Structural Equation Models as Computation Graphs

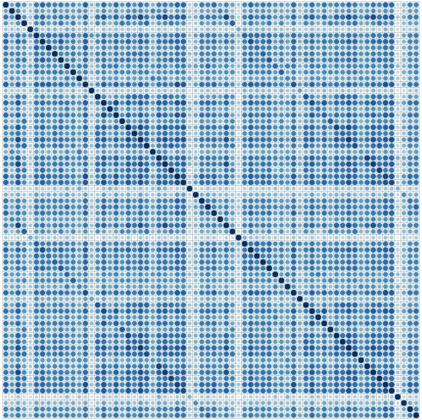
Erik-Jan van Kesteren & Daniel Oberski

Utrecht University, Methodology & Statistics

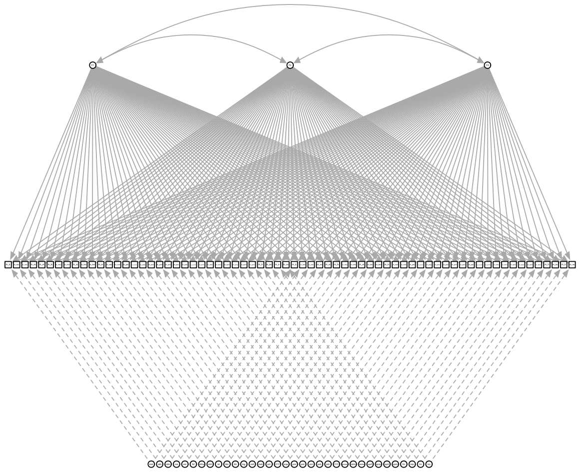
October 16, 2019

Primer: something else I'm working on

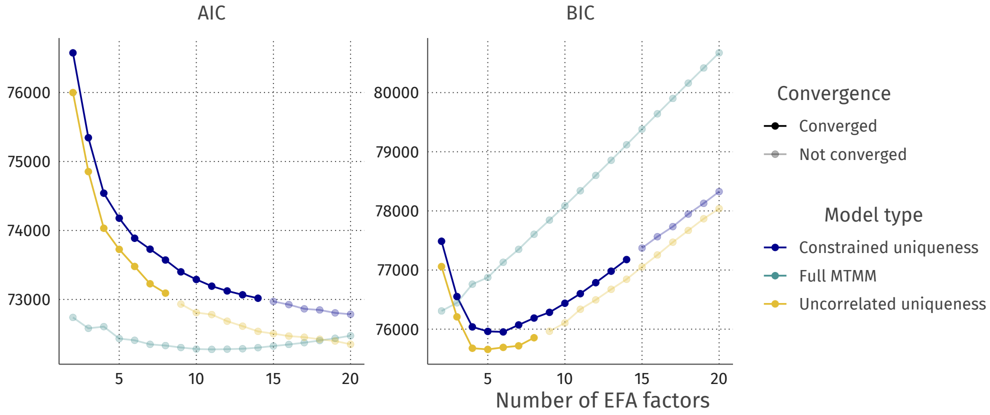
Brain structure covariance matrix (N $>$600)

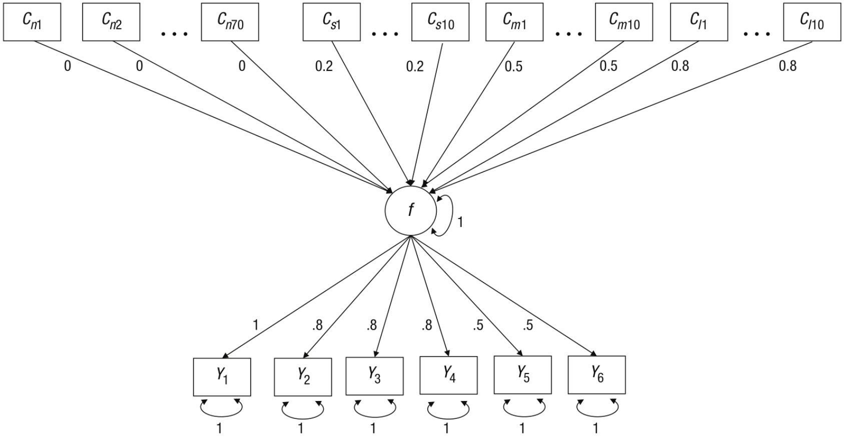


EFA with residual structure: a big model



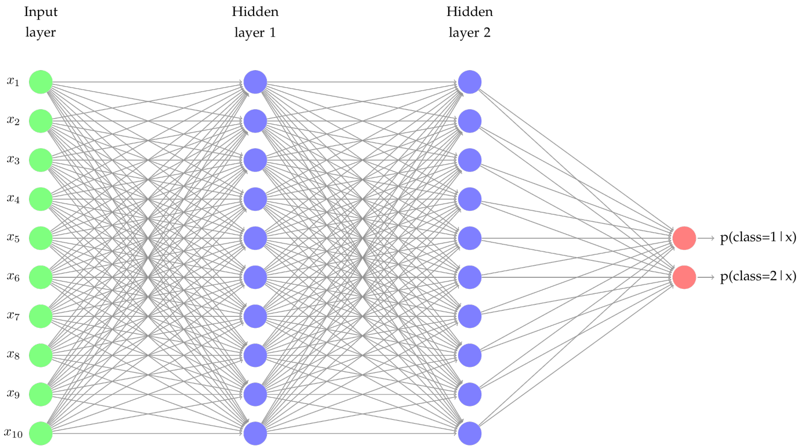
A big model :(


AIC
BIC
76000
75000
74000
73000
80000
79000
78000
77000
76000
5
10
15
20
Number of EFA factors
Convergence
Converged
Not converged
Model type
Constrained uniqueness
Full MTMM
Uncorrelated uniqueness



Jacobucci, Brandmaier & Kievit (2019)

A small model?



http://mdtux89.github.io/2015/12/11/torch-tutorial.html (2015)

- Some SEM models are overparameterized (e.g., when $P > N$)
- We can't estimate these models with default SEM
- Neural networks can be extremely overparameterized
- Deep learning software (e.g., TensorFlow) can still estimate these
- Can we use deep learning methods for SEM?

