HW9

Ines Pancorbo

3/13/2020

2. c)

```
norm <- function(x){
  sqrt(sum(x^2))
}

GramSchmidt <- function(A){
  Q <- matrix(0, nrow=nrow(A), ncol=nrow(A))
  Q[,1] <- A[,1]/norm(A[,1])

  for (i in 2:nrow(A)){
    Qt <- A[,i]
    for (j in 1:(i-1)){
      Qt <- Qt - (t(A[,i]) %*% Q[,j]) * Q[,j]
    }
    Q[,i] <- Qt/norm(Qt)
  }
  return(Q)
}
```

```
A <- matrix(runif(16), nrow=4, ncol=4)
Q <- GramSchmidt(A)
Q
```

```
##           [,1]         [,2]       [,3]       [,4]
## [1,] 0.5632041 -0.447327483 -0.2032331  0.6643761
## [2,] 0.2955326  0.893992740 -0.1201238  0.3146549
## [3,] 0.2447710 -0.024533152 -0.8412914 -0.4813669
## [4,] 0.7318120 -0.008557337  0.4863079 -0.4773705
```

```
qr.Q(qr(A))
```

```
##            [,1]         [,2]       [,3]       [,4]
## [1,] -0.5632041  0.447327483 -0.2032331 -0.6643761
## [2,] -0.2955326 -0.893992740 -0.1201238 -0.3146549
## [3,] -0.2447710  0.024533152 -0.8412914  0.4813669
## [4,] -0.7318120  0.008557337  0.4863079  0.4773705
```

From above, the columns of Q from GramSchmidt() function are the same as the columns of the Q obtained via R (except some columns are multiplied by -1 but this doesn't affect orthonormality or the equality of the spans since -1 is just a scalar).

4

a)

```r
setwd("/Users/inespancorbo/MATH504/HW9")
senators <- read.table("senators_formatted.txt", header = T, stringsAsFactors = F)
votes <- read.table("votes_formatted.txt", header = T, stringsAsFactors = F)
V <- t(as.matrix(votes[,-1]))
```

```r
# center data
mean <- colMeans(V)
V <- V - rep(mean, rep.int(nrow(V), ncol(V)))

# covariance matrix
omega <- t(V) %*% V
```

```r
power_iteration <- function(A, start) {
  start1 <- matrix(start[,1],nrow=ncol(A), ncol = 1)/norm(start[,1])
  start2 <- matrix(start[,2],nrow=ncol(A), ncol = 1)/norm(start[,2])
  RQ1 <- t(start1) %*% A %*% start1
  RQ2 <- t(start2) %*% A %*% start2

  repeat {

    start <- A %*% start
    start <- qr.Q(qr(start))

    start1 <- matrix(start[,1],nrow=ncol(A), ncol = 1)/norm(start[,1])
    start2 <- matrix(start[,2],nrow=ncol(A), ncol = 1)/norm(start[,2])

    new_RQ1 <- t(start1) %*% A %*% start1
    new_RQ2 <- t(start2) %*% A %*% start2

    if (abs(new_RQ1 - RQ1) < 10^-10 && abs(new_RQ2 - RQ2) < 10^-10){
      break
    }
    else{
      RQ1 <- new_RQ1
      RQ2 <- new_RQ2
    }
  }

  return (list(eigenvectors=start, lambda1=RQ1, lambda2=RQ2))
}
```

```r
# Computing the first two dominant eigenvectors
start <- matrix(runif(ncol(omega)*2), nrow=ncol(omega), ncol=2)
power_iteration_result <- power_iteration(omega, start)
v1 <- power_iteration_result$eigenvectors[,1]
v2 <- power_iteration_result$eigenvectors[,2]

# I am going to double check convergence
cat(power_iteration_result$lambda1, power_iteration_result$lambda2, "\n")
```

```
## 14974.9 2542.22
```

```r
cat(eigen(omega)$values[order(abs(eigen(omega)$values), decreasing = T)][1:2])
```

```
## 14974.9 2542.22
```

