Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. http://www.jstatsoft.org/

regsem: Regularized Structural Equation Modeling

Ross Jacobucci
University of Notre Dame

Abstract

The **regsem** package in R, an implementation of Regularized Structural Equation Modeling (Jacobucci, Grimm, and McArdle 2016), was recently developed with the goal incorporating various forms of penalized likelihood estimation with a broad array of structural equations models. The forms of regularization include both the *ridge* (Hoerl and Kennard 1970) and the least absolute shrinkage and selection operator (*lasso*; (Tibshirani 1996), along with sparser extensions. The paper details both the algorithmic details and an overview of the use of **regsem** through both a factor analysis and latent growth curve models.

Keywords: regularization, structural equation modeling, latent variables, R.

1. Introduction

The desire for simplicity in model structure comes by many names, including simple structure (Thurstone 1935), variable complexity (Browne 2001), parsimony (Raykov and Marcoulides 1999; Marsh and Hau 1996), "sparse loadings" in the context of principal components analysis (Zou, Hastie, and Tibshirani 2006), and lastly, "sparsistency", denoting that all parameters in a sparse model that are zero are correctly estimated as zero with probability tending to one (Lam and Fan 2009). The goal is accurately and efficiently estimate a model that is parsimonious in allowing users to easily interpret the model's representation of reality. In Figure 1, this would entail testing whether the cross-loadings are necessary to the fit of the model, or whether they can be pared away without an meaningful increase in bias.

In the context of latent variables, reducing the complexity of models can come in many forms: selecting among multiple predictors of a latent variable, simplifying factor structure by removing cross-loadings, determining whether the addition of nonlinear terms are necessary in longitudinal models, and many others. As a simple running example, Figure 1 depicts a linear latent growth curve model with four time points and ten predictors for a simulated

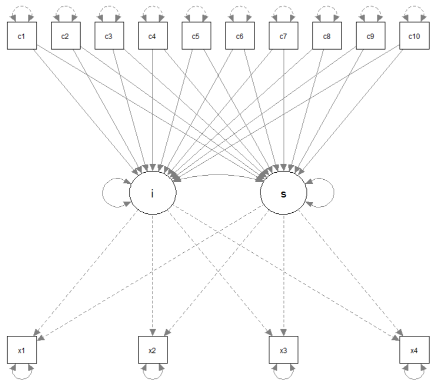


Figure 1: Growth curve model with 10 predictors of both the intercept and slope

dataset.

As an example, a researcher may want to test this model, but may only have a relatively small sample size (e.g. 80). However, there are 29 estimated parameters in this model, resulting in a estimated parameter to sample size ratio as far below even the most liberal recommendations (e.g. 10:1 parameters to sample size; Kline 2015). In lieu of finding additional respondents, reducing the number of parameters estimated is one effective strategy. Specifically, the 20 estimated regressions from *c1-c10* could be reduced to a number that makes the number of parameters estimated to sample size ratio more reasonable.

2. Regularization

Although a whole host of methods exist to perform variable selection, the use of regularization has seen a wide array of application in the context of regression, and more recently, in areas such as graphical modeling, as well as a host of others. The two most common procedures for regularization in regression are the *ridge* (Hoerl and Kennard 1970) and the least absolute shrinkage and selection operator (*lasso*; Tibshirani 1996); however, there are various alternative forms that can be seen as subsets or generalizations of these two procedures. Given an outcome vector y and predictor matrix $X \in R^{n \times p}$, the ridge estimates are defined as

$$\hat{\beta}^{ridge} = argmin\Big\{ \sum_{i=1}^{N}(y_i = \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \Big\}, \tag{1}$$

where β_0 is the intercept, β_j is the coefficient for x_j, and λ is the penalty that controls the amount of shrinkage. Note that when $\lambda = 0$, equation 3 reduces to ordinary least squares regression. As λ is increased, the β parameters are shrunken towards zero. The lasso estimates are defined as

$$\hat{\beta}^{lasso} = argmin\Big\{ \sum_{i=1}^{N}(y_i = \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \Big\}. \tag{2}$$

In lasso regression, the l_1-norm is used, instead of l_2-norm as in ridge, which also shrinks the β parameters, but additionally drives the parameters all the way to zero, thus performing a form of subset selection.

