Prediction with SEM: political democracy example

true

January 25, 2022

Intro

We made an R-function for the SEM based prediction rule and in this note we will analyze the `PoliticalDemocracy` data with this rule.

```
source('~/surfdrive/Predictive-Psychometrics/paper/SEM-Predictive Validity/versie2/Rcode/predict
predicty.lavaan
```

```
## function (object, newdata, xnames, ynames)
## {
##     Sxx = fitted(object)$cov[xnames, xnames]
##     Sxy = fitted(object)$cov[xnames, ynames]
##     mx = fitted(object)$mean[xnames]
##     my = fitted(object)$mean[ynames]
##     Xtest = as.matrix(newdata[, xnames])
##     Xtest = scale(Xtest, center = mx, scale = FALSE)
##     yhat = matrix(my, nrow = nrow(Xtest), ncol = length(ynames),
##         byrow = TRUE) + Xtest %*% solve(Sxx) %*% Sxy
##     return(yhat)
## }
```

Data set

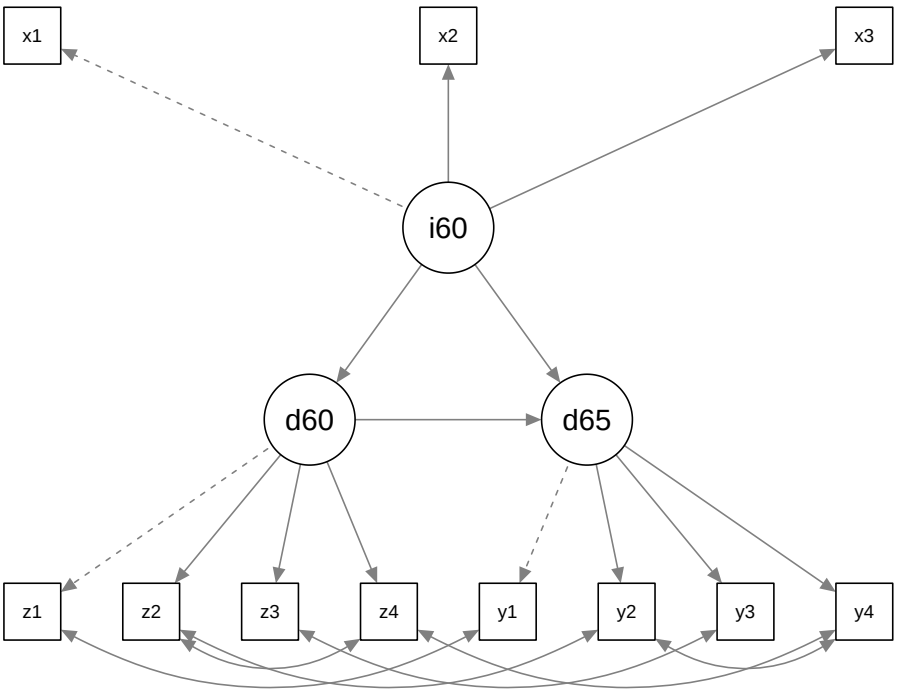
The political democracy data set is the leading data set in lavaan, used for many examples of SEM models.

```
data(PoliticalDemocracy)
colnames(PoliticalDemocracy) = c("z1", "z2", "z3", "z4", "y1", "y2", "y3", "y4", "x1", "x2", "x3
head(PoliticalDemocracy)
```

```
##      z1       z2       z3       z4       y1       y2       y3       y4       x1
## 1  2.50 0.000000 3.333333 0.000000 1.250000 0.000000 3.726360 3.333333 4.442651
## 2  1.25 0.000000 3.333333 0.000000 6.250000 1.100000 6.666666 0.736999 5.384495
## 3  7.50 8.800000 9.999998 9.199991 8.750000 8.094061 9.999998 8.211809 5.961005
## 4  8.90 8.800000 9.999998 9.199991 8.907948 8.127979 9.999998 4.615086 6.285998
## 5 10.00 3.333333 9.999998 6.666666 7.500000 3.333333 9.999998 6.666666 5.863631
## 6  7.50 3.333333 6.666666 6.666666 6.250000 1.100000 6.666666 0.368500 5.533389
```

```
##         x2       x3
## 1 3.637586 2.557615
## 2 5.062595 3.568079
## 3 6.255750 5.224433
## 4 7.567863 6.267495
## 5 6.818924 4.573679
## 6 5.135798 3.892270
```

```
model0 <- '
  # latent variable definitions
  ind60 =~ x1 + x2 + x3
  dem60 =~ z1 + z2 + z3 + z4
  dem65 =~ y1 + y2 + y3 + y4
  # regressions
  dem60 ~ ind60
  dem65 ~ ind60 + dem60
  # residual correlations
  z1 ~~ y1
  z2 ~~ z4 + y2
  z3 ~~ y3
  z4 ~~ y4
  y2 ~~ y4
'

fit <- sem(model0, data = PoliticalDemocracy, meanstructure = TRUE, warn = FALSE)
semPaths(fit, title = FALSE, intercepts = FALSE, residuals = FALSE)
```



