PCA Warmup

Katherine Wilkinson

1/19/2018

The data set fa-data.txt consisting of 500 data points in 7 dimensions. The data are believed to lie mostly neara 2-dimensional linear submanifold.

(a) 3-d visualizations

Produce a few visualization of the data set.

```
setwd("/Users/maraudersmap/Documents/Machine-Learning-in-R/PCA")
library(plotly)
fa_data = read.table('fa_data.txt', header = FALSE)


#Scale data
fa_data <- (scale(fa_data))
library(scatterplot3d)

#Function to get a graph from 3 dimensions selected randomly
random_graph <- function(d){
  rn <- sample(1:dim(d)[2], 3, replace = FALSE)
  r.var = d[,rn]

  p = plot_ly(as.data.frame(d),
              x = r.var[,1],
              y = r.var[,2],
              z = r.var[,3])
  return(p)
}
x <- sample(1:dim(fa_data)[2], 3, replace = FALSE)


#Generate a few 3d plots
#rg1 <- random_graph(fa_data)
#rg2 <- random_graph(fa_data)
#rg3 <- random_graph(fa_data)

#rg1
#rg2
#rg3
```

From our visualization we can see that our data does seem to lie near a 2-dimensional subspace.

(b) Write your own code of PCA to identify the principal components

```
# Create function to calculate first 2 PCAs
pca <- function(data){
```

```
  cov_matrix <- cov(data)
  eg_vals <- eigen(cov_matrix)$values
  eg_sorted <- sort(eg_vals, decreasing = T)

  x1 = which(eg_vals == eg_sorted[1])
  x2 = which(eg_vals == eg_sorted[2])
  x3 = which(eg_vals == eg_sorted[3])
  x4 = which(eg_vals == eg_sorted[4])
  x5 = which(eg_vals == eg_sorted[5])
  x6 = which(eg_vals == eg_sorted[6])
  x7 = which(eg_vals == eg_sorted[7])
  pc1 = eigen(cov_matrix)$vectors[,x1]
  pc2 = eigen(cov_matrix)$vectors[,x2]
  pc3 = eigen(cov_matrix)$vectors[,x3]
  pc4 = eigen(cov_matrix)$vectors[,x4]
  pc5 = eigen(cov_matrix)$vectors[,x5]
  pc6 = eigen(cov_matrix)$vectors[,x6]
  pc7 = eigen(cov_matrix)$vectors[,x7]
  pcs <- (cbind(as.matrix(pc1),as.matrix(pc2),as.matrix(pc3),as.matrix(pc4),
              as.matrix(pc5),as.matrix(pc6),as.matrix(pc7)))

  x <- c(x1,x2,x3,x4,x5,x6,x7)
  return(pcs)
}

e_vals_hl <- function(data){
  cov_matrix <- cov(data)
  eg_vals <- eigen(cov_matrix)$values
  eg_sorted <- sort(eg_vals, decreasing = T)
  x1 = eg_sorted[1]
  x2 = eg_sorted[2]
  x3 = eg_sorted[3]
  x4 = eg_sorted[4]
  x5 = eg_sorted[5]
  x6 = eg_sorted[6]
  x7 = eg_sorted[7]

  x <- c(x1,x2,x3,x4,x5,x6,x7)
  return(x)
}

e_vals_hl(fa_data)
```

```
## [1] 3.59903619 3.26308923 0.05461724 0.03770600 0.02505072 0.01589523
## [7] 0.00460539
```

```
eigen(cov(fa_data))$values
```

```
## [1] 3.59903619 3.26308923 0.05461724 0.03770600 0.02505072 0.01589523
## [7] 0.00460539
```

```
#get first 2 PCAs of fa data
fa_pca <- pca(fa_data)

colnames(fa_pca) <- c('V1','V2','V3','V4','V5','V6','V7')
```