In the context of our example depicted in Figure 1, to use lasso regression to select among the covariates, the growth model would need to be reduced to two factor scores, which neglects both the relationship between both the slope and intercept, reducing both to independent factor scores. Particularly in models with a greater number of latent variables, this becomes increasing problematic. A method that keeps the model structure, while allowing for a penalized estimation of specific parameters is regularized structural equation modeling (RegSEM; Jacobucci, Grimm, and McArdle 2016). RegSEM adds a penalty function to the traditional maximum likelihood cost function for SEMs. The maximum likelihood cost function for SEMs can be written as

$$F_{ML} = log(|\Sigma|) + tr(C * \Sigma^{-1}) - log(|C|) - p. \tag{3}$$

where Σ is the model implied covariance matrix, C is the observed covariance matrix, and p is the number of estimated parameters. RegSem builds in an additional element to penalize certain model parameters yielding

$$F_{regsem} = F_{ML} + \lambda P(\cdot) \tag{4}$$

where λ is the regularization parameter and takes on a value between zero and infinity. When λ is zero, MLE is performed, and when λ is infinity, all penalized parameters are shrunk to zero. $P(\cdot)$ is a general function for summing the values of one or more of the model's parameter matrices. The two common forms of $P(\cdot)$ include both the Lasso ($\|\cdot\|_1$), which penalizes the sum of the absolute values of the parameters, and Ridge ($\|\cdot\|_2$), which penalizes the sum of the squared values of the parameters.

In our example, the twenty regression parameters from the covariates to both the intercept and slope would be penalized. Using lasso penalties, the absolute value of these twenty parameters would be summed and after being multiplied by the penalty λ, added to equation 4, resulting in:

$$F_{lasso} = F_{ML} + \lambda * \begin{Vmatrix} c1 \rightarrow i \\ c2 \rightarrow i \\ \vdots \\ c10 \rightarrow i \\ c1 \rightarrow s \\ \vdots \\ c10 \rightarrow s \end{Vmatrix}_1 \tag{5}$$

Although the fit of the model is easily calculated given a set of parameter estimates, traditional optimization procedures for SEM cannot be used given the non-differentiable nature of lasso penalties, and as detailed later, extensions.

2.1. Optimization

One method that has become popular for optimizing penalized likelihood method is that of proximal gradient descent (e.g. p. 104 in Hastie, Tibshirani, and Wainwright 2015). In comparison to one-step procedures common in SEM optimization, that only involve a method for calculating the step size and the direction (typically using the gradient and an approximation of the Hessian), proximal gradient descent can be formulated as a two-step procedure. With a stepsize of α^t and parameters θ^t at iteration t:

1. First, take a gradient step size $z = \theta^t - \alpha^t \nabla g(\theta^t)$.
2. Second, perform elementwise soft-thresholding $\theta^{t+1} = S_{\alpha^t \lambda}(z)$.

where $S_{\alpha\lambda}(z)$ is the soft-thresholding operator (Donoho 1995) used to overcome non-differentiability of the lasso penalty at the origin:

$$S_{\alpha\lambda}(z_j) = sign(z_j)(|z_j| - \alpha\lambda)_+ \tag{1}$$

where $(x)_+$ is shorthand for max(x,0) and α is the step size. Henceforth, λ is assumed to encompass both the penalty and the step size s^t. This procedure is only used to updated parameters that are subject to penalty. Non-penalized parameters are updated only using step 1 from above.

In contrast to only updating one parameter at a time in a coordinate-wise fashion, for RegSEM the optimization steps can be divided by both the A and S matrices. This block-wise gradient descent manifests itself as:

Algorithm 1 RegSEM Block Coordinate Descent

```
Generate starting values for θ_t
Calculate initial fit F_t
Set step size α. .5 works well at this time.
Set tolerance (tol). e.g. 1e-6
while |F_t − F_{t+1}| > tol do
    Calculate gradient for A: ∇(A) =: ∂A/∂θ_t
    θ_{t+1*,A} =: θ_{t,A} − α∇(A)
    Update penalized parameters: θ_{t+1*,A(pen)} =: S_{αλ}(θ_{t+1*,A(pen)})
    Calculate gradient for S: ∇(S) =: ∂S/∂θ_{t+1*}
    Update S parameters: θ_{t+1*,S} =: θ_{t,S} − α∇(S)
    θ_{t+1} =: θ_{t+1*}
    Update S_{t+1}, A_{t+1}
    Σ_{t+1} = F(I − A_{t+1})^{-1} S_{t+1} (I − A_{t+1})^{-T} F^T
    F_{t+1} = F_{ML}(Σ_{t+1}, C) + λ||A_{t+1}(pen)||
```

