Problem Set 4: Dimension Reduction

Allison Collins

11/8/2019

```
countries <- read_csv(("~/Documents/HW 4/countries.csv"))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   X1 = col_character(),
##   idealpoint = col_double(),
##   gdp.pc.wdi = col_double(),
##   gdp.pc.un = col_double(),
##   pop.wdi = col_double(),
##   cinc = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

Factor Analysis

1. How do CFA and EFA differ?

Confirmatory factor analysis is used when we have a hypothesis we want to confirm re. a number of factors for our variables - for example, we think that there are 3 underlying factors that account for the covariation in our data that contains, say, 15 variables. Exploratory factor analysis seeks to reduce a set of variables to its underlying smaller number of factors that explain covariation among the indicators.

2. Fit three exploratory factor analysis models initialized at 2, 3, and 4 factors. Present the loadings from these solutions and discuss in substantive terms. How does each fit? What sense does this give you of the underlying dimensionality of the space? And so on.

```
#Scale the data
countries[-1] <- scale(countries[-1])

#Fit factor model with 2 variables
factan.1 <- fa(countries[-1],
               nfactors = 2)
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
summary(factan.1)
```

```
##
## Factor analysis with Call: fa(r = countries[-1], nfactors = 2)
##
## Test of the hypothesis that 2 factors are sufficient.
## The degrees of freedom for the model is 169  and the objective function was  51.41
## The number of observations was  107  with Chi Square =  4978.17  with prob <  0
```

```
##
## The root mean square of the residuals (RMSA) is  0.12
## The df corrected root mean square of the residuals is  0.14
##
## Tucker Lewis Index of factoring reliability =  0.081
## RMSEA index =  0.543  and the 10 % confidence intervals are  0.506 NA
## BIC =  4188.46
##  With factor correlations of
##      MR1  MR2
## MR1 1.00 0.41
## MR2 0.41 1.00
```

```
#Fit factor model with 3 variables
factan.2 <- fa(countries[-1],
               nfactors = 3)
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
summary(factan.2)
```

```
##
## Factor analysis with Call: fa(r = countries[-1], nfactors = 3)
##
## Test of the hypothesis that 3 factors are sufficient.
## The degrees of freedom for the model is 150  and the objective function was  46.65
## The number of observations was  107  with Chi Square =  4486.65  with prob <  0
##
## The root mean square of the residuals (RMSA) is  0.06
## The df corrected root mean square of the residuals is  0.07
##
## Tucker Lewis Index of factoring reliability =  0.06
## RMSEA index =  0.549  and the 10 % confidence intervals are  0.509 NA
## BIC =  3785.72
##  With factor correlations of
##       MR1   MR2   MR3
## MR1  1.00  0.38 -0.05
## MR2  0.38  1.00 -0.12
## MR3 -0.05 -0.12  1.00
```

```
#Fit factor model with 3 variables
factan.3 <- fa(countries[-1],
               nfactors = 4)
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
summary(factan.3)
```

```
##
## Factor analysis with Call: fa(r = countries[-1], nfactors = 4)
##
## Test of the hypothesis that 4 factors are sufficient.
## The degrees of freedom for the model is 132  and the objective function was  43.56
## The number of observations was  107  with Chi Square =  4160.02  with prob <  0
##
## The root mean square of the residuals (RMSA) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
```

```
## Tucker Lewis Index of factoring reliability =  0
## RMSEA index =  0.566  and the 10 % confidence intervals are  0.523 NA
## BIC =  3543.21
##  With factor correlations of
##       MR1   MR3   MR4   MR2
## MR1  1.00 -0.06  0.32 -0.29
## MR3 -0.06  1.00  0.00  0.23
## MR4  0.32  0.00  1.00 -0.52
## MR2 -0.29  0.23 -0.52  1.00
```

From a fit perspective, for all of the above runs, the low p-value indicates that the corresponding number of factors was sufficient in explaining the underlying data. However, using three or four factors results in lower root mean squared residuals - we want this value to be close to 0, thus I would be apt to think 3 or 4 factors does a better job explaining the data.

