Chapter 7: Other Modeling Techniques

Tyson S. Barrett

Summer 2017

Utah State University

Introduction

Mediation Modeling

Structural Equation Modeling

Machine Learning Techniques

Conclusions

Introduction

Good Quote

"Simplicity is the ultimate sophistication."
— Leonardo da Vinci

Introduction

We cover, however briefly, modeling techniques that are especially useful to make complex relationships easier to interpret.

We will focus on:

1. mediation and moderation modeling,
2. methods relating to structural equation modeling (SEM), and
3. methods applicable to our field from machine learning.

An Aside

Although machine learning may appear very different than mediation and SEM, they each have advantages that can help in different situations.

- For example, SEM is useful when we know there is a high degree of measurement error or our data has multiple indicators for each construct.
- On the other hand, regularized regression and random forests–two popular forms of machine learning–are great to explore patterns and relationships there are hundreds or thousands of variables that may predict an outcome.

Mediation Modeling

Mediation Modeling

Mediation modeling can be done via several packages.

SEM framework: + `lavaan` (stands for "latent variable analysis")[1]. Although it is technically still a "beta" version, it performs very well especially for more simple models.

Other good ones: + `mediation` + `RMediation`

[1]The `lavaan` package has some great vignettes at http://lavaan.ugent.be/ to help with the other types of models it can handle.

Mediation Modeling

We model the following mediation model:

$$depression = \beta_0 + \beta_1 asthma + \epsilon_1$$

$$time_{Sedentary} = \lambda_0 + \lambda_1 asthma + \lambda_2 depression + \epsilon_2$$

In essence, we believe that asthma increases depression which in turn increases the amount of time spent being sedentary.

Mediation Modeling

```r
library(lavaan)
df$sed_hr = df$sed/60  ## in hours instead of minutes

## Our model
model1 <- '
  dep ~ asthma
  sed_hr ~ dep + asthma
'
## sem function to run the model
fit <- sem(model1, data = df)
summary(fit)
```

Mediation Modeling

```
## lavaan (0.5-23.1097) converged normally after  30 iterations
##
##                                                   Used       Total
##   Number of observations                          4614        4632
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dep ~
##     asthma            1.478    0.183    8.084    0.000
##   sed_hr ~
##     dep               0.044    0.011    3.929    0.000
##     asthma            0.412    0.139    2.965    0.003
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .dep              19.597    0.408   48.031    0.000
##    .sed_hr           11.171    0.233   48.031    0.000
```

Mediation Modeling

From the output we see asthma does predict depression and depression does predict time being sedentary. There is also a direct effect of asthma on sedentary behavior even after controlling for depression. We can further specify the model to have it give us the indirect effect and direct effects tested.

Mediation Modeling

```
## Our model
model2 <- '
  dep ~ a*asthma
  sed_hr ~ b*dep + c*asthma

  indirect := a*b
  total := c + a*b
'
## sem function to run the model
fit2 <- sem(model2, data = df)
summary(fit2)
```

Mediation Modeling

```
## lavaan (0.5-23.1097) converged normally after  30 iterations
##
##                                                   Used       Total
##   Number of observations                          4614        4632
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dep ~
##     asthma     (a)    1.478    0.183    8.084    0.000
##   sed_hr ~
##     dep        (b)    0.044    0.011    3.929    0.000
##     asthma     (c)    0.412    0.139    2.965    0.003
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .dep              19.597    0.408   48.031    0.000
##    .sed_hr           11.171    0.233   48.031    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

Mediation Modeling

We defined a few things in the model.

1. We gave the coefficients labels of `a`, `b`, and `c`.
2. Doing so allows us to define the `indirect` and `total` effects. Here we see the indirect effect, although small, is significant at $p < .001$. The total effect is larger (not surprising) and is also significant.

Also note that we can make the regression equations have other covariates as well if we needed to (i.e. control for age or gender) just as we do in regular regression.

Mediation Modeling