where step 13 is the calculation of the implied covariance matrix using RAM matrices and step 14 calculates the fit of the model with penalizing parameters in the A matrix. Note that parameters from the S matrix can also be penalized, however, this is much less common. This algorithm can be described as first order proximal block coordinate descent. For some SEM models, using block updates has been found to work better than standard gradient descent with the same soft thresholding of penalized parameters.

3. Types of Penalties

Outside of both ridge and lasso penalties, a whole host of additional forms of regularization exist.

3.1. Elastic Net

Most notably, the elastic net (Zou and Hastie 2005) encompasses both the ridge and lasso, reaching a compromise between both through the addition of an additional parameter α.

$$P_{enet}(\theta_j) = 0.5(1-\alpha)\|\theta_j\|_2 + \alpha\|\theta_j\|_2$$

with a soft-thresholding update of

$$S(\theta_j) = \begin{cases} 0, & |\theta_j| < \alpha\lambda \\ \frac{sgn(\theta_j)(|\theta_j|-\alpha\lambda)}{1+(1-\alpha)\lambda}, & |\theta_j| \geq \alpha\lambda \end{cases}$$

When α is zero, ridge is performed, and conversely when α is 1, lasso regularization is performed. This method harnesses the benefits of both methods, particularly when variable selection is warranted (lasso), but there may be collinearity between the variables (ridge).

3.2. Adaptive Lasso

In using lasso penalties, difficulties emerge when the scale of variables differ dramatically. By only using one value of λ, this can add appreciable bias to the resulting estimates (e.g. Fan and Li 2001). One method proposed for overcoming this limitation is the adaptive lasso (Zou 2006). Instead of penalizing parameters directly, each parameter is scaled by the un-penalized estimated (maximum likelihood estimate in SEM). This manifests itself as:

$$F_{alasso} = F_{ML} + \lambda\|\theta_{ML}^{-1} * \theta_{pen}\|_1$$

with, following the same form for the lasso, the soft-thresholding update is: \: ftp://ftp.stat.math.ethz.ch/Teaching/buhlmann/advanced-comput-statist/notes1.pdf

$$S(\theta_j) = sign(\theta_j)(|\theta_j| - \frac{\lambda}{2|\theta_j|})_+$$

In this, larger penalties are given for non-significant (smaller) parameters, limiting the bias in estimating larger, significant, parameters. Note that one limitation of this approach for SEM

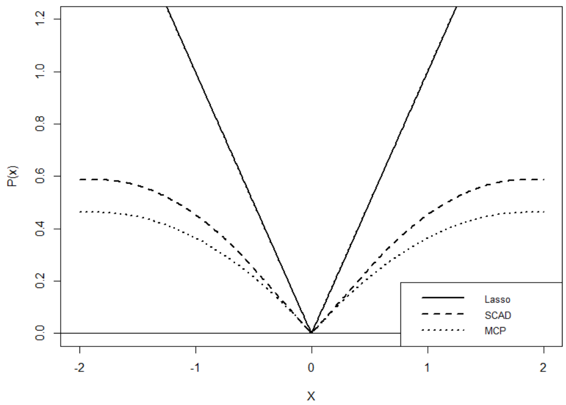


Figure 2: Comparison of types of penalties with $\lambda = 0.5$

models is that the model needs to be estimable with maximum likelihood. Particularly for models with large numbers of variables, in relation to sample size, this may not be possible.

3.3. Smoothly Clipped Absolute Deviation Penalty

Two additional penalties that overcome some of the deficiencies of the lasso, producing sparser solutions, include the smoothly clipped absolute deviation penalty (SCAD; Fan and Li 2001) and the minimax concave penalty (MCP; Zhang 2010). In comparison to the lasso, both the SCAD and MCP have much steeper penalties for smaller parameters, as is evident in Figure **.