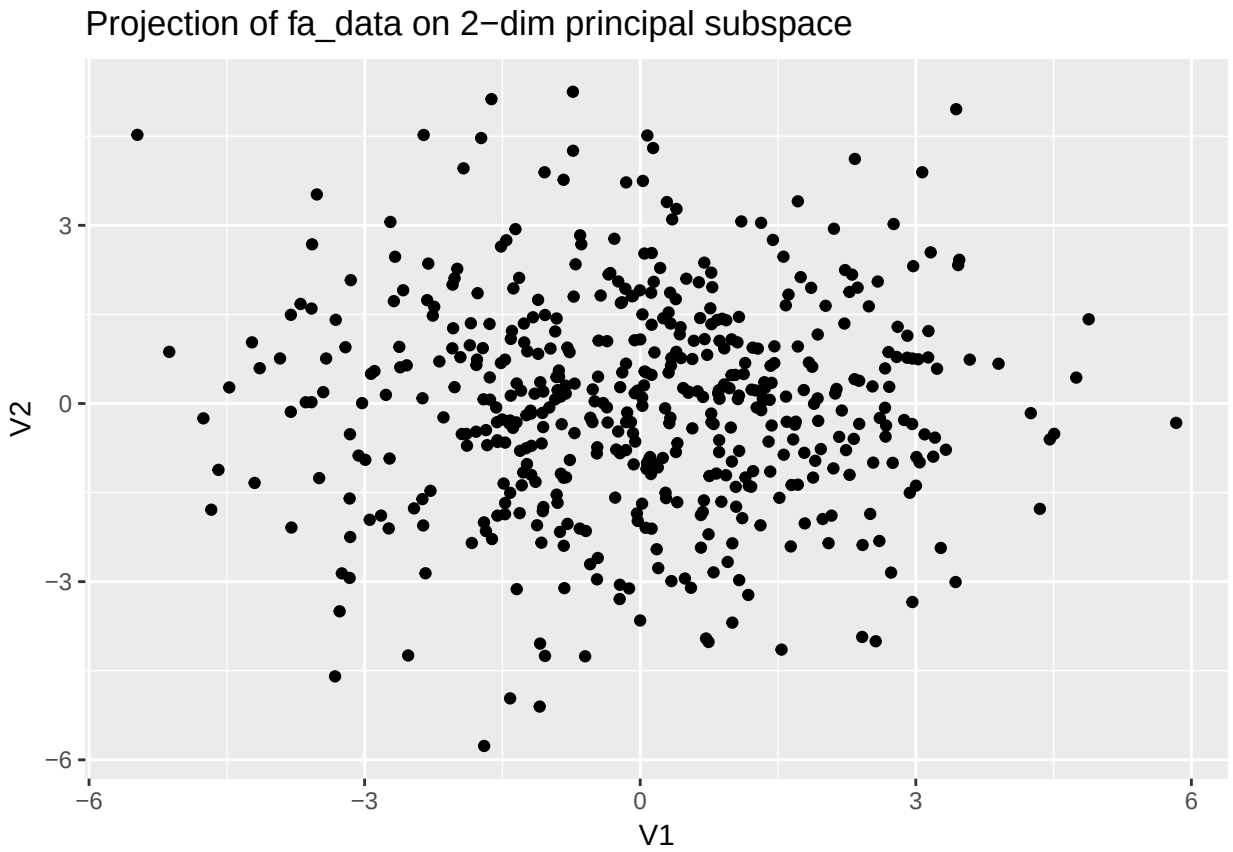
```
fa_pca[,7]
```

```
## [1]  0.03975854  0.10072538  0.24753355  0.70165604  0.22651185 -0.59541169
## [7] -0.16987319
```

```
pca_projections <- as.data.frame(as.matrix(fa_data) %*%
                                   as.matrix(fa_pca))

colnames(pca_projections) <- c('V1','V2','V3','V4','V5','V6','V7')

ggplot(data = pca_projections) +
  geom_point(aes(x = V1, y = V2))  +
  ggtitle('Projection of fa_data on 2-dim principal subspace')
```



Above we can see our principal components calculated from the eigen vectors of the covariance matrix of the standardized fa-data. We standardized the data as we are unsure of how the points are measured and there are 7 dimensions originallly. We can also see the projection on the two dimensional subspace of the first two Principal Components.

```
#Get trace of covariance matrix of fa data
trace <- function(data)sum(diag(data))
cov_fa = cov(fa_data)
tcov_fa <- trace(cov_fa)

#get eigen values and eigen vectors from fa_data
eg_vals <- eigen(cov_fa)$values
eg_sorted <- sort(eg_vals, decreasing = T)
```

```r
# Calculate lambdas
x1 = which(eg_vals == eg_sorted[1])
x2 = which(eg_vals == eg_sorted[2])
lam1 <- eg_vals[x1]

lam2 <- eg_vals[x2]

# Calculate proportion of each lambda the proportion of variance explained by the PCAS two principal co
prop1 <- lam1/(tcov_fa)
prop2 <- lam2/(tcov_fa)

prop1 + prop2
```

```
## [1] 0.9803036
```

The proportion of total variance that is explained by PCA's two principal compoenents is $0.514148 + 0.4661556 = 0.9803036$ or about 98%. Thus, it would be appropriate to use just the first two principal components for PCA.