times 1$ vector $\hat{\boldsymbol{y}}$ is subtracted elementwise from the observed outcome $\boldsymbol{y}$, and the result is squared, then summed to output a single squared error loss value F_{LS}. The nodes in a computation graph may represent scalars, vectors, matrices, or even three- and higher-dimensional arrays. Generally, these nodes are referred to as *tensors*.

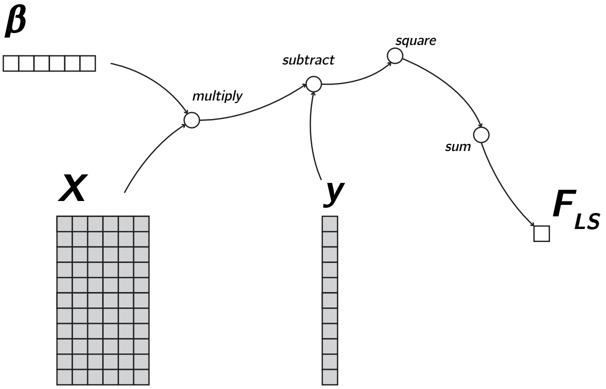

Figure 2.1 Least squares regression computation graph, mapping the regression coefficients ($\boldsymbol{\beta}$) to the least squares objective function F_{LS}. The greyed-out parts contain elements which do not change as the parameters are updated, in this case observed data.

Each of the operations in the graph has a registered derivative function. If it is known that the ”square” operation $f(x) = x^2$ is applied as in Figure 2.1, the derivative $f'(x)$ is $2x$ and the second-order derivative $f''(x)$ is 2. Thus, the gradient of the squared error tensor with respect to the residual tensor $\boldsymbol{r}$ is $2\boldsymbol{r}$.

Although automatic differentiation is an old idea (Wengert, 1964), its combination with state-of the art optimizers (see Appendix A.1) in software such as `Torch` (Collobert, Bengio, & Mariéthoz, 2002; Paszke et al., 2017) and `TensorFlow` (Abadi et al., 2016) have paved the way for the current pace of deep learning research. Before the development and implementation of the computation graph, each neural network configuration (model) required specialized work on the part of the researchers who introduced it to provide a novel estimator. Thanks to computation graphs, researchers can design generic neural nets without needing to invent a bespoke estimator. This development has greatly accelerated progress in this area. For SEM, we see a similar situation at the moment: each development or extension of the model currently requires a new algorithm that is capable of estimating its parameters. By applying computation graphs to SEM, we hope to greatly accelerate the process of developing novel SEM models. In the next section we combine the parameter configuration developed for SEM with the computation graphs and optimizers developed for deep learning to create a more flexible form of SEM.

2.3 Flexible extensions to SEM using computation graphs

In this section, we develop the computation graph and parameter configuration to perform default ML-based structural equation modeling using `PyTorch`

(Paszke et al., 2019). Then, we outline how this computation graph can be edited to extend SEM to novel situations, and how additional penalties can be imposed on any parameter in the model. In the next section, we show examples of such edits.

2.3.1 The SEM computation graph

The SEM computation graph for the LISREL all-y notation is displayed in Figure 2.2. From left to right, a parameter vector $\boldsymbol{\delta}$ is first instantiated with constrained elements, such that the free parameters represent $\boldsymbol{\theta}$. Then, this vector is split into the separate vectors as in Equation 2.2. These vectors are then reshaped into the four SEM all-y matrices, using duplication indices for the symmetric matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\Theta}$.

In the next part, these matrices are transformed to the model-implied covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ by unwrapping Equation 2.4 from the inside outward: $\boldsymbol{B}^{-1}$ is constructed as $(\boldsymbol{I} - \boldsymbol{B}_0)^{-1}$, then $\boldsymbol{\Psi}$ is premultiplied by this tensor and postmultiplied by its transpose. Then, the resulting tensor itself is pre - and postmultiplied by $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^T$, respectively. Lastly, $\boldsymbol{\Theta}$ is added to construct the implied covariance tensor.

The last part is the graphical representation of the ML fit function from Equation 2.4. $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is inverted, then premultiplied by $\boldsymbol{S}$, and the trace of this tensor is added to the log determinant of the inverse of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The resulting tensor, a scalar value, is the $F_{\mathrm{ML}}(\boldsymbol{\theta})$ objective function for SEM.

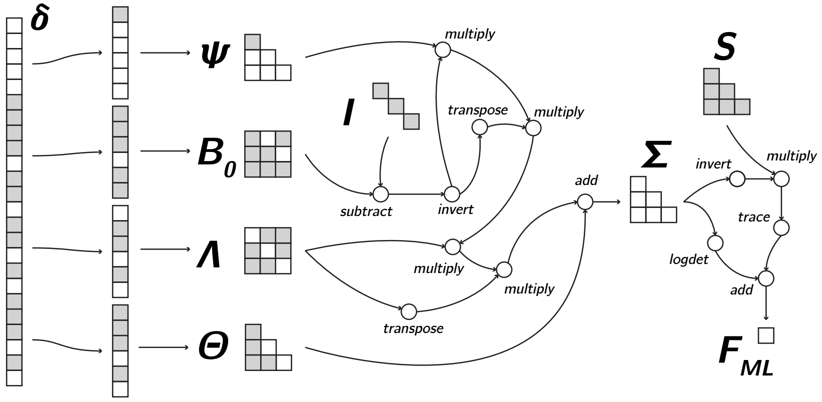

Figure 2.2 Full computation graph for all-y structural equation model, mapping the parameters ($\boldsymbol{\delta}$) to the maximum likelihood fit function F_{ML}. The greyed-out parts contain elements which do not change during model fitting, meaning either observed data or constrained parameters. (NB: The constrained elements in this graph are not representative of a specific model).

Each operation in Figure 2.2 carries with it information about its gradient. PyTorch can therefore automatically compute gradients of to the model parameters with respect to the fit function in the SEM computation graph. The Hessian can also be obtained automatically by applying the same principle to the gradients. Note that these correspond to the observed score and information matrix, rather than their expected versions derived under the null hypothesis of model correctness. PyTorch also provides state-of-the-art optimizers such as Adam Kingma and Ba (2014) to optimize computation graphs using these quantities (see Appendix A.1 for more background on these optimizers). The supplementary material contains a python package which implements this computation graph, along with example code to estimate lavaan models using this package.

2.3.2 Editing the objective function

The computation graph approach allows completely different objective functions to be implemented with relative ease. One such objective was coined by Siemsen and Bollen (2007), who introduce least absolute deviation (LAD) estimation. Their motivation is the performance of the LAD estimator as a robust estimation method in other fields. Note that while Siemsen and Bollen find limited relevance for this SEM estimator in terms of performance, we consider it to be an excellent showcase of the flexibility of our approach. This objective does not fit in the GLS approach of Browne (1974). The LAD estimator implies the following objective:

$$F_{\mathrm{LAD}}(\boldsymbol{\theta}) = \sum_{i,j} |\boldsymbol{\Sigma}(\boldsymbol{\theta})_{i,j} - \boldsymbol{S}_{i,j}| \tag{2.8}$$

A computational advantage of this objective relative to the ML fit function is that there is no need to invert $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The work of Siemsen and Bollen (2007) focuses on developing a greedy genetic evolution numerical estimation algorithm which performs a search over the parameter space. Using this optimization algorithm, they show that the LAD estimator may outperform the ML estimator in very specific situations.

Constructing the LAD estimator in the computation graph framework means replacing the ML fit operations with the LAD operations. This is shown in Figure 2.3. Note that compared to the ML objective, there are fewer operations, and the inversion operation of the implied covariance matrix is removed. This change is trivial to make given the SEM computation graph, and we will show later in the Examples section that such alternative objective functions can be estimated using PyTorch.

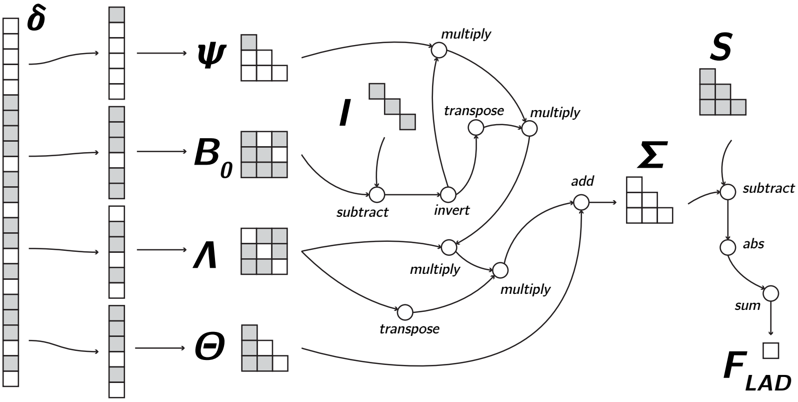

Figure 2.3 Full SEM computation graph for the least absolute deviation (LAD) objective. Compared to the ML fit function, the last part of the graph contains different operations.

2.3.3 Adding parameter penalization

A more useful modification of default SEM is the addition of penalties to the parameters of structural equation models (Holmes Finch & Miller, 2020; P.-H. Huang, Chen, & Weng, 2017; Jacobucci et al., 2016). Such penalties *regularize* the model, which may prevent overfitting and improve generalizability (Hastie, Tibshirani, & Wainwright, 2015). There is a wide variety of parameter penalization procedures, but the most common methods are ridge and LASSO. In regression, the widely used elastic net (Zou & Hastie, 2005) is a combination of the LASSO and ridge penalties. The objective function for elastic net is the following:

$$F_{\text{EN}}(\boldsymbol{\beta}) = F_{\text{LS}}(\boldsymbol{\beta}) + \lambda_1 \left\|\boldsymbol{\beta}\right\|_1 + \lambda_2 \left\|\boldsymbol{\beta}\right\|_1^2 \tag{2.9}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_p |\beta_p|$, and λ_1 and λ_2 are hyperparameters which determine the amount of LASSO and ridge shrinkage, respectively. By setting λ_1 to zero we obtain L2 (ridge) shrinkage, and setting λ_2 to zero yields the L1 (LASSO). Nonzero values for both parameters combines the two approaches, which has been shown to encourage a grouping effect in regression, where strongly correlated predictors tend to be in or out of the model together (Zou & Hastie, 2005).

Friedman et al. (2010) have developed an efficient algorithm for estimating the elastic net for generalized linear models and have implemented this in their package `glmnet`. For SEM, (Jacobucci et al., 2016) have created a package for performing penalization by adding the elastic net penalty to the ML fit function. Their implementation uses the RAM notation (McArdle & McDonald, 1984), and their suggestion is to penalize either the $\boldsymbol{A}$ matrix (factor loadings and regression coefficients), or the $\boldsymbol{S}$ matrix (residual covariances).

In the field of deep learning, parameter penalization is one of the key mechanisms by which massively overparameterized neural networks are estimated (Goodfellow et al., 2016). Regularization is therefore a core component of various software libraries for deep learning, including PyTorch. The optimizers implemented in these libraries, such as Adam (Kingma & Ba, 2014), are tried and tested methods for estimation of neural networks with penalized parameters, which is an active field of research (e.g., Scardapane, Comminiello, Hussain, & Uncini, 2017).

In the SEM computation graph, the LASSO penalty on the regression parameters can be readily implemented by adding a few nodes to the ML fit graph. This is displayed in Figure 2.4. The absolute value of the elements of the $\boldsymbol{B}_0$ tensor are summed, and the resulting scalar is multiplied by the tuning parameter. The resulting value is then added to the maximum likelihood fit tensor to construct the lasso objective $F_{\text{LASSO}}(\boldsymbol{\theta})$.

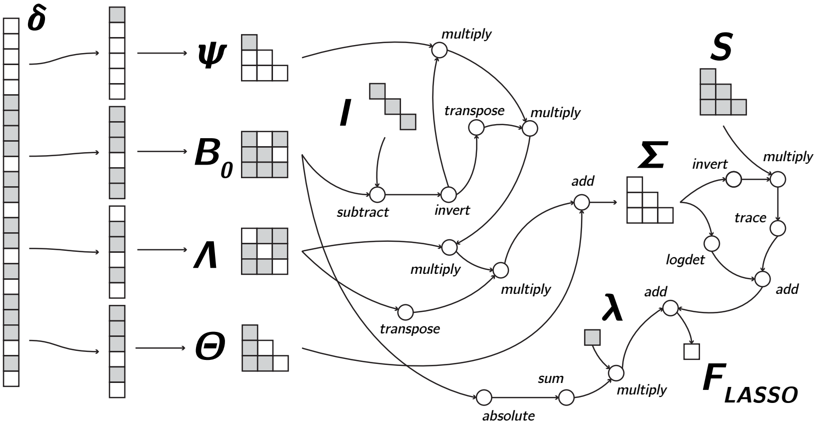

Figure 2.4 $\boldsymbol{B}_0$ LASSO computation graph with a pre-defined λ tuning parameter.

Ridge penalties for the $\boldsymbol{B}_0$ matrix can be implemented in similar fashion, but instead of an "absolute value" operation, the first added node is a "square" operation. These penalties can be added to any tensor in the computation graph, meaning penalization of the factor loadings or the residual covariances, or even a penalty on $\boldsymbol{B}$ is quickly implemented. The elastic net penalty specifically can be implemented by imposing both a ridge and a lasso penalty on the tensor of interest.

Note that each additional penalty comes with its own parameter to be selected – a process called "hyperparameter tuning". Tuning of penalty parameters is traditionally done through cross-validation; glmnet (Friedman et al., 2010) provides a function for automatically selecting the penalization strength in regression models through this method. Another method is through inspecting model fit criteria. For example, Jacobucci et al. (2016) suggest selecting the penalty parameter through the BIC or the RMSEA, where the degrees of freedom is determined by the amount of *nonzero* parameters, which changes as a function of the penalization strength. Another example is penalized network estimation, where Epskamp, Borsboom, and Fried (2018) suggest hyperparameter tuning through an extended version of the BIC. There is another option, used in both deep learning as well as Bayesian statistics: a prior can be set on the hyperparameters. In this way, the parameter itself is learned along with the model: the "full Bayes" approach (van Erp et al., 2019). In the deep learning literature, this is called Bayesian optimization or gradient-based optimization of hyperparameters (Bengio, 2000). In the Examples section, we show how the Bayesian LASSO approach (Park & Casella, 2008) can be leveraged for sparse factor analysis in SEM, a completely novel extension.

2.3.4 Standard errors and model tests

In SEM, standard errors can be calculated through the Fisher information method, requiring only the Hessian of the log-likelihood at the maximum and the assumption that the distribution this log-likelihood is based on (usually Normal-theory) is correct. Additionally, the distributional assumption can be relaxed by using sandwich estimators, in SEM known as Satorra-Bentler (robust) standard errors. These need both the Hessian and the $N \times P$ outer product matrix Δ – the case-wise first derivatives of the parameters w.r.t. the implied covariances $\sigma(\theta)$ (Savalei, 2014). Sandwich estimators also lead to robust test statistics which are not sensitive to deviations from normality. In econometrics, many variations of the sandwich estimator are available, depending on whether the expected or observed information matrix is used (Kolenikov & Bollen, 2012).

Computation graphs as outlined in this section are a general approach for obtaining parameter estimates of structural equation models. Moreover, for the ML computation graph (Figure 2.2) it is also possible to obtain accurate standard errors because the observed information matrix – the inverse of the Hessian of the log-likelihood – is available automatically through the gradient computation in PyTorch. In addition, through the same computation graph but with case-wise entering of the data, the outer product matrix Δ can also be made available. Because these are the observed versions, computation of empirical sandwich (Huber-White) standard errors is possible. Naturally, an established alternative to these procedures is to bootstrap in order to obtain standard errors. Furthermore, the log-likelihood itself is directly available, thus information criteria such as AIC, BIC and SSABIC (Sclove, 1987), as well as normal-theory and robust test statistics (Satorra & Bentler, 1988) can be computed more or less "as usual".

In principle, therefore, standard errors and test statistics are available when using the computation graph approach. However, in practice the computation graph can be edited arbitrarily by introducing penalties or a different fit function. In this case, no general guarantees can be given about the accuracy of the standard errors, the coverage probability of the confidence interval, or the asymptotic behaviour of model fit metrics derived from the obtained model. This is inherent to the flexibility of the computation graph approach: for existing methods in SEM, simulations have shown the performance of the current standard error solutions (including the bootstrap), but as extensions are introduced these results do not necessarily hold. For some extensions, there will be no adequate approximation to the standard error with accurate frequentist properties. For example, there is a large body of literature on standard error approximations for $L1$ penalization (e.g., Fan & Li, 2001), but the problem of obtaining penalized model standard errors is fundamentally unsolvable due to the bias introduced by altering the objective function away from the log-likelihood (Goeman, Meijer, & Chaturvedi, 2018, p. 18). Not even the bootstrap can provide consistent standard error estimates in these situations (Kyung, Gill, Ghosh, & Casella, 2010). Hence, software implementations of penalized regression (e.g., `glmnet`) consciously omit standard errors.

In situations beyond ML, our advice is to pay attention to the behaviour of existing fit criteria and standard errors. Using simulations for each new model and data case, the frequentist properties of the empirical confidence interval can be assessed and the type-I and type-2 errors of the (Satorra-Bentler) χ^2 test can be found. Those values can then be used to adjust the interpretation of the results in the analysis of the real data. If existing standard error approaches fail altogether, a viable solution may be to completely omit standard errors – just as in the $L1$ regression approach.

Note that all of the above holds similarly for Bayesian estimation, where the choice of prior influences the frequentist properties of the posterior, such as the credible interval coverage probability. Just as it is possible with the computation graph approach to create a nonconverging model with bad asymptotic behaviour, it is possible with Bayesian methods to create such a problematic model through the choice of nonsensical priors. Solutions in this case are also based on simulation, e.g., prior predictive checking (Gabry, Simpson, Vehtari, Betancourt, & Gelman, 2019) or leave-one-out cross-validation (Vehtari, Gelman, & Gabry, 2017).

In the next section, we show through a set of examples motivated by existing literature how our implementation of the SEM computation graph can be used to create extensions such as the ones we have introduced in this section.

2.4 Examples

In this section, we implement three completely novel estimation procedures for SEM using our computation graph approach. The first example demonstrates how non-standard extensions to the fit function can be implemented with relative ease: we show how the Least Absolute Deviation (LAD) estimator yields similar parameters to the ML estimator in a factor analysis, even when the covariance matrix is contaminated with wrong values. Then, we perform a structural equation model with a sparse factor, using full Bayesian LASSO regularization (Park & Casella, 2008) for the factor loadings. To our knowledge, the full Bayesian optimization approach with hyperpriors has not previously been performed in the context of factor analysis with a covariate. Then, we perform high-dimensional mediation analysis with ULS optimization and LASSO regularization, using sparsity to select relevant variables among a set of 110 potential mediators. This procedure is also novel, and through our approach it can be implemented relatively simply.

All the computation graphs and estimation methods described in this paper are reproducible through the code in the supplementary material, as well as the python package available at `github.com/vankesteren/tensorsem`. Prior to implementing these examples, we have checked the validity of our `PyTorch` implementation for default and regularized SEM against several other packages. The results of this are shown in Appendix A.2.

2.4.1 LAD estimation

Although LAD estimation was shown to be beneficial only in very specific situations (Siemsen & Bollen, 2007), it is an excellent showcase for the flexibility of the computation graph approach. Because the software developed by Siemsen and Bollen (2007) is not available, we instead compare the LAD estimates to the ML estimates. The `PyTorch` LAD estimator is a completely novel way of estimating SEM.

For this example, we generate data of sample size 1000 from a one-factor model. For this data, we constrain the observed covariance matrix to the covariance matrix implied by the population model in Figure 2.5. Since LAD estimation should be robust to outliers in the observed covariance matrix, which can happen in the trivial case of mistranscribing a covariance matrix into software, we also performed this on data with a “contaminated” covariance matrix: $COV(X_1, X_3) = 2$, $COV(X_2, X_4) = 0.35$.

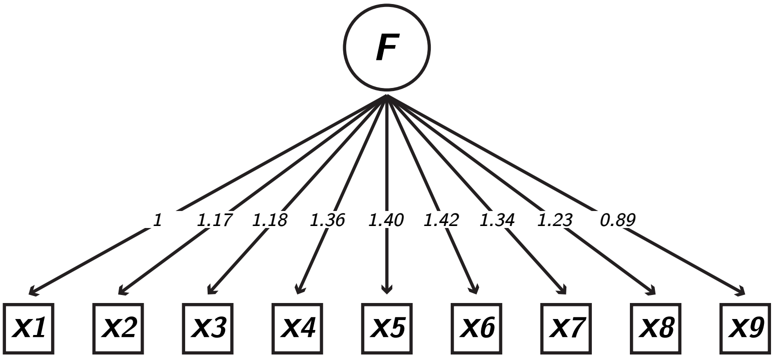

Figure 2.5 Factor analysis model used to generate data for comparing the least absolute deviation (LAD) estimator to the maximum likelihood (ML) estimator in `tensorsem`. Residual variances of the indicators were all set to 1.

The results are shown in Table 2.1. The for ML estimation in `lavaan` and `tensorsem` again agree. With the uncontaminated covariance matrix, the LAD estimates reach the same conclusion as the ML estimates. Note that although unbiased, LAD is relatively less efficient, but this effect is not visible with a sample size of 1000 for this model. With contamination in the covariance matrix, the LAD method shows no bias, whereas the ML method does. Because the Hessian for the LAD objective is not invertible, the standard errors are not available using the previously described ACOV($\boldsymbol{\theta}$) method. Siemsen and Bollen (2007) solve this problem by bootstrapping, which is possible but outside the scope of the current paper.

Table 2.1 Parameter estimates comparing ML estimates to LAD estimates using both uncontaminated (u, top) and contaminated (c, bottom) covariance matrices. LAD is robust to the contamination of the covariance matrix.

	X1	X2	X3	X4	X5	X6	X7	X8	X9
Uncontaminated (ML)	1.00	1.17	1.18	1.36	1.40	1.42	1.34	1.23	0.89
Contaminated (ML)	1.00	0.93	1.15	1.10	1.22	1.24	1.17	1.07	0.78
Uncontaminated (LAD)	1.00	1.17	1.18	1.36	1.40	1.42	1.34	1.23	0.89
Contaminated (LAD)	1.00	1.17	1.18	1.36	1.40	1.42	1.34	1.23	0.89

The results from this example show that the objective function in `PyTorch` can be edited and that Adam still converges to a stable solution with this adjusted objective. The parameter estimates from LAD estimation approximate those obtained from ML estimation in a one-factor model with 9 indicators and 1000 observations. In addition, we have observed that LAD estimation is robust to contamination of the covariance matrix in the contrived example of this section. Note that SEM estimation can be made robust against outliers in the raw data through using a multivariate t likelihood (Asparouhov & Muthén, 2016; Lai & Zhang, 2017; Yuan & Bentler, 1998), which is possible in the computation graph approach but outside the scope of the current paper.

2.4.2 Sparse factor SEM

Obtaining sparsity in factor analysis is a large and old field of research, with methods including rotations of factor solutions in principal component analysis (Kaiser, 1958) and modification indices in CFA (Saris, Satorra, & Sörbom, 1987; Sörbom, 1989). Sparsity is desirable in factor analysis due to the enhanced interpretability of the obtained factors. Recently, penalization has been applied to different factor analysis situations in order to obtain sparse factor loadings and simple solutions (Choi, Oehlert, & Zou, 2010; Jin, Moustaki, & Yang-Wallentin, 2018; Lu, Chow, & Loken, 2016; Pan, Ip, & Dubé, 2017; Scharf & Nestler, 2019). In addition, traditional factor rotations have been combined with SEM in a unified framework called exploratory SEM (ESEM, Asparouhov & Muthén, 2009). Several implementations of factor loading regularization now exist in SEM (e.g. Guo, Zhu, Chow, & Ibrahim, 2012; P.-H. Huang et al., 2017; Jacobucci et al., 2016).

Following these recent developments, in this example we impose sparse structure in a factor by imposing a penalty on the relevant elements of the $\mathbf{\Lambda}$ matrix. We reuse the example of Choi et al. (2010), who created a new lasso estimator for factor analysis and tested their method on an open Parkinson dataset. The example dataset is taken from the UCI Machine Learning repository (Blake & Merz, 1998) and is based on N audio recordings of people with and without Parkinson's disease. Certain biologically inspired features (Little, McSharry, Roberts, Costello, & Moroz, 2007) of these audio recordings can be related to the disease status of the participants. In this example, we seek to find a sparse linear combination of these features which can be explained by the disease status. Note that this feature-based representation is similar in idea to Guo et al. (2012), who used Bayesian LASSO to select among basis functions, creating non-linear spline relations between latent variables and their indicators.

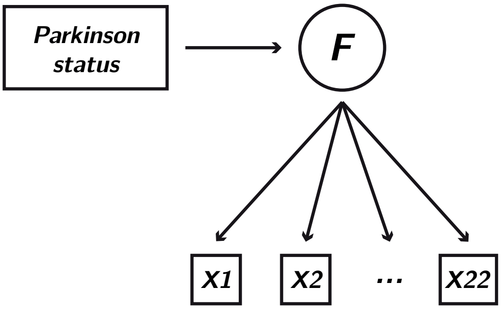

Figure 2.6 Model applied to the Parkinson's data. Parkinson status is a binary variable, X1 - X22 are biologically-inspired features of the audio recording, normalized and standardized to follow a conditional normal distribution.

The model applied to the data is shown in Figure 2.6. After standardization and log-transforming the skewed features (see supplementary material for the full pre-processing pipeline), ML estimates for this model were obtained using standard SEM software (OpenMx; Neale et al., 2016, NB: `lavaan`'s optimization reached a local minimum), as well as via our `PyTorch` implementation. Then, a Bayesian LASSO penalty was added to the model: the objective function was equal to the ML fit function (Equation 2.4) plus a Laplace prior on the factor loadings with a $Gamma(1.78, 1)$ hyperprior on the scale of the double exponential distribution (Park & Casella, 2008). The resulting factor loadings and factor scores are shown in Figure 2.7.

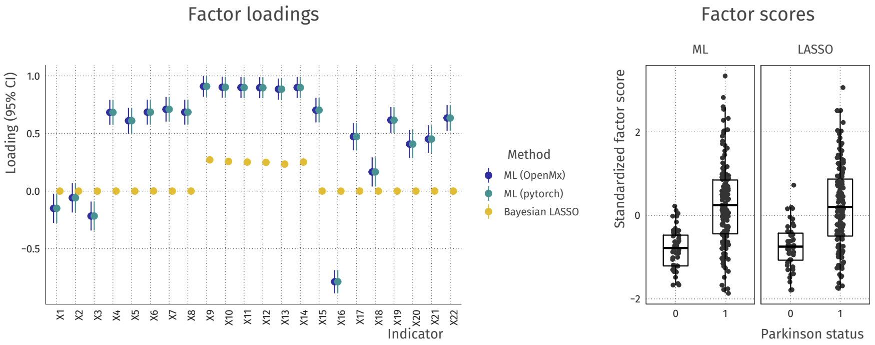

Figure 2.7 Factor loadings (left panel) and factor scores (right panel) in the Parkinson's disease dataset. All but 6 features are set to 0 when estimating the model using a Bayesian LASSO. Error bars are omitted for this method as the quadratic approximation is known to produce inconsistent confidence intervals in this case. The right panel shows that the LASSO factor scores exhibit very similar properties when compared to the ML factor scores despite the sparsity.

The figure shows that the factor scores exhibit very similar class separation, despite the sparsity of the LASSO solution. In other words, using fewer features, a similar amount of information about the disease status is encoded in this factor. In this example, variable selection is informed by the disease status variable. The penalty parameter is learned automatically, along with the remaining variables. Furthermore, in this framework it is easy to extend the penalty to adaptive LASSO, where the strength of penalization is determined on a per-feature basis (Guo et al., 2012), or any of the myriad of alternative penalty functions, some of which are known to exhibit less bias in the nonzero parameters than the LASSO (van Erp et al., 2019).

2.4.3 Sparse high-dimensional mediation

In this last example, we implement high-dimensional mediation analysis. This procedure is becoming more relevant as high-dimensional data becomes accessible due to reductions of cost and increasing availability of complex measurement devices. The motivating example for this high-dimensional mediation procedure can be found in Houtepen et al. (2016) and van Kesteren and Oberski (2019): childhood trauma scores of participants were measured using a standard questionnaire, and their reactivity to stress later in life was measured using their cortisol patterns after a stressor. Gene methylation was measured for each participant and hypothesized to mediate the relation between childhood trauma and stress reactivity. The goal of this study was to identify locations in the genome where methylation has an influence on the relation between childhood trauma and stress reactivity, as a potential target for future research.

Crucially, ML estimation is not available with high-dimensional data, where the parameters outnumber the rows in the dataset, because the model-implied covariance matrix is not invertible. However, other analysis methods such as LAD estimation (Section 2.4.1) and ULS estimation do not need to invert the model-implied covariance matrix to obtain parameter estimates. In this section we use ULS estimation with a LASSO penalty on the paths. In this way, we perform variable selection among the mediators, while taking into account potential residual correlations between the mediators.

To test our approach, we have simulated a dataset following the same pattern as the motivating example. It contains 110 potential mediators, of which only 10 are true mediators with an indirect effect of .25. There are only 40 rows, making the dataset high-dimensional (for more details on data generation, see van Kesteren & Oberski, 2019). Using this data, two mediation models were fit in `PyTorch`, one with only the ULS loss function, and one with the ULS loss function plus the sum of the absolute values of the indirect paths:

$$L(\theta) = (\boldsymbol{s} - \boldsymbol{\sigma}(\boldsymbol{\theta}))^T(\boldsymbol{s} - \boldsymbol{\sigma}(\boldsymbol{\theta})) + \sum_{p \in P} |a_p| + \sum_{p \in P} |b_p|$$

where $\boldsymbol{s}$ and $\boldsymbol{\sigma}(\boldsymbol{\theta}))$ are the half-vectorized observed and implied covariance matrix elements, P is the total number of mediators, a_p is the regression path from the predictor to the p^{th} mediator, and b_p is the regression path from the p^{th} mediator to the outcome. Note that for simplicity, we have not included a multiplicative penalty hyperparameter, but this could be included in future implementations.

The true indirect effects ($a_p b_p$) and their estimates are shown in Figure 2.8. The penalization procedure correctly sets most mediation paths to 0, thus excluding their respective mediators from consideration. If we use this exclusion as a decision rule for variable selection, we obtain one false negative (M.10) and three false positives (M.32, M.52, and M.81), resulting in a respectable positive predictive value (PPV) of 75%.

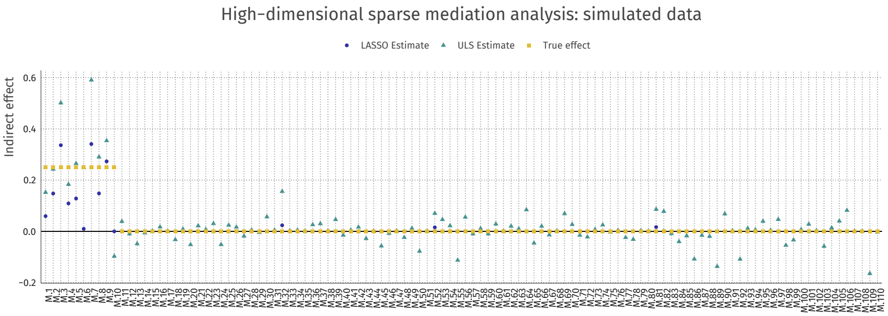

Figure 2.8 True and estimated indirect effects in a high-dimensional mediation analysis. The estimation methods are Unweighted Least Squares (ULS) and penalized ULS (LASSO). Regularized estimation correctly sets most parameters to 0 and shrinks the effect sizes overall.

Applying the same approaches, ULS and penalized ULS estimation, to 1000 preselected mediators from the real dataset (N=85) from the motivating example yields the result shown in Figure 2.9. The top-5 most relevant locations are labelled using their methylation site identifier. This type of penalization approach can be valuable in discovering potential mediation targets for future research, and although a similar procedures have been implemented using LASSO on the b_p paths (Y. T. Huang & Pan, 2015), LASSO on both a_p and b_p paths (Serang, Jacobucci, Brimhall, & Grimm, 2017), or a group LASSO penalty (Schaid & Sinnwell, 2020), it has never been implemented using ULS estimation.

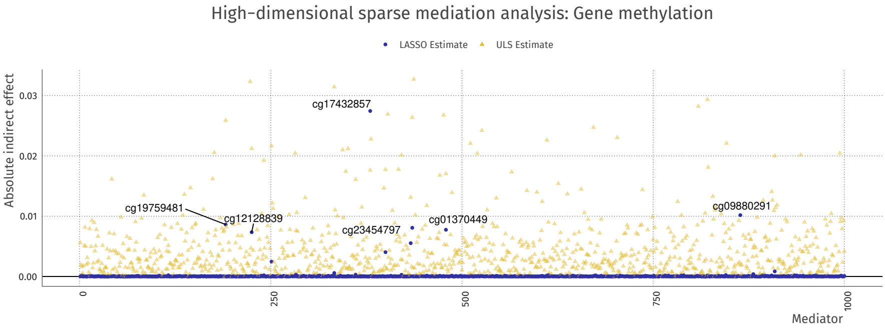

Figure 2.9 ULS and penalized ULS estimated absolute indirect effects in the Houtepen et al. (2016) dataset. Regularized estimation sets most parameters to 0 and shrinks the effect sizes overall, but for some mediators the effect sizes increase with penalization due to correlations among mediators. The top-5 strongest effect sizes are labelled, representing locations in the genome where mediation is strongest.

Together, the examples have shown that estimation and extension of SEM through computation graphs and the Adam optimizer is viable. In a single unified optimization framework, we have implemented several extensions, either suggested in previous literature or completely novel. As previously mentioned, the exact properties of the novel procedures introduced here should be further analyzed in future work for different types of models. This is not the goal of the present work, where these examples have served as an illustration of the flexibility and viability of the computation graph approach in principle.

2.5 Conclusion

Estimation of SEM becomes more challenging as latent variable models become larger and more complex. Traditionally, SEM optimizers have already suffered from nonconvergence and inadmissible solutions (e.g., Chen, Bollen, Paxton, Curran, & Kirby, 2001; Revilla & Saris, 2013), and with the increasing complexity of available datasets these problems are set to become more relevant. We argue that current estimation methods do not fulfil the needs of researchers applying SEM to novel situations in the future.

In this paper, we have introduced a new way of constructing objective functions for SEM by using computation graphs. When combined with a modern optimizer such as Adam, available in the software package `PyTorch`, this approach opens up new directions for SEM estimation. The flexibility of the computation graph lies in the ease with which the graph is edited, after which gradients are computed automatically and optimization can be performed without in-depth mathematical analysis. This holds even for non-convex objectives and objectives which are not continuously differentiable, such as the LASSO objective. We have shown that previously proposed improvements to SEM, such as LAD estimation (Siemsen & Bollen, 2007), follow naturally from this framework, and that our implementation is able to optimize these, yielding parameter estimates that behave according to expectations. In addition, we demonstrated the ease with which extensions can be investigated by implementing a fully Bayesian LASSO and performing high-dimensional variable selection with the ULS loss and a LASSO penalty, both novel penalization methods for SEM.

As the computation graph approach paves the way for a more flexible SEM, researchers can use it to develop theoretical SEM improvements. For example, future research can focus on how penalties may be used to improve the performance and interpretability of specific models (e.g., Jacobucci et al., 2018), or how different objective functions may be used to bring SEM to novel situations such as high-dimensional data (Grotzinger et al., 2019; van Kesteren & Oberski, 2019). A potential extension to SEM is the use of high-dimensional covariates to debias inferences in observational studies (Athey, Imbens, & Wager, 2018). The computation graph may aid in importing such procedures to SEM. An interesting historical note is that Cudeck, Klebe, and Henly (1993) have had similar reasons for creating a general SEM optimization program, where the full Hessian is numerically approximated for any covariance model and the solution is computed using Gauss-Newton iterations. The modern computational tools used here now make such generic SEM programs feasible.

Another topic for future research is exploratory model specification. For example, Brandmaier, von Oertzen, McArdle, and Lindenberger (2013) and Brandmaier, Prindle, McArdle, and Lindenberger (2016) use decision trees to find relevant covariates in SEM, and G. A. Marcoulides and Drezner (2001) use genetic algorithms to perform model specification search. Penalties provide a natural way to automatically set some parameters to 0, which is equivalent to specifying constraints in the model. A compelling example of this is the work by Pan et al. (2017), who used the Bayesian form of LASSO regularization as an alternative to post-hoc model modification in CFA. Their approach penalizes the residual covariance matrix of the indicators, leading to a more sparse selection of residual covariance parameters to be freed relative to the common modification index approach.

There is an opportunity for the SEM computation graph approach to be further developed to expand its range of applications. For example, through applying Adam as a stochastic gradient descent (SGD) optimizer it may be extended to perform full information maximum likelihood (FIML) estimation, batch-wise estimation, or SEM estimation with millions of observations. This will potentially enable SEM to be performed on completely novel types of data, such as streaming data, images, or sounds. Another improvement which may be imported from the deep learning literature is computation of approximate Bayesian posterior credible intervals for any objective function using stochastic gradient descent steps at the optimum (Mandt, Hoffman, & Blei, 2017). The deep learning optimization literature moves fast, and through the connections we have established in this paper the SEM literature could benefit from its pace.

Acknowledgments

We thank Rogier Kievit for his comments on an earlier version of this manuscript, and Maksim Rudnev for his helpful questions regarding `tensorsem`.

Chapter 3
Exploratory Mediation Analysis with Many Potential Mediators

van Kesteren, E. J., & Oberski, D. L. (2019). Exploratory mediation analysis with many potential mediators. *Structural Equation Modeling: A Multidisciplinary Journal, 26*(5), 710-723. doi:10.1080/10705511.2019.1588124

Social and behavioral scientists are increasingly employing technologies such as fMRI, smartphones, and gene sequencing, which yield 'high-dimensional' datasets with more columns than rows. There is increasing interest, but little substantive theory, in the role the variables in these data play in known processes.

This necessitates exploratory mediation analysis, for which structural equation modeling is the benchmark method. However, this method cannot perform mediation analysis with more variables than observations. One option is to run a series of univariate mediation models, which incorrectly assumes independence of the mediators. Another option is regularization, but the available implementations may lead to high false positive rates.

In this paper, we develop a hybrid approach which uses components of both filter and regularization: the 'Coordinate-wise Mediation Filter'. It performs filtering conditional on the other selected mediators. We show through simulation that it improves performance over existing methods. Finally, we provide an empirical example, showing how our method may be used for epigenetic research.

3.1 Introduction

Social and behavioral scientists are increasingly employing technologies such as fMRI, smartphones, and gene sequencing, which yield 'high-dimensional' datasets with more variables than observations. These high-dimensional data are often intended to answer questions such as *"which areas of our brain are relevant for pain perception?"* (Atlas, Lindquist, Bolger, & Wager, 2014) and

Author contributions: EJK & DLO developed the idea. EJK created the software, implemented the experiments, and wrote the manuscript. DLO provided extensive feedback on all components.

"which genes mediate the effect of trauma on stress reactivity?" (Houtepen et al., 2016). These are questions regarding exploratory mediation analysis (EMA).

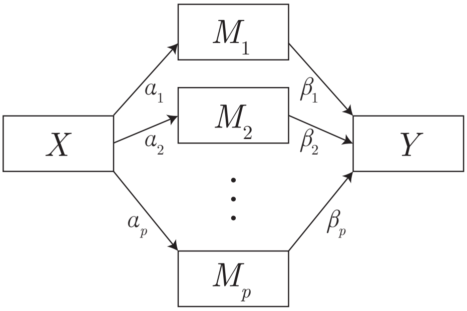

Figure 3.1 Exploratory mediation analysis with a set of p potential mediators $\boldsymbol{M}$. For clarity, we omitted the $P(P+1)/2$ parameters belonging to the residuals of $\boldsymbol{M}$ and their covariances, as well as the residual variance of Y.

Structural equation modeling (SEM) is the preferred method for mediation analysis with multiple mediators (Preacher & Hayes, 2008; Vanderweele & Vansteelandt, 2014). With this method, it is possible to determine to what extent specific M variables mediate the $X \to Y$ effect conditional on the presence of other mediators in the model. However, this method fails when the data is *high-dimensional*, when the variables under investigation outnumber the samples N. In this situation, the observed covariance matrix is rank-deficient, leading to linear dependence in the observed moments and, for the full mediation model, nonconvergence.