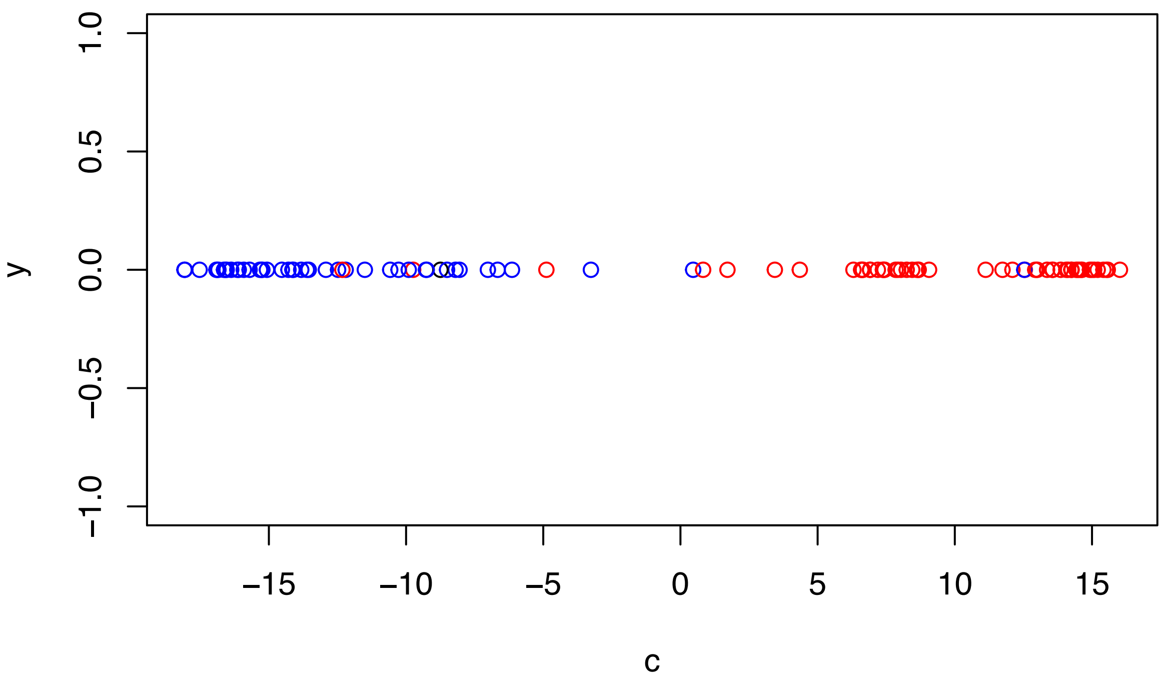
From above one can see that convergence happened. So we can use the two eigenvectors calculated via power_iteration().

b)

```r
# compute projection coefficients
c <- V %*% v1

# coloring according to party affiliation
color <- ifelse(senators$party == "R", 'red', ifelse(senators$party == "D", 'blue', 'black'))

# 1-dim plot
y <- rep(0, length(c))
plot(c, y, col = color)
```




```r
# variance preserved from 1-d projection
eigen(omega)$values[order(abs(eigen(omega)$values), decreasing = T)][1]/sum(eigen(omega)$values)
```