Repeated 10 fold CV for varying models

With the following code we define five different SEM models for predicting the response variables, that are, the indicators for democracy in 1965.

In sum, we define the following modelling and prediction approaches:

1. On the training set we fit the model as defined by Figure **??**. For prediction in the test sample we use variables x_1, x_2, x_3 and z_1, z_2, z_3, z_4 as predictors and y_1, y_2, y_3, y_4 as outcomes;

5. A multivariate linear regression model with x_1, x_2, x_3 and z_1, z_2, z_3, z_4 as predictor variables and y_1, y_2, y_3, y_4 as response variables. For prediction in the test sample we use variables x_1, x_2, x_3 and z_1, z_2, z_3, z_4 as predictors and y_1, y_2, y_3, y_4 as outcomes;

With the following code we define our five structural equation models:

```
model <- '
  # latent variable definitions
  ind60 =~ x1 + x2 + x3
  dem60 =~ z1 + z2 + z3 + z4
  dem65 =~ y1 + y2 + y3 + y4
  # regressions
  dem60 ~ ind60
  dem65 ~ ind60 + dem60
  # residual correlations
  z1 ~~ y1
  z2 ~~ z4 + y2
  z3 ~~ y3
  z4 ~~ y4
  y2 ~~ y4
'
```

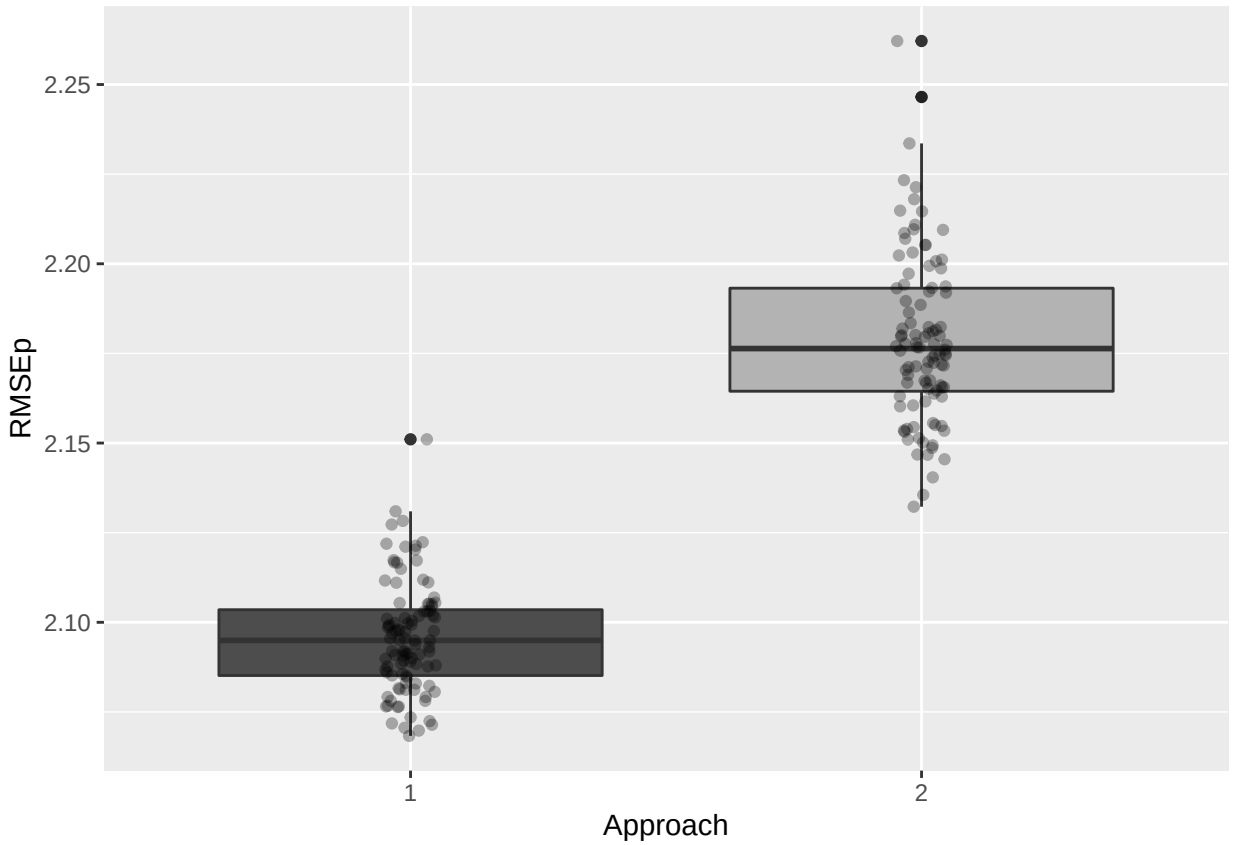
We perform 100 repetitions of 10 fold cross validation and compare the overall prediction error of the models. We also add a simple multivariate multiple linear regression. We focus on the cross-validated prediction error.

```
xnames = colnames(PoliticalDemocracy)[-c(5,6,7,8)]
ynames = colnames(PoliticalDemocracy)[c(5,6,7,8)]

set.seed(1234)
repeats = 100
PE = data.frame(repetition = rep(1:repeats, each = 2),
                model = rep(1:2, repeats),
                pe = rep(0, 2 * repeats))

folds = rep(1:10, length.out = 75)