Several alternative methods for EMA have been proposed to deal with this issue. One option mentioned by Preacher and Hayes (2008) is to select relevant mediators from a series of univariate $X \to M \to Y$ mediation models (e.g., Boca, Sinha, Cross, Moore, & Sampson, 2014; Liu et al., 2013). We call this the "filter" method, following the taxonomy of Guyon and Elisseeff (2003). Its main advantages are that it is simple to explain and run, requiring only P univariate path models. On the other hand, the filter method introduces bias through model misspecification: it takes into account only the *marginal* relationships of M with X and Y. A pitfall of this is that a variable useless by itself can be useful together with others (Guyon & Elisseeff, 2003). In other words, a certain mediator may be marginally irrelevant, but relevant conditional on another set of mediators.

Recently, another multivariate method was introduced by Serang et al. (2017). Their proposal was to perform EMA through regularized estimation of the full structural equation model: "XMed". This method automatically shrinks small regression paths to 0, leading to a selection of potential mediators: mediators are variables for which both the $X \rightarrow M$ path and the $M \rightarrow Y$ path are nonzero after regularization. With this method, it is possible to detect mediators which are only relevant conditionally, while regularization resolves the identification issues of default SEM (Hastie et al., 2015). The disadvantage is that this method finds paths with a large effect rather than the desired subset of mediators: the regularization in XMed shrinks small β paths to 0, irrespective of the value of their associated α paths – shrinkage is performed on all paths equally. This leads to inflated false positive rates as reported by Serang et al. (2017) and Jacobucci et al. (2018). In summary, regularization methods do perform conditional estimation, but they select paths rather than mediators.

In this paper, we propose a hybrid approach to EMA which we call the "Coordinate-wise Mediation Filter" (CMF). This method combines advantages from both the filter and regularization methods: (a) it converges in case of high-dimensional data, (b) it takes into account mediator correlations, leading to conditional selection of mediators, and (c) it selects based on mediation, not paths. CMF performs univariate filtering *conditional* on the other selected mediators by using an algorithm from regularized regression: cyclical coordinate descent on residuals (Breheny & Huang, 2011; Friedman et al., 2010).

The remainder of the article is structured as follows: first, we provide relevant background on exploratory mediation analysis. Then, we outline the Coordinate-wise Mediation Filter as a hybrid method for mediator subset selection. Following this, we show through simulation where each of the discussed methods performs as well as SEM. In addition, we assess the performance of CMF relative to the other available methods in a high-dimensional simulation. Lastly, the CMF procedure is illustrated by applying it to the epigenetic process of trauma and stress reactivity.

3.1.1 Exploratory mediation analysis

The fundamental goal of mediation analysis is to determine the process by which a variable X influences another variable Y (MacKinnon, Lockwood, & Williams, 2004). Exploratory mediation analysis (EMA) in particular is used to explore a dataset for potential mediating variables (MacKinnon, 2008). In other words, EMA pertains to determining among multiple potential mediators which subset is most relevant. Through EMA, researchers can build theory and select variables of interest for further research into the process under investigation.

An example application of EMA is the research by Ammerman et al. (2018), who investigated how childhood maltreatment leads to suicidal behaviour. They defined 46 potential mediators, including psychological counseling, closeness to parents, and self-esteem. The authors did not test a fully specified mediation model about the precise relations of each of these variables to childhood maltreatment and suicidal behaviour. Instead, this study was exploratory, identifying which variables were the most relevant targets for future research. Indeed, the authors conclude that the study “highlights factors that may be potential targets for risk assessment and for treatment among adolescents with a history of childhood maltreatment”.

3.1.1.1 Univariate mediation analysis and the filter method

A common framework for *univariate* mediation analysis is a system of regression equations (Equation (3.1); MacKinnon et al., 2004). The system is displayed graphically in Figure 3.2. In the present paper, we consider only the case where the data from X, M, and Y are continuous and their relations are linear. For nonlinear discrete extensions to mediation analysis, see Hayes and Preacher (2010) and Hayes and Preacher (2014), respectively. For further details, refer to the reviews by MacKinnon, Fairchild, and Fritz (2007) and Preacher (2015).

$$\begin{aligned} M &= \mu_M + \alpha X + e_M \\ Y &= \mu_Y + \tau X + \beta M + e_Y \end{aligned} \tag{3.1}$$

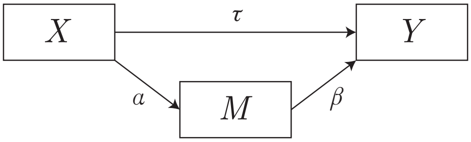

Figure 3.2 Graphical representation of the system of Equation (3.1). For clarity, the residuals are not shown.

Under the standard assumptions of linear SEM, the parameter estimates of this system may be used to determine whether M is a mediator — a dichotomous decision. There are several ways to make this decision, usually based on a quantity of interest q and a measure of uncertainty (MacKinnon, Lockwood, Hoffman, West, & Sheets, 2002). For example, q may represent the size of the indirect effect through the product of its coefficients $q_{\text{prod}} = \alpha\beta$, and uncertainty measures for q_{prod} can be obtained using asymptotic standard error methods (e.g., Olkin & Finn, 1995; Sobel, 1986) or bootstrapping (Preacher & Hayes, 2008).

Combining the quantity of interest q with an uncertainty estimate and a specified alpha level yields a dichotomous decision criterion based on a p-value. We call this a *univariate decision function* $\mathcal{D}$: a function that maps the data of X, M, and Y to a binary decision of whether M should be considered a mediator (1) or not (0).

$$\mathcal{D} : (\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{y}) \mapsto \{0, 1\}$$

Note that any function that follows this specification can be considered a decision function, regardless of complexity. An example of higher complexity decision functions is given by VanderWeele (2015, p. 46), who states exposure-outcome confounding should by default be controlled for when testing for mediation. The decision function encodes the researcher's definition of mediation: a product of coefficients decision function with a p-value cutoff of 0.1 will lead to different results than an exposure-outcome controlled decision function with a stricter cutoff.

This decision function framework thus provides a convenient abstraction, highlighting a key advantage for mediation analysis methods: if the choice of decision functions is flexible, a method is adaptable to the specific needs of a researcher. If researchers want to follow the recommendation of VanderWeele (2015), they can do so by adding an XM interaction term into the decision function.

While these decision functions are univariate, EMA is an inherently multivariate procedure, requiring analysis of multiple indirect effects. To perform EMA, a researcher can apply their chosen decision function to each mediator separately, through P different mediation models as in Figure 3.2. This "filter method" will result in a subset of relevant mediators. However, the implicit assumption is that the M variables are independent of one another. In other words, the selected subset will not include mediators that are relevant only conditionally on another mediator.

3.1.1.2 Multivariate mediation analysis and XMed

To make mediation decisions multivariately, Preacher and Hayes (2008) recommend the SEM approach. In this approach, the quantities of interest $q_1, ..., q_P$ and their uncertainty are estimated directly from a multiple mediation model as in Figure 3.1. A decision can then be made for each individual M_p based on its multivariately estimated quantity q_p. Unlike the filter method, this approach estimates q_p conditional on the other $P-1$ quantities, so that marginally irrelevant true mediators may still be detected.

However, the SEM approach is unavailable in the case of high-dimensional data because SEM parameters are estimated from observed covariances. High dimensional data ($P > N$) leads to a $P \times P$ observed covariance matrix of at most rank N, meaning a linear dependence exists among elements. If dependent elements are mapped to separate parameters in the SEM model, an infinite number of solutions exist for the same log-likelihood, so there is no maximum likelihood solution. This is the case in the full mediation model. As an alternative intuitive explanation, it is possible to view the $M \rightarrow Y$ part of the mediation model as a high-dimensional multiple regression, where ordinary least squares (OLS) estimates are unavailable because the covariance matrix cannot be inverted (Hastie et al., 2015).

XMed (Serang et al., 2017) is an adjustment to the SEM method that not only allows for high-dimensional data, but it also automatically selects a subset of mediators without an explicit decision function. The estimation method for XMed is RegSEM Jacobucci et al. (2016), which applies regularization to a chosen subset of model parameters in a structural equation model. This shrinkage is determined by the hyperparameter λ along with the penalization function $P(\cdot)$ in the objective function of RegSEM:

$$F_{regsem} = F_{ML} + \lambda P(\cdot)$$

where $\cdot$ is a vector of parameters.

In XMed specifically, shrinkage is applied to the vectors of α ($x \rightarrow M$) and β ($M \rightarrow y$) parameters. Subset selection of the mediators occurs through the chosen regularization method; the penalty function $P(\cdot)$ is the LASSO penalty, the ℓ_1 norm of the chosen parameter vector: $P(\cdot) = \| \cdot \|_1$. Depending on the value of λ, The LASSO penalty shrinks the smallest of the chosen parameters to 0 during estimation. This immediately forms the decision rule: for potential mediator $\boldsymbol{M}_p$, if α_p or β_p equals 0, then the estimated indirect effect $\alpha\beta_p$ is 0, thus $\boldsymbol{M}_p$ is not considered to be a true mediator.

A well-known algorithm for computing the LASSO solution, which can also be applied in SEM, is coordinate-wise conditioning or coordinate descent: the conditional solution is well-known and easy to find, in SEM the maximum likelihood estimates, and the penalized solution is found by cyclically updating and soft-thresholding the conditional solution for each parameter in turn, until convergence (Hastie et al., 2015).

A sequential combination of the ideas of filtering and regularization was proposed by Zhang et al. (2016) in a three-step approach called HIMA. First, in the screening step the authors marginally filter irrelevant potential mediators based on the $M \rightarrow Y$ relations. Second, the remaining $M \rightarrow Y$ paths are estimated with regularization. Lastly, the test step performs the joint significance test as introduced by Baron and Kenny (1986) with Bonferroni correction on the remaining mediators.

The main disadvantage of these methods is that there is a pertinent difference between (a) penalized estimation of the paths and (b) finding mediators. For XMed, a relatively small α_p path will be shrunk to 0 before stronger α paths, regardless of the strength of its associated β_p path. This holds for HIMA too, since in the selection stage it considers only β paths. Thus, these methods do not target *mediators with strong indirect effects* $\alpha\beta$, but *intermediate variables with strong* α *or* β *paths.* Even though these methods do work conditionally, they make the implicit assumption that the mediators also have the strongest $X \rightarrow M$ and $M \rightarrow Y$ paths, which need not be so.

Rephrasing this in terms of decision functions, the regularization methods exclude variables which have a relatively weak covariance with X or Y. However, this decision criterion only partially captures theoretically plausible mediators: true mediators may exist for which the covariance with X or Y is relatively weak, but the indirect effect $\alpha\beta$ is relatively strong. The regularization methods will thus underperform in the presence of “noise” variables which are not mediators, but which strongly covary with either X or Y. We illustrate this in the simulation section.

In conclusion, to perform EMA, (a) the SEM method is optimal but unavailable for high-dimensional data, (b) the filter method is simple and flexible but does not select mediators conditionally, and (c) regularization methods do proper conditioning but are estimating paths rather than selecting mediators.

3.2 Coordinate-wise Mediation Filter

We propose a hybrid method, the Coordinate-wise Mediation Filter (CMF), which contains both theory-driven decision functions and conditional estimation of the quantity of interest. Like the filter method, CMF applies a decision function to each of the mediators, but it performs this task conditional on the set of currently selected mediators. The procedure is similar to cyclical coordinate descent, the algorithm underlying regularization procedures in various software implementations – but differs in that mediation rather than separate regression paths are explicitly identified as the target. A key component of this algorithm is the use of residuals to remove dependency among the coordinates (Hastie et al., 2015). CMF generalizes this idea to mediator selection with arbitrary objective functions.

The CMF implementation consists of two components: an inner algorithm, which handles feature selection using the decision function $\mathcal{D}$ through coordinate descent, and an outer algorithm, which performs random starts, feature subsampling, and subsequent aggregation. The combined procedure can be characterized as a stochastic coordinate descent algorithm. The following two sections give a detailed outline of the inner and outer algorithm.

3.2.1 Inner algorithm

First, we initialize a vector of length P which contains the current mediator selection in the form of 0 and 1 values – the starting values. A *step* is then as follows: for each potential mediator M_p, create a data matrix $\boldsymbol{M}_*$, which contains all the mediators currently selected, excluding the variable M_p under consideration. Then, perform the decision function $\mathcal{D}$ on the parts of $\boldsymbol{x}$ and $\boldsymbol{y}$ orthogonal to (conditional on) this matrix. This conditioning is performed through calculating the *residuals* of $\boldsymbol{x}$ and $\boldsymbol{y}$ with respect to $\boldsymbol{M}_*$:

$$\begin{aligned}\boldsymbol{r}_x &= \boldsymbol{x} - \boldsymbol{M}_*(\boldsymbol{M}_*'\boldsymbol{M}_*)^{-1}\boldsymbol{M}_*'\boldsymbol{x}\\ \boldsymbol{r}_y &= \boldsymbol{y} - \boldsymbol{M}_*(\boldsymbol{M}_*'\boldsymbol{M}_*)^{-1}\boldsymbol{M}_*'\boldsymbol{y}\end{aligned}$$

The decision function is thus performed as $\mathcal{D}(\boldsymbol{r}_x, \boldsymbol{M}_p, \boldsymbol{r}_y)$, leading to a binary decision whether mediator p selected, conditional on $\boldsymbol{M}_*$.

The inner algorithm is run continuously, randomly ordering the choice of p in each iteration. It stops either when the mediator selection does not change from one step to the next, or when the prespecified maximum number of iterations is reached. The resulting program, shown in Algorithm 1, is a binary, randomized form of cyclical coordinate descent similar to those in Hastie et al. (2015). The randomization improves stability for very high-dimensional data (Nesterov, 2012). Richtárik and Takáč (2014) show that this method attains relatively fast convergence even with a billion variables in a sparse regression situation.

Algorithm 1 Inner CMF algorithm

```
scale($\boldsymbol{x}$); scale($\boldsymbol{M}$); scale($\boldsymbol{y}$)
P ← ncol($\boldsymbol{M}$)                                          ▷ number of mediators
decvec ← $0_1, 0_2, \dots, 0_P$                                   ▷ initialise 0/1 decision vector
repeat
    for p in 1:P do
        $\boldsymbol{M}_* \leftarrow \boldsymbol{M}$[, decvec & !p]                    ▷ selected mediators excluding $p$
        $\boldsymbol{r}_x \leftarrow \boldsymbol{x} - \boldsymbol{M}_*(\boldsymbol{M}'_*\boldsymbol{M}_*)^{-1}\boldsymbol{M}'_*\boldsymbol{x}$                        ▷ residual of x
        $\boldsymbol{r}_y \leftarrow \boldsymbol{y} - \boldsymbol{M}_*(\boldsymbol{M}'_*\boldsymbol{M}_*)^{-1}\boldsymbol{M}'_*\boldsymbol{y}$                        ▷ residual of y
        decvec[p] ← $\mathcal{D}(\boldsymbol{r}_x, \boldsymbol{M}$[, p]$, \boldsymbol{r}_y)$                  ▷ decision function
    end for
until decvec = decvec$_{prev}$                          ▷ convergence when decvec is stable
```

3.2.2 Outer algorithm

The value of the decision vector resulting from the inner algorithm depends to some extent on the starting values, due to the discrete nature of its coordinates. Therefore, the algorithm is embedded in an outer loop that performs multiple random starts. After aggregating the results from the different starts, the decision vector of length P is continuous: each element p in this vector signifies the proportion of times the potential mediator M_p was selected by the inner algorithm. These proportions, or *empirical selection probabilities*, naturally lead to a mediator ranking. This ranking can then again be dichotomized using a cutoff score.

The second essential part in the outer algorithm is *feature sampling*. With feature sampling, the inner algorithm will loop over only $\lceil\sqrt{P}\rceil$ potential mediators at each iteration. This procedure is similar to how random forest decorrelates its trees (Breiman, 2001a). Zhang, Zhao, Zhang, and Wei (2019) show in a sparse regression setting that feature sampling improves and stabilizes the performance of feature selection. Furthermore, there are links between feature sampling and shrinkage: for linear regression, considering only $\lceil\sqrt{P}\rceil$ variables during training is equivalent to ridge regression on the standardized predictors. This generalizes to more complex methods such as GLM (Wager et al., 2013). Feature sampling in the CMF algorithm thus takes on the crucial role of regularization.

The entire CMF procedure is implemented in the `R` package `cmfilter`, available from `https://github.com/vankesteren/cmfilter`. An example analysis with specific hyperparameters and cutoff score determination is described in the application section to this paper, with accompanying `R` code in the supplementary material.

The CMF method addresses the most important issues associated with both filter and regularization methods: it conditions on the other mediators while simultaneously being flexible to the choice of theoretically relevant decision functions. In the next section, we investigate the performance of CMF through simulation.

3.3 Simulations

This section is subdivided into two parts. The first part aims to show empirically the theoretical advantages and disadvantages of SEM, filter, XMed, HIMA, and CMF. We simulate specific conditions which are theoretically challenging for some but not all methods. The results from the first section are aimed at generating an understanding of the theoretical background in the present paper.

The second part is aimed at simulating real-world performance in a controlled high-dimensional situation. The results from this section indicate to what extent the CMF method outperforms its rival methods in practice, in addition to providing an anchor for the expected absolute level of performance in terms of false positives and true positives in such a situation.

All the simulations were run on R version 3.5.0 (R Core Team, 2018).

3.3.1 Theoretical conditions

The goal of this section is to illustrate when each method performs adequately and when it does not. Two situations are of particular interest: (a) suppression through correlation among mediators, and (b) noise in the α and β paths, overshadowing a potential mediator. Filter methods are likely to underperform in terms of power in the first case, as the effect of a mediator is dependent on another and marginally invisible. In the second case, the regularization methods are theorized to under-perform because the α and β paths are regularized independently whereas it is their combination that indicates mediation.

The data was controlled to behave according to the population, i.e., the data was transformed to exhibit the exact correlation matrix implied by the data-generating model. In each simulation, we show the power and false discovery rates of the three methods in 100 simulated datasets of 400-600 observations. The decision function under consideration for the filter, SEM, and CMF methods was the Sobel test (Sobel, 1986), one of the most common tests in the product of coefficients category (MacKinnon et al., 2002). For these tests, any variable with a p-value below .1 was considered to be a mediator. The SEM and filter methods were implemented using the `lavaan` package (Rosseel, 2012), and CMF was implemented using the accompanying `cmfilter` package. For XMed, the `regsem` package (Jacobucci et al., 2016) was used with cross-validation was to find the optimal penalty parameter, and any variables with nonzero α and β paths were considered mediators. HIMA was run according to its implementation in the R package `HIMA` (Zhang et al., 2016), again with a p-value of .1. Further details on the data generation and precise simulation conditions can be found in the the R code in the supplementary material.

3.3.1.1 Suppression

In the first illustration, the effect of the second β path is 0, but conditional on the first mediator this effect is nonzero. Its data-generating model is shown in Figure 3.3. The power to detect the second mediator thus indicates the robustness of each selection method to a full suppression effect.

$$\begin{aligned}\text{cov}(M_1, M_2) &= -0.44 + -0.4 \cdot 0.4 = -0.6\\ \text{cov}(M_2, Y) &= 0.48 + 0.8 \cdot \text{cov}(M_1, M_2) = 0\end{aligned}$$

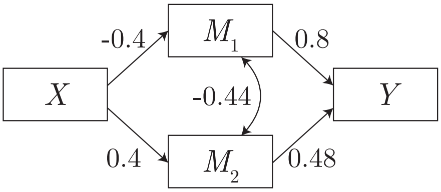

Figure 3.3 Data-generating model for the suppression simulation. Double-headed arrow indicates residual covariance.

The results are shown in Table 3.1, in the form of power to detect each mediator. As expected, the filter method fails to detect M_2 under the marginal suppression in this data, whereas the other methods do detect the suppressed mediator.

Table 3.1 Empirical power, calculated as the proportions of selection for each mediator in the 100 generated datasets.

Method	M_1	M_2
SEM	1	1
Filter	1	0
XMed	1	1
HIMA	1	1
CMF	1	1

3.3.1.2 Noise in the α paths

The second illustration considers noise in the form of variables related to X. In addition to the single mediator, 15 noise variables were generated; the α path was set to 0.8 for 3 of the variables, and 0.4 for the remaining 12. In addition, small residual correlations were induced in this set of variables to more closely resemble real-world patterns. The data-generating mechanism is shown in Figure 3.4.

This situation challenges XMed, which considers the α and β paths separately and is therefore theoretically more likely to select the strong paths rather than the mediating path, which has strength 0.3.

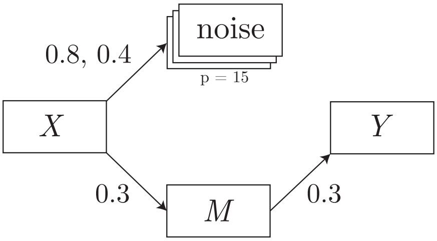

Figure 3.4 Data-generating model for the simulation of noise in the α paths.

The results are displayed in Table 3.2 in the form of rates of detection for each potential mediator. The SEM method performs optimally, as do the HIMA and CMF methods. The filter and XMed methods do not perform as well as these, having relatively strong false positive rates and lower power, respectively.

Table 3.2 Selection rates of each mediator in 100 simulated datasets where the noise variables (2-16) have a nonzero relation with the X variable. M is the true mediator, dot indicates 0.

Method	M	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SEM	99	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Filter	100	.	.	100	.	.	.	.	.	100	.	.	.	76	.	.
XMed	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
HIMA	100	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
CMF	100	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

3.3.1.3 Noise in the β paths

Like the second illustration, the third adds 15 noise variables alongside the true mediator. This time, the noise variables are related to the outcome variable Y. The data-generating mechanism is shown in Figure 3.5.

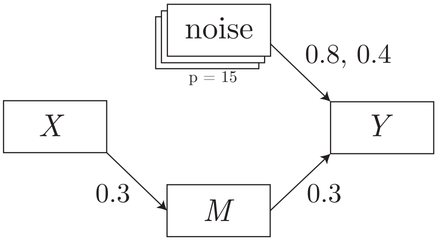

Figure 3.5 Data-generating model for the simulation of noise in the β paths.

The results can be found in Table 3.3. The HIMA method, which in the previous simulations performed as well as the benchmark SEM method, fails to detect the mediator in any of the 100 iterations. The other methods attain a perfect score.

Table 3.3 Selection rates of each mediator in 100 simulated datasets where the noise variables (2-16) have a nonzero relation with the Y variable. M is the true mediator, dot indicates 0.

Method	M	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SEM	99	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Filter	100	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
XMed	92	5	5	6	6	3	4	2	3	4	2	3	4	5	6	3
HIMA	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
CMF	100	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

3.3.1.4 Suppression and noise

The last illustration combines the above simulations into a single data-generating mechanism, where both suppression and noise are present, as shown in Figure 3.6.

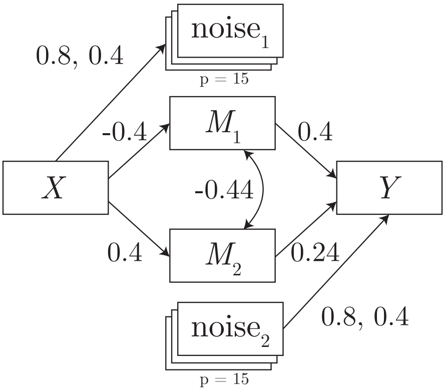

Figure 3.6 Data-generating model for the simulation of suppression with noise in the α and β paths.

The results of this combined simulation, displayed in Table 3.4 show again that CMF performs at benchmark level. An interesting quantity for the imperfect methods is the positive predictive value (PPV): the probability that a mediator selected by a method truly mediates the effect of X on Y. For filter and XMed methods, the PPV is lowered through either a relatively low true positive rate (power) or a high false positive rate (type-I error).

Table 3.4 True positive rates, false positive rates, and positive predictive values (PPV) of the combined suppression and noise simulation. The PPV indicates the probability that a mediator selected by the method is a true mediator.

Method	Power M_1	Power M_2	FPR	PPV
SEM	0.99	0.99	0.000	1.00
Filter	1.00	0.00	0.000	1.00
XMed	0.88	0.87	0.097	0.37
HIMA	1.00	0.00	0.000	1.00
CMF	1.00	1.00	0.000	1.00

3.3.1.5 Interim conclusion

While the considered data-generating mechanisms are very specific, the differences in performance between the methods can be exacerbated and diminished by altering the parameter values while preserving the structure. Overall, CMF is the only method that performs as well as the baseline in all of these data-generating mechanisms. Together, they show that this method is robust to boundary cases where other methods may fail. This is a valuable property of a mediator selection method, because these situations may occur simultaneously, with no way to test them in real-world datasets. In the next part, we explore how well the CMF method performs in high-dimensional circumstances, where the baseline optimal SEM method cannot work.

3.3.2 High-dimensional mediation simulation

In this section, we compare the performance of the available EMA methods in a simplified high-dimensional situation. Due to the wide nature of the dataset ($p = 1000$), the benchmark default SEM method is unavailable.

3.3.2.1 Simulation setup

Following one of the high-dimensional simulation conditions of Zhang et al. (2016), the dataset consists of 100 samples and 1000 potential mediators. These mediators are generated in four uncorrelated blocks: one block with true mediators (M), one with noise variables related to X (A), one noise block covarying with Y (B), and one large "white noise" block without any covariance (I). The general structure can be found in Figure 3.7. For each of the simulations, this structure was created as a sparse block matrix using the `Matrix` package (Bates & Maechler, 2017), after which multivariate normal data was generated using the `sparseMVN` package (Braun, 2018). Specific data generation and simulation `R` code can be found in the supplementary materials.

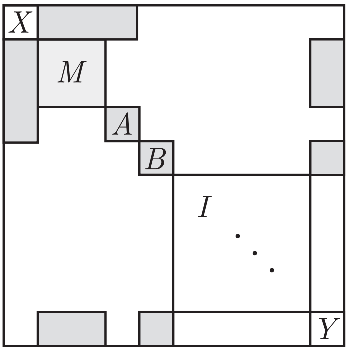

Figure 3.7 General covariance structure for the high-dimensional performance simulation. In the white sections of the matrix, there is no covariance. The true mediator block M is related to both X and Y, whereas the correlating noise blocks are related to either X (block A) or Y (block B). The largest block is the identity matrix block I, which generates only unrelated noise variables.

Note that unlike the illustrative simulations, these data favor the filter method: there is no suppression or excessive interdependence of potential mediators. Therefore, the filter method is the benchmark in this simulation. The XMed method was omitted from this simulation because it requires estimation of the full SEM model before regularizing: it would need to be adjusted to work with high-dimensional data.

3.3.2.2 Results

The results are displayed in Table 3.5. The CMF method has the highest true positive rate, and a medium false positive rate, leading to a similar positive predictive value (PPV) to the filter method. In other words, the mediators selected by CMF are as likely to be true mediators as those selected by the benchmark filter method. As true positive rates and false positive rates can be adjusted by the choice of alpha level, we conclude that the CMF method also performs at benchmark level in this high-dimensional situation.

Table 3.5 True positive rates, false positive rates, and positive predictive values for the high-dimensional data simulation. Note that XMed failed to run as-is for the simulated datasets, as it required running the full SEM model before regularizing.

	Power	Type I Error	PPV
CMF	0.2648	0.00258	0.5068
Filter	0.2412	0.00235	0.5124
HIMA	0.0686	0.00941	0.0323

3.4 Application to epigenetic data

In this section, we show how the CMF method can be used for exploratory mediation analysis in a real-world setting. Aside from the results shown here, the full `R` syntax is available in the supplementary material.

Houtepen et al. (2016) researched which locations in the genome are likely to mediate the relation between childhood trauma and stress reactivity later in life. In order to identify the genomic locations, they measured methylation at CpG sites using array based technology. In a discovery sample, they found a location of interest which they subsequently researched further and related to functional changes in the human prefrontal cortex.

Here, we re-analyze the original discovery sample dataset to investigate whether CMF yields different potentially relevant locations compared to the correlational filter analysis of the original authors.

3.4.0.1 Dataset and preprocessing

The dataset of the discovery sample was obtained from ArrayExpress, the data repository of the European Bioinformatics Institute: `https://www.ebi.ac.uk/`

`arrayexpress/experiments/E-GEOD-77445`. The sample consists of 85 healthy individuals. The X variable is score on a childhood trauma questionnaire and the Y variable is the increase in cortisol after a stress test defined as increase in the area under the curve (iAUC). The 385 884 potential mediators M were taken from the analysis of DNA methylation in the blood, with default preprocessing. From the available respondent characteristics, age and sex were considered to be confounders. For full details of the dataset, see (Houtepen et al., 2016).

Before analysis, X, Y, and M were residualized with respect to their intercept, age, and sex. Since the number of M variables was so large, the last preprocessing step was a straightforward univariate filter. For this, the top 1000 potential mediators in terms of their absolute product of correlations with X and Y were retained. For more details, see the preprocessing `R` code in the supplementary materials.

3.4.0.2 Analysis and Results

The CMF algorithm was performed using the centered X and Y and the 1000 potential mediators M. The Sobel test with a p-value of 0.1 was used as the decision function $\mathcal{D}$ and 10 000 iterations with random starts were run to ensure stability of the results. After inspecting the scree plot of the selection rates, the cutoff for selection was set to 0.075. The resulting selection rates and selected `cg` locations in the genome are shown in Figure 3.8.

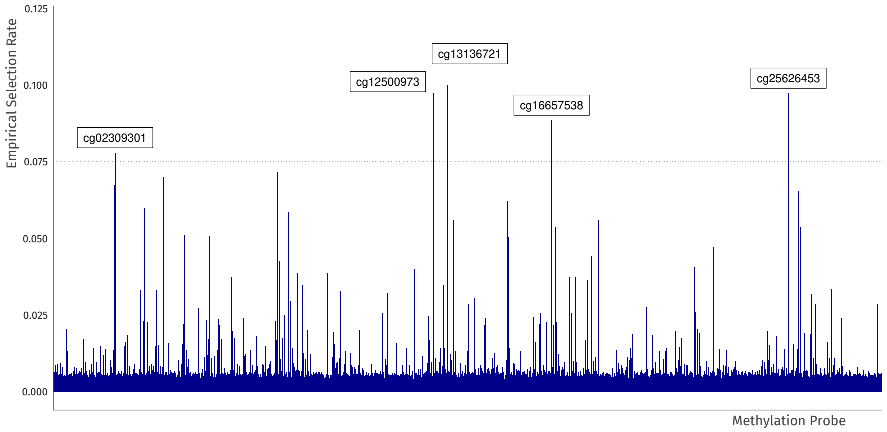

Figure 3.8 Selection rates of the potential mediators in the methylation dataset.

These locations were annotated using the BioConductor package `FDb.InfiniumMethylation.hg18` (Triche, 2014) to find the nearest protein-coding gene. The shortened descriptions were summarized from the GeneCards database (Safran et al., 2002). The result is shown in Table 3.6.

Table 3.6 Annotation of the selected mediators from the CMF algorithm.

Probe	Gene	Description
cg16657538	ZSCAN30	Involved in transcriptional regulation
cg25626453	PRRC2A	Associated with the age-at-onset of diabetes
cg02309301	ARGLU1	Associated with sexual development
cg13136721	RPTOR	Involved in regulation of cell growth and survival
cg12500973	HNRNPF	Involved in regulation of mRNA

Inspecting and comparing these results more closely, two of the locations identified by CMF have been previously associated with development throughout the lifespan: PRRC2A and ARGLU1. These two locations are also in the top 10 of the lists generated by HIMA and filter. In addition, the RPTOR gene has been associated with cell growth and survival – development on a cellular level. Relative to other sites, this last location does not have a strong correlation with either childhood trauma ($r = 0.186$) or stress reactivity ($r = 0.233$), but due to its conditional indirect effect it is deemed relevant by both CMF and HIMA. The ZSCAN30 gene has a small marginal correlation with stress reactivity ($r = 0.096$) which lowers its rank for both the filter and HIMA methods. However, due to its strong correlation with childhood trauma ($r = 0.347$) and its conditional relevance this site is still high on the list for CMA.

In conclusion, CMA has overlap with other methods but can identify relevant locations that other methods may miss. Further research using replication samples could focus on exploring whether and how methylation at these locations may alter stress reactivity after childhood trauma.

3.5 Discussion

Structural equation modeling, the benchmark method for exploratory mediation analysis, is unavailable in the case of high-dimensional data. Several alternative methods exist, but in the current paper we have shown through simulations that these underperform in situations with specific dependence among mediators, noise variables related to either X or Y, or a combination thereof. Taking these situations into account, we have introduced CMF, a hybrid algorithmic method to identify from a set of potential mediators the most likely true mediators.

CMF improves upon the existing methods by combining the estimation method from regularized regression with the theory-based decision functions from classic mediation analysis. It extends EMA with theoretically relevant decision functions to the high-dimensional case. As a full package including a software implementation, it is flexible to the choice of decision function, robust in the tested situations, and it scales to multiple processor cores.

Besides its role as a novel method for EMA, CMF contributes several ideas to the statistical literature. It shows that the use of cyclically calculated residuals is applicable beyond regression into the territory of structural equation modeling. In addition, its performance is greatly improved by feature subsampling, which has regularizing effects on the estimated parameters and thus on the mediator selections. CMF is an example of how combining a deterministic algorithm with a stochastic outer component can lead to adequate performance.

One result of the approach taken in this paper is that there is no formal proof of convergence, and the algorithm may take a long time to stabilize. In addition, the complications introduced in the outer loop make determination of the cutoff for selection nontrivial. In general, the algorithm will output a top-N vector of most selected mediators, and potential options for deciding which cutoff to take are visual inspection of the scree plot or a form of parallel analysis (Horn, 1965). In addition, the error rates (type I and type II errors) are not analytically defined and have a complex relation with the alpha level of the base decision function. This could be investigated empirically in the future.

For this work, we only considered direct feature selection on the set of M variables. Another solution is projecting the available features onto a low-dimensional space before or during estimation. Feature selection can then be performed in this space, leading to variable importance upon reprojection to the original space. Examples are PCA, PLS, or the directions of mediation method by Chén, Crainiceanu, Ogburn, Caffo, and Wager (2017). However, we chose to exclude these methods because they do not select mediators, but rather linear combinations of all mediators.

Our coordinate-wise mediation filter bears resemblance to a class of metaheuristic algorithms in the SEM literature for specification search (K. M. Marcoulides & Falk, 2018). These algorithms perform an exploratory search for the optimal model based on overall model fit, e.g., the BIC objective. CMF could be considered specification search where the objective is not overall model fit but mediation analysis: it is targeted towards determining whether a specific variable is relevant to a process rather than searching for the optimal model. In addition, CMF performs regularization required for high-dimensional data. In the future, other specification search strategies could be implemented for EMA, but they each need to be adjusted to incorporate both a specific mediation objective and regularization.

Future research should focus on embedding mediation analysis theory directly in penalization procedures for these datasets, either in a classical estimation setting (Zhao & Luo, 2016) or using Bayesian estimation with shrinkage priors (van Erp et al., 2019). More generally, enriching structural equation models beyond EMA with embedded feature selection mechanisms will enable social and behavioral scientists to develop and test theories on novel, high-dimensional datasets.

Funding details

This work was supported by the Netherlands Organization for Scientific Research (NWO) under Grant number 406.17.057.

Acknowledgments

We thank Marco Boks, Yves Rosseel, Katrijn van Deun, Milica Miočević, and Ayoub Bagheri for their helpful suggestions at various stages of this work.

Chapter 4

Exploratory Factor Analysis with Structured Residuals for Brain Network Data

Van Kesteren, E. J. & Kievit, R. A., (2020, In Press). Exploratory Factor Analysis with Structured Residuals for Brain Network Data. *Network Neuroscience*. doi:10.1162/netn_a_00162

Dimension reduction is widely used and often necessary to make network analyses and their interpretation tractable by reducing high dimensional data to a small number of underlying variables. Techniques such as Exploratory Factor Analysis (EFA) are used by neuroscientists to reduce measurements from a large number of brain regions to a tractable number of factors. However, dimension reduction often ignores relevant a priori knowledge about the structure of the data. For example, it is well established that the brain is highly symmetric. In this paper, we (a) show the adverse consequences of ignoring a priori structure in factor analysis, (b) propose a technique to accommodate structure in EFA using structured residuals (EFAST), and (c) apply this technique to three large and varied brain imaging network datasets, demonstrating the superior fit and interpretability of our approach. We provide an R software package to enable researchers to apply EFAST to other suitable datasets.

4.1 Introduction

Using modern imaging techniques, it is possible to investigate brain networks involving many regions, across different modalities such as grey matter volume, white matter tracts, and functional connectivity. To examine the relation of these networks with external variables of interest, it is often necessary to summarize them using a small number of dimensions – often called *factors* or *components*. These low-dimensional components representing the networks can be tracked over the lifespan (de Mooij, Henson, Waldorp, & Kievit, 2018; DuPre & Spreng, 2017), compared to behavioural measures (Colibazzi et al.,

Author contributions: EJK & RAK developed the idea, wrote and revised the manuscript. EJK created the software and implemented the experiments. RAK provided datasets.

2008), or related to phenotypes such as intelligence (Ferguson, Anderson, & Spreng, 2017). In the fields of statistics and mathematics, such methods for making analyses tractable and interpretable are collectively called *dimension reduction*.

Many popular dimension reduction techniques make use of *covariance*. For example, principal components analysis (PCA) can be estimated using only a decomposition of the covariance matrix. Covariance underlies many brain imaging and network analysis approaches, too: in analysis of structural connectivity, regions of grey matter volume or white matter tractography which covary across individuals may constitute connected networks (Alexander-Bloch, Giedd, & Bullmore, 2013; Mechelli, Friston, Frackowiak, & Price, 2005), and in resting-state fMRI analysis, regions which covary within an individual over time are considered to have a functional connection (Van Den Heuvel & Pol, 2010). Thus, dimension reduction on the basis of covariance matrices is directly applicable to the field of network neuroscience.

Exploratory factor analysis (EFA) is one such method for dimension reduction based on covariance. EFA models the observed covariance matrix of a set of P variables by assuming there are $M < P$ factors, which predict the values on the observed variables. Although other techniques such as PCA and Independent Component Analysis (ICA) are more common in neuroimaging analysis, EFA has been used since the early days of MRI (see McIntosh and Protzner, 2012 for a review and Machado, Gee, and Campos, 2004 for an early methodological investigation). For instance, Tien et al. (1996) performed an EFA on 60 controls and 44 schizophrenia patients for a selection of regions of interest, explicitly noting the high degree of left/right symmetry and a disruption of this symmetry in patients. Similarly early studies used EFA to model morphology (Stievenart et al., 1997) and width (Denenberg, Kertesz, & Cowell, 1991) of the corpus callosum. Some approaches combined SEM and PCA to model latent factors of grey matter structure in clinical populations (Yeh et al., 2010). These approaches have also been used to study typical population of children and adults (Colibazzi et al., 2008). More recently, EFA has been used to reduce individual differences in white matter microstructure in clinical populations (Herbert et al., 2018), as well as (extremely) large scale population studies (Cox et al., 2016). Hybrid approaches have combined exploratory and confirmatory factor analysis approaches (Baskin-Sommers, Neumann, Cope, & Kiehl, 2016; de Mooij et al., 2018) and used EFA in multimodal structural acquisitions (Mancini et al., 2016). EFA has also been used for functional imaging, including both fMRI (e.g., G. A. James et al., 2009) and EEG (Scharf & Nestler, 2018; Tucker & Roth, 1984). Most excitingly, recent work has used EFA to compare and contrast patterns of individual differences in brain structure at baseline with individual differences in developmental change over time, noting striking differences in dimensionality of change versus cross-sectional differences (Cox et al., 2020). Although the above is not intended to be a comprehensive review, it shows that EFA has been used widely in the imaging literature since early days.

Many related dimension reduction techniques exist beyond EFA, including Partial Least Squares (PLS), Independent Component Analysis (ICA), spectral decomposition, and many more beyond our current scope (see Roweis & Ghahramani, 1999; Sorzano, Vargas, & Montano, 2014). All of these techniques aim to approximate the observed data by means of a lower-dimensional representation. These techniques, although powerful, share a particular limitation, at least in their canonical implementations, namely that they cannot easily integrate prior knowledge of (additional) covariance structure present in the data. In other words, all observed covariation is modeled by the underlying factor structure.