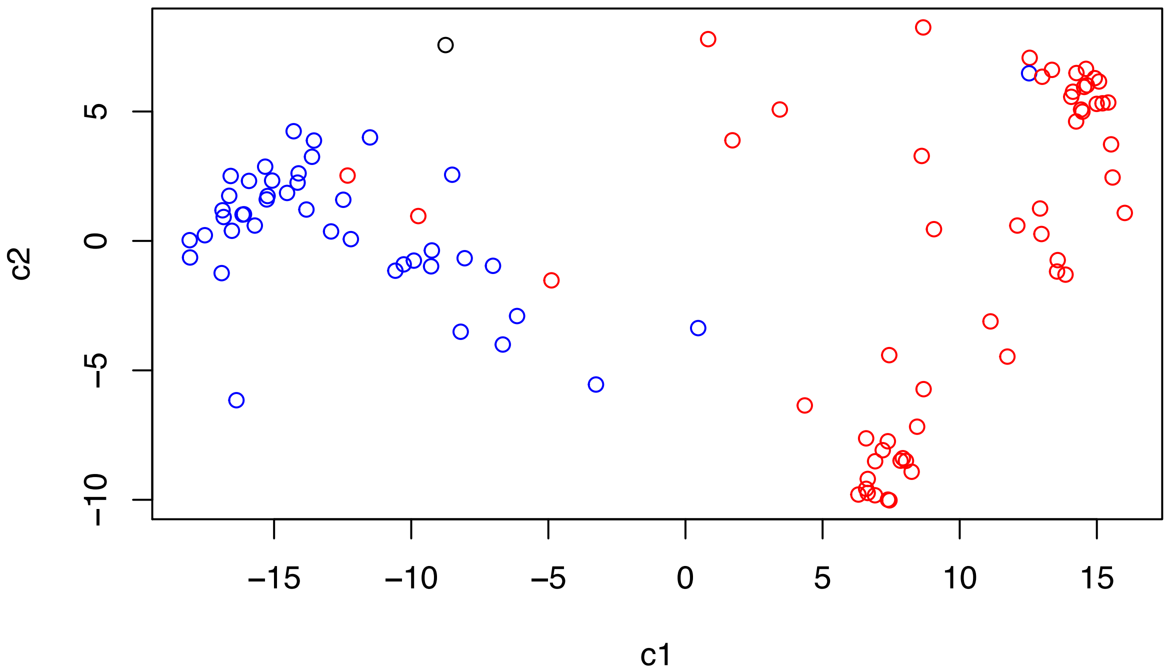
```
## [1] 0.4911232
```

c)

```r
# compute projection coefficients
c <- V %*% power_iteration_result$eigenvectors

# 2-dim plot
c1 <- c[,1]
```

```
c2 <- c[,2]
plot(c1, c2, col = color)
```




```
# variance kept from 2-d projection
sum(eigen(omega)$values[order(abs(eigen(omega)$values), decreasing = T)][1:2])/sum(eigen(omega)$values)
```

```
## [1] 0.5744989
```

HW 9

② a) $\text{Span}(A) = \text{span}(a^{(1)}, a^{(2)}, \ldots, a^{(K)})$, $a^{(i)} \in \mathbb{R}^n$

$q^{(1)} = \frac{a^{(1)}}{\|a^{(1)}\|}$

for $i = 2, 3, \ldots K$ {

$q^{(j)}$ for $j = 1, 2, \ldots, i-1$ has been defined

$\tilde{q}^{(i)} = a^{(i)} - \sum_{j=1}^{i-1} (a^{(i)} \cdot q^{(j)})\, q^{(j)}$

$q^{(i)} = \frac{\tilde{q}^{(i)}}{\|\tilde{q}^{(i)}\|}$

}

b) Lets first prove that the $q^{(i)}$ are orthonormal.

Base case: $q^{(1)} = \frac{\tilde{q}^{(1)}}{\|\tilde{q}^{(1)}\|}$ from above so normalized. It is also trivially orthogonal

Inductive case: Suppose $q^{(i)}$ for $i = 1, 2, 3, \ldots, j-1$ are orthonormal. Take $q^{(j)}$. Since $q^{(j)} = \frac{\tilde{q}^{(j)}}{\|\tilde{q}^{(j)}\|}$ it is normalized. Then, let i, for $i = 1, 2, 3, \ldots, j-1$ be given. Consider $\tilde{q}^{(j)} \cdot q^{(i)} =$

$q^{(i)} \cdot \left(a^{(j)} - \sum^{j-1} (a^{(j)} \cdot q^{(k)})\, q^{(k)} \right) = q^{(i)} \cdot a^{(j)} - \sum_{k=1}^{j-1} (a^{(j)} \cdot q^{(k)})(q^{(k)} \cdot q^{(i)})$

$= q^{(i)} \cdot a^{(j)} - (a^{(j)} \cdot q^{(i)})(q^{(i)} \cdot q^{(i)}) = 0$ since $q^{(i)} \cdot q^{(i)} = 1$ and $q^{(i)} \cdot q^{(k)} = 0$ for $i \neq k$ ■