t = 0
for (r in 1:repeats){
  yhat1 = yhat2 = matrix(NA, 75, 4)
  folds = sample(folds)
  for(k in 1:10){
```

```
    t = t + 1
    idx = which(folds == k)
    # SEM approach
    fit <- sem(model, data = PoliticalDemocracy[-idx, ], meanstructure = TRUE, warn = FALSE)
    yhat1[idx, ] = predicty.lavaan(fit, newdata = PoliticalDemocracy[idx, ], xnames = xnames, yn

    # linear regression model
    fit = lm(cbind(y1,y2,y3,y4) ~ ., data = PoliticalDemocracy[-idx, ])
    yhat2[idx, ]= predict(fit, newdata = PoliticalDemocracy[idx, ])
  }# end folds

  pe1 = sqrt(sum((PoliticalDemocracy[, ynames] - yhat1)^2)/300)
  pe2 = sqrt(sum((PoliticalDemocracy[, ynames] - yhat2)^2)/300)
  PE$pe[((r-1)*2 + 1): (r*2)] = c(pe1, pe2)
} # end repetitions

save(PE, file = "xvalpoldem.Rdata")
#save(coefs1, coefs2, coefs3, coefs4, coefs5, file = "xvalpoldemcoefs.Rdata")
```

We can make prediction error boxplots for the different approaches

```
library(ggplot2)
PE$model = as.factor(PE$model)


p <- ggplot(PE, aes(x=model, y=pe, fill=factor(model))) +
      geom_boxplot(aes(group = factor(model))) +
      geom_jitter(width = 0.05, height = 0, colour = rgb(0,0,0,.3)) +
      xlab("Approach") + ylab("RMSEp") +
      theme(legend.position="none") +
      scale_fill_grey(start=.3,end=.7)

p
```




```
ggsave('~/surfdrive/Predictive-Psychometrics/paper/SEM-Predictive Validity/versie2/Figures/Polde
```

```
## Saving 6.5 x 4.5 in image
```

We can check the number of wins for each of the five approaches:

```
pe = cbind(PE[PE$model == 0, 3], PE[PE$model == 1, 3], PE[PE$model == 2, 3], PE[PE$model == 3, 3
table(apply(pe, 1, which.min))
```

```
##
##   1
## 100
```