This limitation is relevant in the context of structural and functional brain connectivity data because of *symmetry*: Much like other body parts, contralateral (left/right) brain regions are highly correlated due to developmental and genetic mechanisms which govern the gross morphology of the brain. Ignoring this prior information will adversely affect the dimension reduction step, leading to worse representation of the high-dimensional data by the extracted factors. Simple workarounds, such as averaging left and right into a single index per region, have other drawbacks: they throw away information, preclude the discovery of (predominantly) lateralized factors, and prevent the study of (a)symmetry as a topic of interest in and of itself.

Other classes of techniques, developed largely within psychometrics, can naturally accommodate additional covariance structure such as symmetry. These techniques started with multitrait - multimethod (MTMM) matrices (Campbell & Fiske, 1959) and later confirmatory factor analysis (CFA) with residual covariances (e.g., Kenny, 1976). MTMM is designed to extract factors when these factors are measured in different ways: when measuring personality through a self-report questionnaire and behaviour ratings, there are factors that explain correlation among items corresponding to a specific trait such as 'extraversion', and there are factors that explain additional correlation between items because they are gathered using the same methods (self-report and behavioural ratings). Thus MTMM techniques separate the correlation matrix into two distinct, summative parts: correlation due to the underlying traits (factors) of central interest, and correlation due residual structure in the measurements. However, MTMM requires a priori knowledge of the trait structure (e.g., the OCEAN model of personality) for estimation.

In this paper, we combine dimension reduction (e.g., across many brain regions) and prior structure knowledge (e.g., symmetry) by introducing EFA with structured residuals (EFAST). EFAST builds on standard implementations of EFA, CFA, and MTMM, but goes beyond these techniques by simultaneously allowing for exploration and the incorporation of residual structure. We show that EFAST outperforms EFA in empirically plausible scenarios, and that ignoring the problem of structured residuals in these scenarios adversely affects inferences.

This paper is structured as follows. First, we explain why using standard EFA or CFA for brain imaging data may lead to undesirable results, and we develop EFAST based on novel techniques from structural equation modeling (SEM). Then, we show that EFAST performs well in simulations, demonstrating superior performance compared to EFA in terms of factor recovery, factor covariance estimation, and the number of extracted factors when dealing with symmetry. Third, we illustrate EFAST in a large neuroimaging cohort (Cam-CAN; Shafto et al., 2014). We illustrate EFAST for three distinct datasets: Grey matter volume, white matter microstructure and within-subject fMRI functional connectivity. We show how EFAST outperforms EFA both conceptually and statistically in all three datasets, showing the generality of our technique. We conclude with an overview and suggestions for further research.

Accompanying this paper, we provide tools for researchers to use and expand upon with their own datasets. These tools take the form of (a) an R package called `efast` and a tutorial with example code (`https://github.com/vankesteren/efast`), and (b) synthetic data and code to reproduce the empirical examples and simulations (`https://github.com/vankesteren/efast_code`).

4.2 Factor analysis with structured residuals

In this section, we compare and contrast existing approaches in their ability to perform factor analysis in an exploratory way while at the same time accounting for residual structure. We discuss new developments in the field of exploratory structural equation modeling (ESEM) that enable simultaneous estimation of exploratory factors and structured residuals, after which we develop the EFAST model as an ESEM with a single exploratory block. We will use brain morphology data with bilateral symmetry as our working example throughout, although the principles here can be generalized to datasets with similar properties.

EFA, as implemented in software programs such as SPSS, R, and Mplus, models the observed correlation matrix through two summative components: the factor loading matrix $\mathbf{\Lambda}$, relating the predefined number M of factors to the observed variables, and a diagonal residual variance matrix $\mathbf{\Theta}$, signifying the variance in the observed variables unexplained by the factors. Using maximum likelihood, principal axis factoring, or least squares (Harman & Jones, 1966), the factor loadings and residual variances are estimated such that the implied correlation matrix $\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Theta}$ best approximates the observed correlation matrix $\boldsymbol{S}$. After estimation, the factor loadings are rotated to their final interpretable solution using objectives such as oblimin, varimax, or geomin (Bernaards & Jennrich, 2005).

We illustrate the challenge and the rationale behind our approach in Figure 4.1. The true correlation matrix is highlighted on the left, with correlations due to three factors shown as diagonal blocks. However, there is also considerable off-diagonal structure: the secondary diagonals show a symmetry pattern similar to that observed in real-world brain structure data (Taylor et al., 2017). The top panel of the figure shows that a traditional EFA approach will separate this data matrix into two components: (a) covariance due to the hypothesized factor structure and (b) the diagonal residual matrix. The key challenge is that EFA will attempt to approximate all the off-diagonal elements of the correlation matrix through the factors, even if this adversely affects the recovery of the true factor structure. Performing EFA with such a symmetry pattern may affect the factor solution in a variety of ways. For instance, in this toy example, the EFA model requires more than 12 factors to represent the data, instead of the three factors specified (see Appendix B.1). In other words, in such cases it is essential to incorporate the known residual structure via a set of additional assumptions.

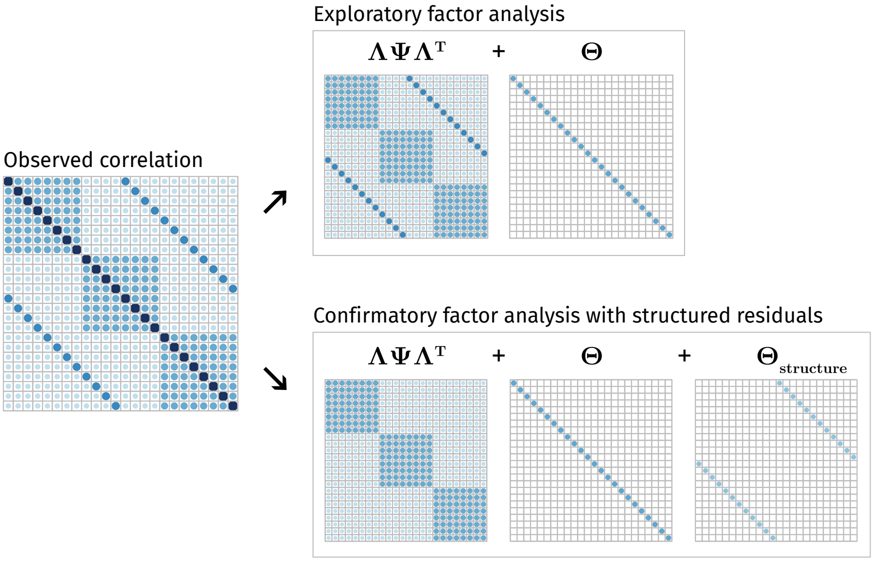
Exploratory factor analysis
$\mathbf{\Lambda\Psi\Lambda}^\mathrm{T}$
+
$\mathbf{\Theta}$
Observed correlation
Confirmatory factor analysis with structured residuals
$\mathbf{\Lambda\Psi\Lambda}^\mathrm{T}$
+
$\mathbf{\Theta}$
+
$\mathbf{\Theta}_{\text{structure}}$

Figure 4.1 Example observed correlation matrix and its associated decomposition according to EFA (top) and according to CFA (bottom) into a factor-implied correlation component ($\mathbf{\Lambda\Psi\Lambda}^T$), residual variance component $\mathbf{\Theta}$, and – in CFA with residual structure only – residual structure component.

As an alternative to EFA, we may implement a Confirmatory Factor Analysis (CFA) instead. In contrast to EFA, CFA imposes a priori constraints on the $\mathbf{\Lambda}$ matrix: some observed variables do not load on some factors. Moreover, in contrast to standard EFA approaches, residual structure can be easily implemented in CFA using standard SEM software such as `lavaan` (Rosseel, 2012). In other words, CFA would allow us to tackle the problem in Figure 1: We can allow for the residual structure known a priori to be present in the data. By allowing for the residual structure in the data, a CFA yields the implied matrices shown in the bottom panel of Figure 4.1, retrieving the correct factor loadings, residual variance, and residual structure. However, this is only possible because in this toy example we *know* the factor structure - In many empirical situations this is precisely what we wish to discover. In the absence of theory about the underlying factors, it is thus not possible to benefit from these features of CFA.

As such, we need an approach that can combine the strengths of EFA (estimating the factor structure in the absence of strong a priori theory) with those from CFA (the potential to allow for a priori residual structure). Here, we propose a hybrid between the two, which we call *exploratory factor analysis with structured residuals*, or EFAST. In order to implement and estimate these models, we make use of recent developments in the field of structural equation modeling (SEM). In the next section, we explain how these developments make EFAST estimation possible.

4.2.1 Exploratory SEM

Exploratory SEM (ESEM) is an extension to SEM which allows for blocks of exploratory factor analysis within the framework of confirmatory SEM (Asparouhov & Muthén, 2009; Brown, 2006; Guàrdia-Olmos, Peró-Cebollero, Benítez-Borrego, & Fox, 2009; Jöreskog, 1969; Marsh, Morin, Parker, & Kaur, 2014; Rosseel, 2019). ESEM is a two-step procedure. In the first step, a regular SEM model is estimated, where each of the EFA blocks have a diagonal latent covariance matrix $\mathbf{\Psi}$ and the $\mathbf{\Lambda}$ matrix of each block is of transposed echelon form, meaning all elements above the diagonal are constrained to 0. For a nine-variable, three-factor EFA block b the matrices would then be:

$$\mathbf{\Psi}_b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{\Lambda}_b = \begin{bmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21} & \lambda_{22} & 0 \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} \\ \lambda_{51} & \lambda_{52} & \lambda_{53} \\ \lambda_{61} & \lambda_{62} & \lambda_{63} \\ \lambda_{71} & \lambda_{72} & \lambda_{73} \\ \lambda_{81} & \lambda_{82} & \lambda_{83} \\ \lambda_{91} & \lambda_{92} & \lambda_{93} \end{bmatrix}$$

This means there are M_b^2 constraints for each EFA block b. This is the same number of constraints as conventional EFA (Asparouhov & Muthén, 2009). The second step in ESEM is to rotate the solution using a rotation matrix $\boldsymbol{H}$. Just as in regular EFA, this rotation matrix is constructed using objectives such as geomin or oblimin. In ESEM, the rotation affects the factor loadings and latent covariances of the EFA blocks, but also almost all other parameters in the model (Asparouhov and Muthén (2009) provide an overview of how rotation changes these parameter estimates). Despite these changes, a key property of ESEM is that different rotation solutions lead to the same overall model fit.

ESEM has long been available only in Mplus (Asparouhov & Muthén, 2009; Muthén & Muthén, 1998). More recently, it has become available in open sourced R packages psych (for specific models, Revelle, 2018) as well as lavaan (since version 0.6-4, Rosseel, 2019) – a comprehensive package for structural equation modeling. An example of a basic EFA model using lavaan syntax with 3 latent variables and 9 observed variables is the following:

```
efa("block1")*F1 =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
efa("block1")*F2 =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
efa("block1")*F3 =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
```

In effect, this model specifies three latent variables (F1, F2, and F3) which are each indicated by all 9 observed variables (x1 to x9). The efa("block1") part is a modifier for this model which imposes the constraints on $\boldsymbol{\Psi}$ and $\boldsymbol{\Lambda}$ mentioned above. For a more detailed explanation of the lavaan syntax, see Rosseel (2012). Figure 4.2 shows a comparison of the factor loadings obtained using conventional factor analysis (factanal() in R) and lavaan's efa() modifier. As shown, the solution obtained is exactly the same, with perfect correlation among the loadings for each of the factors.

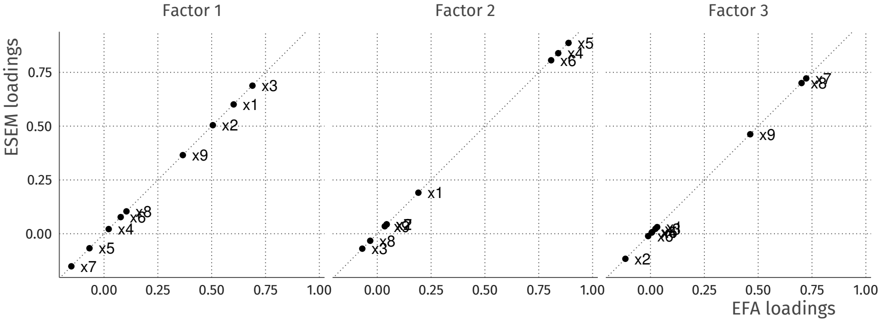

Figure 4.2 Exploratory factor analysis of 9 variables in the Holzinger and Swineford (1939) dataset. On the y-axis are the estimated factor loadings using the oblimin rotation functionality in lavaan version 0.6-4, and the loadings on the x-axis are derived from factanal with oblimin rotation from the GPArotation package (Bernaards & Jennrich, 2005). The loadings are all on the diagonal with a correlation of 1, meaning the solutions obtained from these different methods are equal.

With this tool as the basis for model estimation, the next section provides a detailed development of the construction of EFAST models.

4.2.2 EFAST models

We propose using EFA with corrections for contralateral covariance within the ESEM framework. The corrections we propose are the same as in MTMM models or CFA with residual covariance. In EFAST the method factors use CFA, and the remaining correlations are explained by EFA. Thus, unlike standard MTMM methods, EFAST contains *exploratory* factor analysis on the trait side, as the factor structure of the traits is unknown beforehand: the goal of the analysis is to extract an underlying low-dimensional set of features which explain the observed correlations as well as possible. For our running example of brain imaging data with contralateral symmetry, we consider each ROI a "method" factor, loading on only two regions. Note that in the context of brain imaging, Lövdén et al. (2013, Figure 1, model A) have had similar ideas, but their factor analysis operates on the level of left-right combined ROIs rather than individual ROIs.

The EFAST model has M exploratory factors in a single EFA block, and one method factor per homologous ROI pair, each with loadings constrained to 1 and its own variance estimated. The estimated variance of the method factors then represents the amount of covariance due to symmetry – over and above the covariance represented by the traits. In Figure 4.3, the model is displayed graphically for a simplified example with 6 ROIs in each hemisphere.

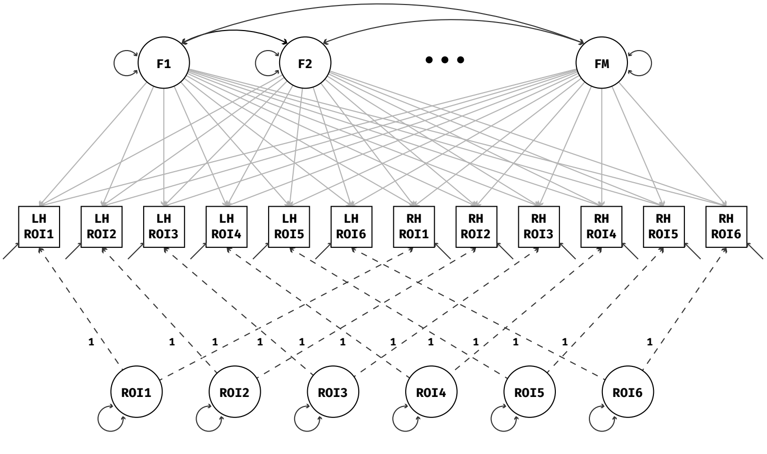

Figure 4.3 EFAST model with morphology of 6 regions of interest measured in the left hemisphere (LH) and right hemisphere (RH). The dashed lines indicate fixed loadings, the two-headed arrows indicate variance/covariance parameters. The method factors are constrained to be orthogonal, and the loadings of the M traits are estimated in an exploratory way.

An alternative parametrization for this model is also available. Specifically, we can use the correlations between the residuals of the observed variables instead of method factors with freely estimated variances. In the SEM framework, this would amount to moving the symmetry structure from the factor-explained matrix ($\mathbf{\Lambda\Psi\Lambda}^T$) to the residual covariance matrix $\mathbf{\Theta}$. This model is exactly equivalent, meaning the same correlation matrix decomposition, the same factor structure, and the same model fit will be obtained. However, we favour the method factor parametrization as it is closer to MTMM-style models, it is easier to extract potentially relevant metrics such as a 'lateralisation coefficient', and easier to extend to other data situations where multiple indicators load on each method factor.

To implement the EFAST model we use the package `lavaan`, which allows for easy scaling of the input data, different estimation methods, missing data handling through full information maximum likelihood, and more. Estimation of the model in Figure 4.3 can be done with a variety of methods. Here we use the default maximum likelihood estimation method as implemented in `lavaan`. Accompanying this paper, we are making available a convenient `R` package called `efast` that can fit EFAST models for datasets with residual structure due to symmetry. For more implementation details, the package and its documentation can be found at `https://github.com/vankesteren/efast`.

In the next section, we show how our implementation of EFAST compares to regular EFA in terms of factor loading estimation, factor covariance estimation, as well as the estimated number of factors.

4.3 Simulations

In this section, we use simulated data to examine different properties of EFAST models when compared to regular EFA in controlled conditions. The purpose of this simulation is not an exhaustive investigation, but rather a pragmatically focused study of data properties (neuro)scientists wishing to use this technique are likely to encounter. First, we explain how data were simulated to follow a specific correlation structure, approximating the general structure of empirical data such as that in the Cam-CAN study (see empirical examples section). Then, we investigate the effects of structured residuals on the extracted factors from EFA and EFAST: in several different conditions, we investigate how the estimation of factor loadings, the covariances between factors, and the number of factors changes with increasing symmetry.

4.3.1 Data generation

Data were generated following a controlled population correlation matrix $\mathbf{\Sigma}_{true}$. This matrix represents the true correlation between measurements of brain structure in 17 left-hemisphere and 17 right-hemisphere regions of interest. An example correlation matrix from our data-generating mechanism is shown in Figure 4.4.

$\mathbf{\Sigma}_{true}$ was constructed through the summation of three separate matrices, as in the lower panel of Figure 4.1:

1. The factor component $\mathbf{\Sigma}_{factor}$ is constructed as $\mathbf{\Lambda}\mathbf{\Psi}\mathbf{\Lambda}^T$, where the underlying factor covariance matrix $\mathbf{\Psi}$ can be either an identity matrix (orthogonal factors) or a matrix with nonzero off-diagonal elements (oblique factors). There are four true underlying factors in this simulation. One

of the factors is completely lateralized (top left, highlighted in green), meaning that it loads only on ROIs in the left hemisphere. An additional illustration of this left-hemisphere factor is shown in Figure 4.5. The remaining 3 factors have both left- and right-hemisphere indicators.

2. The structure component matrix is a matrix with all 0 elements except on the secondary diagonal, i.e., the diagonal elements of the bottom left and top right quadrant are nonzero. The values of these secondary diagonals determine the strength of the symmetry.

3. The residual variance component matrix is a diagonal matrix where the elements are chosen such that the diagonal of $\mathbf{\Sigma}_{true}$ is $\mathbf{1}$.

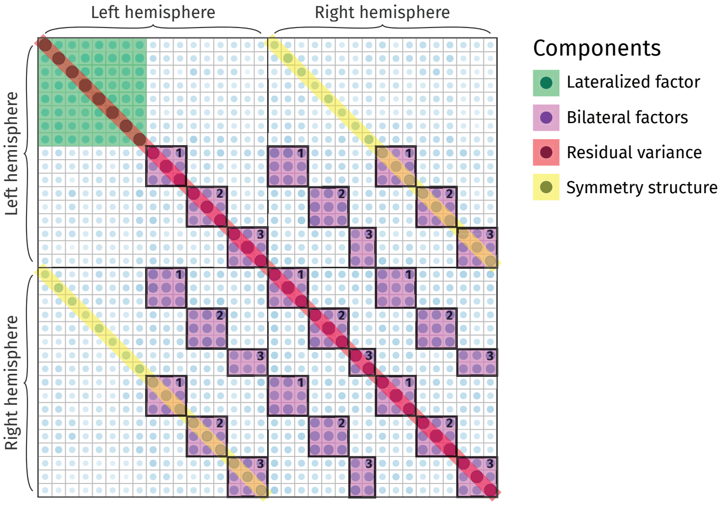

Figure 4.4 Example covariance matrix of the data-generating mechanism used in the simulations. This matrix results from simulated data of 650 brain images, with a factor loading of .595 for the lateralized factor, a loading of .7 for the remaining factors, a factor correlation of .5, and a symmetry correlation of .2. The first 17 variables indicate regions of interest (ROIs) in the left hemisphere, and the remaining variables indicate their contralateral homologues. Note the secondary diagonals, indicating contralateral symmetry, and the block of 8 variables in the top left resulting from the lateralized factor.

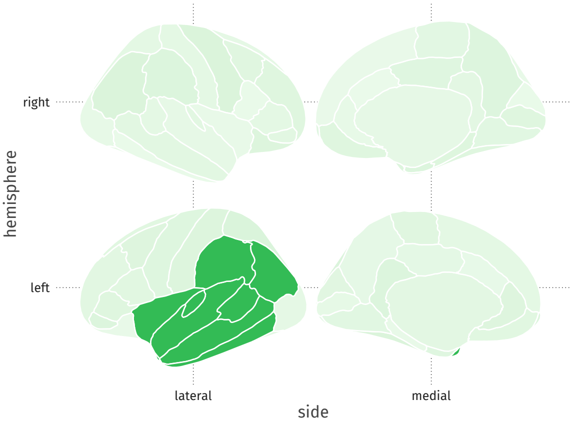

Figure 4.5 Example lateralized factor (the first factor in the simulation). Grey matter volume in 8 left-hemisphere regions of interest are predicted by the value on this factor.

For the following sections, data were generated with a sample size of 650, 130, or 65, a latent correlation of either 0 or 0.5, bilateral factor loadings of 0.5 or 0.7, lateral factor loadings of .425 or .595, and contralateral homology correlations of either 0 (pure EFA), 0.2 (minor symmetry), or 0.4 (major symmetry). These conditions were chosen to be plausible scenarios, similar to the observed data from our empirical examples. In each condition, 120 datasets were generated on which EFA and EFAST models with 4 factors were estimated. Thus, in each analysis the true number of factors is correctly specified before estimation. In the last simulation we then explore different criteria for the choice of number of factors in the case of contralateral symmetry.

4.3.2 Effect of structured residuals on factor loadings

In this section, we compare estimated factor loadings from EFA and EFAST to the true factor loadings from the simulation's data generating process. For each condition, 120 datasets were generated, to which both EFA and EFAST models were fit. The factor loading matrix for each model was then extracted, the columns reordered to best fit the true matrix, and the mean absolute error of the factor loadings per factor was calculated.

As hypothesized, allowing structured residuals affects how well the factor loadings are estimated from the datasets. Notably, as shown in 4.6 when performing regular EFA, the estimation error of the factor loadings increases when the symmetry becomes stronger, whereas the factor loading estimation error for the EFAST model remains at the level of regular EFA when there is no symmetry. Looking at the lateralized factor in particular, the adverse effect of omitting symmetry in dimension reduction becomes even stronger: in EFA, the lateralized factor becomes bilateral, leading to a larger error and an incorrect inference regarding the nature of the thus estimated factor. Although Figure 4.6 shows only the condition with a sample size of 650, factor loadings of 0.5, and factor covariance of 0.5, the pattern is similar for different sample sizes, different factor loading strengths and with no factor covariance (see Appendix B.2 and B.3).

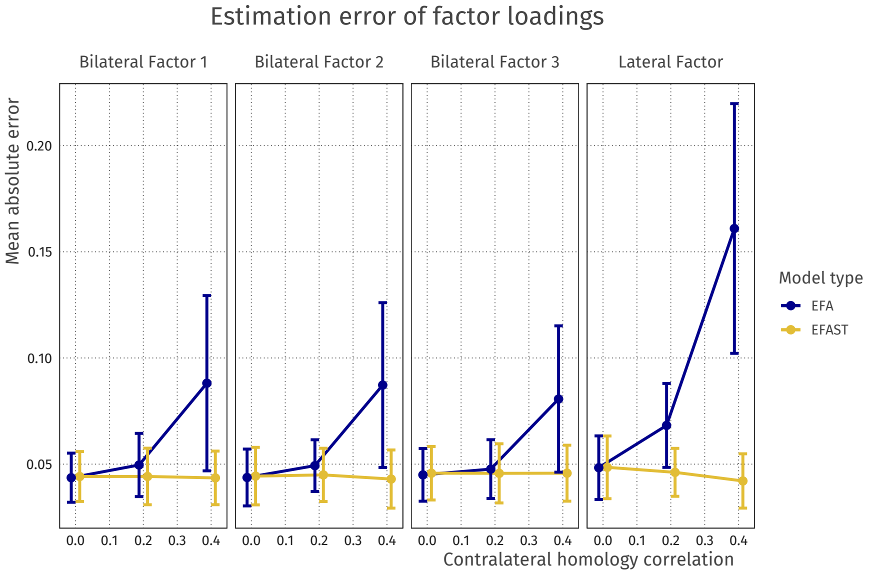

Figure 4.6 Mean absolute error for factor loadings of EFA versus EFAST models with increasing amounts of contralateral symmetry correlation. This plot comes from the condition where the sample size is 650, the covariance of the latent variables is 0.5, and the factor loadings are 0.5. The plot shows that for both bilateral and lateralised factors, EFA starts to exhibit more error as symmetry increases, more so for the lateral factor, whereas EFAST performance is nominal over these conditions. Error bars indicate 95% Wald-type confidence intervals.

In addition, sample size analysis shows that EFAST and EFA show moderate to high convergence rates for small (65) to moderate (130) sample sizes (see Appendix B.4) Although other drawbacks of smaller sample sizes remain (e.g., imprecise estimates, favouring of insufficiently complex models), this shows the feasibility, in principle, of using such analyses in commonly available sample sizes. To assess whether a particular combination of sample size, atlas dimensionality (i.e. number of regions) and strength of factor loadings is feasible for analysis using EFAST, we recommend a simulation approach. Software packages such as `lavaan` offer versatile tools to generate data under various specifications, allowing researchers to see whether a particular analysis is in principle feasible under certain idealized conditions before proceeding with real data.

Results from this section suggest that for the purpose of factor loading estimation, EFA and EFAST perform equally well in the case where a model without residual structure is the true underlying model, but EFAST outperforms EFA when residual structure in the observed data becomes stronger. In other words, implementing EFAST in the absence of residual structure does not seem to have negative consequences for estimation error, suggesting it may also be a useful default if a specific residual structure is thought, but not known, to exist. This is in line with Cole, Ciesla, and Steiger (2007), who argue that in many situations including correlated residuals does not have adverse effects, but omitting them does.

4.3.3 Effect of structured residuals on factor covariances

Here, we compare how well EFA and EFAST retrieve the true factor covariance values. For both methods, we used geomin rotation with an epsilon value of 0.01 as implemented in `lavaan` 0.6.4 (Rosseel, 2019). The matrix product of the obtained rotation matrix $\boldsymbol{H}$ then represents the estimated factor covariance structure of the EFA factors: $\boldsymbol{\Psi}_{EFA} = \boldsymbol{H}^T\boldsymbol{H}$ (Asparouhov & Muthén, 2009, eq. 22).

The mean of the off-diagonal elements of the $\boldsymbol{\Psi}_{EFA}$ matrix were then compared to the true value of 0.5 for increasing symmetry strength. The results are shown graphically in Figure 4.7. Here, it can be seen that with this rotation method the latent covariance is underestimated in all cases, although less so with stronger factor loadings. Furthermore, EFA performs worse as the symmetry increases, whereas the performance of EFAST remains stable regardless of the degree of contralateral homology, again suggesting no adverse effects to implementing EFAST in the absence of contralateral correlations. In the case of uncorrelated factors (not shown), the two methods perform similarly well.

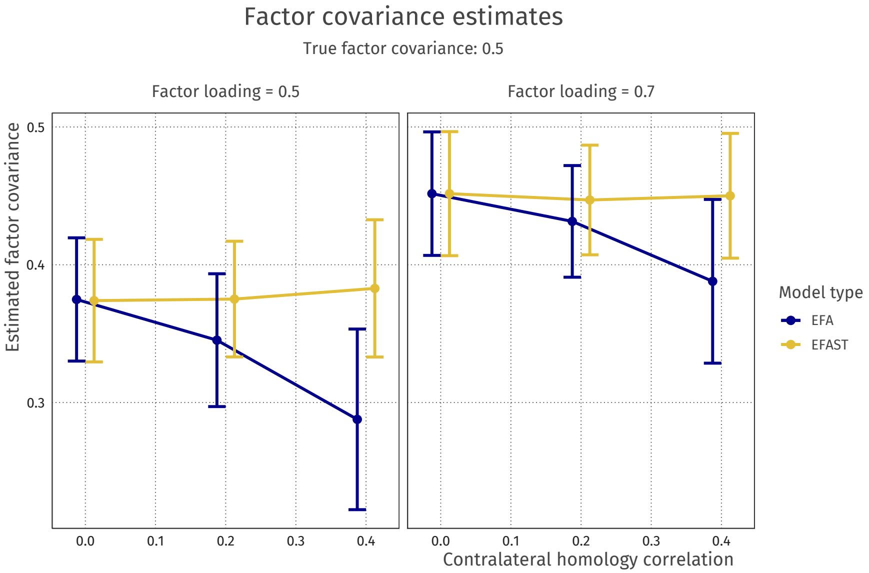

Figure 4.7 Latent covariance estimates for different levels of contralateral homology correlation. The true underlying latent covariance is 0.5; both methods underestimate the latent covariance but EFA becomes more biased as symmetry increases. Error bars indicate 95% Wald-type confidence intervals.

The results from this section shows that in addition to better factor recovery for EFAST, the recovery of factor covariance is also improved relative to EFA. Again, even when the data-generating mechanism does not contain symmetry, EFAST performs at least at the level of the EFA model. Note that in this case the overall model fit in terms of AIC and BIC is slightly better for the EFA model, as it has fewer parameters: for factor loadings of .5 and no symmetry, the mean AIC is 60148 (EFA) versus 60164 (EFAST), and BIC is 60882 (EFA) versus 60974 (EFAST). This, together with the comparable convergence rates for most conditions (Fig S4), suggests that it is viable to use EFAST as a ‘keep it maximal’ strategy (Barr, Levy, Scheepers, & Tily, 2013), where EFAST can be used initially with no drawbacks, but one can use model evidence to favour classical EFA instead.

4.3.4 Effect of structured residuals on model fit

In the above analyses, the number of factors was specified correctly for each model estimation (using either EFA or EFAST). However, in empirical applications the number of factors will rarely be known beforehand, so has to be decided on the basis of some criterion. A common approach to extracting the number of factors, aside from computationally expensive strategies such as parallel analysis (Horn, 1965), is model comparison through information criteria such as the AIC or BIC (e.g. (Vrieze, 2012). In this procedure, models with increasing numbers of factors are estimated, and the best fitting model in terms of these criteria is chosen.

In this simulation, we generated 100 datasets as in Figure 4.4 – i.e., strong loadings and medium symmetry – and we fit EFA and EFAST models with 2 to 10 factors. Across these solutions we then compute the information criteria of interest. Here we choose the two most common information criteria (the AIC and BIC) as well as the sample-size adjusted BIC (SSABIC), as this is the default in the ESEM function of the `psych` package (Revelle, 2018). The results of this procedure are shown in Figure 4.8. Each point indicates a fitted model. The means of the information criteria are indicated by the solid lines.

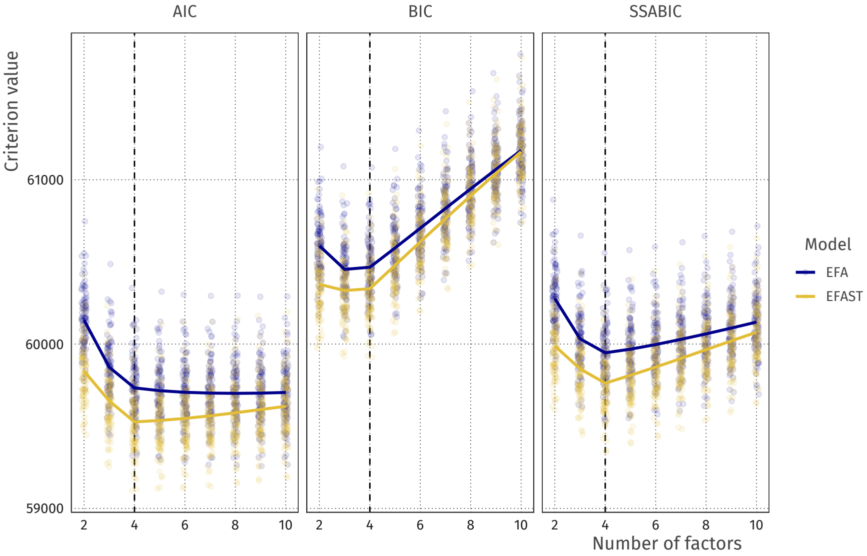

Figure 4.8 AIC and BIC values for increasing number of factors with EFA and EFAST models. Lines indicate expectations: the vertices are at the mean values for these criteria. The true number of factors is 4 (dashed vertical line).

The plot in Figure 4.8 shows that across all factor solutions, EFAST shows better fit than EFA, suggesting the improvement in model fit outweighs the additionally estimated parameters. As the number of requested factors increases beyond optimality, this model fit improvement diminishes as EFA explains more of the symmetry structure through the additional factors. In general, the AIC tends to overextract factors, the BIC slightly underextracts, and the SSABIC shows the best extraction performance (see also Appendix B.5). In practice, therefore, we suggest using SSABIC for determining the number of factors when model fit is of primary concern. Note that a researcher may also wish to determine the number of factors based on other considerations, such as usability in further analysis, estimation tractability, or theory.

4.4 EFAST in practice: Modeling brain imaging data

In the field of cognitive neuroscience, a large body of work has demonstrated close ties between individual differences in brain structure and concurrent individual differences in cognitive performance such as intelligence tasks (Basten, Hilger, & Fiebach, 2015). Moreover, different aspects of brain structure can be sensitive to clinical and pre-clinical conditions such as grey matter for multiple sclerosis (Eshaghi et al., 2018), white matter hyperintensities for cardiovascular factors (Fuhrmann et al., 2019) and white matter microstructure for conditions such as ALS (Bede et al., 2015), Huntingtons (Rosas et al., 2010) and many other conditions.

However, one perennial challenge in imaging is how to deal with the dimensionality of imaging data. Depending on the spatial resolution, a brain image can be divided into as many as 100,000 individual regions, or voxels, rendering mass univariate approaches vulnerable to issues of multiple comparison. An alternative approach is to focus on sections called regions of interest (ROIs) defined either anatomically (e.g., Desikan et al., 2006) or functionally (e.g., Schaefer et al., 2018). However, this only solves the challenge of dimensionality in part, by grouping adjacent voxels into meaningful regions. An emerging approach is therefore to study how neural measures covary across populations or time, either in these ROIs (Sripada et al., 2019) or at the voxel level (DuPre & Spreng, 2017). This offers a promising strategy to reduce the high dimensional differences in brain structure into a tractable number of components, or factors, not limited by spatial adjacency.

However, standard techniques such as EFA or PCA do not easily allow for the integration of a fundamental biological fact: That there exists strong contralateral symmetry between brain regions, such that any given region (e.g. the left lingual gyrus) is generally most similar to the same region on the other side of the brain. Here, we show how we can combine the strengths of exploratory data reduction with the integration of a priori knowledge about the brain into a more sensible, anatomically plausible factor structure which can either be pursued as an object of intrinsic interest or used as the basis for further investigations (e.g. which brain factors are most strongly associated with phenotypic outcomes).

4.4.1 Empirical example: Grey matter volume

4.4.1.1 Data description

The data we use is drawn from the Cambridge Centre for Ageing and Neuroscience (Shafto et al., 2014; Taylor et al., 2017). Cam-CAN is a community derived lifespan sample (ages 18-88) of healthy individuals. Notably, the raw data from the Cam-CAN cohort is freely available through our data portal `https://camcan-archive.mrc-cbu.cam.ac.uk/dataaccess/`. The sample we discuss here is based on 647 individuals. For the purposes of this project we use morphometric brain measures derived from the T1 scans. Specifically, we used the Mindboggle pipeline (Klein et al., 2017) to estimate region based grey matter volume, using the underlying freesurfer processing pipeline. To delineate the regions, we here use the Desikan-Killiany-Tourville atlas for determining the ROIs (Klein & Tourville, 2012) as illustrated in Figure 4.9.

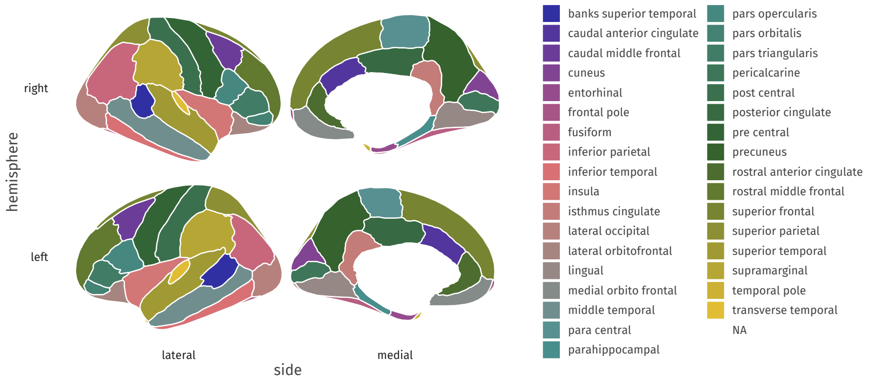

Figure 4.9 Desikan–Killiany–Tourville atlas used in the empirical illustration, as included in the ggseg package (Mowinckel & Vidal-Piñeiro, 2019).

We focus only on grey matter (not white matter) and only on cortical regions (not subcortical or miscellaneous regions such as ventricles) with the above atlas, for a total of 68 brain regions. The correlation matrix of regional volume metrics is shown in Figure 4.10, where the first 34 variables are regions of interest (ROIs) in the left hemisphere, and the last 34 variables are ROIs in the right hemisphere. The presence of higher covariance due to contralateral homology is clearly visible in the darker secondary diagonal 'stripes' which show the higher covariance between the left/right version of each anatomical region. Our goal is to reduce this high-dimensional matrix into a tractable set of 'brain factors', which we may then use in further analyses, such as differences in age sensitivity, in a way that respects known anatomical constraints.

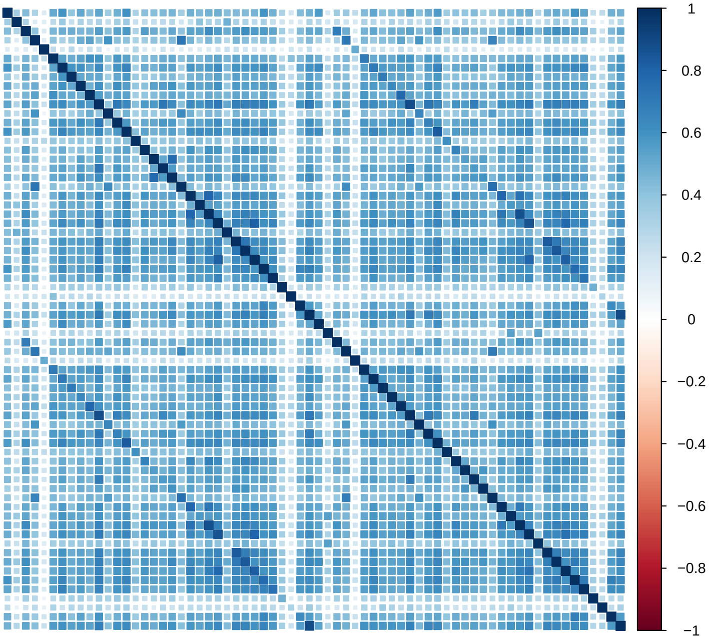

Figure 4.10 Correlation plot of cortical grey matter volume in 647 T1 weighted images of the Cam-CAN sample, estimated through Mindboggle in 34 brain regions in each hemisphere according to DKT segmentation. Numbers on the colour scale indicate the strength of the estimated correlation, with darker blue indicating stronger positive correlations. Secondary diagonal lines are visible indicating correlation due to contralateral homology.