```r
## Our model
model2.1 <- '
  dep ~ asthma + ridageyr
  sed_hr ~ dep + asthma + ridageyr
'
## sem function to run the model
fit2.1 <- sem(model2.1, data = df)
summary(fit2.1)
```

Mediation Modeling

```
## lavaan (0.5-23.1097) converged normally after  33 iterations
##
##                                                   Used       Total
##   Number of observations                          4614        4632
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                0.000
##   Degrees of freedom                                 0
##   Minimum Function Value               0.0000000000000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dep ~
##     asthma            1.462    0.183    7.980    0.000
##     ridageyr         -0.005    0.004   -1.330    0.183
##   sed_hr ~
##     dep               0.044    0.011    3.927    0.000
##     asthma            0.412    0.139    2.956    0.003
##     ridageyr         -0.000    0.003   -0.063    0.950
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .dep              19.590    0.408   48.031    0.000
##    .sed_hr           11.171    0.233   48.031    0.000
```

Mediation Modeling

Although we don't show it here, we can also do moderation ("interactions") as part of the mediation model.

This is best done through packages other than `lavaan`.

Structural Equation Modeling

Structural Equation Modeling

Instead of summing our depression variable, we can use SEM to run the mediation model from above but use the latent variable of depression instead.

```
## Our model
model3 <- '
  dep1 =~ dpq010 + dpq020 + dpq030 + dpq040 + dpq050 + dpq060 + dpq070
  dep1 ~ a*asthma
  sed_hr ~ b*dep1 + c*asthma

  indirect := a*b
  total := c + a*b
'
## sem function to run the model
fit3 <- sem(model3, data = df)
summary(fit3)
```

```
## lavaan (0.5-23.1097) converged normally after  47 iterations
##
##                                                   Used       Total
##   Number of observations                          4614        4632
##
##   Estimator                                         ML
##   Minimum Function Test Statistic             1065.848
##   Degrees of freedom                                43
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dep1 =~
##     dpq010            1.000
##     dpq020            1.096    0.024   45.136    0.000
##     dpq030            1.133    0.031   36.908    0.000
##     dpq040            1.149    0.030   38.066    0.000
##     dpq050            0.933    0.025   36.773    0.000
##     dpq060            0.929    0.022   42.107    0.000
##     dpq070            0.871    0.022   39.760    0.000
##     dpq080            0.686    0.019   36.325    0.000
##     dpq090            0.308    0.011   28.544    0.000
##
## Regressions:
```

We defined `dep1` as a latent variable using `=~`.

Model Fit

Although the model does not fit the data well–“`P-value (Chi-square) = 0.000`”–it is informative for demonstration. We would likely need to find out how the measurement model (`dep1 =~ dpq010 + dpq020 + dpq030 +`) actually fits before throwing it into a mediation model. We can do that via:

Structural Equation Modeling

```
model4 <- '
  dep1 =~ dpq010 + dpq020 + dpq030 + dpq040 + dpq050 + dpq060 + dpq070 + dpq080
'
fit4 <- cfa(model4, data=df)
summary(fit4)
```

Structural Equation Modeling

```
## lavaan (0.5-23.1097) converged normally after  29 iterations
##
##   Number of observations                          4632
##
##   Estimator                                         ML
##   Minimum Function Test Statistic              985.831
##   Degrees of freedom                                27
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dep1 =~
##     dpq010            1.000
##     dpq020            1.097    0.024   45.383    0.000
##     dpq030            1.128    0.031   36.962    0.000
##     dpq040            1.145    0.030   38.136    0.000
##     dpq050            0.927    0.025   36.630    0.000
##     dpq060            0.930    0.022   42.294    0.000
##     dpq070            0.870    0.022   39.941    0.000
##     dpq080            0.681    0.019   36.350    0.000
##     dpq090            0.307    0.011   28.609    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

Lack of fit in the measurement model.

- It is possible that these depression questions could be measuring more than one factor.
- We could explore this using exploratory factor analysis.
- We don't demonstrate that here, but know that it is possible to do in `R` with a few other packages.