Deep learning part 1: adaptive first-order optimizers

Deep learning part 2: computation graphs

The SEM computation graph

Extending SEM

R package showcase

Deep learning part 1: adaptive first-order optimizers

Adaptive first-order optimizers

The loss function is a function of parameters

$$F(\boldsymbol{\theta})$$

Outputs a single number, a distance metric (e.g., expected - observed; sum of squared residuals)
In statistics often log-likelihood $F(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{y})$

SEM maximum likelihood loss function

$$
\begin{aligned}
\theta &= \{B_{\mathbf{0}}, \mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Theta}\} \\
B &= (I - B_{\mathbf{0}}) \\
\mathbf{\Sigma} &= \mathbf{\Lambda} B^{-1} \mathbf{\Psi} B^{-T} \mathbf{\Lambda}^{T} + \mathbf{\Theta} \\
F_{ML}(\theta) &= log|\mathbf{\Sigma}| + tr\left[S\mathbf{\Sigma}^{-1}\right]
\end{aligned}
$$

Optimizers find the parameters $\hat{\boldsymbol{\theta}}$ for which the loss is minimum (maximum)

At the minimum, the gradient of the loss $\nabla F(\boldsymbol{\theta})$ – the vector of partial derivatives $\left[\frac{\partial F}{\partial \theta_1}, \frac{\partial F}{\partial \theta_2}, \frac{\partial F}{\partial \theta_3}, \ldots, \frac{\partial F}{\partial \theta_P}\right]$ – is $\mathbf{0}$

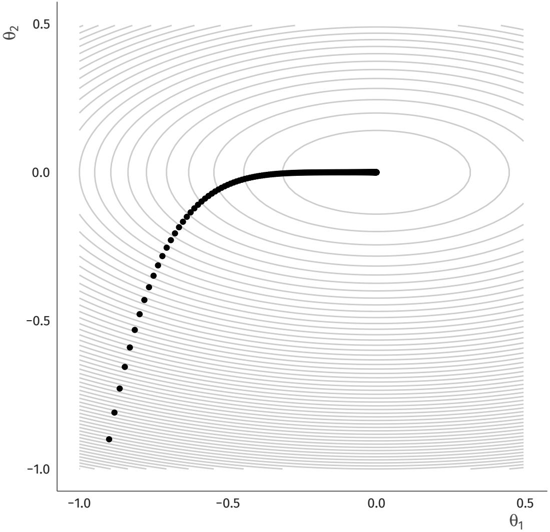
If there is no closed-form solution for $\nabla F(\boldsymbol{\theta}) = \mathbf{0}$, optimizers may still find the minimum in an iterative way: $\hat{\boldsymbol{\theta}}^{(1)}, \hat{\boldsymbol{\theta}}^{(2)}, \hat{\boldsymbol{\theta}}^{(3)}, \ldots, \hat{\boldsymbol{\theta}}^{(R)}$

Gradient descent

Simplest iterative algorithm: take steps of size s in the direction of the negative gradient (Cauchy, 1847)

$$\hat{\boldsymbol{\theta}}^{(i)} = \hat{\boldsymbol{\theta}}^{(i-1)} - s \cdot \nabla F(\hat{\boldsymbol{\theta}}^{(i-1)})$$

Gradient descent for $F(\boldsymbol{\theta}) = \theta_1^2 + 5\theta_2^2$



Gradient descent can get stuck in local minima and determining the step size is difficult

Deep learning field developed two main improvements to gradient descent

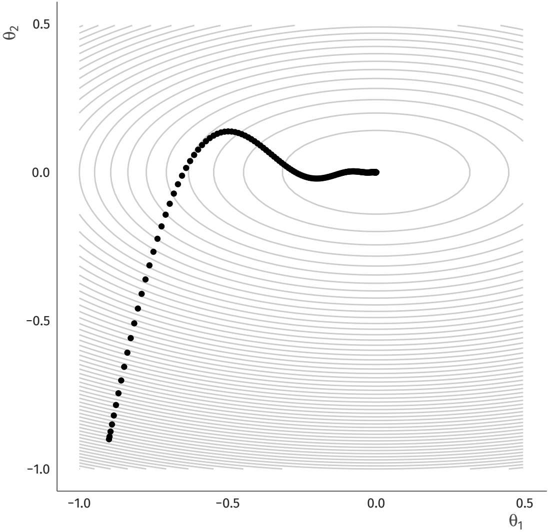
Momentum and **adaptive step-size**

Momentum

Instead of using the gradient, use a *moving average* of the history of gradients, for example with a decay of 0.99:

$$\boldsymbol{v}^{(i)} = 0.99 \cdot \boldsymbol{v}^{(i-1)} + (1 - 0.99) \cdot \nabla F(\hat{\boldsymbol{\theta}}^{(i-1)})$$
$$\hat{\boldsymbol{\theta}}^{(i)} = \hat{\boldsymbol{\theta}}^{(i-1)} - s \cdot \boldsymbol{v}^{(i)}$$

Gradient descent with momentum for $F(\boldsymbol{\theta}) = \theta_1^2 + 5\theta_2^2$



https://distill.pub/2017/momentum/

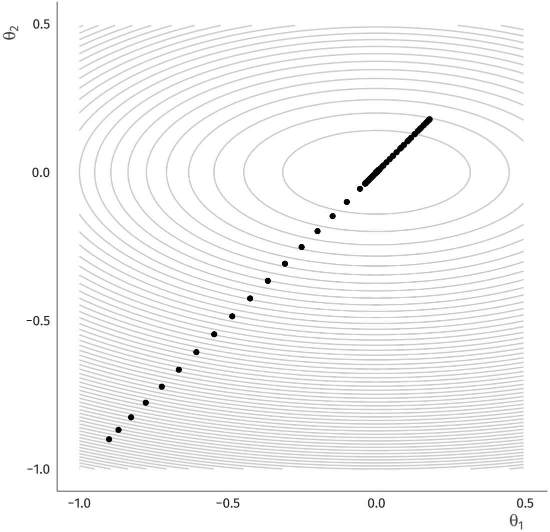
Adaptive step-size

Newer algorithms such as Adam (Kingma & Ba, 2014) also include adaptive step-size

Idea: edit the step size *per parameter* based on how variable the gradient in that direction is

Less variation in the history of gradients $\rightarrow$ larger steps

Adam for $F(\boldsymbol{\theta}) = \theta_1^2 + 5\theta_2^2$



Adam is one of the most popular optimisation algorithms for deep neural networks

Robust against many kinds of loss function abnormalities (e.g., local minima)