```
factan.1$loadings
```

```
##
## Loadings:
##            MR1    MR2
## idealpoint  0.449  0.429
## polity      0.995
## polity2     0.995
## democ       0.931
## autoc      -0.969  0.159
## unreg       0.412 -0.131
## physint            0.782
## speech      0.631  0.154
## new_empinx  0.802  0.197
## wecon              0.509
## wopol       0.551
## wosoc       0.286  0.497
## elecsd      0.852
## gdp.pc.wdi         0.673
## gdp.pc.un          0.671
## pop.wdi     0.204 -0.476
## amnesty           -0.821
## statedept         -0.849
## milper      0.158 -0.468
## cinc        0.211 -0.366
## domestic9   0.288 -0.479
##
##                  MR1   MR2
## SS loadings    6.523 4.527
## Proportion Var 0.311 0.216
## Cumulative Var 0.311 0.526
```

```
factan.2$loadings
```

```
##
## Loadings:
##            MR1    MR2    MR3
## idealpoint  0.432  0.468
## polity      0.992
## polity2     0.992
## democ       0.910  0.144
## autoc      -0.994  0.191
## unreg       0.413 -0.129
## physint            0.737 -0.136
## speech      0.646  0.128
## new_empinx  0.840  0.131 -0.125
## wecon              0.518
## wopol       0.552
## wosoc       0.263  0.547
## elecsd      0.858
## gdp.pc.wdi         0.856  0.158
## gdp.pc.un          0.853  0.157
## pop.wdi                   0.892
## amnesty           -0.715  0.243
## statedept         -0.803  0.144
## milper                    0.949
## cinc                      0.999
## domestic9   0.269 -0.443
##
##                 MR1   MR2   MR3
## SS loadings    6.466 4.275 2.881
## Proportion Var 0.308 0.204 0.137
## Cumulative Var 0.308 0.512 0.649
```

```
factan.3$loadings
```

```
##
## Loadings:
##            MR1    MR3    MR4    MR2
## idealpoint  0.467         0.214 -0.294
## polity      0.995
## polity2     0.995
## democ       0.922         0.127
## autoc      -0.986         0.146
## unreg       0.405                0.165
## physint     0.119               -0.761
## speech      0.658               -0.109
## new_empinx  0.855               -0.145
## wecon       0.105         0.390 -0.170
## wopol       0.555
## wosoc       0.300         0.350 -0.239
## elecsd      0.865
## gdp.pc.wdi                0.986
## gdp.pc.un                 0.979
## pop.wdi            0.923
## amnesty            0.177 -0.197  0.602
## statedept  -0.137        -0.139  0.783
## milper             0.965
## cinc               0.981  0.111
## domestic9   0.247         0.204  0.757
##
##                 MR1   MR3   MR4   MR2
## SS loadings    6.605 2.811 2.426 2.370
## Proportion Var 0.315 0.134 0.116 0.113
## Cumulative Var 0.315 0.448 0.564 0.677
```

To interpret factor loading, we look at which features load highest on each factor (e.g. above .3). For the model with 2 factors we see double loading only on the first feature (idealpoint). This is true for the second model as well, and then for the third model (with four factors) there are no features with multiple factors with loads above .3 although wosoc has a load of .3 and .35. Basically, across all three for the most part we have single loading of features, which is what we are looking for. While with new factors introduced, some of the features will load into different factors, we see that in cases especially with high factors, features which previously mapped to the same factor remain grouped together (for example milper and pop.wdi).

3. Rotate the 3-factor solution using any oblique method you would like and present a visual of the unrotated and rotated versions side-by-side. How do these differ and why does this matter (or not)?

```
r.factan <- fa(countries[,-1],
                 nfactors = 3,
                 rotate = "promax")
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
#summary(r.factan)
r.factan$loadings
```