The default estimation using EFA will attempt to account for the strong covariance among homologous regions seen in this data, meaning it is unlikely for, say, the left insula and the right insula to load on different factors, and/or for a factor to be characterized only/mostly by regions in one hemisphere. To illustrate this phenomenon, we first run a six-factor, geomin-rotated EFA for the above data (the BIC suggests six factors for this data using the EFAST model). The factor loadings for each ROI in the left and right hemispheres are plotted in Figure 4.11. A strong factor loading for a ROI in the left hemisphere is likely to have a strong factor loading in the right hemisphere due to the homologous correlation, as shown by the strong correlations for each of the factors.

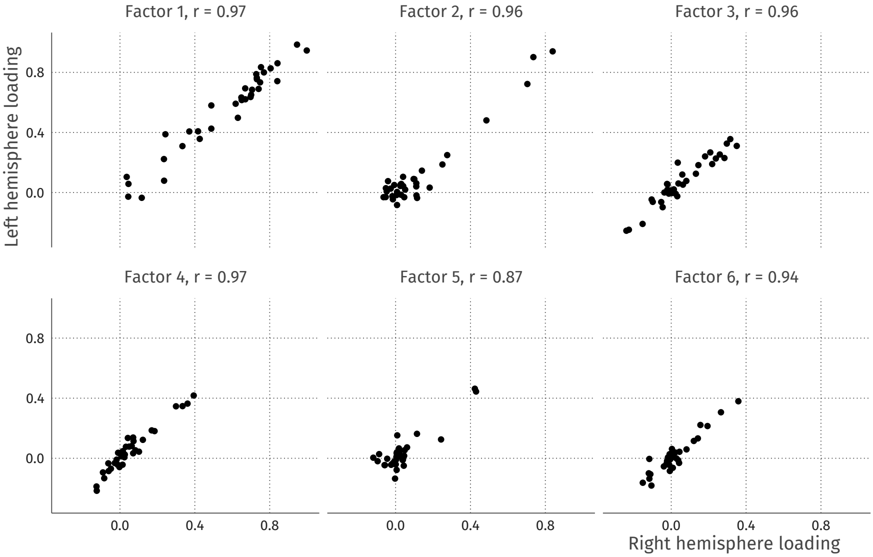

Figure 4.11 Left-right hemisphere factor loading correlations. The correlations between the loadings are high, indicating a strong similarity between the loadings in the left and right hemispheres.

In EFA, the resulting factors thus inevitably capture correlation due to contralateral symmetry, inflating or deflating factor loadings due to these contralateral residual correlations. Most problematically from a substantive neuroscientific standpoint, this distortion means it is effectively impossible to discover *lateralized factors*, i.e. patterns of covariance among regions expressed only, or dominantly, in one hemisphere. This is undesirable, as there is both suggestive and conclusive evidence that some neuroscientific mechanisms may display asymmetry. For instance, typical language ability is associated with an asymmetry in focal brain regions (e.g., Bishop, 2013; Gauger, Lombardino, & Leonard, 1997), whereas structural differences in the right hemisphere may be more strongly associated with face perception mechanisms (Frässle et al., 2016). Developmentally, there is evidence that the degree of asymmetry changes across the lifespan (e.g. Plessen, Hugdahl, Bansal, Hao, & Peterson, 2014; Roe et al., 2020). Within a SEM context, recent work shows that model fit of a hypothesized covariance structure may differ substantially between the right and left hemispheres despite focusing on the same brain regions (Meyer, Garzón, Lövdén, & Hildebrandt, 2019). The ignorance of traditional techniques for the residual structure may cause lateralized covariance factors to appear symmetrical instead, or to not be observed at all.

4.4.1.2 Results

In this section, we compare the model fit and factor solutions of EFA and EFAST for the Cam-CAN data, and we show how EFAST decomposes the correlation matrix in Figure 4.10 into factor, structure, and residual variance components. The full annotated analysis script to reproduce these results is available as supplementary material to this manuscript.

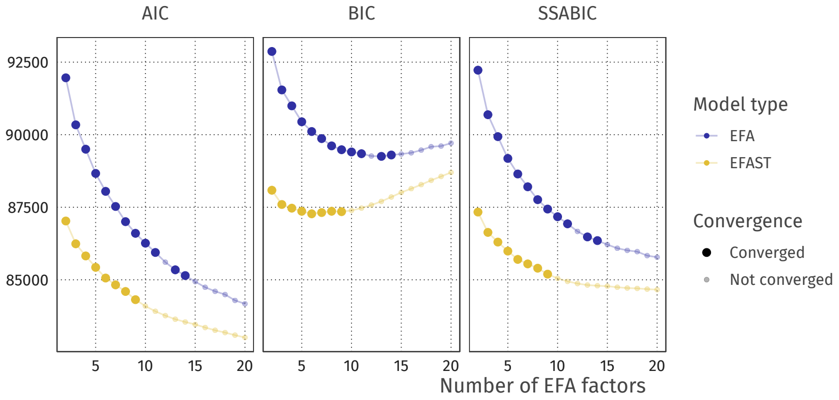

Figure 4.12 AIC and BIC for the with increasing numbers of EFA factors. Semitransparent points indicate models which are inadmissible either due to nonconvergence or convergence to a solution with problems (e.g., Heywood cases). In these cases we plot the information criteria based on the log-likelihood computed at the time the estimation terminated.

Overall, the EFAST model performs considerably better than standard EFA using common information criteria (Figure 12). The BIC criterion, combined with the convergence of the models to an admissible solution, suggests that 6 factors is optimal for this dataset. While both AIC and SSABIC show that more factors may be needed to properly represent the data, we see that this quickly leads to nonconvergence. We here consider 6 factors to be a tractable number for further analysis. First and foremost, this 6-factor solution shows a much better model solution under EFAST (BIC $\approx$ 87500) than under EFA (BIC $\approx$ 90000), emphasizing the empirical benefits of appropriately modeling known biological constraints. Additionally, statistical model comparison through a likelihood ratio test shows that the EFAST model fits significantly better (see Table 4.1). Other fit measures such as CFI, RMSEA, and SRMR paint a similar story. The full factor loading matrix for both EFAST and EFA are shown in Appendix B.6.

The EFAST model decomposes the observed correlation matrix from Figure 4.10 into the three components displayed in Figure 4.13. The most notable observation here is the separation of symmetry structure (last panel) and latent factor-implied structure (first panel): the factor solution (first panel) does

Table 4.1 Comparing the fit of the EFAST and EFA models with 6 factors, using a likelihood ratio test and several fit criteria.

	CFI	RMSEA	SRMR	χ^2	Df	$\Delta\chi^2$	ΔDf	Pr($> \chi^2$)
EFAST	0.912	0.057	0.209	5762.676	1851			
EFA	0.843	0.075	0.342	8818.146	1885	3055.471	34	$< .001$

not attempt to explain the symmetry structure seen in the data (i.e. the characteristic diagonal streaks are no longer present). This indicates that the EFAST model correctly separates symmetry covariance from underlying trait covariance in real-world data.

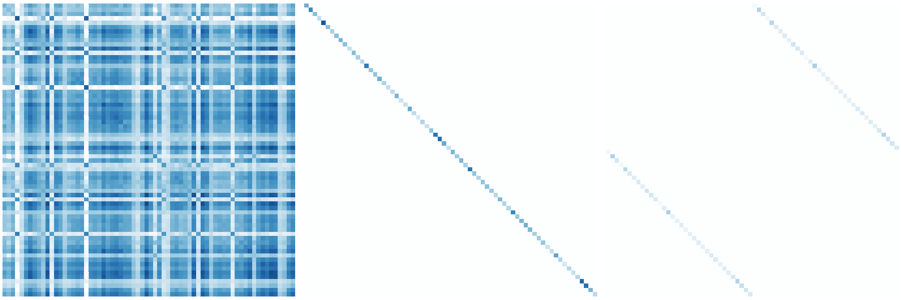

Figure 4.13 Extracted correlation matrix components using a 6-factor EFAST model with unconstrained correlations. Darker blue indicates stronger positive correlation. From left to right: factor-implied correlations, residual variance, and structure matrix.

We also extracted the estimated factor covariance, shown as a network plot in Figure 4.14. For EFA, some latent variables show very strong covariance, clustering them together due to the contralateral symmetry. This effect is not visible in the EFAST model, which shows a more well-separated latent covariance structure. This suggests that one consequence of a poorly specified EFA can be the considerable overestimation of factor covariance, which in turn adversely affects the opportunities to understand distinct causes or consequences of individual differences in these factors.

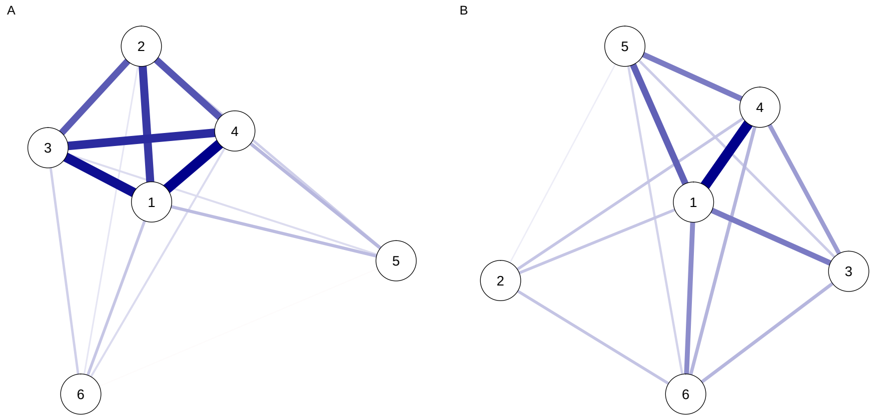

Figure 4.14 Network plots of the latent covariance for EFA (panel A) and EFAST (panel B).

4.4.2 Empirical example: White matter microstructure

4.4.2.1 Data description

Our second empirical example uses white matter structural covariance networks. We use 42 tracts from the ICBM-DTI-81 atlas (Mori et al., 2008), including only those tracts with atlas-separated left/right tracts (i.e. excluding divisions of the corpus callosum – For a full list, see appendix). As anatomical metric we use tract-based mean fractional anisotropy, a summary metric sensitive (but not specific) to several microstructural properties (Jones, Knösche, & Turner, 2013). For more details regarding the analysis pipeline, see (Kievit, Davis, Griffiths, Correia, & Henson, 2016). The same tracts and data were previously analysed in (Jacobucci, Brandmaier, & Kievit, 2019).

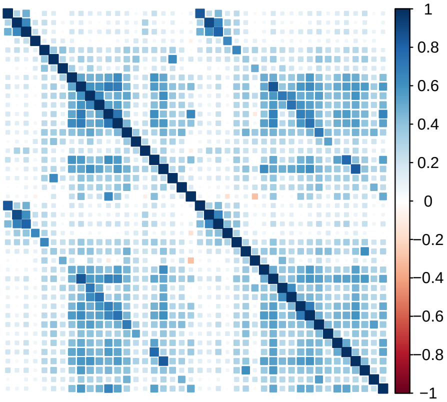

Figure 4.15 Correlation matrix for Cam-CAN white matter tractography data (fractional anisotropy). Numbers on the colour scale indicate the strength of the estimated correlation, with darker blue indicating stronger positive correlations. Secondary diagonal lines are visible indicating correlation due to contralateral homology.

4.4.2.2 Results

We chose 6 factors for the EFAST and EFA models based on the SSABIC in combination with the convergence limitations. In Table 4.2, the two models are compared on various characteristics. From the likelihood ratio test, we can see that the EFAST model represents the white matter data significantly better ($\chi^2(21) = 3632.586$, $p < .001$), and inspecting the SSABIC values (EFA = 59120, EFAST = 55727) leads to the same conclusion. In addition, the CFI, RMSEA, indicate better fit for the EFAST model, too.

Table 4.2 Comparing the fit of the EFAST and EFA models with 6 factors for the white matter data, using a likelihood ratio test and several fit criteria.

	CFI	RMSEA	SRMR	Df	χ^2	$\Delta\chi^2$	ΔDf	Pr($>\chi^2$)
EFAST	0.899	0.081	0.205	603	3137.462			
EFA	0.756	0.123	0.198	624	6770.048	3632.586	21	$< .001$

The only index which indicates slightly poorer fit is the SRMR. The difference is very small in this case, but nonetheless it is relevant to show where these differences lie. A visual representation of the root square residual (observed - implied) correlations – which form the basis of the SRMR fit index – can be found in Figure 4.16. The figure shows that EFAST is able to represent the symmetry better: it has almost no residuals on the secondary diagonals. The remaining residuals are very similar, though slightly higher, leading to a higher SRMR.

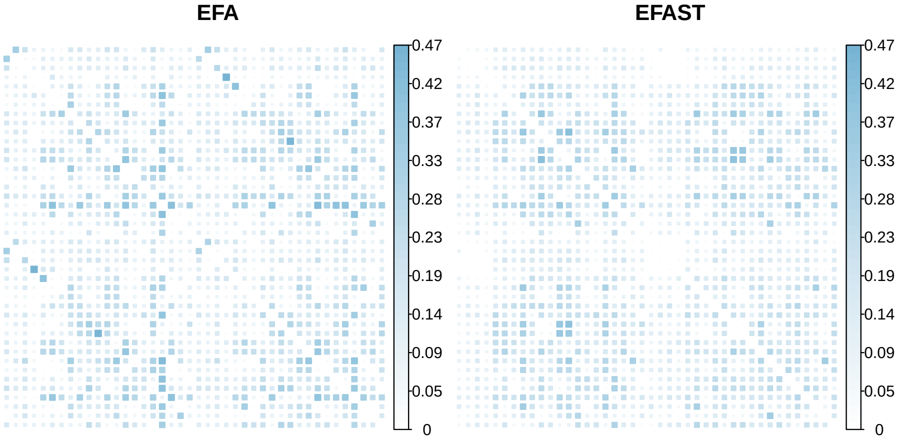

Figure 4.16 Visual representation of the root square residual (observed - implied) correlations, which form the basis of the SRMR fit index. Numbers on the colour scale indicate root square residual correlation, darker blue indicates larger residual.

4.4.3 Empirical example: Resting state Functional connectivity

4.4.3.1 Data description

Our previous examples correlation matrices capturing between- individual similarities across regions. However, the same techniques can be implemented at the within-subject level given suitable data. One such measure is *functional connectivity* which reflects the temporal connectivity between regions during rest or a given task, and captures the purported strength of interactions, or communications, between regions (Van Den Heuvel & Pol, 2010). Here we use functional connectivity matrices from 5 participants in the Cam-CAN study measured during an eyes-closed resting state block. We focus on 90 cortical and sub-cortical regions from the AAL-atlas (Tzourio-Mazoyer et al., 2002). The methodology to compute the connectivity metrics is outlined in (Geerligs, Tsvetanov, & Henson, 2017), and the data reported here have been used in (Lehmann, Henson, Geerligs, & White, 2019).

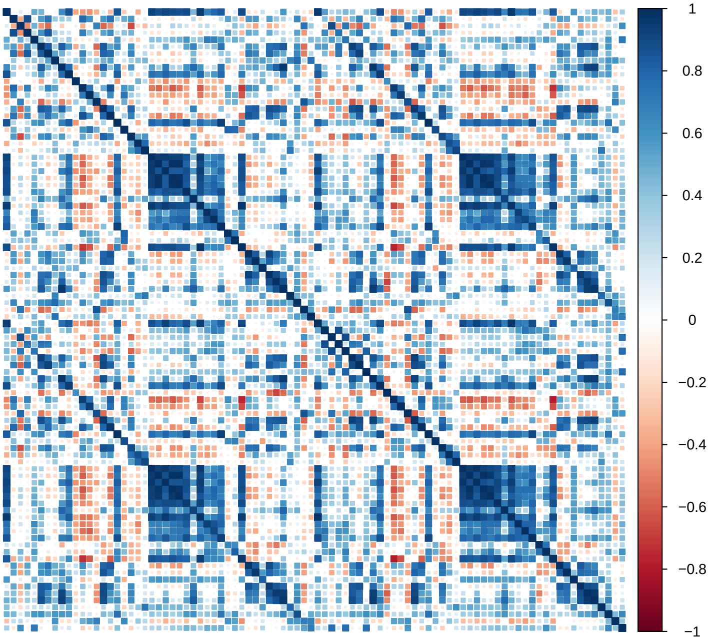

Figure 4.17 Correlation matrix for the first participant in the Cam-CAN resting state functional connectivity dataset. Numbers on the colour scale indicate the strength of the estimated correlation, with darker blue indicating stronger positive correlations. Secondary diagonal lines are visible indicating correlation due to contralateral homology.

4.4.3.2 Results

For this example, data from the first participant was used to perform the model fit assessments. We performed a similar routine as with the previous empirical datasets for determining the number of factors: we fit the EFAST and EFA models for 2-16 factors and compare their information criteria. All of the models converged, and the optimal model based on the BIC is a 13-factor EFAST model. BIC was chosen as a criterion for the number of factors in order to keep the analysis tractable – the other criteria indicated an optimum beyond 16 factors.

The 13-factor EFAST model was then compared to the 13-factor EFA model on various fit indices. The results of this comparison can be found in Table 4.3. Across the board, the EFAST model has better fit, as the EFAST CFI, RMSEA, SRMR and χ^2 fit indices outperform those for the EFA model, demonstrating that accounting for the bilateral symmetry in dimension reduction through factor analysis leads to better fitting model of the data.

This approach also allows for comparing the factor loadings for the dif-

Table 4.3 Comparing the fit of the EFAST and EFA models with 13 factors for the functional resting state data, using a likelihood ratio test and several fit criteria.

	CFI	RMSEA	SRMR	Df	χ^2	$\Delta\chi^2$	ΔDf	Pr($> \chi^2$)
EFAST	0.836	0.093	0.253	2868	9350.278			
EFA	0.774	0.108	0.272	2913	11828.126	2477.848	45	0.000

ferent participants. For illustration, the plot in Figure 4.18 shows the profile of factor loadings for the first three factors (columns) across the five participants (rows). These profile plots can be a starting point for comparison of the connectivity structure across participants, where higher correlation among participants means a more similar connectivity structure, while taking into account the symmetry in the brain. For example, for Factor 1, participant 3 has a quite different functional connectivity factor loading profile than the other participants.

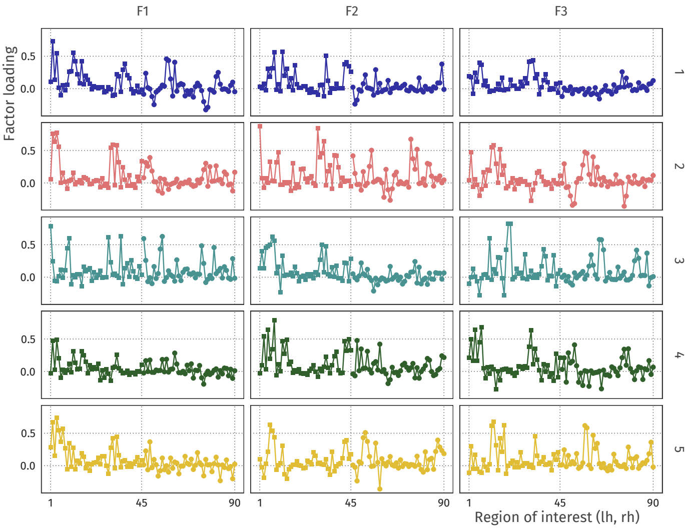

Figure 4.18 Comparison of factor loading profiles for the first three factors (columns) across five participants (rows). The left side of each subplot corresponds to the left hemisphere, the right side corresponds to the right hemisphere.

4.5 Model-based lateralization index

In the simulations, we showed how the EFAST approach yields a more veridical representation of the factor structure than EFA. However, using EFAST yields an additional benefit: our model allows for estimating the extent of symmetry in each ROI, while taking into account the overall factor structure. This enables researchers to use this component of the analysis for further study. The (lack of) symmetry may be of intrinsic interest, such as in language development research (Schuler et al., 2018), intelligence in elderly (Moodie et al., 2019), and age-related changes in cortical thickness asymmetry (Plessen et al., 2014). In the `efast` package, we have implemented a specific form of lateralization which is based on a variance decomposition in the ROIs. Our lateralization index (LI) is a dissimilarity measure representing the proportion of residual variance (given the trait factors) in an ROI that cannot be explained by symmetry. The index value is 0 if the bilateral ROIs are fully symmetric (conditional on the trait factors), and 1 if there is no symmetry:

$$LI_i = 1 - \text{cor}(u_i^{lh}, u_i^{rh}) \tag{4.1}$$

where u_i^{lh} and u_i^{rh} are residuals given the trait factors of interest of the i^{th} ROI in the left and right hemisphere, respectively. The correlation $\text{cor}(\cdot\,,\cdot)$ between these residuals represents the amount of symmetry, so the LI_i represents the *residual dissimilarity* of the i^{th} ROI in the two hemispheres after taking into account the factor structure in the data. When LI_i is 0, the ROIs are fully symmetric given the traits, and a LI_i of 1 indicates no symmetry. Note that LI_i can be larger than 1 if the residuals are negatively correlated.

The LI for each ROI in the grey matter volume example is shown in Figure 4.19. Here, we can see that there is high lateralization in the superior temporal sulcus and medial orbitofrontal cortex, but high symmetry in the lateral orbitofrontal cortex and the insula. In Figure 4.20, we additionally show in the white matter example that LI can naturally be supplemented by standard errors and confidence intervals. Thus, the EFAST procedure not only improves the factor solution under plausible circumstances for such datasets, but in doing so yields an intrinsically interesting metric of symmetry.

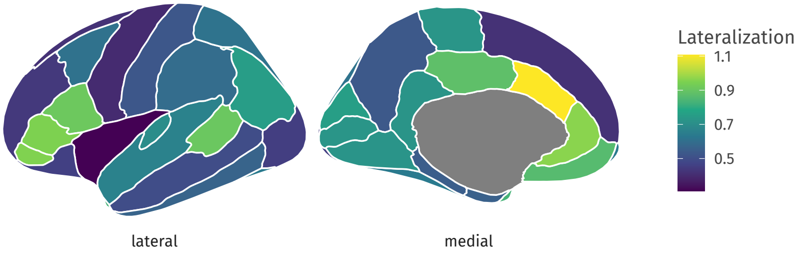

Figure 4.19 Amount of grey matter volume asymmetry per ROI. Dark blue areas are highly symmetric given the previously estimated 6-factor solution, and bright yellow areas are highly asymmetric. Such plots can be made and compared for different groups and statistically investigated for differences in symmetry for a common factor solution. A lateralization index (LI) of 0 means that the regions are fully symmetric conditional on the trait factors.

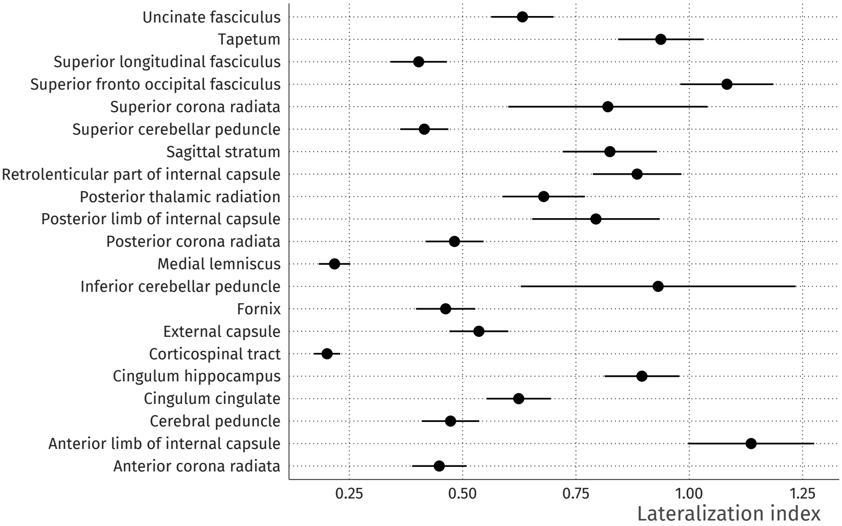

Figure 4.20 White matter lateralization index for a selected set of regions given the previously estimated 6-factor solution. Lower values means that bilateral ROIs are more symmetric conditional on the trait factors, higher values that they are less so. The line ranges indicate 95% confidence intervals, computed as $LI \pm 1.96 \times SE_{LI}$, where the standard error SE_{LI} is computed using the delta method.

4.6 Summary and discussion

In this paper, we have developed and implemented EFAST, a method for performing dimension reduction with residual structure. We show how this new method outperforms standard EFA across three separate datasets, by taking into account hemispheric symmetry in brain covariance data. We have argued through both simulations and real-world data analysis that our method is an improvement in the dimension reduction step of such high-dimensional, structured data, yielding a more veridical factor solution. Such a factor solution can be the basis for further analysis, such as an extension of the factor model to prediction of continuous phenotype variables such as intelligence scores, or the comparison among different age groups. These extensions will be improved by building on a factor solution which appropriately takes into account the symmetry of the brain. Furthermore, we believe that many data reduction problems in social, cognitive, and behavioural sciences have a similar structure: residual structure is known, but precise theory about the underlying factor structure is not (Asparouhov & Muthén, 2009). As such, although we focus on brain imaging data, our approach is likely more widely applicable.

Care is needed in the interpretation of the factor solution as underlying dimensions, as the empirical application has shown that the absolute level of fit for both the EFA and EFAST models is not optimal. In addition, estimation of more complex factor models may lead to nonconvergence or inadmissible solutions. Such problems would need to be further investigated, potentially leading to more stable estimation, for example through a form of principal axis factoring, or potentially through penalization of SEM (Jacobucci et al., 2019; van Kesteren & Oberski, 2019). However, these limitations hold equally for EFA, and when comparing both methods it is clear from the results in this paper that the inclusion of structured residuals greatly improves the representation of the high-dimensional raw data by the low-dimensional factors. In summary, this relatively simple but versatile extension of classical EFA may be of considerable value to applied researchers with data that possess similar qualities to those outlined above. We hope our tool will allow those researchers to easily and flexibly specify and fit such models.

Note that we are not the first to suggest using structured residuals in EFA to take into account prior knowledge about structure in the observed variables. Adding covariances among residuals is a common method to take into account features of the data-generating process (e.g., Cole et al., 2007), and this has been possible in the context of EFA since the release of the ESEM capability in `MPlus` (Asparouhov & Muthén, 2009) and in `lavaan` (Rosseel, 2019). In the context of neuroscientific data, similar methods in accounting for structure in dimension reduction have been researched by De Munck, Huizenga, Waldorp, and Heethaar (2002) in source localization for EEG/MEG. Our goal for this paper has been to provide a compelling argument for the use of such structured residuals from the point of view of neuroscience, as well as a user-friendly, open-source implementation of this method for dimension reduction in real-world datasets.

Acknowledgements

We would like to thank Yves Rosseel for valuable input and the development of key tools, Linda Geerligs for providing the functional connectivity data, and Jonathan Helm and Øystein Sørensen for their helpful comments on an earlier version of this manuscript. E.-J. van Kesteren is supported by the Netherlands Organization for Scientific Research (NWO) under grant number 406.17.057. R. A. Kievit is supported by the UK Medical Research Council SUAG/047 G101400. This project received funding from the European Union's Horizon 2020 research and innovation programme (grant agreement number 732592).

Competing interests

The authors declare no competing interests.

Chapter 5
Privacy-Preserving Generalized Linear Models using Distributed Block Coordinate Descent

Van Kesteren, E. J., Sun, C., Oberski, D. L., Dumontier, M., & Ippel, L. (2019, in preparation). Privacy-Preserving Generalized Linear Models using Distributed Block Coordinate Descent. arXiv:1911.03183

Combining data from varied sources has considerable potential for knowledge discovery: collaborating data parties can mine data in an expanded feature space, allowing them to explore a larger range of scientific questions. However, data sharing among different parties is highly restricted by legal conditions, ethical concerns, and / or data volume. Fueled by these concerns, the fields of cryptography and distributed learning have made great progress towards privacy-preserving and distributed data mining. However, practical implementations have been hampered by the limited scope or computational complexity of these methods. In this paper, we greatly extend the range of analyses available for vertically partitioned data, i.e., data collected by separate parties with different features on the same subjects. To this end, we present a novel approach for privacy-preserving generalized linear models, a fundamental and powerful framework underlying many prediction and classification procedures. We base our method on a distributed *block coordinate descent* algorithm to obtain parameter estimates, and we develop an extension to compute accurate standard errors without additional communication cost. We critically evaluate the information transfer for semi-honest collaborators and show that our protocol is secure against data reconstruction. Through both simulated and real-world examples we illustrate the functionality of our proposed algorithm. Without leaking information, our method performs as well on vertically partitioned data as existing methods on combined data – all within mere minutes of computation time.

EVK was the main author, created the software, and ran the experiments. CS wrote parts of the introduction and related work, found and pre-processed datasets. DLO reviewed and created the proof for the standard errors. MD provided feedback. LI supervised the writing process and reviewed, improved the mathematical formulation, and aided in the experiments.

We conclude that our method is a viable approach for vertically partitioned data analysis with a wide range of real-world applications.

5.1 Introduction

With technological developments in computational power and storage capacity, an increasing amount of data is collected and stored by a variety of data parties (Kaisler, Armour, Espinosa, & Money, 2013). Over the past decades, data mining has been successful in extracting information from such datasets, but it is especially powerful when various data sources are combined: collaborating data parties can mine data in a larger feature space, allowing them to discover knowledge beyond their individual potential. For example, in the medical domain, personal health conditions are significantly affected not only by genetic and biological factors, but also by individual behaviour and social circumstances (World Health Organization, 2008); combining those sources has the potential to improve analytical models for health outcomes (Ancker, Kim, Zhang, Zhang, & Pathak, 2018; Kasthurirathne, Vest, Menachemi, Halverson, & Grannis, 2017).

However, there is a pertinent obstacle to unlocking the potential of combining datasets: integrating various sources may reveal private information about individual data subjects to the collaborating parties. Hence, data sharing is highly restricted by legal and ethical concerns. This highlights the need for privacy-preserving techniques which perform data mining tasks on multiple sources without explicitly sharing their full data (e.g., Du, Han, & Chen, 2004; Gambs, Kégl, & Aïmeur, 2007; Gascón et al., 2017; Karr, Lin, Sanil, & Reiter, 2009). In this paper, we develop a novel algorithm for performing generalized linear modeling (GLM) in a privacy-preserving way in such a partitioned data situation. GLM is a powerful statistical framework for prediction and classification and is at the basis of a wide range of analysis applications including linear, count, and logistic regression (Dobson & Barnett, 2008; McCullagh & Nelder, 1989).

This paper is organized as follows. In Section 5.2, related work is discussed to contextualize our contribution. In Section 5.3, we introduce our proposed method for GLM on vertically partitioned data. Next, we describe in detail the privacy-preserving and information sharing characteristics of this protocol in Section 5.4, and we analyze how the information transfer affects the ability of the partner organisation to recover the collaborator's data. In Section 5.5, we benchmark our implementation of the protocol against full-data analysis using Monte Carlo simulations and we illustrate the functionality of our implementation using three different real-life data sets from the UCI Machine Learning repository (Blake & Merz, 1998). Finally, we discuss the strengths and limitations of our approach in Section 5.6 and we provide suggestions for future research.

All of the methods described here are implemented in `privreg`, an open-source software package for the R programming language (R Core Team, 2018). This implementation includes encryption for all communication across parties based on a pre-shared key, and includes a user-friendly interface based around an object-oriented architecture. The package is available for installation from `https://github.com/vankesteren/privreg`.

5.2 Related work

In practice, there are two main types of data partitioning (Vaidya & Clifton, 2005). Different data sources might collect the same features of different data subjects, e.g., different hospitals collect the same type of information from their own set of patients. This situation is referred to as *horizontally partitioned* data. Alternatively, separate sources might collect different information from the same data subjects, e.g., medical features by the hospital may be combined with socioeconomic features from a government statistics department. This situation is referred to as *vertically partitioned* data, which is the focus of the current paper. There is also a third scenario, where data are both vertically and horizontally partitioned, which may be referred to as *hybrid partitioning.*

Our aim is to analyze data which is vertically partitioned without leaking raw data to the collaborating parties (*Alice* and *Bob*). In order to analyze such data, either the dataset may be combined but hidden from the collaborating parties, or the analytical procedure should prevent leaking of information. The former relies on the inclusion of an 'uninterested' or trusted third party (TTP): Each party sends their raw data encrypted to the TTP, who then performs the required analyses on the combined data sets. Afterwards, the TTP returns the results to all data parties and the raw data of *Alice* stays hidden to *Bob*. However, this solution requires all parties to fully trust the TTP, which might not be possible in the face of restrictive legislation or sensitive data.

There is another class of methods which do not rely on a TTP, instead using cryptography to perform data mining tasks on vertically partitioned data. These methods focus on preventing information leakage by creating protocols which hide the raw data from the collaborator (e.g., for the construction of decision trees, Agrawal & Srikant, 2000). In this class of methods Du and Atallah (2001) and Du et al. (2004) investigated various protocols for secure matrix computation for linear least squares regression and classification problems. Several other authors used and extended more general secure multiparty computation protocols (e.g., the garbled circuit protocol; Yao, 1986) to perform regression on vertically partitioned data (Amirbekyan & Estivill-Castro, 2007; Bloom, 2019; Fang, Zhou, & Yang, 2013; Gascón et al., 2016, 2017; Nikolaenko et al., 2013; Slavkovic, Nardi, & Tibbits, 2007). While their use of these general protocols yields certain privacy guarantees, their practical implementations and use are hindered by requiring semi-trusted third parties, intermediate data sharing, computational complexity, or a limitation to the linear regression situation.

Another line of research leverages the privacy-preserving properties of algorithms from *federated* or *distributed learning*, a field researching data mining on separated datasets (Dobriban & Sheng, 2018; T. Li, Sahu, Talwalkar, & Smith, 2019). A canonical example is by Sanil, Karr, Lin, and Reiter (2004), who developed a method to compute linear regression coefficients iteratively based on an algorithm by Powell (1964). Other authors leverage specific distributed learning algorithms to implement statistical learning for vertically partitioned data (Vaidya & Clifton, 2002, 2003, 2005; Vaidya, Yu, & Jiang, 2008). Our method is closely related to this branch of research. Unlike existing regression methods from the TTP or cryptography fields, our method does not make use of a trust assumption or complex cryptographic protocols, but it is naturally secure due to its reliance on a federated learning algorithm which never moves the data from its original location. In the next section, we explain the concept and implementation behind our proposed privacy-preserving GLM technique.

5.3 Proposed method

Our proposed method uses *block coordinate descent* (BCD) to estimate generalized linear models (GLM) in a situation where data is vertically partitioned across two or more parties. In BCD, parameters are iteratively updated for each block of features, cycling over the blocks until an optimum is found (Hastie et al., 2015). This optimization algorithm can be seen as a form of distributed learning (Bertsekas & Tsitsiklis, 1989; Richtárik & Takáč, 2016) which we exploit as a privacy-preserving method because the features remain in different locations. Only linear predictions need to be transferred across the feature blocks – the full data is never shared.

Note that for the remainder of the paper, we assume that the records of the data subjects are in the same order across databases, in line with Gascón et al. (2017). Furthermore, we only consider the situation where the target attribute is available to both parties (Sanil et al., 2004). In addition, we follow the tradition in the existing literature (e.g., Karr, 2010; Vaidya et al., 2008) to assume semi-honest adversaries: data parties will follow the protocol as described, but will still attempt to learn as much information as possible from other parties. This contrasts with malicious adversaries that can arbitrarily deviate from the protocol (Lindell, 2005).

In this section, we build up the BCD algorithm from the simpler case of linear regression before extending it to full GLM. Therefore, we first explain the necessary background on linear regression, as well as the notation used throughout this paper. Then, coordinate descent estimation is introduced as a means to estimate its maximum likelihood coefficients. In Section 5.3.3, this algorithm is then extended to accommodate a vertically partitioned data structure, and in Section 5.3.4 we generalize it to different outcome families in order to estimate GLMs. Finally, we develop a novel method to obtain standard errors within this framework.

5.3.1 Background

We consider the centered design matrix with features $\boldsymbol{X} \in \mathbb{R}^{N \times P}$ and the centered target variable $\boldsymbol{y} \in \mathbb{R}^{N \times 1}$, where N is the sample size, or number of observations, and P is the number of features. The p^{th} column in $\boldsymbol{X}$ is represented as $\boldsymbol{x}_p$. The columns in $\boldsymbol{X}$ excluding the p^{th} are denoted as $\boldsymbol{X}_{\text{-}p}$.

The basic regression model is then as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{5.1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^P$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, and $\boldsymbol{\epsilon} \perp \boldsymbol{X}$. The well-known closed-form maximum likelihood estimator of the P regression coefficients $\boldsymbol{\beta}$ in this model is:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{5.2}$$

We further define the vector of predicted values as $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ and the vector of residuals as $\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$.

5.3.2 Cyclic coordinate descent estimation

When instead of the full design matrix $\boldsymbol{X}$ we consider only the p^{th} variable, the estimator in Equation 5.1 yields the *marginal* regression coefficient. Thus, by simplifying Equation 5.1 to the univariate case, the marginal coefficient for the p^{th} variable β_p^* is estimated as

$$\hat{\beta}_p^* = \frac{\langle \boldsymbol{x}_p, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}_p \,, \boldsymbol{x}_p \rangle} = \frac{\text{cov}(\boldsymbol{x}_p \,, \boldsymbol{y})}{\text{var}(\boldsymbol{x}_p)} \tag{5.3}$$

where $\langle \cdot \,, \cdot \rangle$ indicates the inner product of two vectors. The covariance/variance notation holds because we assume a centered design matrix $\boldsymbol{X}$ and outcome variable $\boldsymbol{y}$.

If $\boldsymbol{x}_p$ covaries with any of the predictors in $\boldsymbol{X}_{\text{-}p}$, the marginal coefficient $\boldsymbol{\beta}_p^*$ is different from the *conditional* coefficient $\boldsymbol{\beta}_p$. The estimate of this coefficient is an element of $\hat{\boldsymbol{\beta}}$ in Equation 5.1, but it can equivalently be estimated in a coordinate-wise, univariate manner (Hastie et al., 2015) as follows:

$$\hat{\beta}_p = \frac{\langle \boldsymbol{x}_p, \hat{\boldsymbol{\epsilon}}_{\text{-}\boldsymbol{p}} \rangle}{\langle \boldsymbol{x}_p \,, \boldsymbol{x}_p \rangle} = \frac{\langle \boldsymbol{x}_p \,, \boldsymbol{y} - \boldsymbol{X}_{\text{-}p}\hat{\boldsymbol{\beta}}_{\text{-}p} \rangle}{\langle \boldsymbol{x}_p \,, \boldsymbol{x}_p \rangle} = \frac{\langle \boldsymbol{x}_p \,, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}_p \,, \boldsymbol{x}_p \rangle} - \frac{\langle \boldsymbol{x}_p \,, \boldsymbol{X}_{\text{-}p}\hat{\boldsymbol{\beta}}_{\text{-}p} \rangle}{\langle \boldsymbol{x}_p \,, \boldsymbol{x}_p \rangle} \tag{5.4}$$

The residual $\hat{\boldsymbol{\epsilon}}_{\text{-}p} = \boldsymbol{y} - \boldsymbol{X}_{\text{-}p}\hat{\boldsymbol{\beta}}_{\text{-}p}$ is the residual with respect to the variables excluding $\boldsymbol{x}_p$, evaluated at the maximum likelihood (ML) estimates of $\boldsymbol{\beta}$. Equation 5.4 states that the conditional regression coefficient can be obtained by computing the marginal regression coefficient of $\hat{\boldsymbol{\epsilon}}_{\text{-}p}$ on $\boldsymbol{x}_p$. This relation holds because $\hat{\boldsymbol{\epsilon}}_{\text{-}p}$ represents the part of the outcome variable unrelated to $\boldsymbol{X}_{\text{-}p}$ – by definition, $\hat{\boldsymbol{\epsilon}}_{\text{-}p} \perp \boldsymbol{X}_{\text{-}p}$. In addition, the last part of Equation 5.4 shows that the marginal and conditional estimate of the p^{th} regression coefficient are equal if $\boldsymbol{x}_p$ and $\boldsymbol{X}_{\text{-}p}$ do not covary, because the last term drops out.