Now lets prove $\text{span}(A) = \text{span}(Q)$

Base case: $q^{(1)} = \frac{a^{(1)}}{\|a^{(1)}\|}$ so just scalar multiples $\Rightarrow \text{span}(q^{(1)}) = \text{span}(a^{(1)})$

Inductive case: Suppose $\text{span}(a^{(1)}, a^{(2)}, \ldots, a^{(j-1)}) = \text{span}(q^{(1)}, q^{(2)}, \ldots, q^{(j-1)})$

Consider $\tilde{q}^{(j)}$ (since $\tilde{q}^{(j)}$ is just a scalar multiple of $q^{(j)}$).

Need to show $\text{span}(a^{(1)}, a^{(2)}, \ldots, a^{(j)}) = \text{span}(q^{(1)}, q^{(2)}, \ldots \tilde{q}^{(j)})$

Note it suffices to show $a^{(j)} \in \text{span}(q^{(1)}, \ldots, \tilde{q}^{(j)})$ and $\tilde{q}^{(j)} \in \text{span}(a^{(1)}, \ldots, a^{(j)})$ since we supposed $\text{span}(a^{(1)}, \ldots, a^{(j-1)}) = \text{span}(q^{(1)}, \ldots q^{(j-1)})$. $\tilde{q}^{(j)} = a^{(j)} - \sum_{i=1}^{j-1} (a^{(j)} \cdot q^{(i)})(q^{(i)})$

so $\tilde{q}^{(j)}$ is a linear combination of the $a^{(i)}$, $i = 1, 2, \ldots j-1, j$

and $a^{(j)} = \tilde{q}^{(j)} + \sum_{i=1}^{j-1} (a^{(j)} \cdot q^{(i)})(q^{(i)})$ is also a linear comb of the $\tilde{q}^{(j)}$ and $q^{(i)}$, $i = 1, 2, \ldots, j-1$. ■

③

a) The projection of x on $span(v)$ is the closest y in $span(v)$ to x. So in other words we have $\min_{y \in span(v)} \|x-y\|$, or just $\min_{y \in span(v)} \|x-y\|^2$ (since $\min_{y \in span(v)} \|x-y\| \Rightarrow \min_{y \in span(v)} \|x-y\|^2$).

Since $y \in span(v)$ we can write $y = cv \Rightarrow \min_{c \in \mathbb{R}} \|x - cv\|^2$

and now $f(c) = \|x - cv\|^2 = (x-cv)(x-cv) = x^Tx - 2cx^Tv + c^2 v^Tv$

$f'(c) = 2x^Tv + 2cv^Tv$, $f'(c) = 0 \Rightarrow c = \frac{x^Tv}{v^Tv}$, since $\|v\| = 1$

$c = x^Tv = x \cdot v = v \cdot x$. And so we have that the projection of x on $span(v)$ is $cv = (v \cdot x)v$ ∎

b) The projection of $x \in \mathbb{R}^n$ on $span(v^{(1)}, v^{(2)})$ is the closest $y \in span(v^{(1)}, v^{(2)})$ to x. So $\min_{y \in span(v^{(1)}, v^{(2)})} \|x - y\|^2$. Since $y \in span(v^{(1)}, v^{(2)})$ y can be written as $y = Vc$ where $V = (v^{(1)}\ v^{(2)}) \in \mathbb{R}^{n \times n}$ with $V^TV = I$.

So $\min_{c \in \mathbb{R}^2} \|x - Vc\|^2 = \min_{c \in \mathbb{R}^2} (x - Vc)(x - Vc) = \min_{c \in \mathbb{R}^2} x^Tx - 2x^TVc + (Vc)^TVc$

$= x^Tx - 2x^TVc + c^TV^TVc$. This is just minimizing a quadratic and we know the closed form solution: $c = (V^TV)^{-1}V^Tx$

and since $V^TV = I \Rightarrow c = V^Tx \Rightarrow \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} v_1^{(1)} & v_2^{(1)} \\ v_1^{(2)} & v_2^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow c_1 = v^{(1)} \cdot x$, $c_2 = v^{(2)} \cdot x$