```
##
## Loadings:
##            MR1    MR2    MR3
## idealpoint  0.418  0.489
## polity      0.980
## polity2     0.980
## democ       0.895  0.175
## autoc      -0.987  0.167
## unreg       0.410 -0.119
## physint            0.747
## speech      0.638  0.147
## new_empinx  0.831  0.153 -0.126
## wecon              0.528
## wopol       0.546
## wosoc       0.249  0.565
## elecsd      0.848
## gdp.pc.wdi         0.875  0.210
## gdp.pc.un          0.872  0.209
## pop.wdi                   0.892
## amnesty           -0.719  0.202
## statedept         -0.814
## milper                    0.952
## cinc               0.129  1.010
## domestic9   0.272 -0.439
##
##                  MR1   MR2   MR3
## SS loadings    6.311 4.438 2.905
## Proportion Var 0.301 0.211 0.138
## Cumulative Var 0.301 0.512 0.650
```

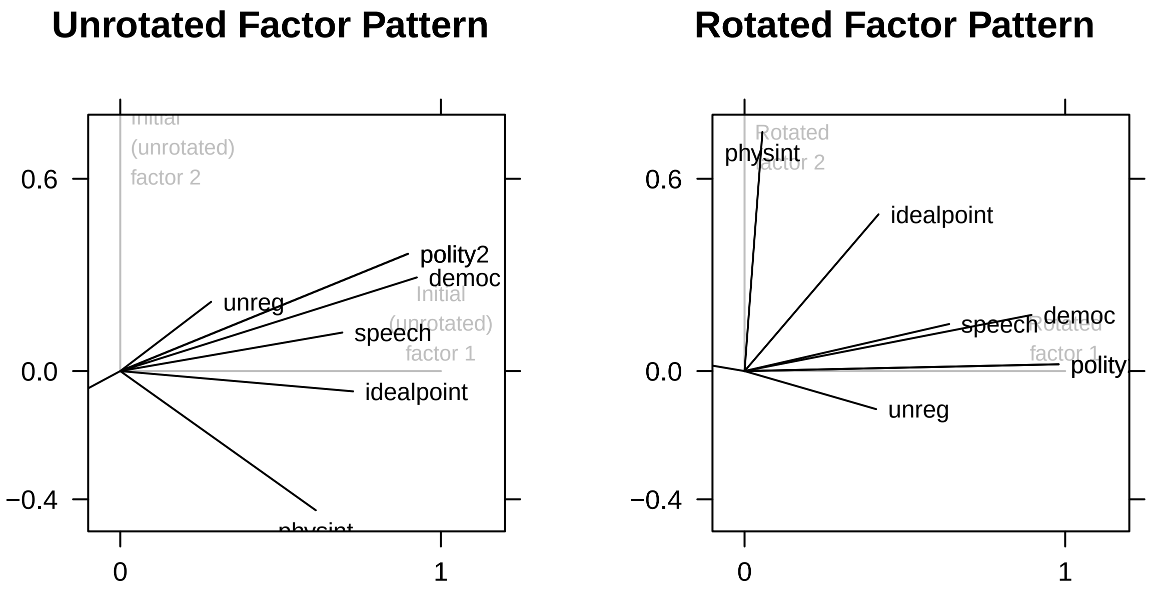
```
#Citation - leveraging code we used in class to plot rotated + nonrotated vectors. I struggled with get

#Non-rotated
nonrotated.factors <- fa(cor(countries[,-1]),
        fm = "pa",
        nfactors = 3,
        rotate = "none",
        residuals = TRUE)
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
nonrot.pattern <- as.data.frame(nonrotated.factors$loadings[1:8,])

p1 <- xyplot(PA2 ~ PA1, data = nonrot.pattern,
       aspect = 1,
       xlim = c(-.1, 1.2),
       ylim = c(-.5, .8),
       panel = function (x, y) {
         panel.segments(c(0, 0), c(0, 0),
             c(1, 0), c(0, 1), col = "gray")
         panel.text(1, 0, labels = "Initial\n(unrotated)\nfactor 1",
                    cex = .65, pos = 3, col = "gray")
         panel.text(0, .7, labels = "Initial\n(unrotated)\nfactor 2",
                    cex = .65, pos = 4, col = "gray")
         panel.segments(rep(0, 8), rep(0, 8), x, y,
            col = "black")
         panel.text(x[-7], y[-7], labels = rownames(nonrot.pattern)[-7],
            pos = 4, cex = .75)
         panel.text(x[7], y[7], labels = rownames(nonrot.pattern)[7],
                    pos = 1, cex = .75)
       },
       main = "Unrotated Factor Pattern",
       xlab = "",
       ylab = "",
       scales = list(x = list(at = c(0, 1)),
                     y = list(at = c(-.4, 0, .6)))
)

#Rotated
rotated.factors <- fa(cor(countries[,-1]),
        fm = "pa",
        nfactors = 3,
        rotate = "promax",
        residuals = TRUE)
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
rot.pattern <- as.data.frame(rotated.factors$loadings[1:8,])