Machine Learning Techniques

Machine Learning Techniques

We are briefly going to introduce some machine learning techniques that may be of interest to researchers. We will quickly introduce and demonstrate:

1. Ridge, Lasso and Elastic Net
2. Random Forests

Ridge, Lasso and Elastic Net

Use the fantastic `glmnet` package.

- Using the `cv.glmnet()` function we can run the ridge ($alpha = 0$), lasso ($alpha = 1$ which is default), and elastic net ($0 \leq alpha \leq 1$).
- It turns out that elastic net is the combination of the ridge and lasso methods and the closer `alpha` is to 1 the more it acts like lasso and the closer it is to 0 the more it acts like ridge.

Ridge, Lasso and Elastic Net

Lasso and Elastic Net

- variable selection
- large number of predictors
- good prediction

Ridge

- handles multi-collinearity
- large number of predictors
- good prediction

To learn more see "Introduction to Statistical Learning" by Daniela Witten, Gareth James, Robert Tibshirani, and Trevor Hastie. A free PDF is available on their website.

glmnet

To use the package, it wants the data in a very specific form.

1. We need to remove any missingness. We use `na.omit()` to do this.
2. We take all the predictors (without the outcome) and put it in a data matrix object. We only include a few for the demonstration but you can include *many* predictors. We name ours `X`.
3. `Y` (a vector) is our outcome.

Prep the Data

```
df2 <- df %>%
  dplyr::select(riagendr, ridageyr, ridreth3, race, famsize
  na.omit
X <- df2 %>%
  dplyr::select(-sed_hr) %>%
  data.matrix
Y <- df2$sed_hr
```

Ridge, Lasso and Elastic Net

Use the `cv.glmnet()` function to fit the different models.

- The "cv" refers to cross-validation[2], which we don't discuss here, but it an important topic to become familiar with. Below we fit a ridge, a lasso, and an elastic net model.
- The elastic net model uses more of the lasso penalty because the `alpha` is closer to 1 than 0.

```r
library(glmnet)

fit_ridge <- cv.glmnet(X, Y, alpha = 0)
fit_lasso <- cv.glmnet(X, Y, alpha = 1)
fit_enet  <- cv.glmnet(X, Y, alpha = .8)
```

[2] Cross-validation is a common way to reduce over-fitting and make sure your model is generalizable. Generally, you split your data into training and testing sets. We recommend using it as often as you can, especially with these methods but also to make sure your other models are accurate on new data as well.

Ridge, Lasso and Elastic Net

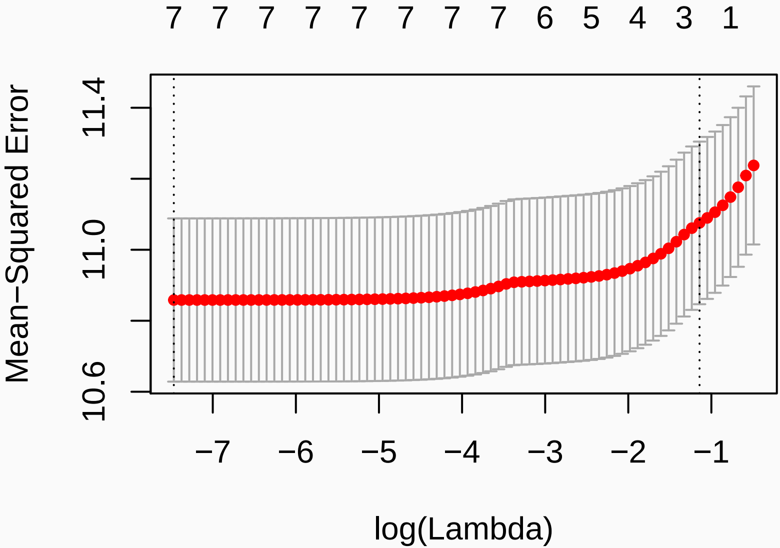
Selecting an appropriate tuning parameter is best done with plots.

- These plots show where appropriate `lambda` values are based on the mean squared error of the cross-validated prediction.
- The vertical dashed lines show a reasonable range of lambda values that can be used.