The coordinate-wise estimation of $\hat{\boldsymbol{\beta}}_p$ (Equation 5.4) requires the maximum likelihood estimates $\hat{\boldsymbol{\beta}}_{\text{-}p}$ of the remaining variables to be known. However, when estimation of $\hat{\boldsymbol{\beta}}$ is the goal, these estimates are not available. This can be solved by an iterative updating procedure of the $\hat{\boldsymbol{\beta}}$ estimates:

Algorithm 1: Cyclic coordinate descent
(Hastie et al., 2015)

```
1. Initialize $\hat{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}}^*$ (marginal coefficients)
2. For each $p \in P$:
    (a) $\hat{\boldsymbol{\epsilon}}_{\text{-}p} \leftarrow \boldsymbol{y} - \boldsymbol{X}_{\text{-}p}\hat{\boldsymbol{\beta}}_{\text{-}p}$
    (b) $\hat{\beta}_p \leftarrow \langle \boldsymbol{x}_p \, , \hat{\boldsymbol{\epsilon}}_{\text{-}p} \rangle \, / \, \langle \boldsymbol{x}_p \, , \boldsymbol{x}_p \rangle$
3. Repeat step (2.) for $R$ iterations until convergence (i.e.,
   the change in parameter estimates over iterations becomes negligible)
```

An advantage of this method is that it does not require storing the full $P \times P$ covariance matrix in memory, and this matrix does not need to be inverted – an $\mathcal{O}(P^3)$ operation. This advantage becomes especially relevant as P grows (Hastie et al., 2015). Another advantage is that this estimation method allows for regularization to be implemented naturally. For example, the ℓ_1 penalized parameters can be computed by soft-thresholding $\langle \boldsymbol{x}_p \, , \hat{\boldsymbol{\epsilon}}_{\text{-}p} \rangle$ in each iteration. This is the approach taken by the popular regularized regression package `glmnet` (Friedman et al., 2010).

A graphical display of the behaviour of the estimated parameters during the cyclical coordinate descent procedure is shown in panel A of Figure 5.1. Here, 9 covarying features $\boldsymbol{X}$ were generated from a multivariate normal distribution. Then random parameter values $\boldsymbol{\beta}$ and random normal errors $\boldsymbol{\epsilon}$ were created and used to generate the target variable $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$.

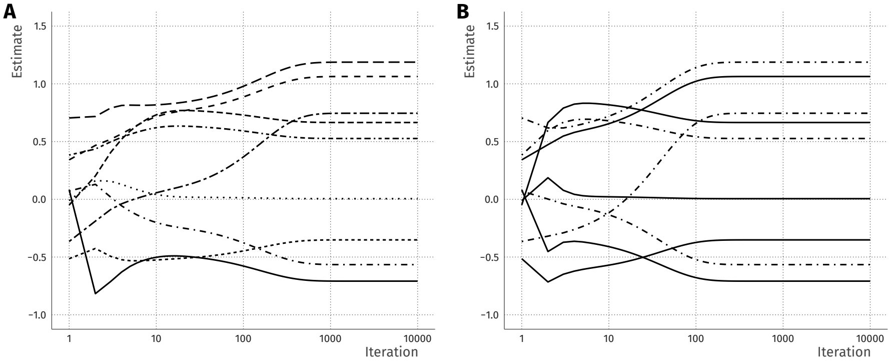

Figure 5.1 Panel A: Coordinate descent paths for linear regression with 9 covarying features, simulated from a multivariate normal distribution. The parameter lines converge from the marginal ML estimates (left) to the conditional ML estimates (right). Note that the x-axis is on a logarithmic scale and convergence happens around iteration 1000. Panel B: Block coordinate descent path for regression with 9 covarying predictors, applied to the same simulated dataset. There are two blocks, indicated by the line types. Note that convergence happens before iteration 500, faster than the cyclic coordinate descent algorithm.

Next, we show how coordinate descent generalizes to blocks of variables, and how it may be used to estimate linear regression coefficients in the vertically partitioned data scenario described above.

5.3.3 Securely estimating coefficients for linear regression

In this section, we develop the framework for analysing vertically partitioned data. Our key contribution is the combination of two observations:

1. Coordinate descent estimation works the same for single features as well as for blocks of features – resulting in a variant called block coordinate descent (BCD; Hastie et al., 2015).
2. Vertically partitioned data is blocked data – the features held by *Alice* can be considered the first block, and those held by *Bob* the second block.

Following these two observations, Algorithm 2 below thus provides an iterative estimator for the parameters of *Alice* ($\boldsymbol{\beta}_a$) and those of *Bob* ($\boldsymbol{\beta}_b$) through sharing of predictions. Predictions from *Alice* are written as $\hat{\boldsymbol{y}}_a = \boldsymbol{X}_a\hat{\boldsymbol{\beta}}_a$, and the working residual with respect to *Alice*, i.e., the part of $\boldsymbol{y}$ not related to the features in $\boldsymbol{X}_a$ is then $\hat{\boldsymbol{\epsilon}}_a = \boldsymbol{y} - \hat{\boldsymbol{y}}_a$.

Algorithm 2: Secure block coordinate descent

```
1. Initialize $\hat{\boldsymbol{y}}_b \leftarrow \mathbf{0}$
2. Alice:
    (a) $\hat{\boldsymbol{\epsilon}}_b \leftarrow \boldsymbol{y} - \hat{\boldsymbol{y}}_b$
    (b) $\hat{\boldsymbol{\beta}}_a \leftarrow (\boldsymbol{X}_a^T \boldsymbol{X}_a)^{-1} \boldsymbol{X}_a^T \hat{\boldsymbol{\epsilon}}_b$
    (c) $\hat{\boldsymbol{y}}_a \leftarrow \boldsymbol{X}_a \hat{\boldsymbol{\beta}}_a$
    (d) Send $\hat{\boldsymbol{y}}_a$ to Bob
3. Bob:
    (a) $\hat{\boldsymbol{\epsilon}}_a \leftarrow \boldsymbol{y} - \hat{\boldsymbol{y}}_a$
    (b) $\hat{\boldsymbol{\beta}}_b \leftarrow (\boldsymbol{X}_b^T \boldsymbol{X}_b)^{-1} \boldsymbol{X}_b^T \hat{\boldsymbol{\epsilon}}_a$
    (c) $\hat{\boldsymbol{y}}_b \leftarrow \boldsymbol{X}_b \hat{\boldsymbol{\beta}}_b$
    (d) Send $\hat{\boldsymbol{y}}_b$ to Alice
4. Repeat step (2.) and (3.) for $R$ iterations until convergence.
```

Upon convergence, the concatenated parameter estimates vector $(\hat{\boldsymbol{\beta}}_a, \hat{\boldsymbol{\beta}}_b)$ is equal (up to a small predetermined tolerance value) to the parameter estimates vector that would be obtained using the standard maximum likelihood estimator in the combined data set (Tseng, 2001). It follows that the element-wise summed prediction $\hat{\boldsymbol{y}}_a + \hat{\boldsymbol{y}}_b$ is equal to the prediction $\hat{\boldsymbol{y}}$ that would be obtained from the combined dataset. Thus, prediction can be done without sharing the parameter estimates. Further analysis of the privacy-preserving properties of this procedure is discussed in Section 5.4.

In panel B of Figure 5.1 we illustrate BCD, applied to the same data set as in panel A. However, instead of P blocks of 1 feature each, now there are two blocks with 5 and 4 features. BCD reaches convergence with fewer iterations than the cyclic version, because it uses more information about the covariance between the features. In general, convergence is obtained faster with fewer blocks, and with less covariance between blocks (Richtárik & Takáč, 2016). In the case of orthogonal blocks, only a single iteration is needed for convergence as the marginal estimates equal the conditional estimates. X. Li, Zhao, Arora, Liu, and Hong (2017, Theorem 8) derived a general result about the iteration complexity of BCD, showing that for smooth convex losses such as the GLM log-likelihood, the number of iterations required for convergence is linear in the number of features P.

In the next section, we show how our BCD approach may be modified to estimate generalized linear model coefficients for a wide range of applications. Then, we provide a way to estimate standard errors within this framework.

5.3.4 Extension to generalized linear models

Extending this procedure to generalized linear models (GLM) requires a slightly different estimation approach: whereas the parameter estimates of full-data linear regression can be found analytically (Equation 5.2), GLM requires an iteratively reweighted least squares (IRLS) procedure (Green, 1984; Wedderburn, 1974). In each iteration i in full-data GLM, the estimates are computed as follows:

$$\hat{\boldsymbol{\beta}}^{(i+1)} = (\boldsymbol{X}^T \boldsymbol{W}^{(i)} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W}^{(i)} \boldsymbol{z}^{(i)} \tag{5.5}$$

Here, $\boldsymbol{W}$ is a diagonal weights matrix and $\boldsymbol{z}$ is a transformation of the target variable called the *working response*, computed as

$$\boldsymbol{z}^{(i)} = \boldsymbol{\eta}^{(i)} + (\boldsymbol{y} - \boldsymbol{\mu}^{(i)}) \left(\frac{d\boldsymbol{\mu}^{(i)}}{d\boldsymbol{\eta}^{(i)}} \right) \tag{5.6}$$

where $\boldsymbol{\eta}^{(i)} = \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(i)}$ and $\boldsymbol{\mu}^{(i)}$ is a function of $\boldsymbol{\eta}^{(i)}$ as predefined in the link function (e.g., logit link for logistic regression; McCullagh & Nelder, 1989). From this working response, a *working residual* needs to be obtained which acts like $\hat{\boldsymbol{\epsilon}}_{\text{-}p}$ in Equation 5.4: a response vector orthogonal to the predictors excluding feature p. We define this working residual as follows (Friedman et al., 2010):

$$\hat{\boldsymbol{\epsilon}}_{\text{-}p} = \boldsymbol{z} - \boldsymbol{X}_{\text{-}p}\hat{\boldsymbol{\beta}}_{\text{-}p} \tag{5.7}$$

Using this working residual and the usual weights matrix from GLM, the coordinate descent algorithm proceeds in a similar fashion to that of linear regression (Algorithm 1). Just as with coordinate descent for linear regression, this algorithm readily extends to a blockwise procedure, meaning it can be adapted for the private regression method as discussed in Section 5.3.3.

5.3.5 Computing standard errors

A key component of inference in regression models is obtaining a measure of sampling uncertainty about the obtained estimates, usually standard errors. Under the assumptions of maximum likelihood theory, the limiting distribution of the deviation of the parameter estimates is the following:

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_N - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\beta) \tag{5.8}$$

where $\boldsymbol{\Sigma}_\beta$ is the asymptotic variance-covariance matrix of $\hat{\boldsymbol{\beta}}$:

$$\boldsymbol{\Sigma}_\beta = \text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} \tag{5.9}$$

In linear regression, $\hat{\sigma}^2 = \langle \hat{\boldsymbol{\epsilon}}, \hat{\boldsymbol{\epsilon}} \rangle / (N - P)$ and the standard errors of $\hat{\boldsymbol{\beta}}$ can be computed as

$$\hat{\text{se}}_{\hat{\boldsymbol{\beta}}} = \sqrt{\text{diag}(\hat{\sigma}^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1})} \tag{5.10}$$

Thus, to compute an estimate of the variance-covariance matrix of the sampling distribution of the $\hat{\boldsymbol{\beta}}$ parameters, the inverse covariance matrix of the features is needed. However, when the data is vertically partitioned, part of this covariance matrix is missing for each party. As a result, computing standard errors using the above information matrix approach is impossible for vertically partitioned data without sharing the features.

We present a novel approach to compute standard errors of the regression coefficient through creating a substitute $\boldsymbol{V}_b$ of the partner's data matrix $\boldsymbol{X}_b$. This substitute is then used as the partner's data in the computation of the asymptotic variance-covariance matrix as in Equation 5.9.

The substitute $\boldsymbol{V}_b$ needs to contain the same information for the parameters of Alice as the real data. This information is in the predictions received from Bob – the parameter estimates of Alice depend only on Bob's linear predictions. Consider the inputs and outputs of Bob, as seen by Alice: as the coordinate descent algorithm progresses along the R iterations, Alice can create two $N \times R$ matrices, $\hat{\boldsymbol{E}}_a$ and $\hat{\boldsymbol{Y}}_b$

$$\begin{aligned} \hat{\boldsymbol{E}}_a &= \left[\hat{\boldsymbol{\epsilon}}_a^{(1)}, \ldots, \hat{\boldsymbol{\epsilon}}_a^{(R)} \right] \\ \hat{\boldsymbol{Y}}_b &= \left[\hat{\boldsymbol{y}}_b^{(1)}, \ldots, \hat{\boldsymbol{y}}_b^{(R)} \right] \end{aligned} \tag{5.11}$$

These are the input and output matrices, respectively, from the projection that Bob applies in each iteration. This projection is commonly known as the *hat matrix* $\boldsymbol{H}_b \in \mathbb{R}^{N \times N}$. The hat matrix relates to Bob's data matrix $\boldsymbol{X}_b$ as follows:

$$\begin{aligned} \hat{\boldsymbol{Y}}_b &= \boldsymbol{H}_b \hat{\boldsymbol{E}}_a \\ \hat{\boldsymbol{Y}}_b &= \boldsymbol{X}_b (\boldsymbol{X}_b^T \boldsymbol{X}_b)^{-1} \boldsymbol{X}_b^T \hat{\boldsymbol{E}}_a \\ \hat{\boldsymbol{Y}}_b &= \boldsymbol{X}_b \boldsymbol{X}_b^+ \hat{\boldsymbol{E}}_a \end{aligned} \tag{5.12}$$

where $\boldsymbol{X}_b^+$ indicates the Moore-Penrose generalized inverse of $\boldsymbol{X}_b$ (Petersen & Pedersen, 2012).

Alice can compute the projection that Bob applies in each iteration $\boldsymbol{H}_b$ as follows:

$$\hat{\boldsymbol{H}}_b = \hat{\boldsymbol{Y}}_b \hat{\boldsymbol{E}}_a^+ \tag{5.13}$$

Across iterations, this minimum-norm solution $\hat{\boldsymbol{H}}_b$ performs the same projection as the true hat matrix of Bob. Using this projection, Alice can then create the data substitute $\boldsymbol{V}_b \in \mathbb{R}^{N \times P_b}$. For this, $\boldsymbol{V}_b$ should have the property

$\hat{\boldsymbol{H}}_b = \boldsymbol{V}_b\boldsymbol{V}_b^+$. Such a $\boldsymbol{V}_b$ has the same effect on the coefficient estimates of Alice that $\boldsymbol{X}_b$ does, because it generates the same predictions that Bob does:

$$\begin{aligned} \hat{\boldsymbol{Y}}_b &= \hat{\boldsymbol{H}}_b\hat{\boldsymbol{E}}_a \\ \hat{\boldsymbol{Y}}_b &= \boldsymbol{V}_b\boldsymbol{V}_b^+\hat{\boldsymbol{E}}_a \end{aligned} \tag{5.14}$$

There is no unique solution to decomposing $\hat{\boldsymbol{H}}_b$ into an $N \times P$ matrix $\boldsymbol{V}_b$ and its pseudoinverse. However, a numerically convenient $\boldsymbol{V}_b$ solution can be found as the first P_b eigenvectors of $\hat{\boldsymbol{H}}_b$. This is a convenient choice, because the columns of $\boldsymbol{V}_b$ are then orthogonal, meaning they also have the following property: $\boldsymbol{V}_b^+ = (\boldsymbol{V}_b^T\boldsymbol{V}_b)^{-1}\boldsymbol{V}_b^T = \boldsymbol{I}^{-1}\boldsymbol{V}_b^T = \boldsymbol{V}_b^T$. As follows from Equations 5.12 and 5.14, the $\boldsymbol{V}_b$ matrix relates to $\boldsymbol{X}_b$ by means of an unknown positive definite rotation matrix $\boldsymbol{V}_b = \boldsymbol{R}\boldsymbol{X}_b$ (Pavel, 2019).

By leveraging this similarity of $\boldsymbol{V}_b$ to $\boldsymbol{X}_b$, Alice can create an augmented data matrix of the following form: $\boldsymbol{Z}_a = [\boldsymbol{X}_a, \boldsymbol{V}_b]$. The augmented data matrix replaces the full data matrix in the computation of the asymptotic covariance matrix: $\boldsymbol{\Sigma}_\beta^{(a)} = \sigma^2(\boldsymbol{Z}_a^T\boldsymbol{Z}_a)^{-1}$. The partition of $\boldsymbol{\Sigma}_\beta^{(a)}$ belonging to $\boldsymbol{\beta}_a$ is then identical to its counterpart from the full data asymptotic covariance matrix $\boldsymbol{\Sigma}_\beta$ (for proof see Appendix C.1). The square root of its diagonal elements are thus the correct standard errors that would be obtained had the full data been available.

Alternative standard error procedures are available, e.g., profile likelihood methods or bootstrapping, but those require additional iterations of the main block coordinate descent algorithm. This yields additional information leakage and dramatically increases time requirements. Conversely, in the novel procedure we suggest here, both parties efficiently leverage the information in the existing iterations to compute standard errors without additional communication.

5.4 Privacy analysis for block coordinate descent

In this section, we analyze the information transfer within our protocol for privacy-preserving regression based on block coordinate descent. In line with previous work on this topic (e.g. Gambs et al., 2007; Gascón et al., 2017; Vaidya & Clifton, 2003, 2005; Vaidya et al., 2008), we take the viewpoint of semi-honest parties: *Alice* and *Bob* follow the protocol accurately, though they may be curious and aim to recover the other party's data. In this section, we aim to identify how well *Bob* can approximate *Alice*'s data using a *model inversion attack* (Fredrikson, Jha, & Ristenpart, 2015; Wang, Si, & Wu, 2015).

5.4.1 Information transfer in vertically partitioned regression

Information about features cannot only leak through dataset sharing, but also via sharing statistics based on this data. For example, a simple method for regression without explicitly sharing the full dataset is that by Karr et al. (2009), who compute the covariance matrix of $\boldsymbol{X}$ using secure inner-product methods and share it between *Alice* and *Bob*. This covariance matrix allows even a semi-honest *Alice* to (a) know how many features are used by *Bob* and – in the case of categorical predictors – know how many categories there are, (b) predict the values of the features held by *Bob* based on the values of the features held by *Alice*, (c) compute standard errors around this prediction, and (d) compute an R^2 value for this prediction. In other words, in a shared covariance matrix setting *Alice* can know up to a certain degree the values on each of *Bob*'s features for each row in the dataset, and *Alice* can know how good this prediction is. Moreover, each additional feature entered by *Alice* improves the prediction of features at *Bob* by definition.

Thus, sharing the full covariance matrix is undesirable for privacy-preserving regression. Newer methods (e.g., Du et al., 2004; Gascón et al., 2017) result in additive shares of $\text{cov}(\boldsymbol{X})$ at *Alice* and *Bob*, without either of them possessing the full covariance matrix. Afterwards, separate secure multi-party matrix inversion protocols or linear system solvers are used to compute the regression parameters according to Equation 5.2. This generally requires complex protocols involving multiple parties, but has been argued to be a secure procedure for obtaining parameter estimates for linear regression with vertically partitioned data. In these protocols, it is clear that information transfer does occur (because the full-data estimates are obtained) but its extent is not made explicit: it is unclear how the additive shares of the covariance matrix (the "statistics") relate to the collaborator's data – and thus it is unclear whether that data can be reconstructed.

Conversely, in our protocol the covariance matrix of the combined data is never explicitly computed. Our method uses a different "statistic": predictions $\hat{\boldsymbol{y}}$ over R iterations. Each of the R predictions are computed as follows by *Alice*:

$$\hat{\boldsymbol{y}}_a^{(r)} = \boldsymbol{X}_a \hat{\boldsymbol{\beta}}_a^{(r)} \tag{5.15}$$

This prediction vector is then sent to *Bob*: the main information transfer. In this protocol, how this information transfer relates to *Alice*'s data is thus explicit. As a result, clear conclusions can be made as to the potential for data recovery.

In the case where *Alice* enters only a single continuous feature in the analysis protocol, the information contained in $\hat{\boldsymbol{y}}_a$ is sufficient for *Bob* to reproduce the values of this feature up to a multiplicative constant: $\hat{\boldsymbol{y}}_a = \boldsymbol{x}_a \cdot \hat{\beta}_a$. With more than one feature per party, $\hat{\boldsymbol{\beta}}_a$ becomes a vector, meaning the problem of recovering the values of any feature at *Alice* is underidentified. Moreover, if the protocol is followed precisely, *Bob* does not know the number of features P entered into the model, meaning there is additional uncertainty about the values of $\boldsymbol{X}_a$ on the part of *Bob*. In its most basic form, the protocol is therefore fully secure for semi-honest parties against reconstruction of the privacy-sensitive data matrices.

5.4.2 Data reconstruction using shared metadata

In practice, there are many situations where the basic algorithm does not suffice and metadata about $\boldsymbol{X}_a$ should be shared with *Bob*. For example, to circumvent multicollinearity and non-convergence, none of the features entered into the model by *Alice* should be entered by *Bob*. Moreover, when distributing the model results is a goal of the analysis, it is relevant to investigate how sharing parameter estimates in addition to the predictions that are already shared leads to information transfer about the original data.

In our protocol, *Alice* sends R predictions to *Bob*. These individual predictions can be appended in a columnwise fashion to create an $N \times R$ matrix $\hat{\boldsymbol{Y}}_a = [\hat{\boldsymbol{y}}_a^{(1)}, \dots, \hat{\boldsymbol{y}}_a^{(R)}]$. Each prediction has an associated set of parameter estimates known only by *Alice* $\hat{\boldsymbol{\beta}}_a^{(r)}$, which can be combined in a similar way to create the matrix $\hat{\boldsymbol{B}}_a \in \mathbb{R}^{P \times R}$. These relate to the data matrix at *Alice* as follows:

$$\hat{\boldsymbol{Y}}_a = \boldsymbol{X}_a \hat{\boldsymbol{B}}_a \tag{5.16}$$

In our protocol, all of $\hat{\boldsymbol{Y}}_a$ is shared with *Bob*, and only the R^{th} column of $\hat{\boldsymbol{B}}_a$ – the final model result – is shared. Using these estimates, *Bob* can make a rank-1 minimum-norm approximation of the data held by *Alice*:

$$\hat{\boldsymbol{X}}_a^{(1)} = \hat{\boldsymbol{y}}_a^{(R)} \hat{\boldsymbol{\beta}}_a^{(R)+} \tag{5.17}$$

where $^+$ indicates the Moore-Penrose inverse (Petersen & Pedersen, 2012). We show empirically in Appendix C.2 that using this method with one set of shared parameter estimates reveals a proportion $1/P_a$ of the variance in the data to *Bob*. Only the combination of predictions and their associated parameter values allows (partial) model inversion and reconstruction of the partner's data.

Furthermore, as presented in Section 5.3.5, the predictions sent to and received from *Alice* can be used to create a minimum-norm approximation of the hat matrix of *Alice* – another statistic which is shared in our protocol. This hat matrix is shown in Appendix C.1 to not contain information about the features of *Alice* directly, but only about a rotation of this data such that the parameter estimates of *Bob* are adequately adjusted towards the conditional estimates.

In conclusion, the protocol is secure against reconstruction of the data in the case of semi-honest parties, and sharing of the final parameter estimates $\hat{\boldsymbol{\beta}}_a$ reveals a proportion $1/P_a$ of the variance in the data to the other data party.

5.4.3 Further privacy considerations

Purposeful attacks to recover data in the case of adversarial collaborations have not been analyzed. It is possible to design such an attack, but it is also possible to design safeguards against such attacks in the implementation of the protocol, for example based on the expected smoothness of the regression paths over iterations. We leave this analysis as a topic for further research.

In addition, because of the explicit link between the shared statistics and the original data, it is possible to limit the information shared with the collaborator in several ways. For example, in each iteration *Alice* may add noise to the computed parameter estimates or to the predictions sent to *Bob* – a technique from the differential privacy literature (Dwork, McSherry, Nissim, & Smith, 2006). Another method is to put an upper bound on the number of iterations based on the number of features in the data. This has two effects: (a) it shrinks (regularizes) the parameter estimates towards the marginal estimates and (b) it creates an upper bound ε on the information shared, depending on the allowed number of iterations.

In the next section, we show how our implementation of the BCD with vertically partitioned data performs in comparison to full-data generalized linear modeling (GLM) in simulated data as well as three real-world datasets.

5.5 Experiments

Our implementation of the BCD algorithm for vertically partitioned data is provided as an `R` package at `https://github.com/vankesteren/privreg`. Here, we use this implementation (version 0.9.5) to estimate models on both simulated data (Section 5.5.1) and real-world data with multiple parties from the UCI data repository (Section 5.5.2). Reproducible code for this section is available in the supplementary material to this paper.

5.5.1 Simulated data

The goal of this section is to compare our proposed privacy-preserving regression method to a benchmark method under controlled conditions. The benchmark method for these experiments is linear and logistic regression with a complete dataset, since the optimum privacy-preserving method would attain the same results with vertically partitioned data. For this section, data with multiple features and one target were simulated in the R programming language (R Core Team, 2018), with the following manipulations:

Target Either a normally distributed or a binomial target variable. In the case of the normal target, the R^2 was set to 0.5.

Dimensionality The total number of features was either 10, 50, or 100.

Covariance The covariance matrix of the features was had 1 on the diagonal and either 0.1 (low covariance) or 0.5 (high covariance) on all off-diagonal elements.

For each condition, 100 datasets were randomly generated. For the privacy-preserving regression method, the generated features were then equally distributed among *Alice* and *Bob*, after which the estimation was started. As a baseline comparison, a generalized linear model was estimated on the full dataset with all the features using the `glm()` function from the base `R stats` package. The exact data-generating mechanism, as well as the estimation method and hyperparameters can be found in the supplementary material.

The empirical convergence rates for the privacy-preserving regression method are shown in Figure 5.2. As expected from the work of X. Li et al. (2017), the number of iterations required increases linearly with the number of features. In addition, the high covariance leads to slower convergence due to the conditional estimates lying further away from the marginal estimates. As mentioned in Section 5.3.3, with no covariance the number of iterations would be 1.

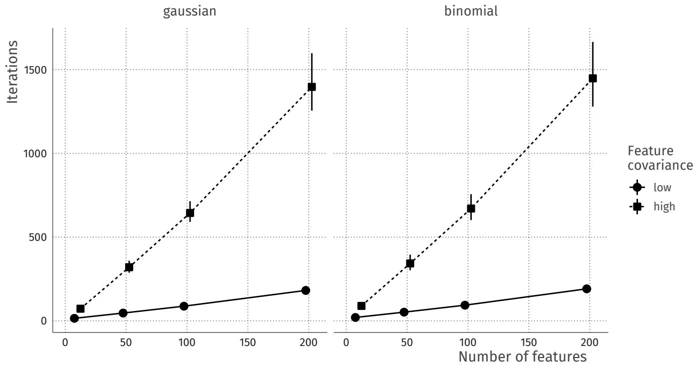

Figure 5.2 Observed amount of iterations required for convergence is approximately linear in the number of features and increases as there is more covariance between the features. Error bars indicate 95% simulation percentile intervals.

The obtained parameter estimates ($\hat{\boldsymbol{\beta}}$) of our method are equal to those found by the baseline comparison method in all simulated conditions, up to a computational tolerance in the convergence of the estimation algorithm (Figure 5.3). This lack of relative bias indicates that the proposed privacy-preserving regression approach performs as well as full-data generalized linear models, at least for the extent of these simulations.

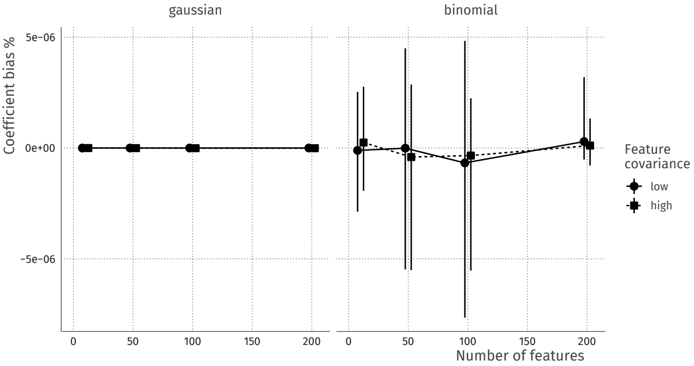

Figure 5.3 The parameter bias relative to the baseline GLM method is negligible for any number of features and feature covariance strength. Note the small y-axis range.

Standard errors indicate uncertainty in the dataset around the coefficient values, and they are the basis for statistical significance tests. Figure 5.4 shows the bias in the standard errors relative to the baseline GLM method for the different conditions. The figure shows that variation of this bias over different datasets increases with the number of features (larger error bars). In addition, there seems to be a very slight relative overestimation of the standard errors on average. This is due to slightly different convergence criteria and tolerances for both methods, which propagates through the standard error procedure (Section 5.3.5). Despite this, the standard error bias is overall small ($< 3\%$).

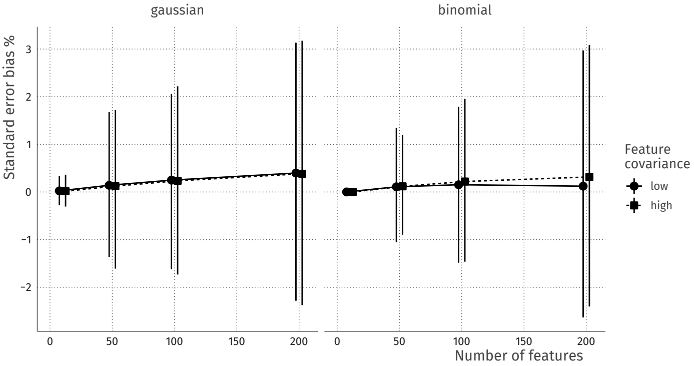

Figure 5.4 Standard error bias in percentage relative to the baseline GLM method. Variation across datasets increases with the number of features, and there is a very slight trend ($< 1\%$) towards overestimation of the standard error for larger datasets.

In conclusion, the simulations have shown that privacy-preserving regression using block coordinate descent on vertically partitioned data has equal performance to established regression methods on full data. However, in this section the data has been simulated to behave according to specification. In the next section, we compare the performance of these two methods on real-world datasets.

5.5.2 UCI datasets

In this section, we tested our proposed method on three different real-life data sets from the UCI (University of California at Irvine) Machine Learning repository (Blake & Merz, 1998). The datasets were chosen because they can be naturally partitioned into two sources, and their size and targets are different (Table 5.1). As before, the full preprocessing and analysis code for this section is available in the supplementary materials. Analyses were run on two separate computers (an Intel Core i7-8750H at 2.20 GHz and an Intel Xeon E5-2650 v4 at 2.20GHz) connected via a gigabit Ethernet connection on a university network.

Dataset	Features	Instances	Task	Parties
Forest fire	13	517	Regression	Weather & Fire dept.
HCC	49	165	Classification	Lab & Clinic
Diabetes	43	15 000	Classification	Clinic & Pharmacy

Table 5.1 Properties of the datasets used from the UCI machine learning repository after dataset cleaning and pre-processing. Code can be found in the supplementary materials.

5.5.2.1 Forest fires data

The forest fire data comes from the Montesinho natural park in Portugal (Cortez & Morais, 2007). It contains several weather observations by a meteorological station (e.g. wind speed, temperature, relative humidity, etc) as well as fire department risk assessments. In this dataset, the target is to predict the area of forest burned by a particular fire using the features from the aforementioned parties.

We performed linear regression where the target was log-transformed to normalize the residuals. Continuous features were standardized before they were entered into the analysis. The analysis took 450 BCD iterations in the privacy-preserving regression case. Including encryption and networking overhead, estimation took 14.51 seconds and computing standard errors took 0.61 seconds. Figure 5.5 shows that the coefficients and their 95% confidence intervals are equal for the full-data analysis and the privacy-preserving procedure. Several months show a significant positive effect on the log-area, meaning that – conditional on the ratings of the fire department – fires in these months (e.g., August and December) burn larger areas of forest.

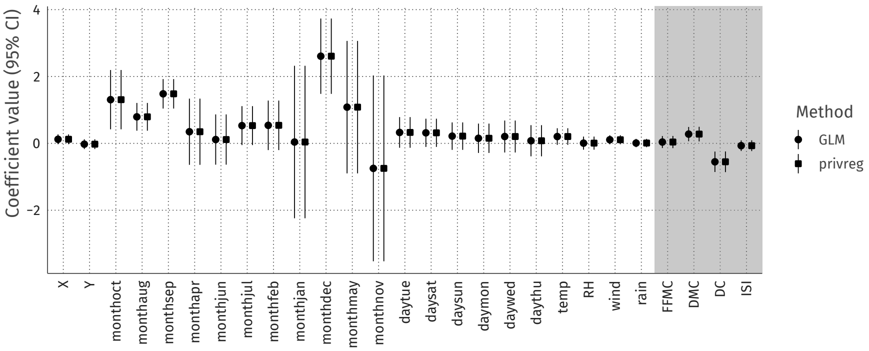

Figure 5.5 The coefficients and standard errors for the forest fire analysis are exactly the same for the GLM and our privacy-preserving regression estimation methods. The shading indicates data partitioning into the weather service (light) and fire department (dark).

5.5.2.2 Hepatocellular carcinoma data

This dataset was collected by Coimbra's Hospital and University Centre in Portugal for studying an epithelial cell cancer of the liver called hepatocellular carcinoma (HCC) (Santos, Abreu, García-Laencina, Simão, & Carvalho, 2015). It contains heterogeneous data on demographics, risk factors, laboratory and overall survival features from HCC patients. The goal of the analysis is to use lab results for a tissue sample as well as clinical data for the patient to predict survival after diagnosis. Since survival is a binary target, a binomial family GLM (logistic regression) was performed. For this analysis, continuous features were standardized before the analysis, which improved the convergence characteristics. The privacy-preserving GLM converged in 1636 iterations. Including encryption and networking overhead, estimation took 3 minutes and 16 seconds and computing standard errors took 0.63 seconds.

The results of the analysis (Figure 5.6) show that the estimates are exactly equal across the full-data and the privacy-preserving analyses, meaning survival probability predictions for new incoming patients based on these models will be the same. Despite slight deviations in the width of the confidence intervals, conclusions about the effects of the features on survival are also the same in this dataset.

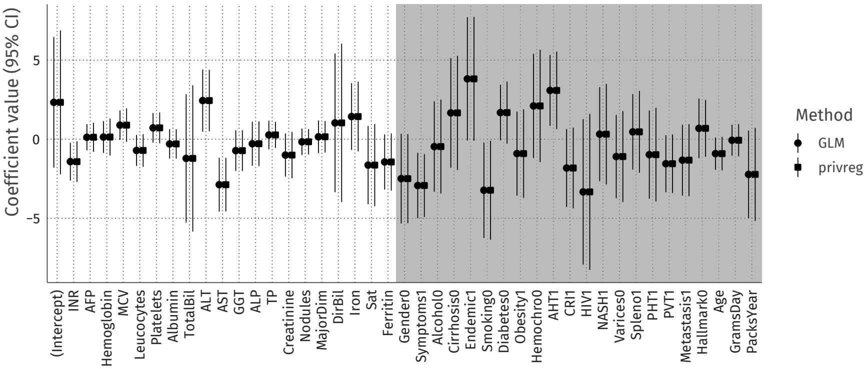

Figure 5.6 The coefficients and standard errors for the carcinoma analysis are very similar for the GLM and our privacy-preserving regression estimation methods. The shading indicates data partitioning into the lab results (light) and clinic (dark).

5.5.2.3 Diabetes

The diabetes dataset is an extract representing 10 years (1999-2008) of clinical diabetes care at 130 hospitals and integrated delivery networks throughout the United States (Strack et al., 2014). It is a large and also heterogeneous data set including encounter data (emergency, outpatient, and inpatient), provider speciality, demographics, laboratory data, pharmacy data, in-hospital mortality, and hospital characteristics. In this dataset, we predict readmission to the hospital using both administrative features and pharmaceutical features. To keep the computation of the standard errors for this analysis possible, 15000 patients were randomly selected from the dataset. Features were re-coded where necessary, and categorical features with only a single category in the sample were excluded from the analysis. The full pre-processing pipeline can be found in the supplementary material.

Since readmission is a binary target, a binomial family GLM (logistic regression) was performed. The diabetes data analysis required 284 iterations of the BCD algorithm. Including encryption and networking overhead, estimation took 1 minute and 37 seconds and computing standard errors took 42 seconds. This analysis is particularly interesting with respect to the effect of insulin (`insulinYes`) on the readmission probability. In the analysis of only the medication data, insulin has a significant positive effect on readmission (OR $= 1.20$, $p < .001$), whereas conditional on the administrative data, insulin significantly reduces the readmission probability (OR $= 0.88$, $p < .001$). This is a strong argument for including the data of both parties in the analysis.

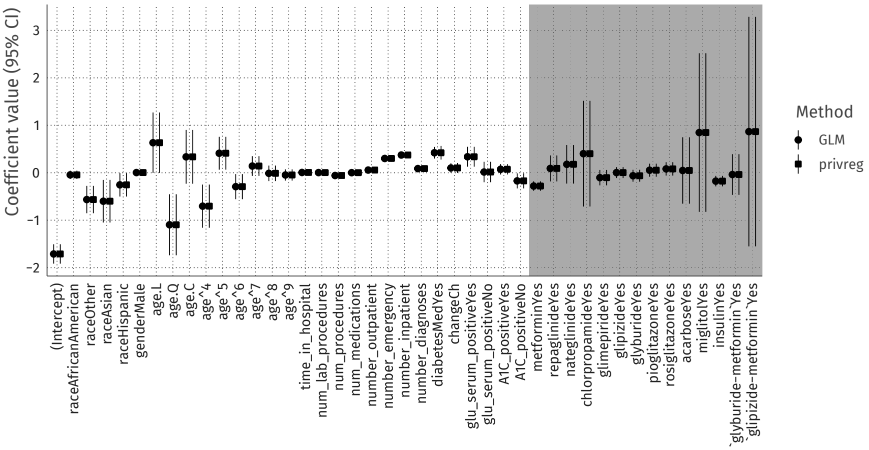

Figure 5.7 The coefficients and standard errors for the diabetes analysis are exactly the same for the GLM and our privacy-preserving regression estimation methods. The shading indicates data partitioning into the clinical data (light) and pharmaceutical data (dark).

In this section, we have shown that privacy-preserving regression using block coordinate descent is not only a theoretical possibility, but also a viable implementation of GLM for analyzing data with varied characteristics – both in simulated data under controlled conditions (Section 5.5.1) and in real-world prediction and analysis problems with various targets (Section 5.5.2). The time constraints on the real-world analyses are manageable, with all example analyses converging in under 4 minutes. We have shown that the parameter estimates exactly match those of the existing reference methods, and that our novel estimation method for the standard errors generally agrees with its full-data counterpart – and where it did not the difference was so small that it lead to the same conclusions in the analysis.

5.6 Discussion

In this paper, we have argued that block coordinate descent is a general method for estimating conditional parts of a generalized linear model (GLM) in a vertically partitioned data situation. Using this approach, two or more data parties can collaboratively estimate a GLM without sharing their features. This is useful when the features are not allowed to be shared, for example when there are privacy issues.

Our method falls within the category of federated learning algorithms. This means it can be implemented for situations when data mining is to be performed over remote devices or siloed data centers (T. Li et al., 2019), where aggregating the data tables is prohibitively expensive in terms of time, computation, or storage costs. This work aligns with several recent contributions that seek to exploit the privacy-preserving aspects of federated learning algorithms (see, e.g., Bonawitz et al., 2016; Geyer, Klein, & Nabi, 2017).

Due to the accessibility of our protocol and its similarity to existing regression estimation methods, extensions are relatively simple to implement. First and foremost, our framework can be extended to multiple parties as coordinate descent naturally extends to multiple blocks. In addition, our algorithm could include penalties for regularized estimation of the regression parameters through thresholding (Friedman et al., 2010). Through further research into combining coordinate descent with missing data methods such as full information maximum likelihood (Enders, 2001), our protocol could even be extended for a hybrid partitioning situation where data is both horizontally and vertically partitioned.

Our novel approach is a natural modification of the familiar linear modeling framework – without changes in the assumptions. We argue that our protocol restricts statistical information sharing as much as possible, while being explicit in how the shared information relates to the original data. Because of this, data parties know how much information they share, and the protocol could even incorporate methods from the differential privacy literature – such as additive noise or early stopping – to put a restriction on the amount of information shared with the partner institution (Dwork et al., 2006).

The main tradeoff of this flexibility compared to existing methods is relatively high communication cost: each iteration requires N prediction values to be sent to the partner institution. In addition, like other methods for this situation the block coordinate descent assumes (probabilistic) linkage of the individual records – both parties need to have their records in the same order. Lastly, this method is possible only when the target can be shared, although in absence of a shareable target collaborators could still perform some form of transfer learning, e.g., by predicting a shareable feature *related* to the true target.