Note: for well-behaved* functions, Fisher scoring is still *way* better

*convex, twice continuously differentiable, not too many parameters, etc.
(please don't quote this fuzzy statement)

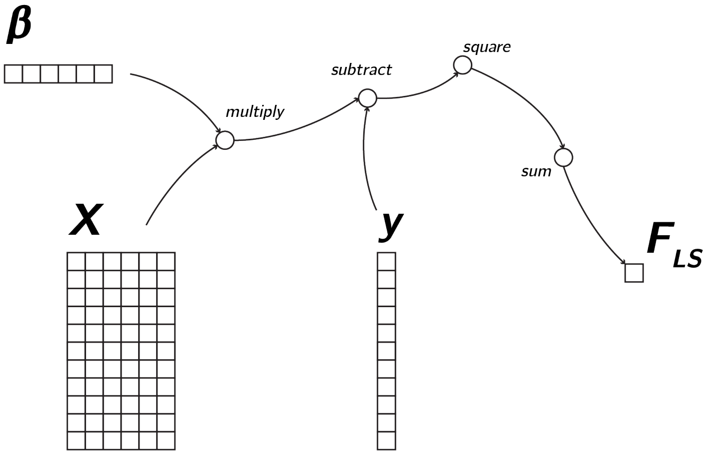
Deep learning part 2: computation graphs

Computation graphs

Describe operations from parameters to loss function

$$\boldsymbol{\theta} \to F(\boldsymbol{\theta})$$

Least squares regression computation graph



Computation graphs

Software can automatically compute $\nabla F(\boldsymbol{\theta})$ (autograd)
Software implements optimisation algorithms (e.g., Adam)

TensorFlow

Computation graphs

Computation graph $+$ software $\rightarrow$ parameter estimation

The SEM computation graph

SEM computation graph

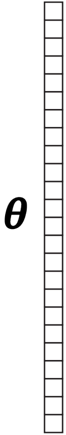
Describe operations from parameters to loss function

$$\boldsymbol{\theta} \to F(\boldsymbol{\theta})$$

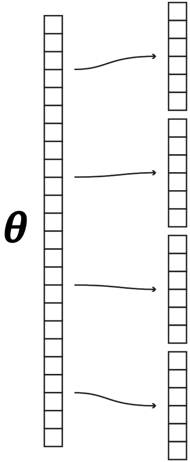
SEM computation graph

$$
\begin{aligned}
\theta &= \{B_{\mathbf{0}}, \mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Theta}\} \\
B &= (\boldsymbol{I} - B_{\mathbf{0}}) \\
\mathbf{\Sigma} &= \mathbf{\Lambda} B^{-1} \mathbf{\Psi} B^{-T} \mathbf{\Lambda}^{T} + \mathbf{\Theta} \\
F_{ML}(\theta) &= log|\mathbf{\Sigma}| + tr\left[\boldsymbol{S}\mathbf{\Sigma}^{-1}\right]
\end{aligned}
$$