The SCAD takes the form of:

$$pen_{\lambda,\gamma}(\theta_j) = \lambda\big\{I(\theta_j \leq \lambda) + \frac{(\gamma\lambda - 0)_+}{(\gamma - 1)\lambda} I(\theta_j > \lambda)\big\}$$

with a soft-thresholding update of

$$S(\theta_j) = \begin{cases} S(\theta_j, \lambda), & |\theta_j| \geq 2\lambda \\ \frac{\gamma-1}{\gamma-2} S(\theta_j, \frac{\lambda\gamma}{\gamma-1}), & 2\lambda < |\theta_j| \leq \alpha\lambda \\ \theta_j & |\theta_j| > \lambda\gamma \end{cases}$$

As the the penalty in equation 11 is non-convex (as is the MCP), this makes computation more difficult. However, in the context of SEM this can be seen as less problematic, as equation 3 is also non-convex.

3.4. Minimax Concave Penalty

Zhang (2010) proposed the minimax concave penalty:

$$pen_{\lambda,\gamma}(\theta_j) = \lambda\left(|\theta_j| - \frac{\theta_j^2}{2\lambda\gamma}\right)I(|\theta_j| < \lambda\gamma) + \frac{\lambda^2\gamma}{2}I(|\theta_j| \geq \lambda\gamma)$$

with a soft-thresholding update of

$$S(\theta_j) = \begin{cases} \frac{\gamma}{\gamma-1}S(\theta_j,\lambda), & |\theta_j| \leq \lambda\gamma \\ \theta_j & |\theta_j| > \lambda\gamma \end{cases}$$

As seen in Figure **, this results in similar amount of shrinkage for smaller estimates in comparison to the SCAD, however, less for larger estimates.

4. Implementation

RegSEM is implemented as the **regsem** package in the R statistical environment. To estimate the maximum likelihood fit of the model, **regsem** uses *Reticular Action Model* (RAM; J. J. McArdle and McDonald 1984, McArdle (2005)) notation to derive an implied covariance matrix. The parameters of each SEM are translated into three matrices: the *filter* (F), the *asymmetric* (A; directed paths; e.g. factor loadings or regressions), and the *symmetric* (S; undirected paths; e.g. covariances or variances). See Jacobucci, Grimm, and McArdle (2016) for more detail on RAM notation.

Syntax for using the **regsem** is based on the **lavaan** package (Rosseel 2012) for structural equation models. **lavaan** is a general SEM software program that can fit a wide array of models with various estimation methods. To use **regsem**, the user has to first fit the model in **lavaan**. As an example, below is the code for a one factor model fit in lavaan with the Holzinger-Swineford (Holzinger and Swineford 1939) dataset.

```
library(lavaan)
mod <- "
f1 = ~ NA*x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
f1~~1*f1
"
out <- cfa(mod,HolzingerSwineford1939)
```

After a model is run in lavaan, using `lavaan()` or any of the wrapper functions for fitting a model (i.e. `sem()`, `cfa()`, or `growth()`), the object is then used by the regsem package to translate the model into RAM notation and run using one of three functions: `regsem()`, `multi_optim()`, or `cv_regsem`. The `regsem()` function runs a model with one penalty value, whereas `multi_optim()` does the same but allows for the use of random starting values. However, the main function is `cv_regsem()` as this not only runs the model, but runs it across a vector of varying penalty values. For instance in the above one-factor model, each of the factor loadings can be tested with lasso penalties to determine whether each indicator is a necessary component of the latent factor.

```
library(regsem)
extractMatrices(out)["A"]
```

```
out.reg <- cv_regsem(out, type="lasso",
                     pars_pen=c(1:9),n.lambda=15,jump=.05)
```

In this, the function `extractMatrices()` allows the user to examine at how the lavaan model is translated into RAM matrices. Further, by looking at the A matrix, the parameter numbers corresponding to the factor loadings of interest for regularization can be identified. For this model, the factor loadings represent parameter numbers one through nine, of which we pass directly to the `pars_pen` argument of the `cv_regsem()` function (if `pars_pen=NULL` then all directed effects, outside of intercepts, are penalized). Additionally, we pass the arguments of how many values of penalty we want to test (`n.lambda=15`), how much the penalty should increase for each model (`jump=.05`), and finally that lasso estimation is used (`type="lasso"`).