p2 <- xyplot(PA2 ~ PA1, data = rot.pattern,
       aspect = 1,
       xlim = c(-.1, 1.2),
       ylim = c(-.5, .8),
       panel = function (x, y) {
         panel.segments(c(0, 0), c(0, 0),
             c(1, 0), c(0, 1), col = "gray")
         panel.text(1, 0, labels = "Rotated\nfactor 1",
                    cex = .65, pos = 3, col = "gray")
         panel.text(0, .7, labels = "Rotated\nfactor 2",
                    cex = .65, pos = 4, col = "gray")
         panel.segments(rep(0, 8), rep(0, 8), x, y,
            col = "black")
         panel.text(x[-7], y[-7], labels = rownames(rot.pattern)[-7],
            pos = 4, cex = .75)
         panel.text(x[7], y[7], labels = rownames(rot.pattern)[7],
                    pos = 1, cex = .75)
       },
       main = "Rotated Factor Pattern",
       xlab = "",
       ylab = "",
       scales = list(x = list(at = c(0, 1)),
                     y = list(at = c(-.4, 0, .6)))
)
grid.arrange(p1, p2, ncol = 2)
```



Above, we performed an oblique rotation, which means that axes are rotated based on maximum correlation of loadings within factors across features. Rotation is used to make it easier to interpret the results of a factor analysis - the axes are rotated so that the clusters of items fall as closely as possible to try to make the patterns more clear. We can visually identify differences looking at how some of the features are displayed, e.g. rotate over the x-axis like unreg, and see how some become closer to other features.

However, rotation does not change key aspects, such as the amount of variance accounted for. Additionally (from printing the new loadings), while the load values may have slightly shifted, we see the same features loading to the same factors - for example, pop.wdi, cinc, and milper all load in 3rd factor pre-and post rotation (and this holds spot checking against some others as well). Again, I went the route of just displaying two of the three factors.

Principal Components Analysis

1. What is the statistical difference between PCA and FA? Describe the basic construction of each approach using equations and then point to differences that exist across these two widely used methods for reducing dimensionality.

Factor analysis and PCA have in common that they both allow you to reduce the amount variables by capturing variance in a smaller set of variables - but fundamentally the main difference is that while PCA is a linear combination of variables, factor analysis uses latent (non-observed) variables that get at the underlying dimensions of the feature space because they are thought to CAUSE the variance among observed features..

PCA works by finding a sequence of linear combinations across the p features that have maximal variance and are mutually uncorrelated. The first component is the dimension on which observations vary the most; the second is the one on which observations vary second-most, and so on with the constraint that the vector is orthogonal to the previous vector.

The overall mathematical set-up for PCA (apologies, I wrote in symbols due to issues trying to knit!): We are solving for the optimal loading vector for each component e.g. maximizing s.t. sump of squares equals one - because the loading vector is defining a direction in the feature space where the data vary the most zij =phi11Xi1 + phi21Xi2 + [...] +phip1Xip

And then when we have n observations, x1 [...] xn projected onto directional vector, the projected values are the principal component scores z11 [...] zn1.

For Factor Analysis: We start looking at the relationship between the factor, error variance , and X: X1 =b1F+d1U1 X2 =b2F+d2U2 cov(F,U1) = cov(F,U2) = cov(U1,U2) = 0

Then we build out a series of regression equations where b's are the factor loadings - acting as the regression coefficients. We work through by assuming a mathematical combination of the features that maximizes predicted variance across features to get the first factor, and then so on.

Recapping differences in the two approaches from above, one seeks to apply weights to existing features to build up to fewer principal components whereas the other approach seeks to identify the simplest model to observe covariance in the population through the underlying factors thought to be causing this covariance.

2. Fit a PCA model. Present the proportion of explained variance across the first 10 components. What do these values tell you substantively (e.g., how many components likely characterize these data?)?