And so the projection of x onto $span(v^{(1)}, v^{(2)})$ is $(v^{(1)} \cdot x)v^{(1)} + (v^{(2)} \cdot x)v^{(2)}$ ∎

c) Since M is symmetric we know we can write $v \in \mathbb{R}^n$ as $v = Qc$ where $Q = [q^{(1)}, \dots, q^{(n)}] \in \mathbb{R}^{n \times n}$, $q^{(i)}$ eigenvectors of M

$q^{(1)}\ q^{(2)} \dots q^{(n)}$, $|\lambda_1| > |\lambda_2| \dots |\lambda_n|$

So $\max_{Qc \in \mathbb{R}^n, \|Qc\| = 1} (Qc)^T QDQ^T (Qc)$. Now $\|Qc\| = \|c\| = 1$ since Q is orthogonal matrix. So we want $\max_{c \in \mathbb{R}^n, \|c\| = 1} c^T D c =$

$\max_{c \in \mathbb{R}^n, \|c\| = 1} (c_1^2\lambda_1 + c_2^2\lambda_2 + \dots + c_n^2\lambda_n)$

Now $\lambda_1 \left(c_1^2 + c_2^2\frac{\lambda_2}{\lambda_1} + \dots + c_n^2\frac{\lambda_n}{\lambda_1}\right) \leq \lambda_1 \cdot 1 = \lambda_1$ since $\lambda_1 > \lambda_2 \geq \dots > \lambda_n$

If $c_1 = 1$ and $c_i = 0$ $\forall i = 2, 3, \dots, n$ we have maximized $c_1^2\lambda_1 + c_2^2\lambda_2 + \dots + c_n^2\lambda_n$. So we want $c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n$

$\Rightarrow v = Qc = Q\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = q^{(1)}$. So we must choose $v = q^{(1)}$ (dominant eigenvector) ∎

d) Let $x^{(i)} \in \mathbb{R}^n$ for $i = 1, 2, \dots, N$

then the projection of $x^{(i)}$ onto a 1-d linear space given by span (v) is $c_i v = (x^{(i)} \cdot v) v$ by a) b)

And we want $\min_{v \in \mathbb{R}^n} \sum_{i=1}^{N} \| x^{(i)} - (x^{(i)} \cdot v) v \|^2$

$$= \min_{v \in \mathbb{R}^n} \sum_{i=1}^{N} (x^{(i)} - (x^{(i)} \cdot v) v)(x^{(i)} - (x^{(i)} \cdot v) v)$$

$$= \min_{v \in \mathbb{R}^n} \sum_{i=1}^{N} \left[x^{(i)} \cdot x^{(i)} - 2 (x^{(i)} \cdot v)^2 + (x^{(i)} \cdot v)^2 v \cdot v \right]$$

$$= \min_{v \in \mathbb{R}^n} \sum_{i=1}^{N} - (x^{(i)} \cdot v)^2$$ since $x^{(i)} \cdot x^{(i)}$ doesn't affect the min and we can assume $\|v\| = 1$

$$= \max_{v \in \mathbb{R}^n} \sum_{i=1}^{N} (x^{(i)} \cdot v)^2$$

$$= \max_{v \in \mathbb{R}^n} \sum_{i=1}^{N} (v \cdot x^{(i)})(x^{(i)} \cdot v) = \max_{v \in \mathbb{R}^n} \sum_{i=1}^{N} v^T x^{(i)} x^{(i)T} v$$

$$= \max_{v \in \mathbb{R}^n} v^T \left(\sum_{i=1}^{N} x^{(i)} x^{(i)T} \right) v .$$

Now $\sum_{i=1}^{N} x^{(i)} x^{(i)T}$ is symmetric by c) we know

$\max_{v \in \mathbb{R}^n} v^T \left(\sum_{i=1}^{N} x^{(i)} x^{(i)T} \right) v = q^{(1)}$ where $q^{(1)}$ is the dominant eigenvector of $\sum_{i=1}^{N} x^{(i)} x^{(i)T}$. So the "best" v is $q^{(1)}$.