The `out.reg` object contains two components, `out.reg[[1]]` has the parameter estimates for each of the 15 models, while `out.reg[[2]]` has the fit of each model. For the code above, using the **xtable** package to turn the output from `out.reg[[2]]` into a table

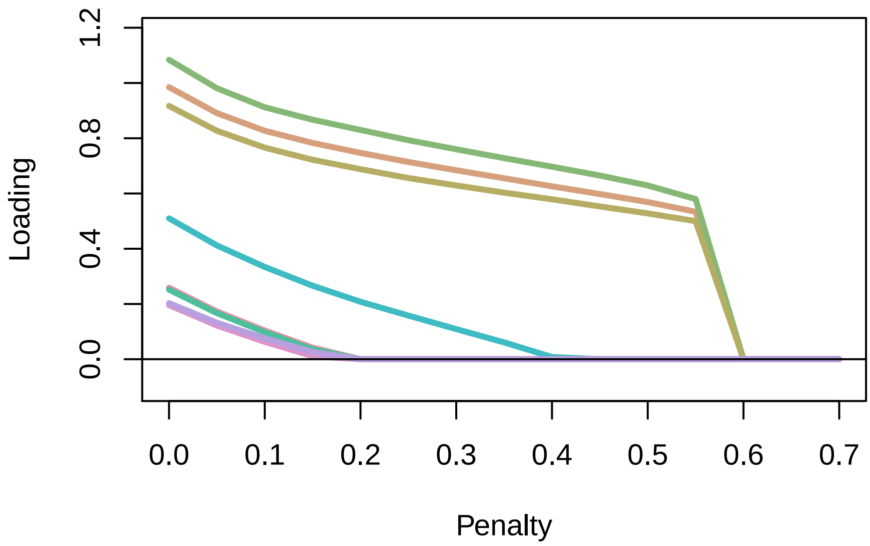
```
round(out.reg[[2]],3)
```

```
      lambda conv rmsea      BIC
 [1,]   0.00    0 0.188 7805.177
 [2,]   0.05    0 0.191 7815.389
 [3,]   0.10    0 0.199 7840.590
 [4,]   0.15    0 0.210 7875.842
 [5,]   0.20    0 0.203 7886.147
 [6,]   0.25    0 0.210 7911.816
 [7,]   0.30    0 0.214 7934.454
 [8,]   0.35    0 0.220 7960.670
 [9,]   0.40    0 0.227 7991.033
[10,]   0.45    0 0.229 8009.416
[11,]   0.50    0 0.234 8034.805
[12,]   0.55    0 0.242 8069.487
[13,]   0.60    0 0.286 8360.539
[14,]   0.65    0 0.286 8360.530
[15,]   0.70    0 0.286 8360.525
```

In this, the user can examine the penalty (lambda), whether the model converged ("conv"), with values of 0 indicating convergence, and the fit of each model. By default, two fit indices are output, both the root mean square error of approximation (RMSEA; Steiger and Lind 1980), and the Bayesian information criteria (BIC; Schwarz 1978). Both the RMSEA and BIC take into account the degrees of freedom of the model, an important point for model selection in the presence of lasso penalties (and other penalties that set parameters to zero). This concurs with work done on degrees of freedom for lasso regression, as Zou, Hastie, and Tibshirani (2007) proved that the number of nonzero coefficients is an unbiased estimate of the degrees of freedom for regression. As the penalty increases, select parameter are set to zero, thus the degrees of freedom increases, which for fit indices that include the degrees of freedom in the calculation, means that although the likelihood of the model may only get worse (increase), both the RMSEA and BIC can improve (decrease) due to the impact of increased degrees of freedom.

Instead of examining the `out.reg[[1]]` output matrix of parameter estimates, it is easiest to plot the trajectory of each of the penalized parameters. This is accomplished with `plot_cv(out.reg,pars=1:9)`, resulting in:

```
plot_cv(out.reg,pars=1:9)
```



5. Comparison

To compare the different types of penalties in **regsem**, we return to the the initial example of the latent growth curve model. Using the same simulated data (see Appendix), the model can be run in **lavaan** as

```
mod1 <- "
i =~ 1*x1 + 1*x2 + 1*x3 + 1*x4
s =~ x1 + 1*x2 + 2*x3 + 3*x4
i ~ c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8+ c9+ c10
s ~ c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8+ c9+ c10
"
lav.growth <- growth(mod1,dat,fixed.x=T)
```

To compare different types of penalties in **regsem**, it requires a different specification of the `type` argument. The options currently include maximum likelihood (`"none"`), ridge

(`"ridge"`), lasso (`"lasso"`), adaptive lasso (`"alasso"`), elastic net (`"enet"`), SCAD (`"scad"`), and MCP (`"mcp"`). For the elastic net, there is an additional hyperparameter, α that controls the tradeoff between ridge and lasso penalties. This is specified as `alpha=` , which has a default of 0.5. Additionally, both the SCAD and MCP have the additional hyper parameter of γ, which is specified as `gamma=` and defaults to 3.7 per Fan and Li (2001).