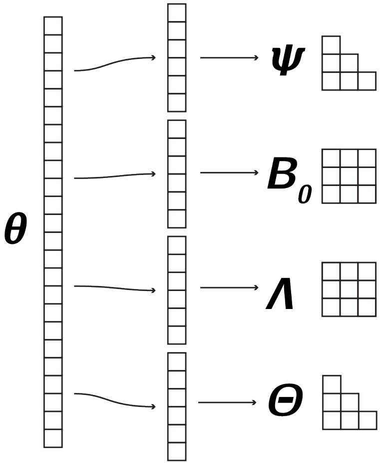
SEM computation graph



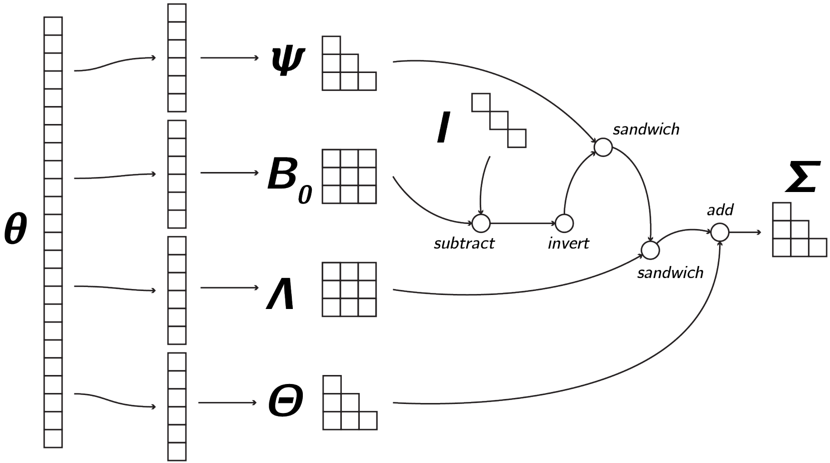
SEM computation graph



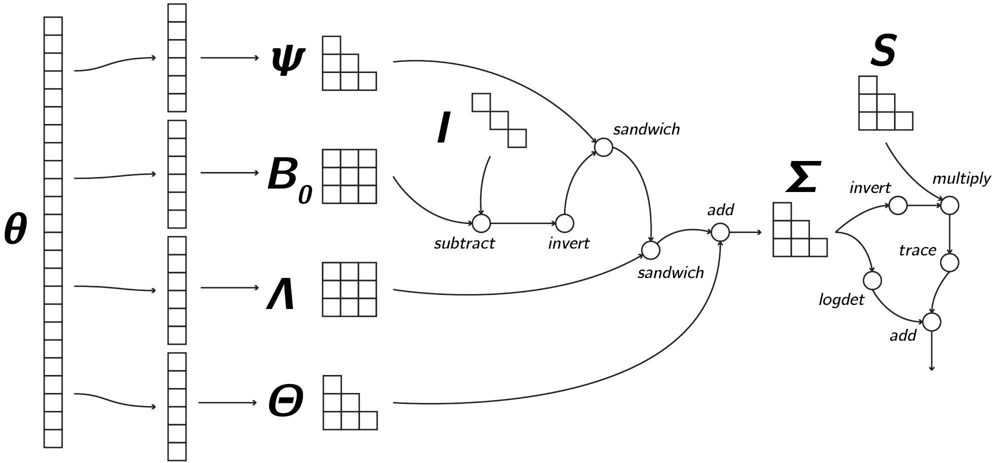
SEM computation graph



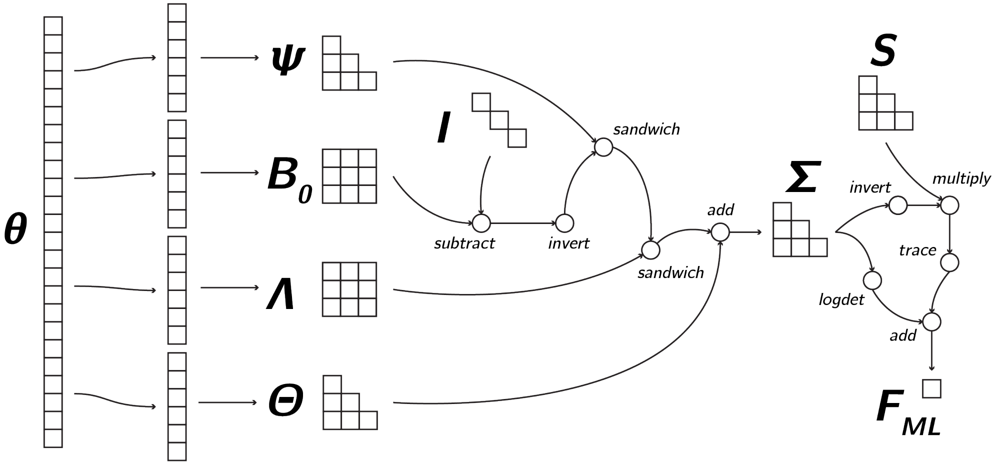
SEM computation graph



SEM computation graph



SEM computation graph

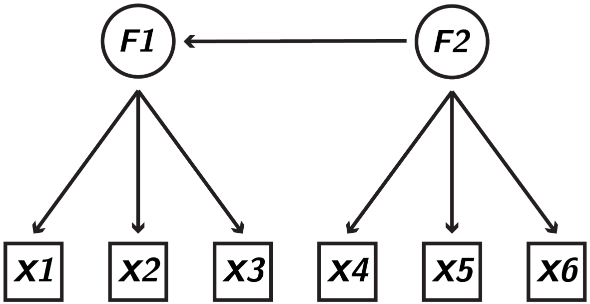


SEM computation graph

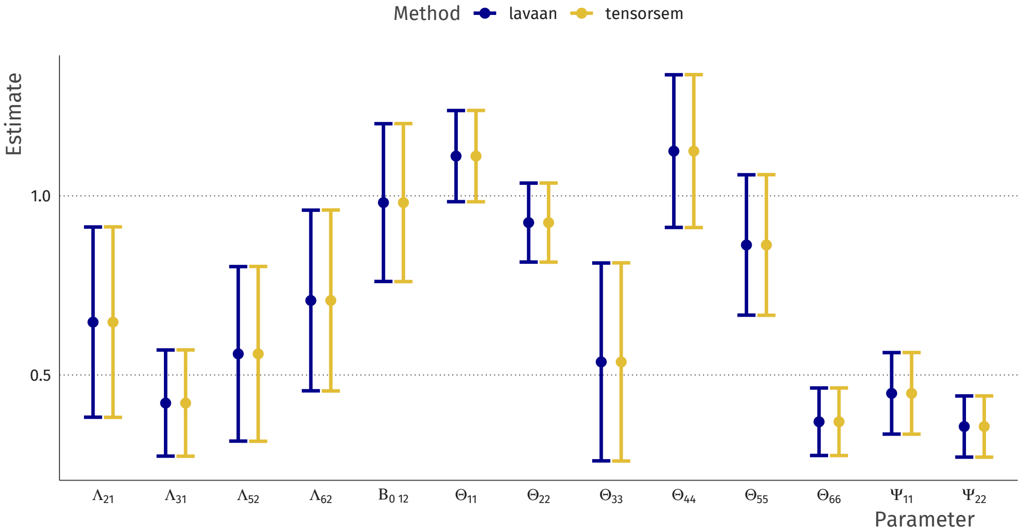
What can we do now

- Get gradient $\nabla F_{ML}(\boldsymbol{\theta})$
- Get Hessian $H_\theta = H\left[F_{ML}(\boldsymbol{\theta})\right]$
- Get standard errors: $SE_\theta \approx \sqrt{diag\left[H_\theta^{-1}\right]}$
- Fit SEM models using smart optimiser (e.g., Adam)

Example



Example



Method
lavaan
tensorsem
Estimate
1.0
0.5
Λ_{21}
Λ_{31}
Λ_{52}
Λ_{62}
$B_{0\,12}$
Θ_{11}
Θ_{22}
Θ_{33}
Θ_{44}
Θ_{55}
Θ_{66}
Ψ_{11}
Ψ_{22}
Parameter

Extending SEM

Extending SEM

Now we can edit the objective:

- Different objective
- Penalise structural paths
- Penalise factor loadings

Sociological Methods & Research
Volume 36 Number 2
November 2007 227-265

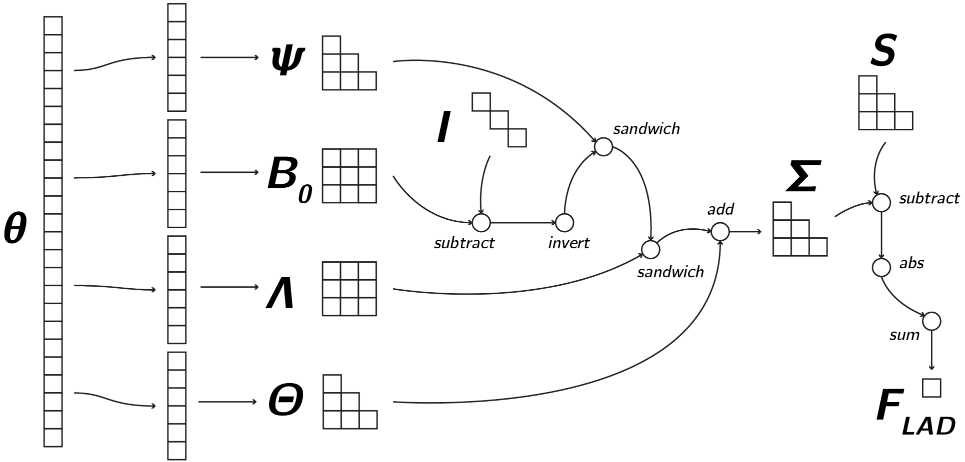
10.1177/0049124107301946
http://smr.sagepub.com
hosted at
http://online.sagepub.com