Considering the prospect of these extensions and the availability of an accessible open-source implementation, we believe the proposed block coordinate descent protocol can be a springboard for future developments in the privacy-preserving data mining field.

Compliance with Ethical Standards

Funding: this work was supported by the Netherlands Organization for Scientific Research (NWO) under grant number 406.17.057 and by the Dutch National Research Agenda (NWA) under project number 400.17.605.

Conflict of Interest: The authors declare that they have no conflict of interest.

Acknowledgments: We thank Ayoub Bagheri for his helpful comments on an earlier version of this manuscript.

Chapter 6

Fair inference on error-prone outcomes

Boeschoten, L.[1], van Kesteren, E. J.[1], Bagheri, A., & Oberski, D. L. (2020, accepted). Fair inference on error-prone outcomes. *ECAI conference 2020 workshop: Artificial Intelligence for a Fair, Just and Equitable World.* arXiv:2003.07621.

Fair inference in supervised learning is an important and active area of research, yielding a range of useful methods to assess and account for fairness criteria when predicting ground truth targets. As shown in recent work, however, when target labels are error prone, potential prediction unfairness can arise from measurement error. In this paper, we show that, when an error-prone proxy target is used, existing methods to assess and calibrate fairness criteria do not extend to the true target variable of interest. To remedy this problem, we suggest a framework resulting from the combination of two existing literatures: fair ML methods, such as those found in the counterfactual fairness literature on the one hand, and, on the other, measurement models found in the statistical literature. We discuss these approaches and their connection resulting in our framework. In a healthcare decision problem, we find that using a latent variable model to account for measurement error removes the unfairness detected previously.

6.1 Introduction

Supervised learning is used to guide human decisions across a wide range of different fields. In sensitive areas such as healthcare or criminal justice, a key issue is that these decisions are equitable and fair. To this end, an active area of research investigates how fairness criteria can be incorporated into supervised learning (Berk, Heidari, Jabbari, Kearns, & Roth, 2018; Corbett-Davies, Pierson, Feller, Goel, & Huq, 2017; Dwork, Hardt, Pitassi, Reingold, & Zemel, 2012; Kleinberg, Mullainathan, & Raghavan, 2016; Kusner, Loftus, Russell, &

[1] Shared first authorship

Author contributions: LB and EJK wrote the largest part of the manuscript, DLO wrote the related work section, AB provided feedback throughout. LB, EJK, and AB prepared the data, EJK implemented the analysis and created the figures using feedback from LB.

Silva, 2017; Verma & Rubin, 2018). This literature has focused on supervised learning for a single objective, assumed to be the target variable of interest. Recently, however, Obermeyer, Powers, Vogeli, and Mullainathan (2019) observed that, even when substantial care has been taken to develop a prediction algorithm, unfairness in the predictions can still result due to *measurement error*. Intuitively, calibrating decisions to be fair for an error-prone proxy does not imply the decision is fair for the true variable of interest. Such effects can be substantial; for example, Obermeyer et al. (2019) demonstrated a large differential in predicted risk scores between black and white patients with equal values on a new proxy measurement. However, these authors did not suggest a method for dealing with this problem. This issue cannot be ignored because fairness is generally conceptualized on a level more abstract than the proxy label; for example, it is reasonable to require that fairness in a healthcare need prediction system should extend to a person's true health status.

This paper addresses the problem of prediction unfairness arising from measurement error. By considering the supervised learning problem at the level of a latent variable of interest, we reformulate the problem as one of adequate *measurement modeling*. In effect, instead of requiring perfect measurement to achieve fairness, we propose that researchers developing a prediction model to be used for decision-making collect several independent, possibly error-prone, measures of the variable of interest (e.g. health). We then suggest to combine measurement models from the statistical literature with techniques from the literature on fair ML to assess and ameliorate the problem of unfair predictions in the face of measurement error.

Our contributions are as follows:

- We illustrate that existing methods to examine unfairness in error-prone outcomes are insufficient;
- We suggest a framework, based on the existing measurement modeling literature, to investigate and ameliorate such issues;
- We perform an example analysis to demonstrate the suggested approach. In an existing healthcare application, this demonstrates that replacing one proxy with another does not lead to parity, while our approach does.

In Section 6.2, we provide a summary of basic concepts in fairness. In Section 6.3 prior approaches with respect to fair inference are discussed. In Section 6.4, the failure of these approaches is discussed when making use of proxies, and the proposed framework is introduced based on existing measurement models. In Section 6.5 the proposed framework is then applied to the exemplary data set provided by Obermeyer et al. (2019).

6.2 Problem definition

We consider probabilistic classification and regression problems with a set of features $\mathbf{X}$ and true outcome Y^*. Among the features, there is a sensitive feature $S \in \mathbf{X}$ (e.g. race, gender), with respect to which discriminatory predictions are to be avoided. Furthermore, although the prediction problem is with respect to the true outcome Y^* – e.g. "health" or "crime" – this outcome is not directly observed; instead, we have observed a set of error-prone proxy variables $\mathbf{Y}$. For example, in practice a proxy for "health", $Y \in \mathbf{Y}$, might be the costs of healthcare or the number of chronic conditions experienced by the patient, whereas, instead of "crime", the number of arrests might be measured. Following Nabi and Shpitser (2018), we represent the goal of the regression or classification problem as a query on the (generative) joint distribution $p(Y^*, \mathbf{X})$, potentially after conditioning on a set of "fixed" covariates $\mathbf{C}$, i.e. the (discriminative) conditional joint $p(Y^*, \mathbf{X}\backslash\mathbf{C} \mid \mathbf{C})$. Typically, this query will be the point prediction $\hat{Y}^* := E(Y^* \mid \mathbf{X})$.

Following standard social-scientific measurement theory (Borsboom, 2006), the fact that $\mathbf{Y}$ is a measurement proxy for Y^* is reflected by a *causal model*, in the sense of Pearl (2013); Spirtes, Glymour, Scheines, and Heckerman (2000), in which $Y^* \rightarrow \mathbf{Y}$, i.e. the true outcome is a common cause of all available proxy variables. Because Y^* is an unobserved latent variable, our causal model will be identifiable only through additional assumptions of conditional independence; we discuss these assumptions later. The key point to note here is that, generally, $E(Y^* \mid \mathbf{X}) \neq E(Y \in \mathbf{Y} \mid \mathbf{X})$, i.e. predictions using error-prone proxies as labels, $\hat{Y}$, will, of course, differ from the $\hat{Y}^*$ that would have been obtained had the true labels been available.

6.3 Related work

A large and growing literature on fairness of predictions for the error-free outcome Y^* exists, with divergent and sometimes mutually exclusive definitions of the notion of algorithmic fairness. An excellent overview of this literature can be found in Verma and Rubin (2018), which identified 20 separate definitions. Broadly, a distinction can be made between statistical metrics, distance-based measures, and causal reasoning (Verma & Rubin, 2018).

Statistical metrics define fairness as the presence or absence of a (conditional) independence in the joint distribution $p(Y^*, \hat{Y}^*, S)$. For example, take a classification problem in which the decision is taken as $d := I(\hat{Y}^* > \tau)$, where I is the indicator function and τ is some threshold on the predicted score. *Statistical parity* ("group fairness") is then defined as $p(d = 1 \mid S = s) = p(d = 1 \mid S = s')$ for all $s \neq s'$, i.e., the decision should not depend on the sensitive attribute, whereas *predictive parity* is defined as $p(Y^* = 1 \mid d = 1, S = s) = p(Y^* = 1 \mid d = 1, S = s')$ for all $s \neq s'$—i.e. the positive predictive value should not depend on the sensitive attribute. Further definitions include conditional statistical parity (Corbett-Davies et al., 2017), overall accuracy equality (Berk et al., 2018), and well calibration (Kleinberg et al., 2016).

Distance-based measures of fairness account for the non-sensitive predictors $\mathbf{X}\backslash S$, in addition to the observed and predicted outcomes and sensitive attribute. The well-known "fairness through awareness" framework (Dwork et al., 2012) generalises several of the preceding notions, such as statistical parity, by defining fairness as "similar decisions for similar people". Consider a population of potential applicants P, and consider any randomised output from the prediction algorithm, $M(x \in P)$. Fairness is achieved whenever the distance among the decisions M made for two people is at least as small as the distance between these people, i.e. when $D(M(x), M(y)) \leqslant d(x, y)$ for any $x, y \in P$. Here, D and d are arbitrary metrics on the distance between outputs and people, respectively. Careful choice of these metrics can yield some of the above definitions as special cases. Since the fairness condition can be trivially achieved, for example by always outputting a constant regardless of the input, the prediction model should be trained by minimising a loss function under the above constraint.

Finally, in recent years, results from the causal modelling literature have been leveraged to define and achieve "counterfactual" fairness (Kusner et al., 2017; Nabi & Shpitser, 2018). In these definitions one first considers a causal model involving Y, $X \backslash S$, and S such as Panel A of Figure 6.1. This causal model then induces a counterfactual distribution $p_{do(s)}(\hat{Y}^* \mid X)$, i.e. the distribution we would observe if S were *set* to the value s (Pearl, 2013). Kusner et al. (2017) then defined counterfactual fairness as $p_{do(s)}(\hat{Y}^* \mid X) = p_{do(s')}(\hat{Y}^* \mid X)$. Note that this definition looks superficially similar to the definition of statistical parity (group fairness), but is distinct because it refers to an individual. This definition has as a disadvantage that *any* causal effect of the sensitive attribute on the prediction is deemed illegitimate. Based on the same framework, Nabi and Shpitser (2018) suggested a more general definition: some causal *pathways* originating in S are denoted discriminatory, while others are not. Fairness is then achieved by performing inference on a distribution $p^*(Y^*, \mathbf{X})$, in which the "fair world" distribution $p^*(Y^*, \mathbf{X})$ is close in a Kullback-Leibler sense to the original $p(Y^*, \mathbf{X})$, but all discriminatory pathways have been blocked (up to a tolerance) using standard causal inference techniques. Note that, if all causal pathways originating in S are deemed discriminatory and the tolerance set to zero, the counterfactual fairness criterion by Kusner et al. (2017) will be satisfied.

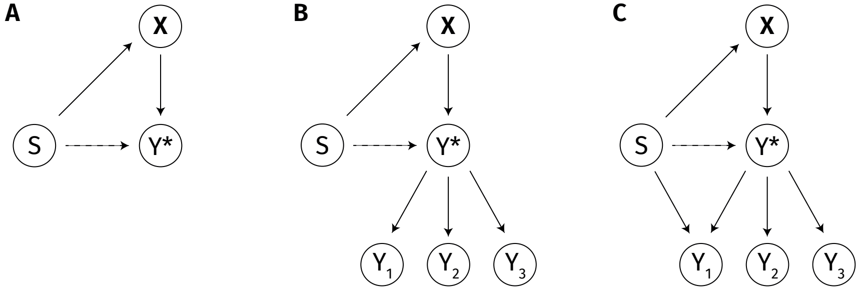

Figure 6.1 Graphical representation of causal relations between the sensitive feature (S), the predictors ($\mathbf{X}$), and the error-prone outcome ($\hat{Y}*$) in the naive case (A), in the measurement error framework (B), and in the measurement error framework with differential item functioning on the Y_1 proxy (C). The dotted arrow indicates the discriminatory causal pathway (as in Nabi and Shpitser (2018)) which is blocked when performing fair inference, evaluating $E[Y* \mid \mathbf{X}, S]$ to compute a risk score $\hat{Y}*$.

6.4 Proposed framework

6.4.1 Fair inference in error-prone outcomes

The existing methods from Section 6.3 do not consider the target Y^* to be error-prone. However, in practice, the target feature $Y \in \mathbf{Y}$ in the data set is not a perfect representation of the true underlying outcome Y^*. There can be several sources for this imperfect representation. For example, the true underlying outcome of interest may not be directly measurable at all (i.e., $Y^* \neq Y$ for any possible Y). In this case, the outcome of interest will only partially explain any feature used as its proxy. For example, in using healthcare costs Y as a proxy for health Y^*, the observed value will in part be determined by other factors besides Y^*, such as the location of residence of the patient. Then, even if the outcome of interest were “true healthcare costs” – thus in principle measurable – the observed feature will in practice still not be an infallible proxy, because health records are never perfect observations and always contain some form of noise (Brakenhoff et al., 2018). Together, such sources of noise in the observation process are termed “measurement error”, and any outcome Y^* containing measurement error can be considered *latent* (Borsboom, 2008) and modelled as such.

Crucially, the presence of measurement error may result in unfair inferences for the error-prone outcome, even after applying the procedures presented in Section 6.3 to account for unfairness. This is shown in a compelling example by Obermeyer et al. (2019), who concluded that commercial algorithms used by insurance companies for patient referral contain a fundamental racial bias. In the algorithm under consideration, healthcare costs $Y \in \mathbf{Y}$ are used as a proxy for health Y^*. Obermeyer et al. (2019) illustrated that although there is no bias in healthcare costs, there is strong racial bias in other proxies of health such as whether patients have chronic conditions. Specifically, in order to be referred to a primary care physician, the true underlying health status Y^* of black patients was worse than that of white patients.

Obermeyer et al. (2019) concluded that fair inference requires selecting a better proxy for health as the outcome variable Y^*. Indeed, their analyses were possible precisely due to the availability of different proxies of health, such as the number of chronic conditions. However, we note that solving racial bias in a new proxy does not guarantee the absence of racial bias in other proxies indicating other aspects of health. Instead, here we suggest incorporating several proxies, or *indicators* $\mathbf{Y}$ in a measurement model for the unobserved, error-prone outcome Y^* (Kilbertus et al., 2017). In the next section, we introduce the existing literature on measurement models and its approach to fair inference.

6.4.2 Fair inference in measurement models

When outcomes are thought to be error-prone, an existing literature suggests the use of measurement models (Brakenhoff et al., 2018; Fuller, 2009). At their core, measurement models describe the causal relationship between observed scores $\mathbf{Y}$ and error-prone unobserved "true scores" Y^* as $Y^* \rightarrow \mathbf{Y}$. A measurement model adequately represents the empirical conditions of measurement if conditional independence can be assumed (Blalock & Blalock, 1968). More specifically, measurement models assume that Y_1 and Y_2 are conditionally independent given Y^* (i.e., $p(Y_1, Y_2 \mid Y^*) = p(Y_1 \mid Y^*)p(Y_2 \mid Y^*)$). A plethora of variations of measurement models assuming conditional independence have been developed, such as latent class models (McCutcheon, 1987), item response models (Rasch, 1993), mixture models (McLachlan & Basford, 1988), factor models (Lord, 2012), structural equation models (Bollen, 1989), and generalized latent variable models (Skrondal & Rabe-Hesketh, 2004).

Measurement models are suggested here as a convenient way to account for a latent variable's relationship to sensitive features. The measurement error of a proxy variable (e.g. Y_1) is then assumed to differ over different groups of S. To account for group differences in proxy variables, a large body of literature is available where this issue is known under different labels. Generally, these approaches are applied within the structural equation modelling (SEM) framework (Jöreskog, 1993), as SEM explicitly separates the measurement model ($Y^* \rightarrow \mathbf{Y}$) from the structural model ($\mathbf{X} \rightarrow Y^*$). Approaches for investigating how features S influence Y^* are investigating item bias (Mellenbergh, 1989), Differential Item Functioning (DIF) (Holland & Wainer, 1993) and measurement invariance (Schmitt & Kuljanin, 2008). For an extensive overview of the different approaches and their benefits and drawbacks, we refer to (Flore, 2018; Schmitt & Kuljanin, 2008; Steenkamp & Baumgartner, 1998; Vandenberg & Lance, 2000).

6.4.3 Proposed method for fair inference on latent variables

We propose our framework for fair inference on outcomes which are measured only through error-prone proxies. To clarify the framework and make it more comparable to earlier work, we use the running example of health risk score prediction from Obermeyer et al. (2019). Their healthcare data set contains several clinical features $\mathbf{X}$ at time point $t-1$ (e.g., age, gender, care utilisation, biomarker values and comorbidities) which are used to predict healthcare cost Y^* at time t. In addition, the patient's race is the sensitive feature S, coded as $S = b$ for black patients and $S = w$ for white patients. The relations between these features are shown in panel A of Figure 6.1.

Based on $\mathbf{X}$, the expectation of a persons' healthcare cost is used as a risk score $\hat{Y}^* := E[Y^* \mid \mathbf{X}, S]$. The risk score is used to make a decision D to refer a patient to their primary care physician to consider program enrolment. More specifically $d = 1$ if $\hat{Y}^*$ is above the 55^{th} percentile. In this setting, attributes $\mathbf{X}$ can be legitimately controlled. However, conditional on $\mathbf{X}$ both groups in S should have equal probability of being referred: $\mathrm{P}(d = 1 \mid \mathbf{X} = x, S = b) = \mathrm{P}(d = 1 \mid \mathbf{X} = x, S = w)$. As mentioned in Section 6.4.1 and shown by Obermeyer et al. (2019), this procedure leads to bias in other proxies of Y^*, such as a patient's number of chronic conditions.

Our proposed framework is a SEM implementation of the second and third panels of Figure 6.1. The general structure of the model is that of a Multiple Indicator, Multiple Causes (MIMIC) model: the outcome variable Y^* (e.g., health) has multiple proxy indicators (e.g., chronic conditions, healthcare costs, hypertension), and the $\mathbf{X}$ features predict Y^* directly (thus the proxies only indirectly). A graphical representation of the MIMIC SEM model is shown in Figure 6.2. This implementation imposes additional assumptions on the general causal graphs, most notably linear relationships between the variables and the multivariate Gaussian residuals.

Fair inference on Y^* can be performed in the following way: during estimation of the regression parameters ($\mathbf{X} \to Y^*$), health is conditioned on race, but during prediction the path from Race to Health is blocked by setting $S = b$. Following the notation of Nabi and Shpitser (2018), this yields a "fair world" distribution $p^*(Y^*, \mathbf{X})$. The expectation $\hat{Y}^* = E[Y^* \mid \mathbf{X}, S]$ is then computed from this distribution, meaning for two participants who differ only on S but not on $\mathbf{X}$, the risk score $\hat{Y}^*$ will be exactly the same. Because the latent outcome Y^* is modelled as a linear combination of the different proxies, the risk score is a reflection of the underlying health rather than only health cost.

6.5 Experiments

In this section, we evaluate the proposed framework on an application of the procedures discussed in this paper. We first prepare the data set as provided by Obermeyer et al. (2019) to create a basic risk score based on healthcare cost similar to the commercial risk score reported in their paper. Then, we illustrate our argument from Section 6.4.1: we perform fair inference on the

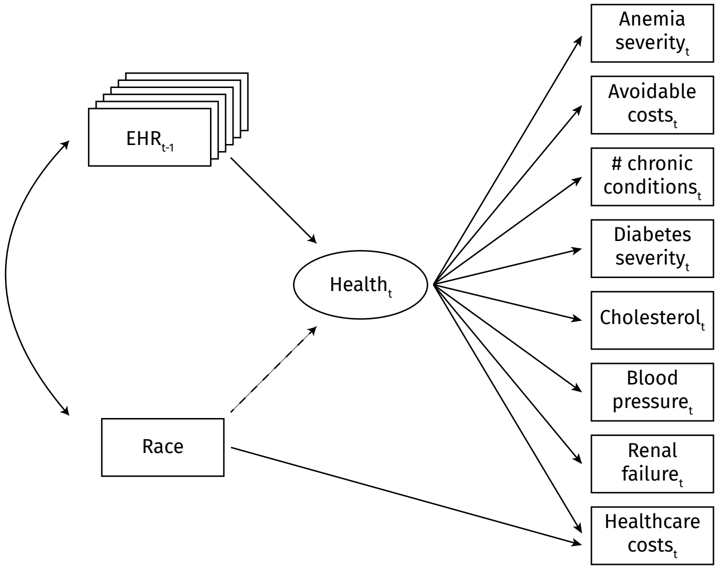

Figure 6.2 Structural equation model for the proposed framework on the healthcare data set. For clarity, residual variances of the endogenous variables are not drawn in the diagram. For more information on the variables used in the model, see Obermeyer et al. (2019).

proxy measure for health (healthcare cost) to show that this does not solve the issue of unfairness in other proxy measures. This is a reproduction of the results shown by Obermeyer et al. (2019). Next, we use the SEM framework from Section 6.4.3 to show how including a formal measurement model for Y^* – as in panel B of Figure 6.1 – can largely solve the issue of unfairness in the proxies. Last, we show how existing differential item functioning (DIF) methods in the SEM framework – panel C of Figure 6.1 – can aid in interpreting the extent to which proxy measures contain unfairness. Fully reproducible R code for this section is available as supplementary material to this paper at the following DOI: `10.5281/zenodo.3708150`.

6.5.1 Data preparation and feature selection

Log-transformations are applied to highly skewed variables at time-point t, such as costs, to meet the assumption of normally distributed residuals in regression procedures. As an additional normalisation step, the predictors at time-point $t-1$ are re-scaled to homogenise their levels of variance. The data set is then split into a training and a test set. In this section, estimation is always done on the training set and inference is done on the test set.

To simplify our proposed framework for the purpose of this application, we select a subset of features at time-point $t-1$ for prediction of the target of interest at time point t, health. We want our procedure to be comparable to the commercial algorithm which produces the risk scores described in Obermeyer et al. (2019). If the features we select are the same features used by the commercial algorithm, then our procedure would yield very similar results upon generating a risk score. Unfortunately, the predicted risk scores used by Obermeyer et al. (2019) cannot be replicated exactly using the provided data set.

To select the subset of predictor features for further use in our procedure, we performed a LASSO regression (Tibshirani, 1996) where all available features at time-point $t-1$ are used as predictor variables, and the provided algorithmic risk score at time-point t is used as a target. Following the guidelines by Hastie, Tibshirani, and Friedman (2009), we used cross-validation to select the optimal λ penalty value. This yields a set of non-zero predictors which predict the algorithmic risk score well.

Spearmans rank correlation between the commercial and the replicated risk score is high $\rho = .82$, indicating that the commercial and replicated risk scores perform similarly in the rank-based cutoff applied in Obermeyer et al. (2019). The predictors selected in this model are used as predictors $\boldsymbol{X}$ in the structural equation models of the following sections.

6.5.2 Fair inference on cost as a proxy of health

Pane A of Figure 6.1 illustrates conditional statistical parity as defined by Verma and Rubin (2018). Here, the outcome Y^* is conditioned on sensitive feature S when estimating the coefficients of the prediction model ($\boldsymbol{X} \rightarrow Y^*$),

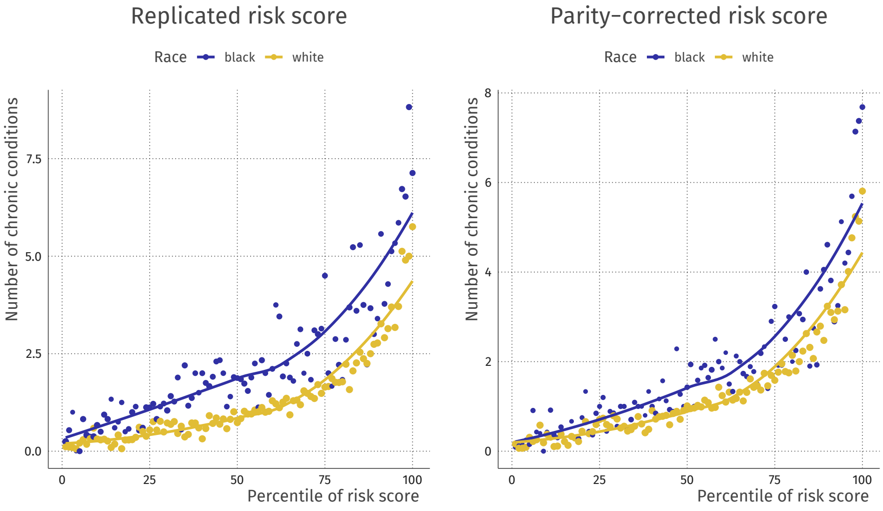

Figure 6.3 Effect of parity correction in one proxy of health (healthcare cost) on the race differences in another proxy of health (the number of chronic conditions). From the replicated risk score to the parity-corrected risk score, the cross-race difference becomes slightly smaller but does not disappear.

and during prediction all subjects are assumed to have the same level of S, e.g., $S = b$, such that $\mathrm{P}(Y^* = y^* \mid \mathbf{X} = x, S = b) = \mathrm{P}(Y^* = y^* \mid \mathbf{X} = x, S = w)$.

Accounting for sensitive feature ‘Race’ by conditioning the outcome ‘Replicated Risk score’ on ‘Race’ when estimating the model and being excluded during prediction reduces the extent of the problem. Figure 6.3 illustrates that although the results improve compared to not including ‘Race’ at all, conditional statistical parity is still not met. As a consequence, individuals belonging to $S = b$ will still have a lower health status when being selected for intervention.

6.5.3 Fair inference on latent health

A cause for the fact that conditional statistical parity is not met when following Pane A of Figure 6.1 can be that $\hat{Y}^*$ is a (bad) proxy. Instead of using one bad proxy, it is better to use multiple (bad) proxies as indicators of an unobserved latent variable measuring ‘true health’. How such a model can be specified is illustrated in Pane B of Figure 6.1. Similarly to Verma and Rubin (2018), the sensitive feature is excluded during prediction.

Figure 6.4 shows the effect of including a measurement model in constructing risk scores. This figure illustrates that using a measurement model with multiple imperfect measurements of health as indicators for ‘true health’ substantially improves conditional statistical parity. The improvement has been more compared to accounting for the sensitive feature. By using this mea-

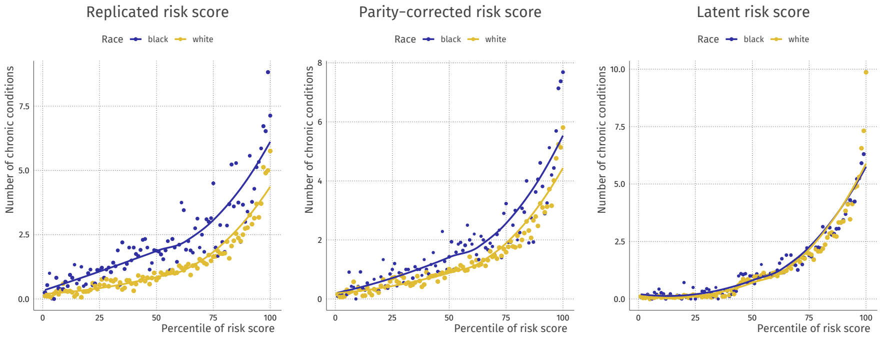

Figure 6.4 Effect of including a measurement model in constructing risk scores. The first panel shows the uncorrected risk score based on healthcare cost, the middle panel shows the same risk score but corrected for the sensitive feature, and the third panel shows the corrected risk score based on the latent health outcome using a measurement model.

surement model, the problem that individuals belonging to $S = b$ had a lower health status when being selected for intervention is minimised.

6.5.4 Investigating unfairness in proxies

When using a measurement model with multiple imperfect measurements of health as indicators of 'true health', differences in measurement error over the different groups of the sensitive feature can still be present. Panel C of Figure 6.1 illustrates how differences over the sensitive feature groups in the error prone indicator variables can be incorporated directly when estimating 'true health'. For example, differences in measurement error of healthcare cost can be present for the different groups of Race.

Including a DIF parameter δ on the healthcare cost variable yields a model which fits significantly better on the test set than the model without the DIF parameter ($\chi^2(1) = 50$, $p < 0.001$). The value of the DIF parameter on cost is estimated as $\delta = 0.198$ (95% CI = $[0.172, 0.225]$). This means that for the same level of health, the log-healthcare costs of the white race class in this data set is estimated to be 0.198 higher. This means that the cost of healthcare for white patients is $(e^{0.198} - 1) \cdot 100\% = 21.9\%$ higher than that for black patients, *given an equal level of health* as measured by the measurement model (95% CI $= [18.7, 25.2]$).

Applying the same procedure to the other indicators leads to estimates of DIF for those indicators. The results are shown in Table 6.1. This table shows that some proxies have stronger DIF than others, meaning some proxies are more unfair than other proxies. Notable, the avoidable healthcare cost and the renal failure items have low levels of DIF for Race, whereas the healthcare cost and the number of active chronic conditions have strong DIF.

Table 6.1 Estimated differential item functioning parameters for each indicator (proxy) of health. δ parameters should be interpreted as the mean deviation of the black patients compared to the white patients given health.

Indicator	δ	2.5%	97.5%
No. active chronic conditions	0.453	0.364	0.541
Mean blood pressure	-0.262	-0.320	-0.204
Diabetes severity (HbA1c)	-0.343	-0.391	-0.296
Anemia severity (hematocrit)	0.250	0.231	0.268
Renal failure (creatinine)	-0.019	-0.025	-0.014
Cholesterol (mean LDL)	-0.235	-0.317	-0.153
Healthcare cost (log)	0.198	0.172	0.225
Avoidable healthcare cost (log)	-0.052	-0.096	-0.008

6.6 Conclusion

In this paper, we have argued that when measurement error is at play, performing fair inference on a proxy measure of the outcome is insufficient to achieve a fair inference on the true outcome. This manifests itself, as shown in Obermeyer et al. (2019), as unfairness in other proxy measures of the outcome of interest. Alternatively, in this study we proposed to make use of existing measurement models containing multiple error-prone proxies for the outcome of interest. In addition, fair inference can be accounted for in each of these proxies simultaneously if needed by allowing for measurement error in proxies to differ over groups of a sensitive feature. We provided a framework to perform these estimations and applied this framework to the exemplary data set provided by Obermeyer et al. (2019). Here, it was concluded that fair inference was accounted for when multiple proxies were used in a measurement model instead of a single proxy. Additionally accounting for differences in measurement error over race groups was not needed to further improve fairness in predicted risk scores, although substantive group differences were found for some proxies.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … Isard, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283).

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. In *Acm sigmod record* (Vol. 29, pp. 439–450).

Alexander-Bloch, A., Giedd, J. N., & Bullmore, E. (2013). Imaging structural co-variance between human brain regions. *Nature Reviews Neuroscience*, *14*(5), 322.

Amirbekyan, A., & Estivill-Castro, V. (2007). Privacy-preserving regression algorithms. In *Proceedings of the 7th wseas international conference on simulation, modelling and optimization* (pp. 37–45).

Ammerman, B. A., Serang, S., Jacobucci, R., Burke, T. A., Alloy, L. B., & McCloskey, M. S. (2018). Exploratory analysis of mediators of the relationship between childhood maltreatment and suicidal behavior. *Journal of adolescence*, *69*, 103–112.

Ancker, J. S., Kim, M.-H., Zhang, Y., Zhang, Y., & Pathak, J. (2018). The potential value of social determinants of health in predicting health outcomes. *Journal of the American Medical Informatics Association*, *25*(8), 1109–1110.

Asparouhov, T., Hamaker, E. L., & Muthén, B. (2018). Dynamic structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(3), 359–388.

Asparouhov, T., & Muthén, B. (2009). Exploratory structural equation modeling. *Structural equation modeling: a multidisciplinary journal*, *16*(3), 397–438.

Asparouhov, T., & Muthén, B. (2016). Structural equation models and mixture models with continuous nonnormal skewed distributions. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(1), 1–19.

Athey, S., Imbens, G. W., & Wager, S. (2018). Approximate residual balancing: debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *80*(4), 597–623.

Atlas, L. Y., Lindquist, M. A., Bolger, N., & Wager, T. D. (2014, aug). Brain mediators of the effects of noxious heat on pain. *Pain*, *155*(8), 1632–1648. doi: 10.1016/j.pain.2014.05.015

Baron, R. M., & Kenny, D. a. (1986). The Moderator-Mediator Variable Distinction in Social The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations. *Journal of Personality and Social Psychology*, *51*(6), 1173–1182. doi: 10.1037/0022-3514.51.6.1173

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, *68*(3), 255–278.

Baskin-Sommers, A. R., Neumann, C. S., Cope, L. M., & Kiehl, K. A. (2016). Latent-variable modeling of brain gray-matter volume and psychopathy in incarcerated offenders. *Journal of abnormal psychology*, *125*(6), 811.

Basten, U., Hilger, K., & Fiebach, C. J. (2015). Where smart brains are different: A quantitative meta-analysis of functional and structural brain imaging studies on intelligence. *Intelligence*, *51*, 10–27.

Bates, D., & Maechler, M. (2017). *Matrix: Sparse and dense matrix classes and methods.*

Bede, P., Elamin, M., Byrne, S., McLaughlin, R., Kenna, K., Vajda, A., … Hardiman, O. (2015). Patterns of cerebral and cerebellar white matter degeneration in als. *J Neurol Neurosurg Psychiatry*, *86*(4), 468–470.

Bengio, Y. (2000). Gradient-based optimization of hyperparameters. *Neural computation*, *12*(8), 1889–1900.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., & Roth, A. (2018). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 0049124118782533.

Bernaards, C. A., & Jennrich, R. I. (2005). Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, *65*, 676–696.

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods* (Vol. 23). Prentice hall Englewood Cliffs, NJ.

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*.

Bishop, D. V. (2013). Cerebral asymmetry and language development: cause, correlate, or consequence? *Science*, *340*(6138), 1230531.

Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases, 1998.*

Blalock, H. M., & Blalock, A. B. (1968). *Methodology in social research.*

Bloom, J. M. (2019). *Secure multi-party linear regression at plaintext speed.*

Boca, S. M., Sinha, R., Cross, A. J., Moore, S. C., & Sampson, J. N. (2014). Testing multiple biological mediators simultaneously. *Bioinformatics*, *30*(2), 214–220. doi: 10.1093/bioinformatics/btt633

Bock, R. D., & Bargmann, R. E. (1966). Analysis of covariance structures. *Psychometrika*, *31*(4), 507–534.

Bollen, K. A. (1989). *Structural equations with latent variables.* John Wiley & Sons.

Bollen, K. A., & Curran, P. J. (2006). *Latent curve models: A structural equation perspective* (Vol. 467). John Wiley & Sons.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., … Seth, K. (2016). *Practical secure aggregation for federated learning on user-held data.*

Borsboom, D. (2006). When does measurement invariance matter? *Medical care*, *44*(11), S176–S181.

Borsboom, D. (2008). Latent variable theory. *Measurement: Interdisciplinary Research and Perspectives*, *6*(1-2), 25-53. doi: 10.1080/15366360802035497

Brakenhoff, T. B., Mitroiu, M., Keogh, R. H., Moons, K. G., Groenwold, R. H., & van Smeden, M. (2018). Measurement error is often neglected in medical literature: a systematic review. *Journal of clinical epidemiology*, *98*, 89–97.

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. doi: 10.1037/met0000090

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

Braun, M. (2018). *sparseMVN: Multivariate normal functions for sparse covariance and precision matrices.*

Breheny, P., & Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, *5*(1), 232–253. doi: 10.1214/10-AOAS388

Breiman, L. (2001a, Oct 01). Random forests. *Machine Learning*, *45*(1), 5–32. doi: 10.1023/A:1010933404324

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

Brown, T. A. (2006). *Confirmatory factor analysis for applied research.*

Browne, M. W. (1974). Generalized least squares estimators in the analysis of covariance structures. *South African Statistical Journal*, *8*(1), 1–24.

Browne, M. W., & Cudeck, R. (1989). Single sample cross-validation indices for covariance structures. *Multivariate behavioral research*, *24*(4), 445–455.

Campbell, D. T., & Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological bulletin*, *56*(2), 81.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., … Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, *76*(1).

Casella, G., & Berger, R. L. (2002). *Statistical inference* (Vol. 2). Duxbury Pacific Grove, CA.

Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, *25*(1847), 536–538.

Cernat, A., & Oberski, D. L. (2019). Extending the within-persons experimental design: The multitrait-multierror (MTME) approach. In P. J. Lavrakas, M. W. Traugott, C. Kennedy, A. L. Holbrook, & E. de Leeuw (Eds.), *Experimental methods in survey research: Techniques that combine random sampling with random assignment.* New York: John Wiley & Sons.

Chen, F., Bollen, K. A., Paxton, P., Curran, P. J., & Kirby, J. B. (2001). Improper solutions in structural equation models: Causes, consequences, and strategies. *Sociological methods & research*, *29*(4), 468–508.

Chén, O. Y., Crainiceanu, C., Ogburn, E. L., Caffo, B. S., & Wager, T. O. R. D. (2017). High-dimensional multivariate mediation with application to neuroimaging data. *Biostatistics*(September), 1–16. doi: 10.1093/biostatistics/kxx040

Choi, J., Oehlert, G., & Zou, H. (2010). A penalized maximum likelihood approach to sparse factor analysis. *Statistics and its Interface*, *3*(4), 429–436.

Cole, D. A., Ciesla, J. A., & Steiger, J. H. (2007). The insidious effects of failing to include design-driven correlated residuals in latent-variable covariance structure analysis. *Psychological methods*, *12*(4), 381.

Colibazzi, T., Zhu, H., Bansal, R., Schultz, R. T., Wang, Z., & Peterson, B. S. (2008). Latent volumetric structure of the human brain: exploratory factor analysis and structural equation modeling of gray matter volumes in healthy children and adults. *Human brain mapping*, *29*(11), 1302–1312.

Collobert, R., Bengio, S., & Mariéthoz, J. (2002). *Torch: a modular machine learning software library* (Tech. Rep.). Rue Marconi 19 CH - 1920 Martigny Switzerland: Idiap.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. (2017). Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 797–806).

Cortez, P., & Morais, A. d. J. R. (2007). A data mining approach to predict forest fires using meteorological data. In *Proceedings of 13th portuguese conference on artificial intelligence* (pp. 512–523). Associação Portuguesa para a Inteligência Artificial (APPIA).

Cox, S. R., Harris, M. A., Ritchie, S. J., Buchanan, C. R., Hernández, M. d. C. V., Corley, J., … Whalley, H. C. (2020). Three major dimensions of human brain cortical ageing in relation to cognitive decline across the 8th decade of life. *BioRxiv*.

Cox, S. R., Ritchie, S. J., Tucker-Drob, E. M., Liewald, D. C., Hagenaars, S. P., Davies, G., … Deary, I. J. (2016). Ageing and brain white matter structure in 3,513 uk biobank participants. *Nature communications*, *7*(1), 1–13.

Cudeck, R., Klebe, K. J., & Henly, S. J. (1993). A simple gauss-newton procedure for covariance structure analysis with high-level computer languages. *Psychometrika*, *58*(2), 211–232.

de Mooij, S. M., Henson, R. N., Waldorp, L. J., & Kievit, R. A. (2018). Age differentiation within gray matter, white matter, and between memory and white matter in an adult life span cohort. *Journal of Neuroscience*, *38*(25), 5826–5836.

De Munck, J. C., Huizenga, H. M., Waldorp, L. J., & Heethaar, R. (2002). Estimating stationary dipoles from meg/eeg data contaminated with spatially and temporally correlated background noise. *IEEE Transactions on Signal Processing*, *50*(7), 1565–1572.

Denenberg, V. H., Kertesz, A., & Cowell, P. E. (1991). A factor analysis of the human's corpus callosum. *Brain research*, *548*(1-2), 126–132.

Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., … Hyman, B. T. (2006). An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, *31*(3), 968–980.

Dobriban, E., & Sheng, Y. (2018). *Distributed linear regression by averaging.*

Dobson, A. J., & Barnett, A. G. (2008). *An introduction to generalized linear models.* Chapman and Hall/CRC.

Du, W., & Atallah, M. J. (2001). Privacy-preserving cooperative scientific computations. In *csfw* (p. 0273).

Du, W., Han, Y. S., & Chen, S. (2004). Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 siam international conference on data mining* (pp. 222–233).

DuPre, E., & Spreng, R. N. (2017). Structural covariance networks across the life span, from 6 to 94 years of age. *Network Neuroscience*, *1*(3), 302–323.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference* (pp. 214–226).

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference* (pp. 265–284).

Enders, C. K. (2001). The performance of the full information maximum likelihood estimator in multiple regression models with missing data. *Educational and Psychological Measurement*, *61*(5), 713–740.

Epskamp, S., Borsboom, D., & Fried, E. I. (2018). Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, *50*(1), 195–212.