For the purposes of comparison, each of the 20 covariate regressions were penalized using the lasso, adaptive lasso, SCAD, and MCP, and compared to the maximum likelihood estimates (see Appendix for code). In this model, the data were simulated to have two large effects (both *c1* parameters), two small effects (both *c2* parameters) and sixteen true zero effects (*c3*-*c10* parameters). Note that the covariates were simulated to have zero covariance among each variable. If there was substantial collinearity among covariates, the elastic net would be more appropriate to simultaneously select predictors while also accounting for the collinearity. The parameter estimates corresponding the the best fit of the BIC are displayed in Table *

	ML	lasso	alasso	SCAD	MCP
c1 -> i	0.17	0.00	0.00	0.00	0.00
c2 -> i	-0.04	0.00	0.00	0.00	0.00
c3 -> i	0.01	0.00	0.00	0.00	0.00
c4 -> i	0.06	0.00	0.00	0.00	0.00
c5 -> i	0.20	0.00	0.00	0.00	0.00
c6 -> i	-0.52*	0.00	0.00	0.00	-0.24
c7 -> i	0.20	0.00	0.00	0.00	0.00
c8 -> i	-0.18	0.00	0.00	0.00	0.00
c9 -> i	0.03	0.00	0.00	0.00	0.00
c10 -> i	-0.12	0.00	0.00	0.00	0.00
c1 -> s	1.47*	1.44	1.60	1.62	1.61
c2 -> s	0.32*	0.20	0.29	0.35	0.34
c3 -> s	0.21	0.10	0.00	0.00	0.00
c4 -> s	-0.07	0.00	0.00	0.00	0.00
c5 -> s	-0.25	0.00	0.00	0.00	0.00
c6 -> s	0.37*	0.00	0.00	0.00	0.00
c7 -> s	-0.21	0.00	0.00	0.00	0.00
c8 -> s	0.27*	0.04	0.00	0.00	0.00
c9 -> s	-0.09	0.00	0.00	0.00	0.00
c10 -> s	0.18	0.00	0.00	0.00	0.00
BIC	3530.61	3484.27	3475.40	3474.85	3476.96

Table 1: Parameter estimates for the final models across five estimation methods. Note that * represent significant parameters at $p < .05$ for maximum likelihood estimation.

While every regularization method erroneously set both simulated true intercept effects as zero (as in maximum likelihood, which accounts for these errors), both the adaptive lasso and SCAD correctly identified every true zero effect. The lasso and MCP mistakenly identified one false effect each. As expected given the small ratio between number of estimated parameters and sample size, maximum likelihood mistakenly identified 3 false effects as significant.

To compare the performance of each penalization method further, particularly in the presence of small number of parameters to sample size ratios, a small scale simulation study was conducted. The same model and effects was kept, but the sample size was varied to include 80, 200, and 1000 to demonstrate how maximum likelihood estimation of effects improves as sample size increases, while each of the regularization methods performs well regardless of sample size. Each run was replicated 200 times, with the corresponding code in the Appendix. The results are displayed in Table *

	N	ML	lasso	alasso	SCAD	MCP
	80.00	0.09	0.05	**0.04**	0.05	0.05
False Positives	200.00	0.06	0.05	**0.02**	0.03	0.04
	1000.00	0.05	0.05	**0.01**	**0.01**	0.19
	80.00	0.45	**0.41**	0.43	0.45	0.54
False Negatives	200.00	0.30	**0.24**	0.29	0.29	0.45
	1000.00	0.10	**0.09**	0.17	0.19	0.24

Table 2: Results from the simulation using the model in Figure 1. Each condition was replicated 200 times. False positives represent concluding that the simulated regressions of zero were concluded as nonzero. False negatives are concluding that either the simulated regression values of 1 or 0.2 are in fact zero. Bolded values represent the smallest error per condition.