Least Absolute Deviation Estimation in Structural Equation Modeling

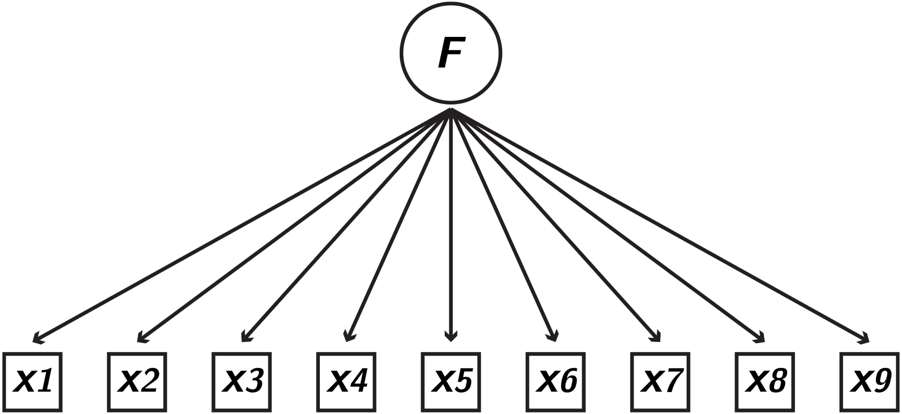
Enno Siemsen
University of Illinois, Urbana-Champaign
Kenneth A. Bollen
University of North Carolina, Chapel Hill

Least absolute deviation (LAD) is a well-known criterion to fit statistical models, but little is known about LAD estimation in structural equation modeling (SEM). To address this gap, the authors use the LAD criterion in SEM by minimizing the sum of the absolute deviations between the observed and the model implied covariance matrices. Using Monte Carlo

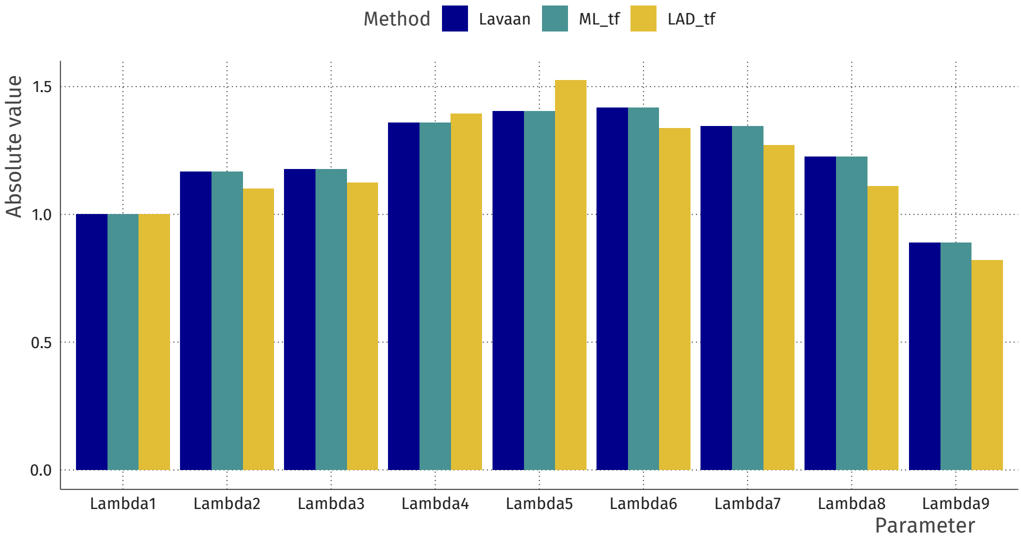
Least absolute deviation estimation



Least absolute deviation estimation



Least absolute deviation estimation



Structural Equation Modeling: A Multidisciplinary Journal, 23: 555–566, 2016

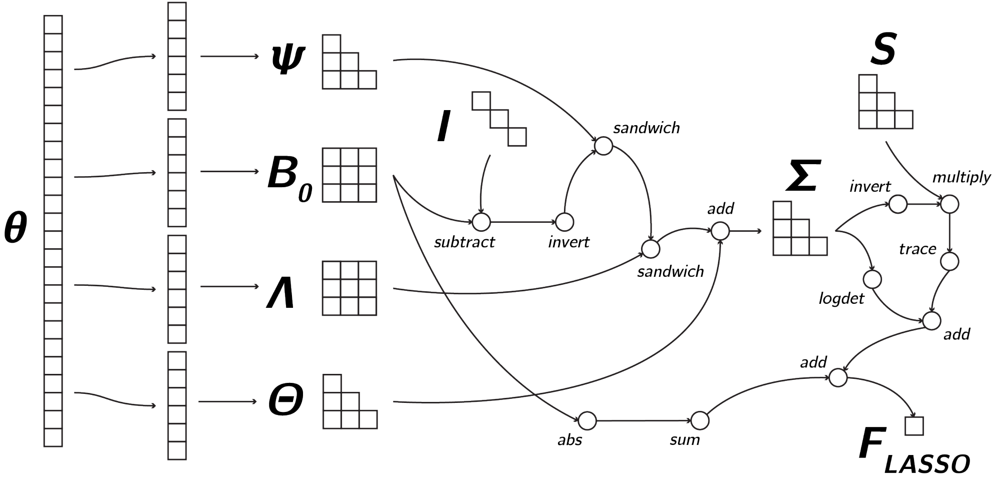
ISSN: 1070-5511 print / 1532-8007 online
DOI: 10.1080/10705511.2016.1154793