For example

```
plot(fit_enet)
```

Ridge, Lasso and Elastic Net



Ridge, Lasso and Elastic Net

We can get the coefficients at a reasonable `lambda`.

- Specifically, we use the "1-SE rule" (near the right hand side vertical dashed lines in the above plots) by `s = "lambda.1se"`.
- You can directly tell it what `lambda` value you'd like but this is a simple rule of thumb.

```
coef(fit_ridge, s = "lambda.1se")
coef(fit_lasso, s = "lambda.1se")
coef(fit_enet, s = "lambda.1se")
```

Ridge, Lasso and Elastic Net

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)  6.039337e+00
## riagendr     8.351244e-03
## ridageyr    -9.484185e-05
## ridreth3     1.145168e-02
## race         1.777208e-02
## famsize     -7.278221e-03
## dep          1.890571e-03
## asthma       1.891846e-02
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 5.7065357
## riagendr    .
## ridageyr    .
## ridreth3    .
## race        0.1354815
## famsize     .
## dep         .
## asthma      .
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 5.4775407
## riagendr    .
## ridageyr    .
## ridreth3    .
```

Ridge, Lasso and Elastic Net

Although we briefly introduce these regression methods, they are indeed very important. We highly recommend learning more about them.

Random Forests

Random forests is another machine learning method that can do fantastic prediction.

It is built in a very different way than the methods we have discussed up to this point. It is not built on a linear modeling scheme; rather, it is built on classification and regression trees (CART).

Again, "Introduction to Statistical Learning" is a great resource to learn more.

Random Forests

Use the `randomForest` package.

- We specify the model by the formula `sed_hr ~ .` which means we want `sed_hr` to be the outcome and all the rest of the variables to be predictors.

```
library(randomForest)

fit_rf <- randomForest(sed_hr ~ ., data = df2)
fit_rf
```

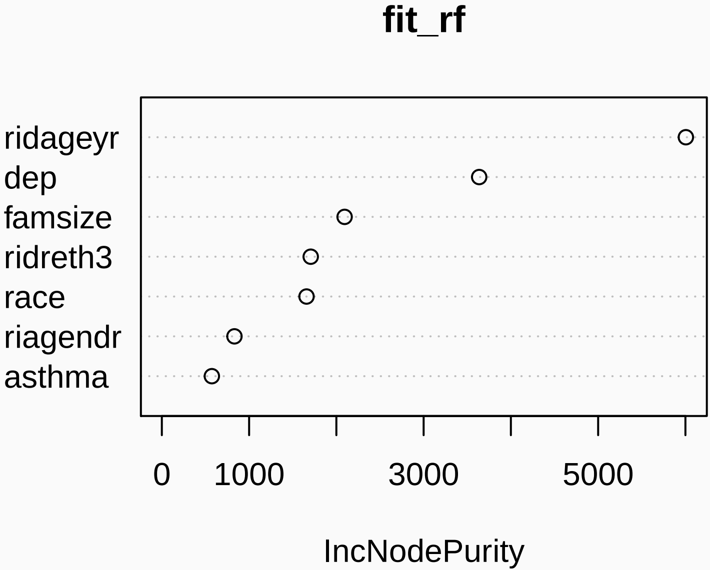
Random Forests

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

##
## Call:
##  randomForest(formula = sed_hr ~ ., data = df2)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 10.82932
##                     % Var explained: 3.64
```

Random Forests

We can find out which variables were important in the model via:

```
par(mfrow=c(1,1))  ## back to one plot per page
varImpPlot(fit_rf)
```


fit_rf
ridageyr
dep
famsize
ridreth3
race
riagendr
asthma
0
1000
3000
5000
IncNodePurity

Random Forests

We can see that age (`ridageyr`) is the most important variable, depression (`dep`) follows, with the family size (`famsize`) the third most important in the random forests model.

Conclusions

Conclusions

Although we only discussed these methods briefly, that does not mean they are less important. On the contrary, they are essential upper level statistical methods. This brief introduction hopefully helped you know what `R` is capable of across a wide range of methods.