6. Discussion

6.1. Future Directions

6.2. Limitation

Conclusion

Browne, Michael W. 2001. "An Overview of Analytic Rotation in Exploratory Factor Analysis." *Multivariate Behavioral Research* 36 (1). Taylor & Francis: 111–50.

Donoho, David L. 1995. "De-Noising by Soft-Thresholding." *IEEE Transactions on Information Theory* 41 (3). IEEE: 613–27.

Fan, Jianqing, and Runze Li. 2001. "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties." *Journal of the American Statistical Association* 96 (456). Taylor & Francis: 1348–60.

Hastie, Trevor, Robert Tibshirani, and Martin Wainwright. 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Hoerl, Arthur E, and Robert W Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12 (1). Taylor & Francis Group: 55–67.

Holzinger, Karl John, and Frances Swineford. 1939. "A Study in Factor Analysis: The

Stability of a Bi-Factor Solution." *Supplementary Educational Monographs.*

Jacobucci, Ross, Kevin J Grimm, and John J McArdle. 2016. "Regularized Structural Equation Modeling." *Structural Equation Modeling: A Multidisciplinary Journal* 23 (4). Taylor & Francis: 555–66.

Kline, Rex B. 2015. *Principles and Practice of Structural Equation Modeling.* Guilford publications.

Lam, Clifford, and Jianqing Fan. 2009. "Sparsistency and Rates of Convergence in Large Covariance Matrix Estimation." *Annals of Statistics* 37 (6B). NIH Public Access: 4254.

Marsh, Herbert W, and Kit-Tai Hau. 1996. "Assessing Goodness of Fit: Is Parsimony Always Desirable?" *The Journal of Experimental Education* 64 (4). Taylor & Francis: 364–90.

McArdle, J. Jack, and Roderick P. McDonald. 1984. "Some Algebraic Properties of the Reticular Action Model for Moment Structures." *British Journal of Mathematical and Statistical Psychology* 37 (2). Wiley-Blackwell: 234–51.

McArdle, John J. 2005. "The Development of the Ram Rules for Latent Variable Structural Equation Modeling." *Contemporary Psychometrics: A Festschrift for Roderick P. McDonald.* Erlbaum. Mahwah, NJ, 225–73.

Raykov, Tenko, and George A Marcoulides. 1999. "On Desirability of Parsimony in Structural Equation Model Selection." *Structural Equation Modeling: A Multidisciplinary Journal* 6 (3). Taylor & Francis: 292–300.

Rosseel, Yves. 2012. "Lavaan: An R Package for Structural Equation Modeling." *Journal of Statistical Software* 48 (2): 1–36.

Schwarz, Gideon. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics* 6 (2). Institute of Mathematical Statistics: 461–64.

Steiger, James H, and John C Lind. 1980. "Statistically Based Tests for the Number of Common Factors." In *Annual Meeting of the Psychometric Society, Iowa City, Ia.* Vol. 758.

Thurstone, L. L. 1935. *The Vectors of Mind.* Chicago, IL: University of Chicago Press.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1). Wiley for the Royal Statistical Society: 267–88.

Zhang, Cun-Hui. 2010. "Nearly Unbiased Variable Selection Under Minimax Concave Penalty." *The Annals of Statistics* 38 (2). Institute of Mathematical Statistics: 894–942.

Zou, Hui. 2006. "The Adaptive Lasso and Its Oracle Properties." *Journal of the American Statistical Association* 101 (476). Taylor & Francis: 1418–29.

Zou, Hui, and Trevor Hastie. 2005. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2). Wiley Online Library: 301–20.

Zou, Hui, Trevor Hastie, and Robert Tibshirani. 2006. "Sparse Principal Component Analysis." *Journal of Computational and Graphical Statistics* 15 (2). Taylor & Francis: 265–86.

———. 2007. "On the Degrees of Freedom of the Lasso." *The Annals of Statistics* 35

(5). Institute of Mathematical Statistics: 2173–92. doi:10.1214/009053607000000127.

Affiliation:

Ross Jacobucci
University of Notre Dame
First line Second line
E-mail: rcjacobuc@gmail.com
URL: rjacobucci.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume VV, Issue II *Submitted:*
MMMMMM YYYY *Accepted:* yyyy-mm-dd