DA410_Assignment8_MattGraham

Exploratory Factor Analysis

Used to reduce large number of variables to fewer numbers of factors. Extracts common variance and puts in common score as index of all variables. Use score for further analysis.

Part of general linear model (correct) - must check all assumptions

Many methods to extract factors - most common = principal component analysis – extracts max variance and puts into first factor. Then removes var explained by first factor, then extracts max variance for second through n factors. - Second preferred – Common factor analysis = extracts common variance and puts into factors. Does not include unique variance of all variables - Third – Image factoring - based on correlation matrix; predictions use ordinary least squared method. Maximum likelihood method for factoring

This week: - Exploratory factor analysis (EFA) - assume indicator or variable is associated with any factor – Most common factor analysis used by researchers, not based on prior theory — Determine factor and factory

Next week: - Confirmatory factor analysis (CFA) - Assume each factor is associated with subset of measured variables

EFA is a statistical technique used to reduce smaller set of variables and explore theoretical structure and identify relationship between variables and response

```
library(nnspat)  # used for dist2full()
library("dplyr")  # used to select numeric datatypes
library("ggplot2")
library(reshape)  # used for melting matricies
library(klaR)
library(ggvis)
library(class)
library(gmodels)
library(MASS)
library(readxl)
library(psych)
```

Get data

```
words <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/T5_9_ESSAY.DAT", header
=FALSE)

words <- words[, -1]  # remove student ID
colnames(words) <- c("y1", 'y2', 'x1', 'x2')
words
```

y1 <int>	y2 <int>	x1 <int>	x2 <int>
148	20	137	15

y1 <int>	y2 <int>	x1 <int>	x2 <int>
159	24	164	25
144	19	224	27
103	18	208	33
121	17	178	24
89	11	128	20
119	17	154	18
123	13	158	16
76	16	102	21
217	29	214	25

1-10 of 15 rows Previous **1** 2 Next

13.7

a

Step 1 - Find correlation matrix

```
words.cor <- cor(words)
words.cor
```

```
##           y1        y2        x1        x2
## y1 1.0000000 0.7660725 0.5953551 0.2173378
## y2 0.7660725 1.0000000 0.5600505 0.4427548
## x1 0.5953551 0.5600505 1.0000000 0.7202028
## x2 0.2173378 0.4427548 0.7202028 1.0000000
```

While one pair (y1, x2) has a correlation below .30, we will still proceed with exploratory factor analysis.

Step 2 - Find eigenvaue D and eigenvectors C

```
D <- eigen(words.cor)$values
D
```

```
## [1] 2.6657459 0.8993358 0.3276382 0.1072801
```

Finding C (vectors)

```
C <- eigen(words.cor)$vectors
C
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,] -0.4914201  0.5642628  0.3208298  0.5806737
## [2,] -0.5241023  0.3441774 -0.6604771 -0.4130722
## [3,] -0.5409202 -0.2848265  0.6020434 -0.5136371
## [4,] -0.4372967 -0.6942790 -0.3136590  0.4778769
```

Step 3 - Finding C1 and D1

```
C1 <- C[,1:2]
C1
```

```
##            [,1]       [,2]
## [1,] -0.4914201  0.5642628
## [2,] -0.5241023  0.3441774
## [3,] -0.5409202 -0.2848265
## [4,] -0.4372967 -0.6942790
```

D1

```
D1 <- diag(D[1:2])
D1
```

```
##          [,1]      [,2]
## [1,] 2.665746 0.0000000
## [2,] 0.000000 0.8993358
```

Step 4 - Finding lambda

```
words.lambda <- C1 %*% sqrt(D1)
words.lambda
```

```
##            [,1]       [,2]
## [1,] -0.8023471  0.5351091
## [2,] -0.8557077  0.3263949
## [3,] -0.8831664 -0.2701104
## [4,] -0.7139792 -0.6584078
```

```
C1[1,]
```

```
## [1] -0.4914201  0.5642628
```

Step 5 - Obtain the loadings

```
loading <- c('loading.lambda.1j', 'loading.lambda.2j', 'H.i2', 'psi.j')

Y1 <- c(C1[1,], (C1[1,1]**2 + C1[1,2]**2), 1-(C1[1,1]**2 + C1[1,2]**2))
loading <- rbind(loading, Y1)
Y2 <- c(C1[2,], (C1[2,1]**2 + C1[2,2]**2), 1-(C1[2,1]**2 + C1[2,2]**2))
loading <- rbind(loading, Y2)
X1 <- c(C1[3,], (C1[3,1]**2 + C1[3,2]**2), 1-(C1[3,1]**2 + C1[3,2]**2))
loading <- rbind(loading, X1)
X2 <- c(C1[4,], (C1[4,1]**2 + C1[4,2]**2), 1-(C1[4,1]**2 + C1[4,2]**2))
loading <- rbind(loading, X2)

loading <- as.data.frame(loading)
colnames(loading) <- loading[1,]
loading <- loading[-1, ]

loading
```

	loading.lambda.1j <chr>	loading.lambda.2j <chr>	H.i2 <chr>	psi.j <chr>
Y1	-0.491420117708461	0.564262789405499	0.559886227596273	0.440113772403727
Y2	-0.524102315489297	0.344177438109905	0.393141346005141	0.606858653994859
X1	-0.540920166213667	-0.284826462867299	0.373720740166118	0.626279259833882
X2	-0.437296700872004	-0.694278965286239	0.673251686232469	0.326748313767531

4 rows

Obtain variance analysis

```
prop.vars <- c('loading.lambda.1j', 'loading.lambda.2j', 'H.i2')

# get two largest eigenvalues and prop of total var is eigen value / sum of all, cum prop = s
um of prop
var.acc <- c(D[1], D[2], sum(D[1], D[2]))
prop.vars <- rbind(prop.vars, var.acc)
prop.tot.var <- c(D[1] / sum(D), D[2] / sum(D), ((D[1] / sum(D)) + (D[2] / sum(D))))
prop.vars <- rbind(prop.vars, prop.tot.var)
cumul.prop <- c(D[1] / sum(D),((D[1] / sum(D)) + (D[2] / sum(D))), ((D[1] / sum(D)) + (D[2] /
sum(D))))
prop.vars <- rbind(prop.vars, cumul.prop)
prop.vars <- as.data.frame(prop.vars)
colnames(prop.vars) <- prop.vars[1,]
prop.vars <- prop.vars[-1, ]

prop.vars
```

	loading.lambda.1j <chr>	loading.lambda.2j <chr>	H.i2 <chr>

	loading.lambda.1j <chr>	loading.lambda.2j <chr>	H.i2 <chr>
var.acc	2.66574588400473	0.899335844890742	3.56508172889548
prop.tot.var	0.666436471001184	0.224833961222686	0.89127043222387
cumul.prop	0.666436471001184	0.89127043222387	0.89127043222387

3 rows

Overall, we can conclude that 89% of total variance is present across 2 factors and represent our variables well. We can see across our commonalities, there are not very many. Our Y1 and X2 being our strongest with Y2 and X1 being our weakest and nearly even.