Routledge
Taylor & Francis Group

Regularized Structural Equation Modeling

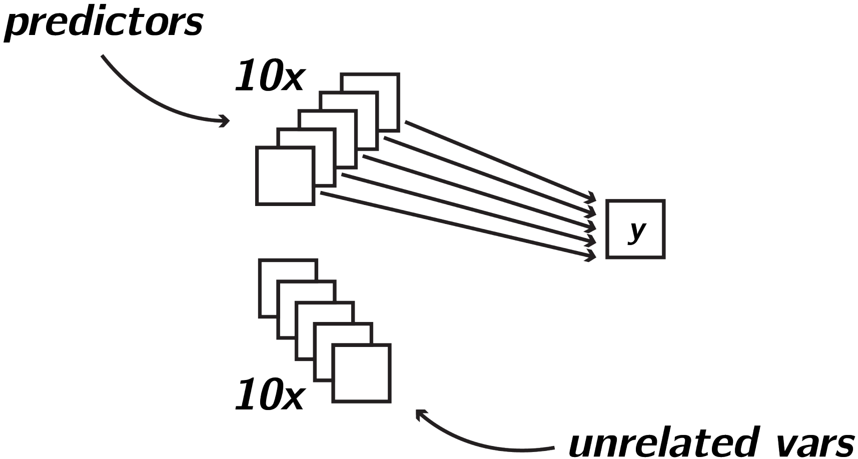
Ross Jacobucci,[1] Kevin J. Grimm,[2] and John J. McArdle[1]
[1]*University of Southern California*
[2]*Arizona State University*

A new method is proposed that extends the use of regularization in both lasso and ridge regression to structural equation models. The method is termed regularized structural equation modeling (RegSEM). RegSEM penalizes specific parameters in structural equation models, with the goal of creating easier to understand and simpler models. Although regularization has gained wide adoption in regression, very little has transferred to models with latent variables. By adding penalties to specific parameters in a structural equation model, researchers have a high level of flexibility in reducing model complexity, overcoming poor fitting models, and the creation of models that are more likely to generalize to new samples. The proposed method was evaluated through a simulation study, two illustrative examples involving a measurement model, and one empirical example involving the structural part of the model to demonstrate RegSEM's utility.

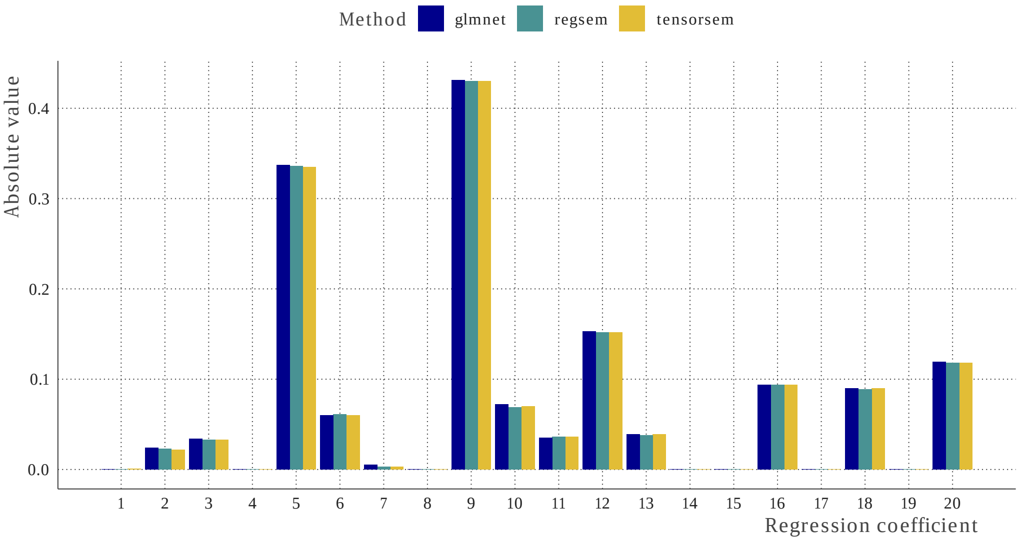


θ
ψ
B0
Λ
Θ
I
S
Σ
sandwich
subtract
invert
sandwich
add
invert
multiply
trace
logdet
add
add
abs
sum
F LASSO

Regularized regression



Regularized regression



PSYCHOMETRIKA—VOL. 83, NO. 3, 628–649
SEPTEMBER 2018
https://doi.org/10.1007/s11336-018-9623-z