We want the loss function to be $\sum_{i=1}^{N} \| x^{(i)} - (x^{(i)} \cdot v) v \|^2$ b/c we want to minimize the sum of the squared distance from the $x^{(i)}$ to their projections, $(x^{(i)} \cdot v) v \in$ span(v), where span(v) is a 1d linear space

Therefore to transform the dataset into 1-d dataset we let each $x^{(i)} \in \mathbb{R}^n \longmapsto x^{(i)} \cdot q^{(1)} \in \mathbb{R}$, where $q^{(1)}$ is the dominant eigenvector of $\sum_{i=1}^{N} x^{(i)} x^{(i)T}$.

e) Let $x^{(i)} \in \mathbb{R}^n$ for $i = 1, 2, \dots, N$.

Then the projection of each $x^{(i)}$ onto span$(v^{(1)}, v^{(2)})$ is $Vc^{(i)}$ where $c^{(i)} \in \mathbb{R}^2$ and $V = [v^{(1)}\ v^{(2)}] \in \mathbb{R}^{2\times 2}$

As shown in a) b), $Vc^{(i)} = VV^T x^{(i)}$. We want our loss function to be the sum of squared distances b/w $x^{(i)}$ and $VV^T x^{(i)}$ and minimize this loss function to find the appropriate V

So $\min\limits_{V \in \mathbb{R}^{2\times2}} \sum_{i=1}^{N} \| x^{(i)} - VV^T x^{(i)} \|^2 = \min\limits_{V \in \mathbb{R}^{2\times2}} \sum_{i=1}^{N} (x^{(i)} - VV^T x^{(i)})(x^{(i)} - VV^T x^{(i)})$

$= \min\limits_{V \in \mathbb{R}^{2\times2}} \sum_{i=1}^{N} \left[x^{(i)} \cdot x^{(i)} - 2x^{(i)} \cdot VV^T x^{(i)} + (VV^T x^{(i)})^T (VV^T x^{(i)}) \right]$

$= \min\limits_{V \in \mathbb{R}^{2\times2}} \sum_{i=1}^{N} \left[-2x^{(i)T} VV^T x^{(i)} + x^{(i)T} VV^T VV^T x^{(i)} \right]$ since $x^{(i)} \cdot x^{(i)}$ won't influence the minimization

$= \min\limits_{V \in \mathbb{R}^{2\times2}} \sum_{i=1}^{N} \left(- x^{(i)T} VV^T x^{(i)} \right)$ since we can assume $v^{(1)}, v^{(2)}$ are orthonormal

$= \max\limits_{V \in \mathbb{R}^{2\times2}} \sum_{i=1}^{N} x^{(i)T} VV^T x^{(i)} = \max\limits_{V \in \mathbb{R}^{2\times2}} V^T \left(\sum_{i=1}^{N} x^{(i)} x^{(i)T} \right) V$

The above can be seen also as: (if we split $V = [v^{(1)}, v^{(2)}]$ into $v^{(1)}$ and $v^{(2)}$)

$\max\limits_{v^{(1)}, v^{(2)} \in \mathbb{R}^2} v^{(1)T} \left(\sum_{i=1}^{N} x^{(i)} x^{(i)T} \right) v^{(1)} + v^{(2)T} \left(\sum_{i=1}^{N} x^{(i)} x^{(i)T} \right) v^{(2)}$

And so we can choose $v^{(1)}$ to be the dominant eigenvector of $\sum_{i=1}^{N} x^{(i)} x^{(i)T}$ and given this choice, we can choose $v^{(2)}$ to be the eigenvector corresponding to the next largest eigenvalue.

Therefore to transform the dataset into a 2-d dataset you take each $x^{(i)} \in \mathbb{R}^n \longmapsto (x^{(i)} \cdot v^{(1)}, x^{(i)} \cdot v^{(2)}) \in \mathbb{R}^2$

dominant eigenvector of $\sum_{i=1}^{N} x^{(i)} x^{(i)T}$

eigenvector w/ second largest eigenvalue in abs value of $\sum_{i=1}^{N} x^{(i)} x^{(i)T}$