```r
#first let's fix labeling on the countries to feed into prcomp

countries <- countries %>%
  column_to_rownames("X1")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```r
pca1 <- prcomp(countries,
               scale=TRUE,
               center = TRUE); summary(pca1)
```

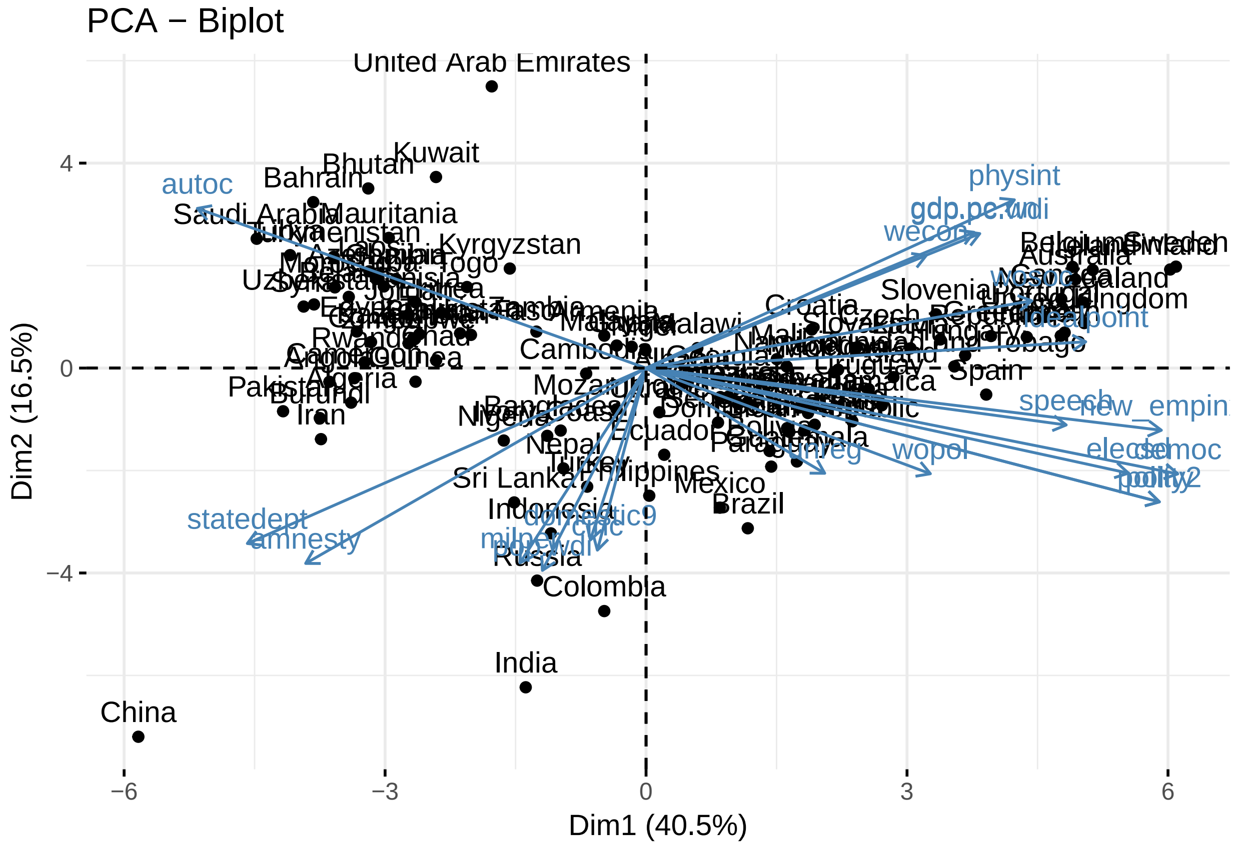
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.9173 1.8600 1.6439 1.10713 1.07631 0.91289
## Proportion of Variance 0.4053 0.1648 0.1287 0.05837 0.05516 0.03968
## Cumulative Proportion  0.4053 0.5700 0.6987 0.75708 0.81225 0.85193
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.78181 0.72948 0.64421 0.58703 0.55164 0.49341
## Proportion of Variance 0.02911 0.02534 0.01976 0.01641 0.01449 0.01159
## Cumulative Proportion  0.88104 0.90638 0.92614 0.94255 0.95704 0.96864
##                           PC13    PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.46337 0.3995 0.32765 0.29011 0.24347 0.18215
## Proportion of Variance 0.01022 0.0076 0.00511 0.00401 0.00282 0.00158
## Cumulative Proportion  0.97886 0.9865 0.99157 0.99558 0.99840 0.99998
##                           PC19      PC20      PC21
## Standard deviation     0.01990 5.378e-16 2.786e-16
## Proportion of Variance 0.00002 0.000e+00 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00 1.000e+00
```

From the above output, we can see that the first component explains 40% of the variance; the second explains an additional 17% of variance; the third explains 13%; and the fourth drops to 5%. If we took the top 8 components we could explain over 90% of the variance, and once we get to the 15th component we have 99% of the variance explained.

I would need a bit more domain expertise to understand the optimal number of components, as there is no one objective way to decide, but the way I would do so would be through looking at the proportion of variance explained (and/or a scree plot as discussed in class). In the above case, I would think that when we drop off below 5% for variance explained by each incremental component (e.g. between 5 and 6) we have hit the number that likely characterize these data.

3. Present a biplot of the PCA fit from the previous question. Describe what you see (e.g., which countries are clustered together? Which input features are doing the bulk of the explaining? How do you know this?

```r
fviz_pca_biplot(pca1)
```



(A bit easier to see expanding into a separate window) - from the above chart, we can see that in the upper left quadrant there are a number of middle eastern countries, as well as African countries when we zoon in. The upper right quadrant seems to have a number of Eastern and Western European countries; South American countries appear in the lower right quadrant but close to the origin, and the lower left quadrant seems to have a number of Asian countries, though China is a bit removed from the grouping.

The way we interpret the arrows is that low values of PC1 for example mean higher levels of "amnesty", as it is pointing to the lower level of the dimension 1. Same with autocracy for instance, whereas higher levels of PC1 would mean higher levels of "speech".

To see which features influence the most, we want to see how far the arrows extend from where they are pinned at the origin. Democracy for instance heaviy influences dimension 1; one like pop_wdi which extends far vertically can be thought to heavily influence component 2.