APPROXIMATED PENALIZED MAXIMUM LIKELIHOOD FOR EXPLORATORY FACTOR ANALYSIS: AN ORTHOGONAL CASE

SHAOBO JIN

UPPSALA UNIVERSITY

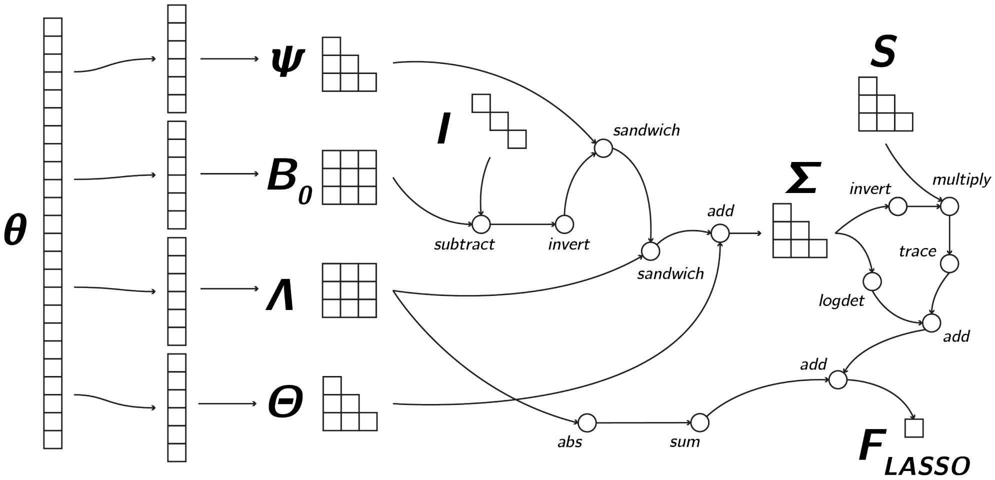
IRINI MOUSTAKI

LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

FAN YANG- WALLENTIN

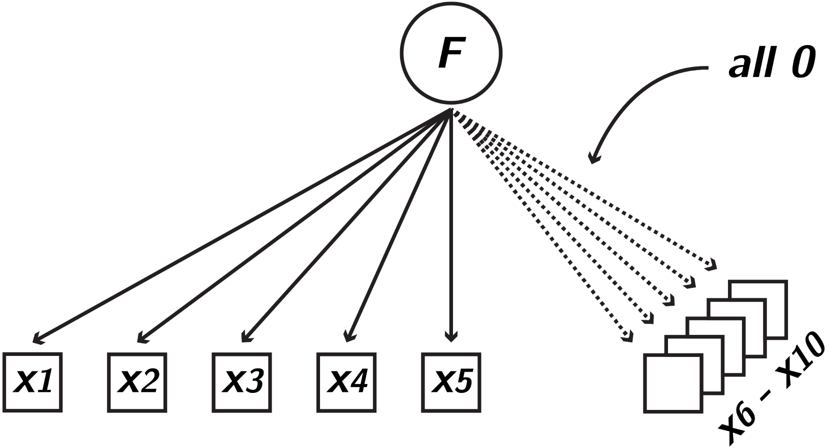
UPPSALA UNIVERSITY

The problem of penalized maximum likelihood (PML) for an exploratory factor analysis (EFA) model is studied in this paper. An EFA model is typically estimated using maximum likelihood and then the estimated loading matrix is rotated to obtain a sparse representation. Penalized maximum likelihood simulta-

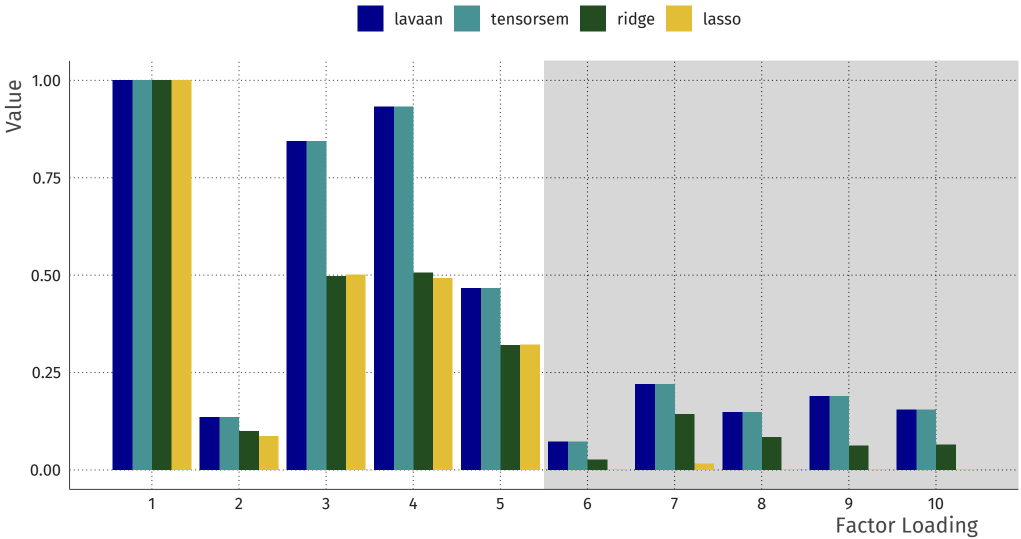


θ
ψ
B0
Λ
Θ
I
S
Σ
sandwich
subtract
invert
sandwich
add
invert
multiply
trace
logdet
add
add
abs
sum
F LASSO

Regularized exploratory factor analysis



Regularized exploratory factor analysis



R package showcase

tensorsem

R package showcase

```r
# Install tensorsem
devtools::install_github("vankesteren/tensorsem@computationgraph")
library(tensorsem)
```

R package showcase

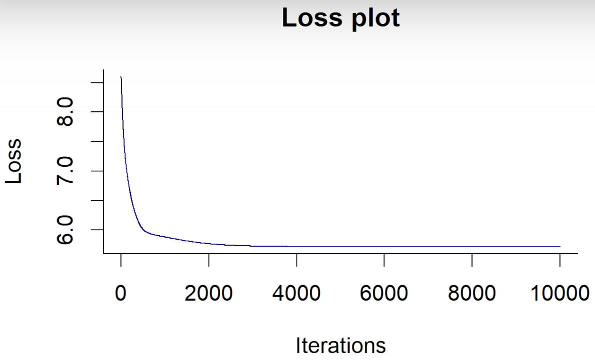
```r
# Create a model using lavaan syntax
mod <- "
F1 =~ x1 + x2 + x3
F2 =~ x4 + x5 + x6
F1 ~ F2
"
dat <- lavaan::HolzingerSwineford1939
```

R package showcase

```
# create a tf_sem object, similar to lavaan
tensem ← tf_sem(lav_model = mod, data = dat)

# optimise / compute parameter estimates
tensem$train(niter = 10000)


> [loss: 5.71860] [==========================>--------] 65%
```

R package showcase

```r
# plot the loss over iterations
tensem$plot_loss()
```



R package showcase

```
tensem$summary()


TensorFlow SEM session
----------------------

Loss: 5.718596


Sigma:
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 1.3583698 0.4594367 0.5818981 0.4136796 0.4595856 0.3828115
[2,] 0.4594367 1.3817839 0.3252543 0.2312279 0.2568872 0.2139740
[3,] 0.5818981 0.3252543 1.2748649 0.2928609 0.3253597 0.2710081
[4,] 0.4136796 0.2312279 0.2928609 1.3506645 1.0901316 0.9080243
[5,] 0.4595856 0.2568872 0.3253597 1.0901316 1.6597858 1.0087876
[6,] 0.3828115 0.2139740 0.2710081 0.9080243 1.0087876 1.1963584


Psi:
          [,1]      [,2]
[1,] 0.6475559 0.0000000
[2,] 0.0000000 0.9812432
```