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

Eshaghi, A., Marinescu, R. V., Young, A. L., Firth, N. C., Prados, F., Jorge Cardoso, M., … Brownlee, W. J. (2018). Progression of regional grey matter atrophy in multiple sclerosis. *Brain*, *141*(6), 1665–1677.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, *96*(456), 1348–1360.

Fang, W., Zhou, C., & Yang, B. (2013). Privacy preserving linear regression modeling of distributed databases. *Optimization Letters*, *7*(4), 807–818.

Ferguson, M. A., Anderson, J. S., & Spreng, R. N. (2017). Fluid and flexible minds: Intelligence reflects synchrony in the brain's intrinsic network architecture. *Network Neuroscience*, *1*(2), 192–207.

Fletcher, R. (2013). *Practical methods of optimization*. John Wiley & Sons.

Flore, P. (2018). *Stereotype threat and differential item functioning: A critical assessment* (Unpublished doctoral dissertation). Tilburg University.

Frässle, S., Paulus, F. M., Krach, S., Schweinberger, S. R., Stephan, K. E., & Jansen, A. (2016). Mechanisms of hemispheric lateralization: Asymmetric interhemispheric recruitment in the face perception network. *Neuroimage*, *124*, 977–988.

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd acm sigsac conference on computer and communications security* (pp. 1322–1333).

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical software*, *33*(1), 1.

Fuhrmann, D., Nesbitt, D., Shafto, M., Rowe, J. B., Price, D., Gadie, A., … Kievit, R. A. (2019). Strong and specific associations between cardiovascular risk factors and white matter micro-and macrostructure in healthy aging. *Neurobiology of aging*, *74*, 46–55.

Fuller, W. A. (2009). *Measurement error models* (Vol. 305). John Wiley & Sons.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *182*(2), 389–402.

Gambs, S., Kégl, B., & Aïmeur, E. (2007). Privacy-preserving boosting. *Data Mining and Knowledge Discovery*, *14*(1), 131–170.

Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J., Zahur, S., & Evans, D. (2016). Secure linear regression on vertically partitioned datasets. *IACR Cryptology ePrint Archive*, *2016*, 892.

Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J., Zahur, S., & Evans,

D. (2017). Privacy-preserving distributed linear regression on high-dimensional data. *Proceedings on Privacy Enhancing Technologies*, *2017*(4), 345–364.

Gauger, L. M., Lombardino, L. J., & Leonard, C. M. (1997). Brain morphology in children with specific language impairment. *Journal of Speech, Language, and Hearing Research*, *40*(6), 1272–1284.

Geerligs, L., Tsvetanov, K. A., & Henson, R. N. (2017). Challenges in measuring individual differences in functional connectivity using fmri: the case of healthy aging. *Human brain mapping*, *38*(8), 4125–4156.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis.* CRC press.

Geyer, R. C., Klein, T., & Nabi, M. (2017). *Differentially private federated learning: A client level perspective.*

Goeman, J., Meijer, R., & Chaturvedi, N. (2018). L1 and l2 penalized regression models. *Vignette R Package Penalized..*

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press. (http://www.deeplearningbook.org)

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, *46*(2), 149–170.

Grotzinger, A. D., Rhemtulla, M., de Vlaming, R., Ritchie, S. J., Mallard, T. T., Hill, W. D., … Deary, I. J. (2019). Genomic structural equation modelling provides insights into the multivariate genetic architecture of complex traits. *Nature human behaviour*, *3*(5), 513.

Guàrdia-Olmos, J., Peró-Cebollero, M., Benítez-Borrego, S., & Fox, J. (2009). Using sem library in r software to analyze exploratory structural equation models. In *59th isi world statistics congress.*

Guo, R., Zhu, H., Chow, S.-M., & Ibrahim, J. G. (2012). Bayesian lasso for semiparametric structural equation models. *Biometrics*, *68*(2), 567–577.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research (JMLR)*, *3*(3), 1157–1182. doi: 10.1016/j.aca.2011.07.027

Harman, H. H., & Jones, W. H. (1966). Factor analysis by minimizing residuals (minres). *Psychometrika*, *31*(3), 351–368.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations.* Boca Raton: CRC Press. doi: 10.1201/b18401-1

Hayes, A. F., & Preacher, K. J. (2010). Quantifying and testing indirect effects in simple mediation models when the constituent paths are nonlinear. *Multivariate Behavioral Research*, *45*(4), 627–660. doi: 10.1080/00273171.2010.498290

Hayes, A. F., & Preacher, K. J. (2014). Statistical mediation analysis with a multicategorical independent variable. *British Journal of Mathematical and Statistical Psychology*, *67*, 451–470. doi: 10.1111/bmsp.12028

Herbert, E., Engel-Hills, P., Hattingh, C., Fouche, J.-P., Kidd, M., Lochner, C., … van Rensburg, S. J. (2018). Fractional anisotropy of white matter, disability and blood iron parameters in multiple sclerosis. *Metabolic brain disease*, *33*(2), 545–557.

Holland, P. W., & Wainer, H. (1993). *Differential Item Functioning.* New York:

Routledge.

Holmes Finch, W., & Miller, J. (2020). A comparison of regularized maximum-likelihood, regularized 2-stage least squares, and maximum-likelihood estimation with misspecified models, small samples, and weak factor structure. *Multivariate Behavioral Research*, 1–19.

Holzinger, K. J., & Swineford, F. (1939). A study in factor analysis: The stability of a bi-factor solution. *Supplementary Educational Monographs.*

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

Houtepen, L. C., Vinkers, C. H., Carrillo-Roa, T., Hiemstra, M., van Lier, P. A., Meeus, W., … Boks, M. P. M. (2016). Genome-wide DNA methylation levels and altered cortisol stress reactivity following childhood trauma in humans. *Nature communications*, *7*, 10967. doi: 10.1038/ncomms10967

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

Huang, Y. T., & Pan, W. C. (2015). Hypothesis test of mediation effect in causal mediation model with high-dimensional continuous mediators. *Biometrics*(June), 402–413. doi: 10.1111/biom.12421

Jacobucci, R., Brandmaier, A. M., & Kievit, R. A. (2018). Variable selection in structural equation models with regularized MIMIC models. *PsyArXiv Preprint*, 1–40. doi: 10.17605/OSF.IO/BXZJF

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., & McArdle, J. J. (2016). Regularized structural equation modeling. *Structural Equation Modeling*, *23*(4), 555–566. doi: 10.1080/10705511.2016.1154793.Regularized

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). *Introduction to Statistical Learning With Applications in R* (6th ed.). New York: Springer.

James, G. A., Kelley, M. E., Craddock, R. C., Holtzheimer, P. E., Dunlop, B. W., Nemeroff, C. B., … Hu, X. P. (2009). Exploratory structural equation modeling of resting-state fmri: applicability of group models to individual subjects. *Neuroimage*, *45*(3), 778–787.

Jin, S., Moustaki, I., & Yang-Wallentin, F. (2018). Approximated penalized maximum likelihood for exploratory factor analysis: An orthogonal case. *Psychometrika*, *83*(3), 628–649.

Jones, D. K., Knösche, T. R., & Turner, R. (2013). White matter integrity, fiber count, and other fallacies: the do's and don'ts of diffusion mri. *Neuroimage*, *73*, 239–254.

Jöreskog, K. G. (1966). Testing a simple structure hypothesis in factor analysis. *Psychometrika*, *31*(2), 165–178.

Jöreskog, K. G. (1967). Some contributions to maximum likelihood factor analysis. *Psychometrika*, *32*(4), 443–482.

Jöreskog, K. G. (1969). A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika*, *34*(2), 183–202.

Jöreskog, K. G. (1993). Testing structural equation models. *Sage focus editions*, *154*, 294–294.

Jöreskog, K. G., & Sörbom, D. (1993). *Lisrel 8: Structural equation modeling with* *the simplis command language*. Scientific Software International.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*(3), 187–200.

Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big data: Issues and challenges moving forward. In *2013 46th hawaii international conference on system sciences* (pp. 995–1004).

Karr, A. F. (2010). Secure statistical analysis of distributed databases, emphasizing what we don't know. *Journal of Privacy and Confidentiality*, *1*(2).

Karr, A. F., Lin, X., Sanil, A. P., & Reiter, J. P. (2009). Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, *25*(1), 125.

Kasthurirathne, S. N., Vest, J. R., Menachemi, N., Halverson, P. K., & Grannis, S. J. (2017). Assessing the capacity of social determinants of health data to augment predictive models identifying patients in need of wraparound social services. *Journal of the American Medical Informatics Association*, *25*(1), 47–53.

Kenny, D. A. (1976). An empirical application of confirmatory factor analysis to the multitrait-multimethod matrix. *Journal of Experimental Social Psychology*, *12*(3), 247–252.

Kievit, R. A., Davis, S. W., Griffiths, J., Correia, M. M., & Henson, R. N. (2016). A watershed model of individual differences in fluid intelligence. *Neuropsychologia*, *91*, 186–198.

Kilbertus, N., Carulla, M. R., Parascandolo, G., Hardt, M., Janzing, D., & Schölkopf, B. (2017). Avoiding discrimination through causal reasoning. In *Advances in neural information processing systems* (pp. 656–666).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, A., & Tourville, J. (2012). 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in neuroscience*, *6*, 171.

Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

Kolenikov, S., & Bollen, K. A. (2012). Testing negative error variances: Is a heywood case a symptom of misspecification? *Sociological Methods & Research*, *41*(1), 124–167.

Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual Fairness. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4066–4076). Curran Associates, Inc.

Kyung, M., Gill, J., Ghosh, M., & Casella, G. (2010). Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, *5*(2), 369–411.

Lai, M. H., & Zhang, J. (2017). Evaluating fit indices for multivariate t-based structural equation modeling with data contamination. *Frontiers in psychology*, *8*, 1286.

Lee, S.-Y., & Jennrich, R. (1979). A study of algorithms for covariance structure analysis with specific comparisons using factor analysis. *Psychometrika*, *44*(1), 99–113.

Lehmann, B. C., Henson, R. N., Geerligs, L., & White, S. R. (2019). Characterising group-level brain connectivity: a framework using bayesian exponential random graph models. *bioRxiv*, 665398.

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2019). Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*.

Li, X., Zhao, T., Arora, R., Liu, H., & Hong, M. (2017). On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization. *The Journal of Machine Learning Research*, *18*(1), 6741–6764.

Lindell, Y. (2005). Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of data warehousing and mining* (pp. 1005–1009). IGI Global.

Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A., & Moroz, I. M. (2007). Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical engineering online*, *6*(1), 23.

Liu, Y., Aryee, M. J., Padyukov, L., Fallin, M. D., Hesselberg, E., Runarsson, A., … Feinberg, A. P. (2013). Epigenome-wide association data implicate DNA methylation as an intermediary of genetic risk in rheumatoid arthritis. *Nature Biotechnology*, *31*(2), 142–147. doi: 10.1038/nbt.2487

Lord, F. M. (2012). *Applications of item response theory to practical testing problems*. Routledge.

Lövdén, M., Laukka, E. J., Rieckmann, A., Kalpouzos, G., Li, T.-Q., Jonsson, T., … Bäckman, L. (2013). The dimensionality of between-person differences in white matter microstructure in old age. *Human brain mapping*, *34*(6), 1386–1398.

Lu, Z.-H., Chow, S.-M., & Loken, E. (2016). Bayesian factor analysis as a variable-selection problem: Alternative priors and consequences. *Multivariate behavioral research*, *51*(4), 519–539.

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.

Machado, A. M., Gee, J. C., & Campos, M. F. (2004). Structural shape characterization via exploratory factor analysis. *Artificial Intelligence in Medicine*, *30*(2), 97–118.

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Routledge.

MacKinnon, D. P., Fairchild, A. J., & Fritz, M. S. (2007). Mediation analysis. *Annual Review of Psychology*, *58*(1), 593–614. doi: 10.1146/annurev.psych.58.110405.085542

MacKinnon, D. P., Lockwood, C. M., Hoffman, J. M., West, S. G., & Sheets, V. (2002, mar). A comparison of methods to test mediation and other intervening variable effects. *Psychological methods*, *7*(1), 83–104.

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research*, *39*(1), 99–128. doi: 10.1207/s15327906mbr3901

Mancini, M., Giulietti, G., Spanò, B., Bozzali, M., Cercignani, M., & Conforto, S. (2016). Estimating multimodal brain connectivity in multiple sclerosis: an exploratory factor analysis. In *2016 38th annual international conference of the ieee engineering in medicine and biology society (embc)* (pp. 1131–1134).

Mandt, S., Hoffman, M. D., & Blei, D. M. (2017). Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, *18*(1), 4873–4907.

Marcoulides, G. A., & Drezner, Z. (2001). Specification searches in structural equation modeling with a genetic algorithm. *New developments and techniques in structural equation modeling*, 247–268.

Marcoulides, K. M., & Falk, C. F. (2018). Model specification searches in structural equation modeling with r. *Structural Equation Modeling: A Multidisciplinary*

Journal, *25*(3), 484–491.

Marsh, H. W., Morin, A. J., Parker, P. D., & Kaur, G. (2014). Exploratory structural equation modeling: An integration of the best features of exploratory and confirmatory factor analysis. *Annual review of clinical psychology*, *10*, 85–110.

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology*, *37*(2), 234–251.

McCullagh, P., & Nelder, J. (1989). *Generalized linear models.* New York: Chapman and Hall/CRC.

McCutcheon, A. L. (1987). *Latent class analysis* (No. 64). Sage.

McElreath, R. (2020). *Statistical rethinking: A bayesian course with examples in r and stan.* CRC press.

McIntosh, A. R., & Protzner, A. B. (2012). Handbook of structural equation modeling. In R. H. Hoyle (Ed.), (p. 636–649). The Guilford Press.

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering* (Vol. 38). M. Dekker New York.

Mechelli, A., Friston, K. J., Frackowiak, R. S., & Price, C. J. (2005). Structural covariance in the human cortex. *Journal of Neuroscience*, *25*(36), 8303–8310.

Mellenbergh, G. J. (1989, January). Item bias and item response theory. *International Journal of Educational Research*, *13*(2), 127–143. doi: 10.1016/0883-0355(89)90002-5

Merkle, E. C., & Rosseel, Y. (2015). blavaan: Bayesian structural equation models via parameter expansion. *arXiv preprint arXiv:1511.05604*.

Meyer, K., Garzón, B., Lövdén, M., & Hildebrandt, A. (2019). Are global and specific interindividual differences in cortical thickness associated with facets of cognitive abilities, including face cognition? *Royal Society open science*, *6*(7), 180857.

Moodie, J., Ritchie, S. J., Cox, S. R., Harris, M. A., Maniega, S. M., Hernández, M. C. V., … Starr, J. (2019). Structural brain asymmetry and general intelligence in 73-year-olds. *PsyArXiv*. doi: 10.31234/osf.io/ea7tz

Mori, S., Oishi, K., Jiang, H., Jiang, L., Li, X., Akhter, K., … Woods, R. (2008). Stereotaxic white matter atlas based on diffusion tensor imaging in an icbm template. *Neuroimage*, *40*(2), 570–582.

Mowinckel, A. M., & Vidal-Piñeiro, D. (2019). *Visualisation of brain statistics with r-packages ggseg and ggseg3d.*

Muthén, L. K., & Muthén, B. O. (1998). Mplus user's guide (version 7). *Los Angeles, CA: Muthén & Muthén, 2004.*

Nabi, R., & Shpitser, I. (2018, April). Fair Inference on Outcomes. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

Nalisnick, E., Hernandez-Lobato, J. M., & Smyth, P. (2019, 09–15 Jun). Dropout as a structured shrinkage prior. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 4712–4722). Long Beach, California, USA: PMLR. Retrieved from http://proceedings.mlr.press/v97/nalisnick19a.html

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. doi: 10.1007/s11336-014-9435-8

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, *22*(2), 341–362.

Neudecker, H., & Satorra, A. (1991). Linear structural relations: Gradient and hessian of the fitting function. *Statistics & Probability Letters*, *11*(1), 57–61.

Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., & Taft, N. (2013). Privacy-preserving ridge regression on hundreds of millions of records. In *2013 ieee symposium on security and privacy* (pp. 334–348).

Norwegian Centre for Research Data. (2018). *Ess round 9: European social survey round 9 data.* Data Archive and distributor of ESS data for ESS ERIC. (Data file edition 2.0) doi: 10.21338/NSD-ESS9-2018

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019, October). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, *366*(6464), 447–453. doi: 10.1126/science.aax2342

Oberski, D. L. (2014). Evaluating sensitivity of parameters of interest to measurement invariance in latent variable models. *Political analysis*, 45–60.

Olkin, I., & Finn, J. D. (1995). Correlations redux. *Psychological Bulletin*, *118*(1), 155–164. doi: 10.1037/0033-2909.118.1.155

Pan, J., Ip, E. H., & Dubé, L. (2017). An alternative to post hoc model modification in confirmatory factor analysis: The bayesian lasso. *Psychological Methods*, *22*(4), 687.

Pan, J., & Tompkins, W. J. (1985). A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*(3), 230–236.

Park, T., & Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, *103*(482), 681–686. doi: 10.1198/016214508000000337

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., … Lerer, A. (2017). Automatic differentiation in pytorch..

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pavel, A. (2019). *Decompose projection matrix into a matrix and its pseudoinverse.* Mathematics Stack Exchange. (2019-08-12)

Pearl, J. (2009). *Causality.* Cambridge university press.

Pearl, J. (2013). *Causality models, reasoning, and inference.* Cambridge: Cambridge University Press. (OCLC: 956314447)

Petersen, K., & Pedersen, M. (2012). The matrix cookbook, version 20121115. *Technical Univ. Denmark, Kongens Lyngby, Denmark, Tech. Rep*, *3274*.

Plessen, K. J., Hugdahl, K., Bansal, R., Hao, X., & Peterson, B. S. (2014). Sex, age, and cognitive correlates of asymmetries in thickness of the cortical mantle across the life span. *Journal of Neuroscience*, *34*(18), 6294–6302.

Powell, M. J. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, *7*(2), 155–162.

Preacher, K. J. (2015). Advances in mediation analysis: A survey and synthesis of new developments. *Annual Review of Psychology*, *66*(1), 825–852. doi: 10.1146/annurev-psych-010814-015258

Preacher, K. J., & Hayes, A. F. (2008). Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods*, *40*(3), 879–891. doi: 10.3758/BRM.40.3.879

R Core Team. (2018). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing.

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests.* ERIC.

Revelle, W. (2018). psych: Procedures for psychological, psychometric, and personality research [Computer software manual]. Evanston, Illinois. Retrieved from `https://CRAN.R-project.org/package=psych` (R package version 1.8.12)

Revilla, M., & Saris, W. E. (2013). The split-ballot multitrait-multimethod approach: Implementation and problems. *Structural Equation Modeling: A Multidisciplinary Journal*, *20*(1), 27–46.

Richtárik, P., & Takáč, M. (2014, Apr 01). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, *144*(1), 1–38. doi: 10.1007/s10107-012-0614-z

Richtárik, P., & Takáč, M. (2016). Distributed coordinate descent method for learning with big data. *The Journal of Machine Learning Research*, *17*(1), 2657–2681.

Roe, J. M., Vidal-Piñeiro, D., Sørensen, Ø., Brandmaier, A. M., Düzel, S., Gonzalez, H. A., … Lindenberger, U. (2020). Asymmetric thinning of the cerebral cortex across the adult lifespan is accelerated in alzheimer's disease. *bioRxiv*.

Rosas, H. D., Lee, S. Y., Bender, A. C., Zaleta, A. K., Vangel, M., Yu, P., … Cha, J.-H. (2010). Altered white matter microstructure in the corpus callosum in huntington's disease: implications for cortical "disconnection". *Neuroimage*, *49*(4), 2995–3004.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36.

Rosseel, Y. (2019). lavaan version 0.6-4 [Computer software manual]. (R package version 0.6-4)

Roweis, S., & Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural computation*, *11*(2), 305–345.

Safran, M., Solomon, I., Shmueli, O., Lapidot, M., Shen-Orr, S., Adato, A., … Lancet, D. (2002). Genecards™ 2002: Towards a complete, object-oriented, human gene compendium. *Bioinformatics*, *18*, 1542-1543. doi: 10.1093/bioinformatics/18.11.1542

Sanil, A. P., Karr, A. F., Lin, X., & Reiter, J. P. (2004). Privacy preserving regression modelling via distributed computation. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 677–682). New York, NY, USA: ACM. doi: 10.1145/1014052.1014139

Santos, M. S., Abreu, P. H., García-Laencina, P. J., Simão, A., & Carvalho, A. (2015). A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. *Journal of biomedical informatics*, *58*, 49–59.

Saris, W. E., Satorra, A., & Sörbom, D. (1987). The detection and correction of specification errors in structural equation models. *Sociological methodology*, 105–129.

Satorra, A., & Bentler, P. (1988). Scaling corrections for statistics in covariance structure analysis. In *Asa 1988 proceedings of the business and economic statistics section* (p. 308–313). Alexandria, VA.

Savalei, V. (2014). Understanding robust corrections in structural equation modeling.

Structural Equation Modeling: A Multidisciplinary Journal, *21*(1), 149–160.

Scardapane, S., Comminiello, D., Hussain, A., & Uncini, A. (2017). Group sparse regularization for deep neural networks. *Neurocomputing*, *241*, 81–89.

Schaefer, A., Kong, R., Gordon, E. M., Laumann, T. O., Zuo, X.-N., Holmes, A. J., … Yeo, B. T. (2018). Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity mri. *Cerebral cortex*, *28*(9), 3095–3114.

Schaid, D. J., & Sinnwell, J. P. (2020). Penalized models for analysis of multiple mediators. *Genetic Epidemiology*.

Scharf, F., & Nestler, S. (2018). Principles behind variance misallocation in temporal exploratory factor analysis for erp data: Insights from an inter-factor covariance decomposition. *International Journal of Psychophysiology*, *128*, 119–136.

Scharf, F., & Nestler, S. (2019). Should regularization replace simple structure rotation in exploratory factor analysis? *Structural Equation Modeling: A Multidisciplinary Journal*, 1–15.

Schmitt, N., & Kuljanin, G. (2008). Measurement invariance: Review of practice and implications. *Human resource management review*, *18*(4), 210–222.

Schuler, A.-L., Bartha-Doering, L., Jakab, A., Schwartz, E., Seidl, R., Kienast, P., … Kasprian, G. (2018, Nov 01). Tracing the structural origins of atypical language representation: consequences of prenatal mirror-imaged brain asymmetries in a dizygotic twin couple. *Brain Structure and Function*, *223*(8), 3757–3767. doi: 10.1007/s00429-018-1717-y

Sclove, S. L. (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika*, *52*(3), 333–343.

Serang, S., Jacobucci, R., Brimhall, K. C., & Grimm, K. J. (2017). Exploratory mediation analysis via regularization. *Structural Equation Modeling*, *24*(5), 733–744. doi: 10.1080/10705511.2017.1311775

Shafto, M. A., Tyler, L. K., Dixon, M., Taylor, J. R., Rowe, J. B., Cusack, R., … Dalgleish, T. (2014). The cambridge centre for ageing and neuroscience (cam-can) study protocol: a cross-sectional, lifespan, multidisciplinary examination of healthy cognitive ageing. *BMC neurology*, *14*(1), 204.

Siemsen, E., & Bollen, K. A. (2007). Least absolute deviation estimation in structural equation modeling. *Sociological Methods & Research*, *36*(2), 227–265.

Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models.* CRC Press.

Slavkovic, A. B., Nardi, Y., & Tibbits, M. M. (2007). Secure logistic regression of horizontally and vertically partitioned distributed databases. In *Proceedings of the seventh ieee international conference on data mining workshops* (pp. 723–728). Washington, DC, USA: IEEE Computer Society. doi: 10.1109/ICDMW.2007.84

Sobel, M. E. (1986). Some new results on indirect effects and their standard errors in covariance structure models. *Sociological Methodology*, *16*(16), 159–186.

Sörbom, D. (1974). A general method for studying differences in factor means and factor structure between groups. *British Journal of Mathematical and Statistical Psychology*, *27*(2), 229–239.

Sörbom, D. (1989). Model modification. *Psychometrika*, *54*(3), 371–384.

Sorzano, C. O. S., Vargas, J., & Montano, A. P. (2014). A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*.

Spirtes, P., Glymour, C. N., Scheines, R., & Heckerman, D. (2000). *Causation, prediction, and search.* MIT press.

Sripada, C., Angstadt, M., Rutherford, S., Kessler, D., Kim, Y., Yee, M., & Levina, E. (2019). Basic units of inter-individual variation in resting state connectomes. *Scientific reports*, *9*(1), 1–12.

Steenkamp, J.-B. E., & Baumgartner, H. (1998). Assessing measurement invariance in cross-national consumer research. *Journal of consumer research*, *25*(1), 78–90.

Stievenart, J.-L., Iba-Zizen, M.-T., Tourbah, A., Lopez, A., Thibierge, M., Abanou, A., & Cabanis, E. (1997). Minimal surface: A useful paradigm to describe the deeper part of the corpus callosum? *Brain research bulletin*, *44*(2), 117–124.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 44–47.

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, *2014*.

Taylor, J. R., Williams, N., Cusack, R., Auer, T., Shafto, M. A., Dixon, M., … Henson, R. N. (2017). The cambridge centre for ageing and neuroscience (cam-can) data repository: structural and functional mri, meg, and cognitive data from a cross-sectional adult lifespan sample. *Neuroimage*, *144*, 262–269.

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

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, *4*(2), 26–31.

Tien, A. Y., Eaton, W. W., Schlaepfer, T. E., McGilchrist, I. K., Menon, R., Richard, P., … Pearlson, G. D. (1996). Exploratory factor analysis of mri brain structure measures in schizophrenia. *Schizophrenia Research*, *19*(2-3), 93–101.

Triche, T. (2014). *FDb.InfiniumMethylation.hg18: Annotation package for Illumina Infinium DNA methylation probes.*

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, *109*(3), 475–494.

Tucker, D. M., & Roth, D. L. (1984). Factoring the coherence matrix: Patterning of the frequency-specific covariance in a multichannel eeg. *Psychophysiology*, *21*(2), 228–236.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., … Joliot, M. (2002). Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage*, *15*(1), 273–289.

Vaidya, J., & Clifton, C. (2002). Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the eighth acm sigkdd international conference on knowledge discovery and data mining* (pp. 639–644).

Vaidya, J., & Clifton, C. (2003). Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining* (pp. 206–215).

Vaidya, J., & Clifton, C. (2005). Privacy-preserving decision trees over vertically partitioned data. In *Ifip annual conference on data and applications security and privacy* (pp. 139–152).

Vaidya, J., Yu, H., & Jiang, X. (2008). Privacy-preserving svm classification. *Knowledge and Information Systems*, *14*(2), 161–178.

van Erp, S., Oberski, D. L., & Mulder, J. (2019). Shrinkage Priors for Bayesian Penalized Regression. *Journal of Mathematical Psychology*, *89*, 31–50. doi: 10.31219/osf.io/cg8fq

van Kesteren, E.-J., & Oberski, D. L. (2019). Exploratory mediation analysis with many potential mediators. *Structural Equation Modeling: A Multidisciplinary Journal*, 1–14.

Vandenberg, R. J., & Lance, C. E. (2000). A review and synthesis of the measurement invariance literature: Suggestions, practices, and recommendations for organizational research. *Organizational research methods*, *3*(1), 4–70.

Van Den Heuvel, M. P., & Pol, H. E. H. (2010). Exploring the brain network: a review on resting-state fmri functional connectivity. *European neuropsychopharmacology*, *20*(8), 519–534.

VanderWeele, T. J. (2015). *Explanation in Causal Inference: Methods for Mediation and Interaction.* New York: Oxford University Press.

Vanderweele, T. J., & Vansteelandt, S. (2014). Mediation analysis with multiple mediators. *Epidemiologic Methods*, *2*(1), 95–115. doi: 10.1515/em-2012-0010 .Mediation

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

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, *27*(5), 1413–1432.

Verma, S., & Rubin, J. (2018, May). Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness* (pp. 1–7). Gothenburg, Sweden: Association for Computing Machinery. doi: 10.1145/3194770.3194776

Voelkle, M. C., & Oud, J. H. (2013). Continuous time modelling with individually varying time intervals for oscillating and non-oscillating processes. *British Journal of Mathematical and Statistical Psychology*, *66*(1), 103–126.

Vrieze, S. I. (2012). Model selection and psychological theory: a discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic). *Psychological methods*, *17*(2), 228.

Wager, S., Wang, S., & Liang, P. S. (2013). Dropout training as adaptive regularization. In *Advances in neural information processing systems* (pp. 351–359).

Wang, Y., Si, C., & Wu, X. (2015). Regression model fitting under differential privacy and model inversion attack. In *Twenty-fourth international joint conference on artificial intelligence.*

Wedderburn, R. W. (1974). Quasi-likelihood functions, generalized linear models, and the gauss—newton method. *Biometrika*, *61*(3), 439–447.

Wengert, R. E. (1964). A simple automatic derivative evaluation program. *Communications of the ACM*, *7*(8), 463–464.

World Health Organization. (2008). *Closing the gap in a generation: Health equity through action on the social determinants of health.* World Health Organization.

Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, *151*(1), 3–34.

Yao, A. C.-C. (1986). How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (sfcs 1986)* (pp. 162–167).

Yeh, P.-H., Zhu, H., Nicoletti, M. A., Hatch, J. P., Brambilla, P., & Soares, J. C. (2010). Structural equation modeling and principal component analysis of gray matter volumes in major depressive and bipolar disorders: differences in latent volumetric structure. *Psychiatry Research: Neuroimaging*, *184*(3), 177–185.

Yuan, K.-H., & Bentler, P. M. (1998). Structural equation modeling with robust covariances. *Sociological methodology*, *28*(1), 363–396.

Zhang, H., Zheng, Y., Zhang, Z., Gao, T., Joyce, B., Yoon, G., … Liu, L. (2016). Estimating and testing high-dimensional mediation effects in epigenetic studies. *Bioinformatics*, *32*(20), 3150–3154. doi: 10.1093/bioinformatics/btw351

Zhang, J., Zhao, Z., Zhang, K., & Wei, Z. (2019, March). A feature sampling strategy for analysis of high dimensional genomic data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *16*(2), 434-441. doi: 10.1109/TCBB.2017.2779492

Zhao, Y., & Luo, X. (2016). Pathway Lasso: Estimate and Select Sparse Mediation Pathways with High Dimensional Mediators. *arXiv preprint*, 1–26.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic-net. *Journal of the Royal Statistical Society*, *67*(2), 301–320. doi: 10.1111/j.1467-9868.2005.00503.x

Appendices

Appendix A
Flexible Extensions to Structural Equation Models using Computation Graphs: Supplementary Material

A.1 Adaptive first-order optimizers

We suggest using adaptive first-order optimizers to extend SEM beyond the existing estimation methods. Adaptive first-order optimizers are a class of optimization algorithms designed to work even under nonconvexity and non-smoothness. Some early algorithms such as RMSProp (Tieleman & Hinton, 2012) were originally developed with deep learning in mind, where nonconvexity, non-smoothness, and high-dimensional parameter spaces are common. Therefore, we consider these methods excellent candidates for estimating an expanding class of SEM models, as they have historically done for neural networks. The idea of using first-order optimizers for SEM is by no means new (Lee & Jennrich, 1979), but the recent developments in this area have made it a feasible approach.

The simplest first-order optimizer is gradient descent, which uses the gradient $\boldsymbol{g}(\boldsymbol{\theta})$ of the objective with respect to the parameters to guide the direction that each parameter should move towards. The gradient is combined with a step size s so that in each iteration i of gradient descent the parameters are moved a small amount towards the direction of the negative gradient evaluated at the current parameter values:

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - s \cdot \boldsymbol{g}(\boldsymbol{\theta}^{(i)}) \tag{A.1}$$

This algorithm has a similar structure to the Newton-Raphson method shown in Equation 2.6. In that algorithm, the step size s in each iteration is replaced by the inverse of the Hessian matrix. Gradient descent is thus a simplified version of the methods currently in use for optimizing SEM. Because it does not use the Hessian, it continues to function when the objective is not smooth or not convex. Computationally, it is also more tractable, foregoing the need to compute the full Hessian matrix. However, it is necessary to determine the correct step size s. This is not a trivial problem: with an improperly tuned step size, the algorithm may never converge.

One of the state-of-the art adaptive first-order optimizers is Adam (Kingma & Ba, 2014). It introduces two improvements to the framework of gradient descent (Figure A.1). Firstly, it introduces momentum, where the direction in each iteration is not only the negative gradient of that iteration, but a *moving average* of the entire history of gradients. Momentum allows Adam to move through local minima in the search for a global minimum by smoothing the path it takes in the parameter space. Secondly, Adam introduces a self-adjusting step size for each parameter, which is adjusted based on the *variability* of the gradients over time: if the variability of the gradient of a parameter is smaller, Adam will take larger steps as it has more certainty about the direction the parameter should move in (and vice versa). This self-adjusting step size takes the place of computing and inverting the Hessian matrix. By using both the first and second moments of the history of the gradients, Adam is an adaptive optimizer capable of reliably optimizing a wide variety of objectives.

A relevant parallel to the development of adaptive first-order optimizers for deep learning is the recent advances in Bayesian SEM (Merkle & Rosseel, 2015) and Bayesian posterior sampling in general. Here, too, the objective function may be nonconvex, e.g., in hierarchical models and with nonconjugate priors. Such objective functions may lead to inefficient behaviour for the Markov Chain Monte Carlo (MCMC) methods used to approximate posterior expectations. For this problem, Hamiltonial Monte Carlo (HMC) (Betancourt, 2017) has been developed, which introduces momentum in the proposal of a sample, thereby more efficiently exploring the posterior. This is the method implemented in Stan (Carpenter et al., 2017), which works for situations with many parameters and hyperparameters.

Adaptive first-order optimizers are one part of a pair of improvements that have enabled rapid growth of the deep learning field. The other is the development of computation graphs, an intuitive way of specifying the objective such that gradients can be computed automatically. Automatic gradient computation can enable a wide range of extensions to SEM without having to analytically derive the gradient and Hessian for each separate extension. In the next section, we explain the concept behind computation graphs and how they can be combined with optimizers such as Adam.

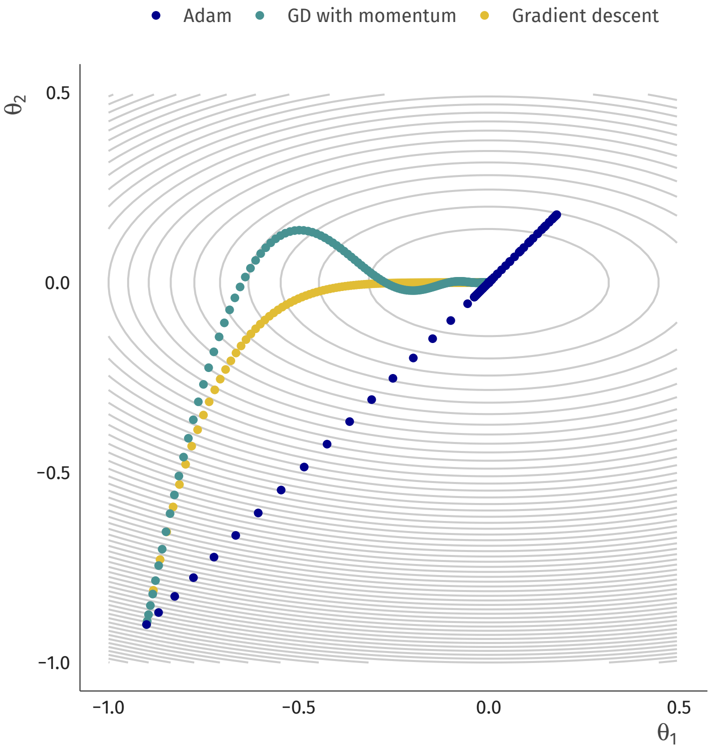

Figure A.1 Three first-order algorithms finding the minimum of $F(\boldsymbol{\theta}) = \theta_1^2 + 5\theta_2^2$ with starting value $\hat{\boldsymbol{\theta}} = [-0.9, -0.9]$. Gradient descent uses the gradient and a fixed step size ($s = 0.01$) to update its parameter estimates. Gradient descent with momentum instead uses an exponential moving average of the gradients (decay of 0.9) with the same s. Finally, Adam adds a moving average of the squared gradient (decay of 0.999) to adjust the step size per parameter, leading to a straight line to the minimum with an overshoot and return due to momentum. In this example, Adam converges fastest, and gradient descent is slowest.

A.2 PyTorch estimation validation

A.2.1 ML-SEM estimation

We first validated our `PyTorch` implementation of default SEM through comparing the parameter estimates and their standard errors to two example models from the `lavaan` package: the Holzinger-Swineford model and the Political Democracy model. For more information about these models, see Rosseel (2012). The reproducible code for these models can be found in the supplementary material. The results are shown in Figure A.2.

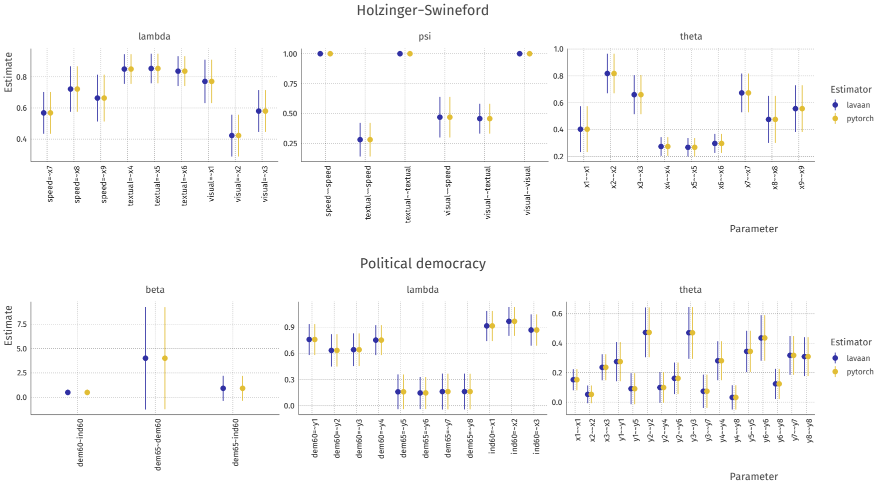

Figure A.2 Comparison of parameter estimates and their 95% confidence interval for the Holzinger-Swineford and Political Democracy models. The plots show that both methods arrive at the same solution.

From this validation, we conclude that computation graphs and Adam optimization are together capable of estimating structural equation models. In addition, as the solution obtained by `PyTorch` is the same as with other packages, it is possible compute the value of the log-likelihood objective function and its derivative fit measures such as χ^2, AIC, and BIC.

A.2.2 LASSO regularization

In this example, we show how LASSO penalization on the regression parameters in `tensorsem` compares to `regsem` (Jacobucci et al., 2016) and `glmnet` (Friedman et al., 2010). For this, we generate data with a sample size of 1000 from a regression model with a single outcome variable, 10 true predictors, and 10 unrelated variables. The resulting parameter estimates for the three different estimation methods are shown in Table A.1. The table shows that with the chosen penalty parameter (0.11 for `regsem` and `PyTorch`, 0.028 for `glmnet` due to a difference in scaling), the estimates are very close in value. As expected, some parameters are shrunk to 0 for all three methods.

Table A.1 Regularization with `glmnet`, `regsem`, and `PyTorch`. Table indicates parameter estimates for a LASSO penalized regression model with 20 predictors. `PyTorch` is compared to existing approaches and shown to provide similar parameter estimates. (dot indicates 0)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
`glmnet`	.	.07	.10	.07	.20	.23	.34	.13	.31	.17	-.03	.02	.05	.	.	.	.	.	.	.
`regsem`	.	.07	.10	.08	.20	.23	.34	.13	.31	.17	-.03	.02	.05	.	.	.	.	.	.	.
`PyTorch`	.	.07	.10	.07	.20	.23	.34	.13	.31	.17	-.03	.02	.05	.	.	.	.	.	.	.