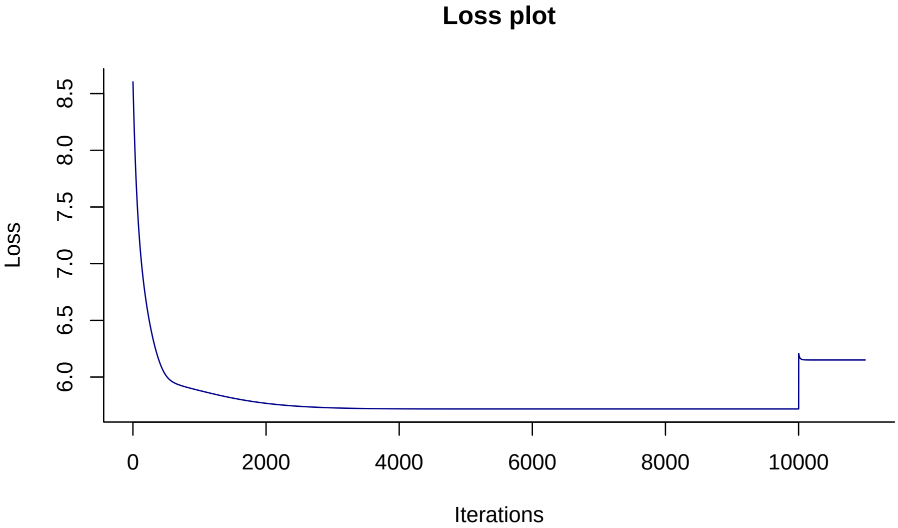
R package showcase

```
Beta:
     [,1]      [,2]
[1,]    0 0.4215872
[2,]    0 0.0000000


Lambda:
          [,1]      [,2]
[1,] 1.0000000 0.0000000
[2,] 0.5589541 0.0000000
[3,] 0.7079415 0.0000000
[4,] 0.0000000 1.0000000
[5,] 0.0000000 1.1109698
[6,] 0.0000000 0.9253816


Theta:
          [,1]    [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.5364119 0.00000 0.0000000 0.0000000 0.0000000 0.0000000
[2,] 0.0000000 1.12498 0.0000000 0.0000000 0.0000000 0.0000000
[3,] 0.0000000 0.00000 0.8629151 0.0000000 0.0000000 0.0000000
[4,] 0.0000000 0.00000 0.0000000 0.3694213 0.0000000 0.0000000
[5,] 0.0000000 0.00000 0.0000000 0.0000000 0.4486825 0.0000000
[6,] 0.0000000 0.00000 0.0000000 0.0000000 0.0000000 0.3560894
```

R package showcase

```r
# add ridge penalty to lambda and refit
tensem$penalties$ridge_lambda <- 0.1
tensem$train(1000)
tensem$plot_loss()
```

R package showcase



References

- Allaire J.J. & Tang, Y. (2019). tensorflow: R Interface to 'TensorFlow'. R package version 1.10.0.9000. https://github.com/rstudio/tensorflow
- Bollen, K. A. (1989). *Structural Equations with latent variables*. New York, NY: Wiley.
- Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris, 25*(1847), 536-538.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- 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*
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software, 48*(2), 1-36.

e.vankesteren1@uu.nl

github.com/vankesteren/tensorsem

@ejvankesteren

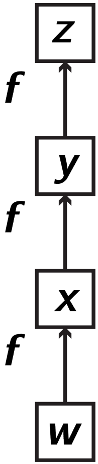
arxiv.org/abs/1905.04492

Automatic gradients

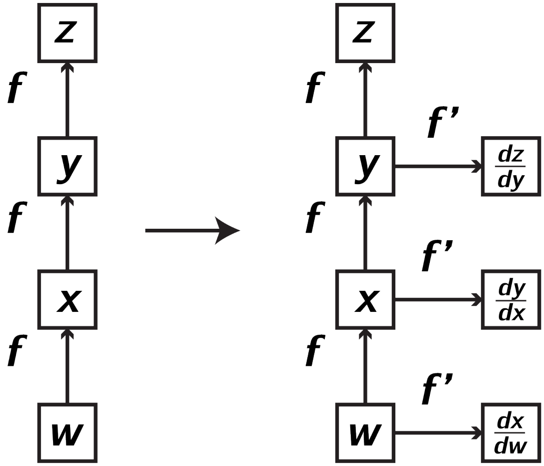
Computation Graphs: gradient computation

Autograd: use the chain rule to traverse the graph from objective back to parameters
Deep learning book section 6.5.1, figure 6.10

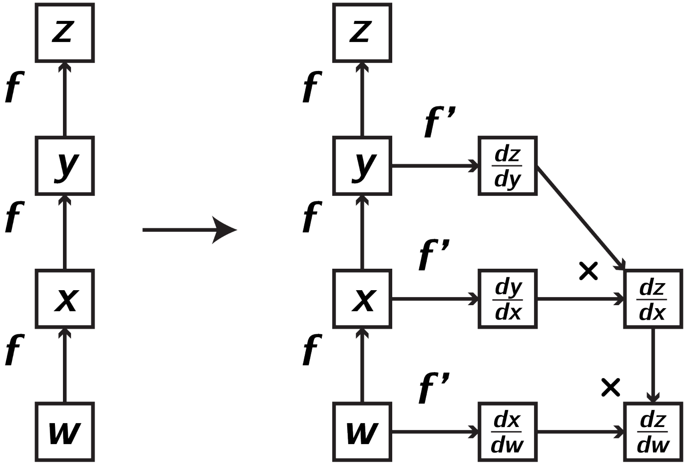
Gradient



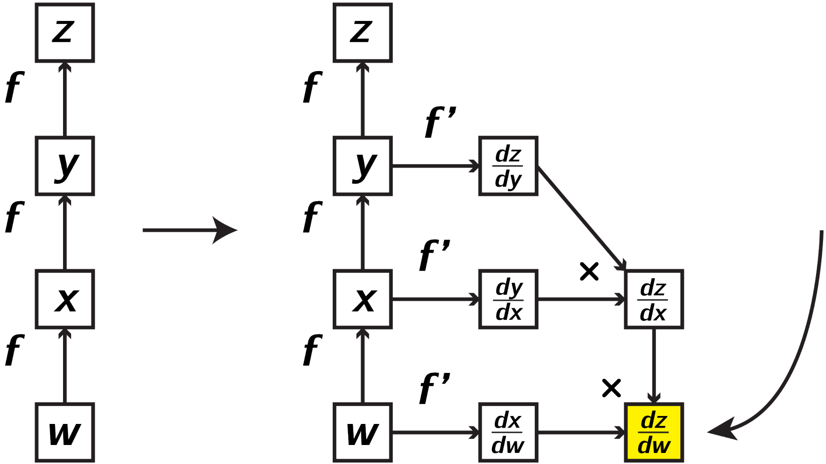
Gradient



Gradient



Gradient



Parameter path for LASSO regression. (Early stopping showcase)