Appendix B

Exploratory Factor Analysis with Structured Residuals for Brain Network Data: Supplementary Figures and Tables

B.1 Symmetry pattern recovery with default EFA

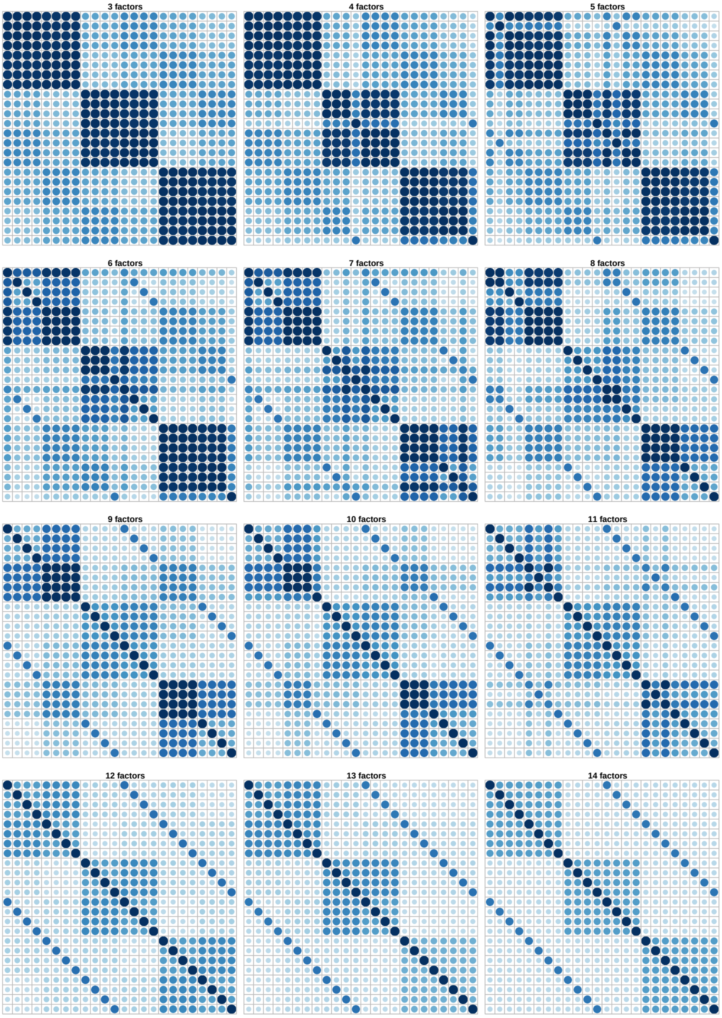

Figure B.1 Predicted correlation matrix for EFA models with M factors for the example observed correlation matrix of Figure 1 in the main text. Proper recovery of the observed pattern happens around 12 factors (bottom left frame).

B.2 Comparing EFA and EFAST in factor loading estimation error

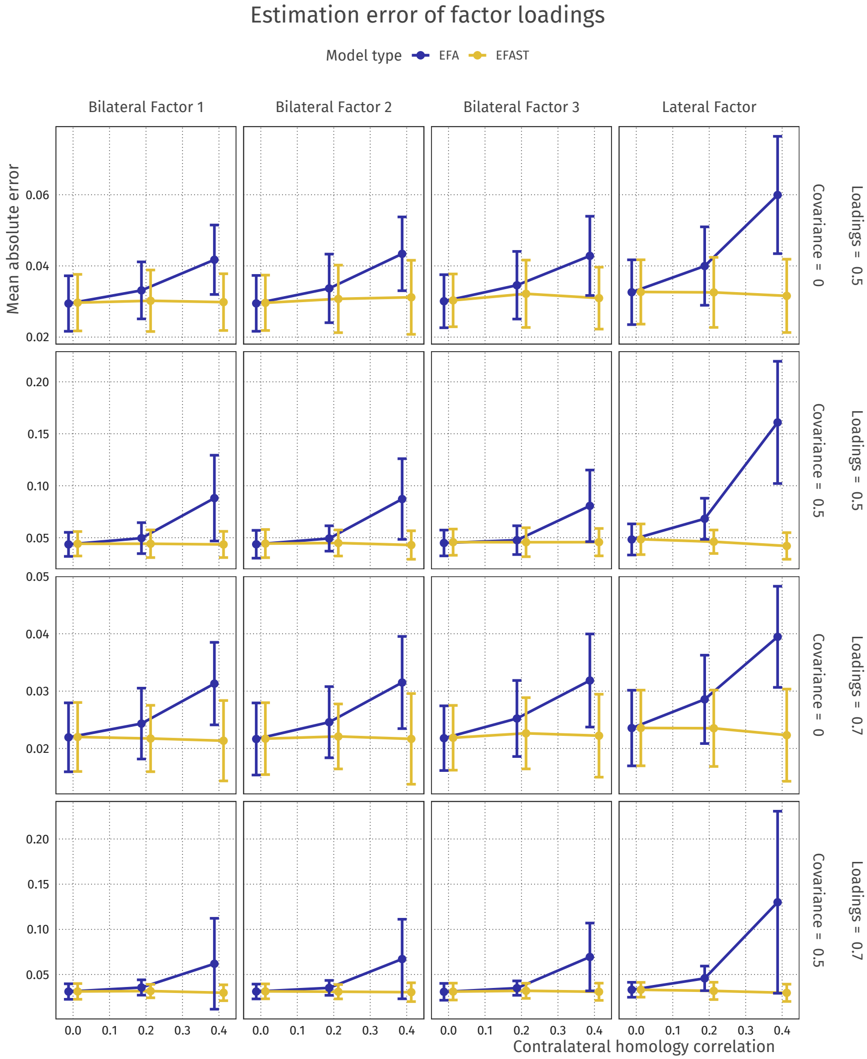

Figure B.2 Factor loading median absolute error over different conditions of factor loading and factor correlation strength (top-to-bottom, see labels on the right) and different factors (left-to-right, see labels on top).

B.3 Sample size in factor loading estimation error

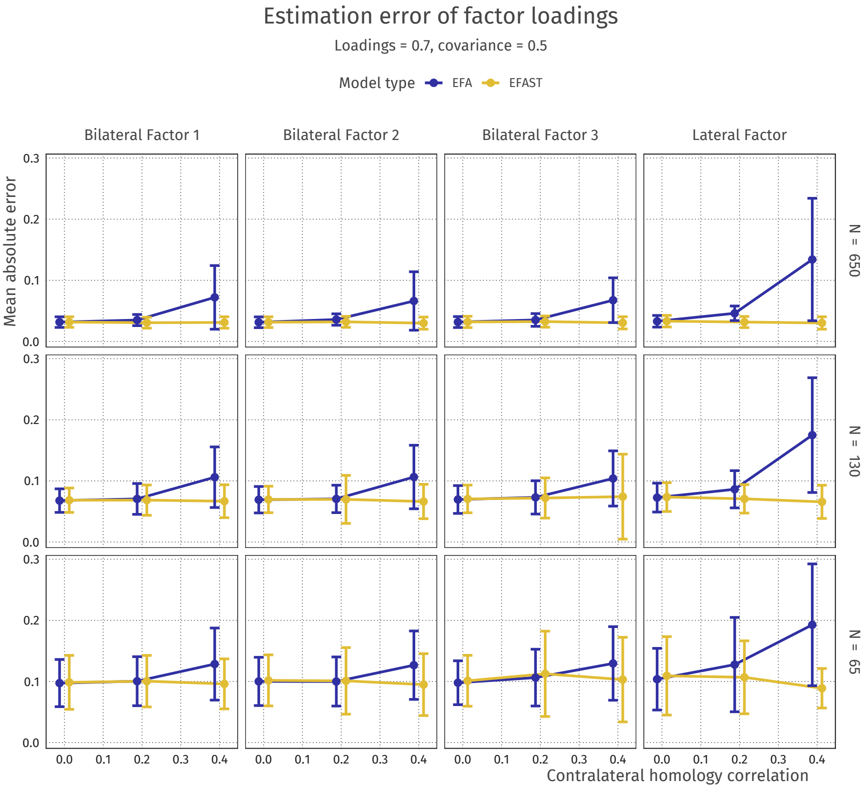

Figure B.3 Factor loading median absolute error over different sample sizes (top-to-bottom, see labels on the right) and different factors (left-to-right, see labels on top).

B.4 Sample size and model estimation convergence

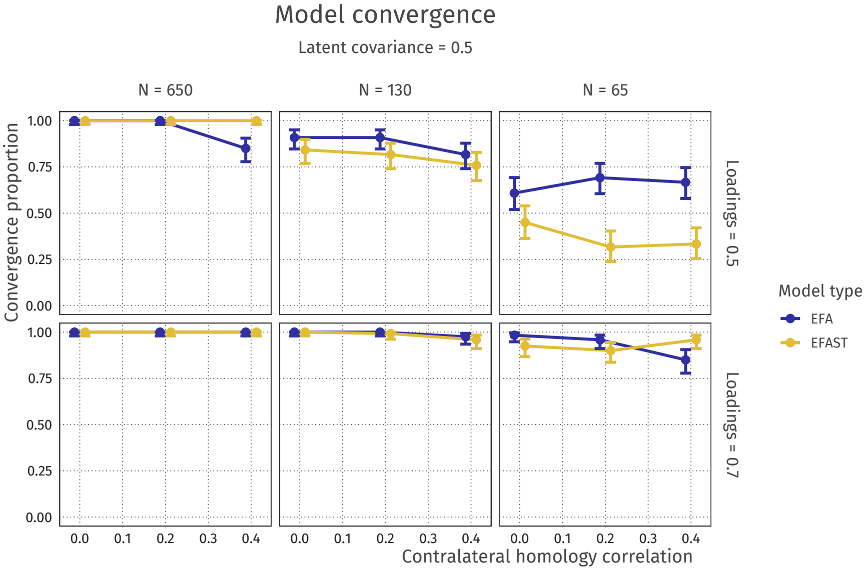

Figure B.4 Convergence rates of EFA and EFAST for different sample sizes (left-to-right, see labels on top). Convergence probability is not only determined by the sample size, but also by other factors such as the amount of latent covariance, the strength of the factor loadings, and the amount of symmetry.

B.5 Information criterion factor extraction performance

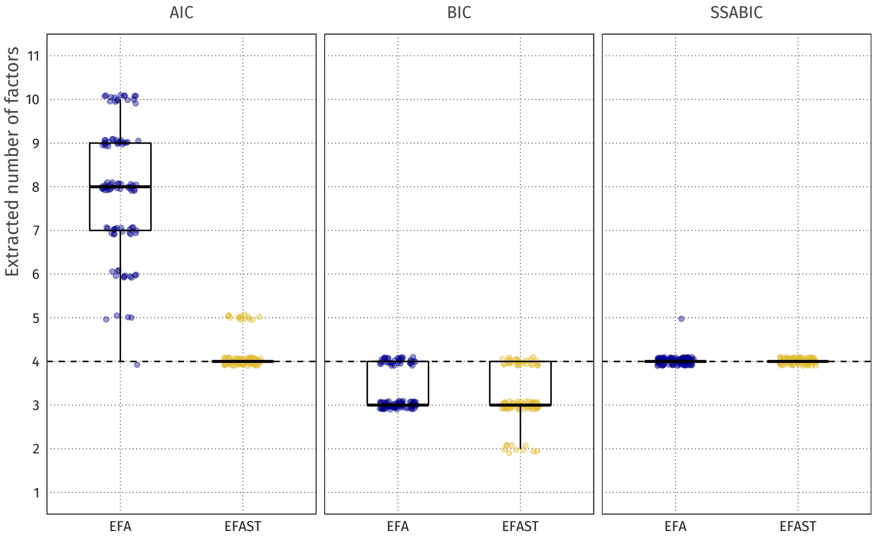

Figure B.5 Number of extracted factors using the AIC (left panel), BIC (middle panel), and sample-size adjusted BIC (right panel) criterion. AIC works well for the EFAST method but not for the EFA method. BIC slightly underextracts for both methods. SSABIC shows excellent performance for both methods. The true number of factors is 4 (dashed line), for which this result holds; different simulation situations may show different factor extraction patterns.

B.6 Factor loadings for empirical application

Table B.1 Factor loadings for 6-factor model fitted using EFAST and EFA on the Cam-CAN volume data. Loadings with absolute value below 0.3 not shown.

	EFAST						EFA					
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6
lh__bankssts	0.38				0.6		0.79					
lh__caudalanteriorcingulate				-0.4	0.55							
lh__caudalmiddlefrontal					0.73						0.74	
lh__cuneus		0.91						0.87				
lh__entorhinal			0.31									
lh__fusiform					0.4		0.38					
lh__inferiorparietal					0.69		0.67					
lh__inferiortemporal					0.4		0.54					
lh__isthmuscingulate					0.44							
lh__lateraloccipital		0.45					0.37	0.46				
lh__lateralorbitofrontal					0.78						0.71	
lh__lingual		0.71						0.7				
lh__medialorbitofrontal					0.65						0.63	
lh__middletemporal	0.31				0.66		0.81					
lh__parahippocampal					0.37							
lh__paracentral					0.83						0.73	
lh__parsopercularis					0.79						0.72	
lh__parsorbitalis					0.61						0.63	
lh__parstriangularis					0.84						0.89	
lh__pericalcarine		0.91						0.91				
lh__postcentral					0.8		0.37				0.33	
lh__posteriorcingulate					0.72						0.52	
lh__precentral			-0.3		0.87				-0.34		0.72	
lh__precuneus					0.76							0.74
lh__rostralanteriorcingulate					0.59						0.49	
lh__rostralmiddlefrontal					0.78						0.82	
lh__superiorfrontal					0.88						0.93	
lh__superiorparietal					0.72							0.77
lh__superiortemporal					0.77		0.52				0.41	
lh__supramarginal					0.77		0.32					
lh__frontalpole					0.32						0.51	
lh__temporalpole			0.39						0.34			
lh__transversetemporal					0.72					0.43	0.53	
lh__insula					0.78						0.7	
rh__bankssts					0.71		0.69					
rh__caudalanteriorcingulate				0.73							0.48	
rh__caudalmiddlefrontal					0.76						0.74	
rh__cuneus		0.71						0.72				
rh__entorhinal			0.34						0.34			
rh__fusiform					0.5		0.4					
rh__inferiorparietal	0.31				0.7		0.67					
rh__inferiortemporal	0.31				0.38		0.58					
rh__isthmuscingulate					0.45							
rh__lateraloccipital		0.47					0.37	0.47				
rh__lateralorbitofrontal					0.73						0.72	
rh__lingual		0.66						0.68				
rh__medialorbitofrontal					0.74						0.67	
rh__middletemporal					0.66		0.75					
rh__parahippocampal					0.39							
rh__paracentral					0.83						0.65	
rh__parsopercularis					0.76						0.68	
rh__parsorbitalis					0.62						0.82	
rh__parstriangularis					0.71						0.8	
rh__pericalcarine		0.77						0.82				
rh__postcentral					0.77		0.31				0.35	
rh__posteriorcingulate					0.57						0.44	
rh__precentral					0.96				-0.31		0.83	
rh__precuneus					0.76							0.72
rh__rostralanteriorcingulate				0.44							0.51	
rh__rostralmiddlefrontal					0.7						0.68	
rh__superiorfrontal					0.95						0.81	
rh__superiorparietal					0.82							0.71
rh__superiortemporal					0.91		0.39				0.46	
rh__supramarginal					0.82		0.36					0.43
rh__frontalpole												
rh__temporalpole			0.34									
rh__transversetemporal					0.78					0.39	0.43	
rh__insula					0.73						0.69	

Appendix C

Privacy-Preserving Generalized Linear Models using Distributed Block Coordinate Descent: Supplementary proof and simulation

C.1 Proof for recovery of standard errors

Let $A = X^T X$, partitioned into four submatrices A_{11} (held by Alice), A_{22} (held by Bob), and A_{22} (unknown to either). The standard inverse of such a partitioned, positive definite symmetric matrix is

$$\begin{aligned} A^{-1} &= \begin{pmatrix} B_{11} & B_{12} \\ B_{22} & B_{22} \end{pmatrix} \\ &= \begin{pmatrix} \left(A_{11} - A_{12}A_{22}^{-1}A_{12}^T\right)^{-1} & -A_{11}^{-1}A_{12}\left(A_{22} - A_{12}^T A_{11}^{-1}A_{12}\right)^{-1} \\ -A_{22}^{-1}A_{12}^T\left(A_{11} - A_{12}A_{22}^{-1}A_{12}^T\right)^{-1} & \left(A_{22} - A_{12}^T A_{11}^{-1}A_{12}\right)^{-1} \end{pmatrix} \end{aligned} \tag{C.1}$$

Following the procedure outlined in Section 3.3, Alice replaces X_2 with $V_2 = R_2 X_2$, and Bob replaces X_1 with $V_1 = R_1 X_1$, where R_j are unknown orthogonal rotation matrices. This gives two new matrices, $A^{(1)}$ and $A^{(2)}$, and their inverses, $B^{(1)}$ and $B^{(2)}$. By substition,

$$\begin{aligned} A_{12}^{(1)} &= X_1^T R_2 X_2 \\ A_{22}^{(1)} &= X_2^T R_2^T R_2 X_2 \end{aligned} \tag{C.2}$$

So that

$$\begin{aligned} B_{11}^{(1)} &= \left(A_{11}^{(1)} - A_{12}^{(1)}(A_{22}^{(1)})^{-1}(A_{12}^{(1)})^T\right)^{-1} \\ &= \left((X_1^T X_1) - (X_1^T R_2 X_2)(X_2^T R_2^T R_2 X_2)^{-1}(X_1^T R_2 X_2)^T\right)^{-1} \\ &= \left((X_1^T X_1) - (X_1^T X_2)(X_2^T X_2)^{-1}(X_1^T X_2)^T\right)^{-1} \\ &= \left(A_{11} - A_{12}A_{22}^{-1}A_{12}^T\right)^{-1} \\ &= B_{11} \end{aligned} \tag{C.3}$$

This shows that the part of the usual ACOV to do with $\hat{\beta}_1$ can be estimated correctly, and therefore the standard errors are available: $\mathrm{ACOV}(\hat{\beta}_j) = \sigma^2 B_{jj}$. Moreover,

$$\begin{aligned} B_{21}^{(1)} &= -(A_{22}^{(1)})^{-1}(A_{12}^{(1)})^T B_{11} \\ &= -(R_2^T R_2)^{-1} R_2 B_{21} \end{aligned} \tag{C.4}$$

so that

$$\begin{aligned} \left[(Z^T Z)^{-1} Z^T y\right]_{p_1} &= B_{11} X_1^T y - (R_2^T R_2)^{-1} R_2^T R_2 B_{21}^T X_2^T y \\ &= B_{11} X_1^T y - B_{21}^T X_2^T y \\ &= \hat{\beta}_1 \end{aligned} \tag{C.5}$$

This shows that the exact same estimates are obtained for $\hat{\beta}_1$. The same proof can be given for Bob and $\hat{\beta}_2$.

Note further that:

1. Alice cannot get $\hat{\beta}_2$ right because R_2 does not drop out in the other's part of the vector
2. We cannot get the ACOV of $(\hat{\beta}_1, \hat{\beta}_2)$ for this same reason

C.2 MSE of rank-R data approximation

The goal of this appendix is to show empirically the amount of explained variance when a set of parameters and their associated predictions are shared with another party. From Equation 11, but assuming all in-between parameter estimates $\hat{\boldsymbol{\beta}}_a^{(r)}$ are shared, *Bob* can create the following approximation:

$$\begin{aligned} \hat{\boldsymbol{Y}}_a &= \boldsymbol{X}_a \hat{\boldsymbol{B}}_a \\ \hat{\boldsymbol{X}}_a &= \hat{\boldsymbol{Y}}_a \hat{\boldsymbol{B}}_a^+ \end{aligned} \tag{C.6}$$

where $\hat{\boldsymbol{Y}}_a \in \mathbb{R}^{N \times R}$, $\boldsymbol{B}_a \in \mathbb{R}^{P \times R}$, $\boldsymbol{X}_a \in \mathbb{R}^{N \times P}$, all matrices are full rank, and A^+ indicates the Moore-Penrose pseudoinverse of A. For simplicity, but without loss of generality, we assume here that the variance of all the features in $\boldsymbol{X}_a$ is the same, σ_a^2, and these features are uncorrelated.

The relation between P, R, and the accuracy of the approximation $\hat{\boldsymbol{X}}_a$ is as follows: as $R \to P$, the MSE improves linearly, with perfect approximation being achieved when $R = P$. As mentioned in-text, when $P = 1$, sharing one set of parameters ($R = 1$) means the data can be recovered completely. Empirical simulations show that the relation between R, P, and expected mean square error of approximation is MSE $= \sigma_a^2(1 - R/P)$, where σ_a^2 is the variance of the features in $\boldsymbol{X}_a$ (see Figure C.1).

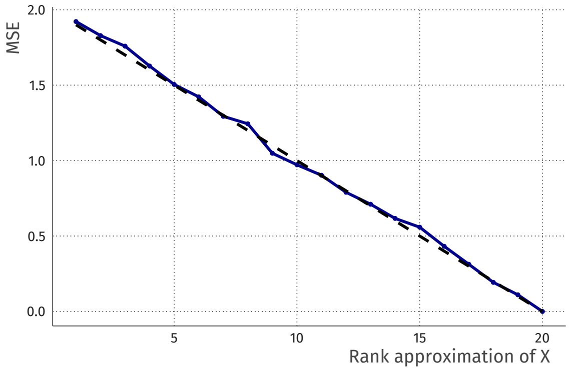

Figure C.1 Mean square error (MSE) of the approximation of the data $\boldsymbol{X}_a$ at Alice by Bob if $\hat{\boldsymbol{B}}_a$ is known. $\boldsymbol{X}_a$ was simulated as having $P = 20$ uncorrelated features with variance $\sigma_a^2 = 2$. Note that the approximation linearly improves as the rank of $\hat{\boldsymbol{B}}_a$ increases, with a perfect approximation reached when $R = P$. Dashed line indicates expected MSE, using the formula $E[\text{MSE}] = \sigma_a^2(1 - R/P)$.

Phrasing the above in terms of information sharing and privacy preservation: in sharing R sets of parameter estimates $\hat{\boldsymbol{\beta}}_a^{(r)}$ with their associated predictions $\hat{\boldsymbol{y}}_a^{(r)}$, Alice reveals a proportion of at least R/P of variance in the data. This proportion is a lower bound: in case there are correlations among the features of Alice, this proportion increases. When $R = P$ the data of Alice can be reconstructed by Bob. When either of a pair $(\hat{\boldsymbol{\beta}}_a^{(r)}, \hat{\boldsymbol{y}}_a^{(r)})$ are shared but not the other, no information is revealed.

Developed software

Software is a vital part of the current open-access, open-source, reproducible workflow of scientific research – especially in the field of statistics. Throughout my PhD, I have worked on several software projects, either directly or partially related to my research. In this chapter, I shortly outline the different software projects I have worked on. All projects are released under a permissive open-source license and easily accessible for others to use and improve upon.

JASP

`https://jasp-stats.org/`

During my PhD, I have contributed one day per week to the `JASP` statistics project, an open-source, free, friendly statistics program. At `JASP`, among other things I have implemented logistic regression, confirmatory factor analysis, and structural equation models, I have improved exploratory factor analysis, principal components analysis and descriptive statistics, and I have made a few improvements to the underlying infrastructure of the program.

tensorsem

`https://github.com/vankesteren/tensorsem`

`R` and `Python` package for structural equation modeling using Torch. The `R` interface parses `lavaan` code to create a structural equation model native to the `pytorch` machine learning framework. This model can then be estimated and extended using techniques from the field of deep learning, as implemented in `pytorch`.

cmfilter

`https://github.com/vankesteren/cmfilter`

`R` package to discover mediators among many potential variables. The internal routines are optimized `C++` code using the `Armadillo` linear algebra library and `OpenMP` parallelization pragmas. Because of this, it scales well to many cores, and has even been used on high-performance computing facilities such as the SurfSara Lisa compute cluster.

efast

`https://github.com/vankesteren/efast`

`R` package for exploratory factor analysis with structured residuals. Built on top of `lavaan` for maximum compatibility with existing structural equation models, but with a few stability and speed improvements applied, such as boundaries for the estimated residual covariances.

privreg

`https://github.com/vankesteren/privreg`

`R` package for privacy-preserving generalized linear models using coordinate descent. Full two-party privacy-preserving computation, using AES encryption for the communication. Built on native `R` `glm` procedures.

firatheme

`https://github.com/vankesteren/firatheme`

`R` package for a clean and colourful `ggplot2` theme using open-source Fira fonts. I have used `firatheme` throughout this thesis, and it has been used in publications by several other researchers.

rpeaks

`https://github.com/vankesteren/rpeaks`

`R` package for fast detection of heartbeats in electrocardiogram data. A reimplementation (in optimized `C++` code) of the famous Pan-Tompkins algorithm (Pan & Tompkins, 1985) to analyze ambulant ECG measurements of several days.

Massign

`https://github.com/vankesteren/Massign`

`R` package for simple matrix construction for prototyping. Introduces into `R` the `%<-%` operator for matrix assignment, with convenient shortcuts for symmetric matrices.

vennvis

`https://github.com/vankesteren/vennvis`

`R` package for Venn diagram visualisation of variable covariances. Displays two to three variables as Venn diagrams, where the area of the circles is proportional to the variance of each variable, and the overlap is proportional to the covariance. The layout is computed using line search optimization.

Rijkspalette

`https://github.com/vankesteren/rijkspalette`

`R` package to generate colour palettes for graphs using Rijksmuseum paintings. Interfaces with the excellent API of the Rijksmuseum to fetch paintings, and then performs k-means clustering on their colour pixels in `a*b*` space for determining the colour palette.

Nederlandse samenvatting

Structurele vergelijkingsmodellering (Structural Equation Modeling; SEM) is een flexibele en populaire methode voor data-analyse in de sociale- en gedragswetenschappen. SEM is vooral geschikt voor onderzoekssituaties waarin concepten niet direct gemeten kunnen worden, of waarin de meetinstrumenten foutgevoelig zijn. Denk hierbij aan concepten als "vertrouwen" of "welvaart" die indirect gemeten worden met vragenlijsten. Maar het moderne datalandschap verandert, en SEM bereikt daarin haar grenzen: klassieke vragenlijstonderzoeken en experimenten worden aangevuld met (en soms zelfs vervangen door) metingen uit registerdata, draagbare sensoren, foto's, internetdatabases, genetische sequenties, geavanceerde brein-beeldvormingstechnieken, en meer. De SEM methode is hiervoor niet altijd beschikbaar, maar de problemen van foutgevoelige metingen zijn in dit soort moderne data onverminderd groot en veel onderzoeksvragen gaan nog altijd over relaties tussen moeilijk meetbare concepten. Analyses die gebruik maken van SEM zijn hierdoor van grote waarde voor onderzoek met zulke nieuwe meetinstrumenten. Het doel van dit proefschrift is dan ook om SEM analyses beschikbaar te maken voor een groter bereik aan moderne datasets. Om dit doel te bereiken presenteer ik verschillende oplossingen voor problemen die men tegenkomt bij het toepassen van SEM op zulke datasets.

In Hoofdstuk 2 introduceer ik ten eerste een methode om structurele vergelijkingsmodellen op een nieuwe manier te specificeren en schatten. Hierbij leen ik methodologie uit het veld van "deep learning" en neurale netwerken. Met deze methode worden aanpassingen zoals regularisatie – wat veel gebruikt wordt voor moderne data-analyse – in één keer beschikbaar voor SEM. Ik toon dit aan door middel van drie verschillende voorbeelden waarin ik nuttige, niet eerder vertoonde uitbreidingen maak voor klassieke structurele vergelijkingsmodellen.

In Hoofdstuk 3 ontwikkel ik een algoritme om mediatie-analyse (een speciaal geval van SEM) uit te voeren op hoog-dimensionele, epigenetische sequentiedata. Het probleem bij deze data is de grote hoeveelheid metingen per observatie, tot wel honderdduizenden waarden. Het algoritme dat ik ontwikkel is een alternatief voor de klassieke SEM schattingsmethode, welke überhaupt niet met dit soort situaties om kan gaan. Ik maak gebruik van de grote hoeveelheid computerkracht die beschikbaar is om een benadering te maken van "gewone" mediatieanalyse en ik toon aan dat dit in bepaalde situaties beter werkt dan andere beschikbare methoden voor zulk onderzoek.

In Hoofdstuk 4 ontwikkel ik een uitbreiding voor exploratieve factoranalyse (EFA; een ander speciaal geval van SEM) voor brein-beeldvormingsdata. Procedures als factoranalyse worden veel gebruikt bij dit soort data omdat het de hoeveelheid data vermindert met zo min mogelijk verlies van informatie – een belangrijke stap voor vervolgonderzoek naar bijvoorbeeld de ontwikkeling van het brein. De uitbreiding die ik presenteer maakt gebruik van de specifieke voorkennis dat het brein nagenoeg symmetrisch is, iets wat nog niet eerder is gedaan in de context van EFA. Met verschillende voorbeelden van structurele en functionele breindata toon ik de flexibiliteit van de uitbreiding, en ik laat zien dat deze methode een verbetering is ten opzichte van standaard EFA.

In Hoofdstuk 5 presenteer ik een oplossing voor het probleem van data-analyse in de context van verticaal gedistribueerde data, dat wil zeggen tabellen waarbij de kolommen op verschillende plaatsen zijn opgeslagen. Dit komt bijvoorbeeld voor in privacygevoelige situaties met medische data. De oplossing stelt twee partijen in staat om samen een gegeneraliseerd regressiemodel te schatten – inclusief standaardfouten – door enkel hun lineaire voorspelling van de uitkomstvariabele te delen. Met enkele toegepaste voorbeelden presenteer ik een implementatie van deze oplossing, inclusief encryptie om de gedistribueerde berekeningen veilig uit te voeren.

Tot slot stel ik in Hoofdstuk 6 een structureel vergelijkingsmodel voor om een ander modern dataprobleem aan te pakken: algoritmische rechtvaardigheid. Dit hoofdstuk is gebaseerd op een situatie waarin medische voorspellingen op basis van registerdata leiden tot een bevooroordeelde behandeling van blanke patiënten ten opzichte van zwarte patiënten. Door gebruik te maken van een klassiek latente variabelemodel in combinatie met bestaande technieken voor rechtvaardige *machine learning* verdwijnt het probleem in een toegepaste dataset vrijwel volledig.

Curriculum vitae

2004 – 2010 Walburg College Zwijndrecht

Erik-Jan attended secondary school in Zwijndrecht, a town between Dordrecht and Rotterdam. In addition to the required programme (e.g., Dutch, English, mathematics) his curriculum included a profile of economics, arts, physics, chemistry, German, and Latin.

2010 – 2013 University College Utrecht

Erik-Jan pursued a bachelor's degree in Liberal Arts & Sciences at University College Utrecht, with courses in social science (mainly psychology, economics, and sociology) and minors in neuroscience and statistics. His bachelor's thesis was in the field behavioural economics, on the improvement of pro-social behaviour using "nudges" in economic games.

2013 – 2014 London School of Economics & Political Science

Erik-Jan's first master's degree was in Organisational and Social Psychology. He wrote a dissertation on the effects of virtual teamwork on interpersonal trust in organizations, collecting quantitative and qualitative data in a large multinational consultancy. The survey data was analyzed using a structural equation model.

2014 – 2015 Douwe Egberts Master Blenders 1753 (Utrecht)

At Douwe Egberts, Erik-Jan was a "data professional", managing data concerning business-to-business customers of coffee machines. Here, he later worked in a project team, designing and executing several technical improvements for data processes within the company. Examples of such improvements are data-based segmentation of the Netherlands for the sales division, and live dashboards for on-demand coffee machine maintenance.

2015 – 2017 Utrecht University

Erik-Jan pursued a second master's degree in Methodology and Statistics for the Behavioural, Biomedical, and Social Sciences. In the second year, he did an elective internship at JASP, implementing new statistical procedures. His thesis was written at UMC Utrecht (Julius Center) on the effect of bivariate covariance on feature selection for class prediction models in high-dimensional genetic data. He also became a board member of *Young Statisticians*, the early-career section of the Netherlands Society for Statistics and Operations Research (VVSOR).

2017 – 2020 JASP (University of Amsterdam)

Erik-Jan was a part-time (one day per week) statistical programmer for JASP during his PhD period, implementing procedures such as logistic regression, confirmatory factor analysis, structural equation modeling, MIMIC models, latent growth curve models, exploratory factor analysis, principal components analysis and descriptive statistics. He also contributed to the overall software architecture of JASP, both in the front-end and the back-end.

2017 – 2020 Utrecht University

PhD candidate at the department of Methodology & Statistics under daily supervision of Dr. Daniel L. Oberski. Erik-Jan was funded by the Netherlands Organization for Scientific Research (NWO), under talent grant project *New Dimensions in Social Science: Extending Structural Equation Models to Accommodate Novel Data Sources*. During this time, Erik-Jan worked with Dr. Rogier Kievit on a research visit to the UK, at the University of Cambridge Cognition and Brain Sciences Unit. His teaching duties included master's thesis supervision, a yearly recurring master's course on data analysis and visualisation, and an `R` programming summer course.

List of publications

Boeschoten, L.[1], **Van Kesteren, E. J.**[1], Bagheri, A., & Oberski, D. L. (2020, accepted). Fair inference on error-prone outcomes. *ECAI conference 2020 workshop: Artificial Intelligence for a Fair, Just and Equitable World.* `arXiv:2003.07621`

Ly, A., Stefan, A., van Doorn, J., Dablander, F., van den Bergh, D., Sarafoglou, A., ..., **Van Kesteren, E. J.**, ... & Wagenmakers, E. J. (2020). The Bayesian Methodology of Sir Harold Jeffreys as a Practical Alternative to the P-value Hypothesis Test. *Computational Brain & Behavior, 3*(2), 153-161. `doi:10.1007/s42113-019-00070-x`

Van den Bergh, D., Van Doorn, J., Marsman, M., Draws, T., **Van Kesteren, E. J.**, Derks, K., ... & Sarafoglou, A. (2020). A Tutorial on Conducting and Interpreting a Bayesian ANOVA in JASP. *L'Annee psychologique, 120*(1), 73-96. `doi:10.3917/anpsy1.201.0073`

Van Kesteren, E. J. & Kievit, R. A., (2020, in press). Exploratory Factor Analysis with Structured Residuals for Brain Network Data. *Network Neuroscience.* `doi:10.1162/netn_a_00162`

Van Kesteren, E. J., & Oberski, D. L. (2019). Exploratory mediation analysis with many potential mediators. *Structural Equation Modeling: A Multidisciplinary Journal, 26*(5), 710-723. `doi:10.1080/10705511.2019.1588124`

Van Kesteren, E. J., & Oberski, D. L. (2019, in preparation). Structural equation models as computation graphs. *arXiv preprint.* `arXiv:1905.04492.`

Van Kesteren, E. J., Sun, C., Oberski, D. L., Dumontier, M., & Ippel, L. (2019, in preparation). Privacy-Preserving Generalized Linear Models using Distributed Block Coordinate Descent. *arXiv preprint.* `arXiv:1911.03183`

Wagenmakers, E. J., Love, J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., ..., **Van Kesteren, E. J.**, ... & Morey, R.D. (2018). Bayesian inference for psychology. Part II: Example applications with JASP. *Psychonomic Bulletin & Review, 25*(1), 58-76. `doi:10.3758/s13423-017-1323-7`

[1] Shared first authorship

Dankwoord

Dit proefschrift zou er niet geweest zijn zonder een groot aantal mensen in mijn leven, niet alleen in de afgelopen jaren maar ook daarvoor. Hier wil ik al deze mensen bedanken.

Daniel, je was er nog voor het begin van mijn PhD bij, zelfs voordat je in Utrecht werkte. Ik weet nog dat we elkaar in 2016 ontmoetten bij Kriterion om de beursaanvraag te bespreken, en in die ontspannen eerste meeting wist ik al dat we goed bij elkaar zouden passen en dat ik ontzettend veel van je kon leren. Dat is ook gebleken, want als ik dit proefschrift tijdens die eerste meeting had kunnen lezen was ik van mijn stoel gevallen. Bijna alles wat je hier leest is op één of andere manier ook jouw product, van de meest overkoepelende thema's tot de kleinste details in de zinsopbouw. Bedankt daarvoor, en ook voor je losse humor en begeleiding op alle niveaus.

Onze eerste meeting bleek overigens zó ontspannen dat we de beursaanvraag uiteindelijk urenlang samen in je kantoor hebben moeten afmaken op de dag van de deadline.

Irene, je hebt me geholpen om überhaupt te beginnen aan het PhD traject, en samen met jou heb ik de eerste scriptiebegeleiding gedaan – dit was zo'n goede ervaring dat ik dacht dat een academische baan zo slecht nog niet zou zijn. Bedankt dat je er altijd voor hebt gezorgd dat het allemaal goed ging. René, mijn hoog-dimensionele masterscriptie bij jou in het UMC was de inspiratie voor mijn onderwerp. Bedankt dat je er was bij mijn eerste stappen in de wereld van statistische simulatie.

Rogier, je hebt mijn academische uitwisseling niet alleen mogelijk gemaakt, maar ook ontzettend verrijkt. Onze samenwerking verliep ongelooflijk vloeiend, en ik heb echt veel van je geleerd. Je bent een voorbeeld voor zo velen, bedankt dat je dat ook voor mij was – ik kijk uit naar onze vloeiende samenwerkingen in de toekomst.

Chang & Lianne, in Maastricht heb ik eigenlijk ook een soort mini-uitwisseling gedaan. Door jullie is het distributed estimation idee een volwaardig onderzoek geworden: bedankt voor de geweldige samenwerking.

Laura, we werken al zo lekker samen en dat gaan we in de toekomst nog meer doen. Bedankt natuurlijk voor die samenwerking, maar ook zeker voor alle chats – de gekke thuiswerktijd was gezelliger door jou.

Ayoub & Oisín, jullie zijn de aller-allerbeste C1.22 kamergenoten. In de afgelopen jaren zijn we niet alleen hele goede collega's geworden, maar ook hele goede vrienden. Jullie zijn allebei fantastisch, en ik kijk uit naar alle dingen die we nog samen gaan doen. Ayoub, رفيق من, je moet minder werken en wat vaker nee zeggen; Oisín, je moet je wat minder opwinden over dingen. Ik hoop dat we snel weer terug kunnen naar onze kamer en daar samen koffie drinken, brainstormen over allerlei ideeën, zeuren over van alles, elkaar op deadlines wijzen, en gewoon kletsen.

Ik wil ook graag de anderen bij methoden en statistiek bedanken – wat een gezellige afdeling. Anastasia, Anne (bedankt voor de dropjes), Duco (bedankt voor de planks), Ellen (bedankt voor het boek van Bollen dat ik gedurende mijn PhD van je heb gestolen), Els, Fayette, Gerko, Herbert, Jeroen, Karlijn, Kevin, Lientje (bedankt voor de liters koffie en wandelingen in de botanische tuinen), Peter L, Peter vd H, Qixiang, Sanne, Shiva, en alle anderen. Ik heb zin om iedereen weer te kunnen zien en horen bij de lunch.

De mensen van het JASP team op de UvA zijn ook onmisbaar geweest. In de afgelopen jaren hebben jullie meer bijgedragen aan deze scriptie dan jullie denken. Die dag per week bij jullie gaf me altijd nieuwe energie en ideeën om aan het proefschrift verder te gaan. Eric-Jan, Bruno, Joris, Frans, Tim, Alexandra, Akash, Don, Alexander, Koen, Quentin, Johnny, en alle anderen, bedankt!

Jesse, Merel, Iris, Anne, Marc – jullie hebben me sinds 2010 altijd gesteund in alles wat ik doe. Jullie zullen altijd een deel van mijn leven zijn en ik koester onze vriendschap. Er zijn nog veel meer mensen die ik zou willen bedanken omdat ze een deel zijn van mijn leven; Ruben, Roline, Kees, Lisa, Oğuzhan, Rob – allemaal bedankt. Als je jezelf hier niet terug vindt maar wel had verwacht, dan hoor je er gewoon bij en dus natuurlijk ook bedankt.

Mama, papa, Nadia, Danique, bij jullie voel ik me thuis en is al het andere even onbelangrijk. Ik hoop dat we (als het weer kan) nog heel vaak samen eten, skiën, chillen, en geforceerd lachen om de stomme grappen van papa.

Lara, je bent mijn partner, mijn beste vriendin, mijn thuiswerkcollega, en nu ook (hoe kan het ook anders) mijn paranimf. In twee jaar tijd hebben we samen al zoveel meegemaakt! Ik voel me ontzettend bevoorrecht om mijn leven met jou te mogen delen. Je bent de allerleukste persoon die er is, en ik kijk heel erg uit naar alles wat nog komt voor ons twee.