Untitled

Yijie Wu

5/28/2021

1/a/i.

```
library(Matrix)
V=Diagonal(x = seq(0.8,1.25,0.05))
r=rep(0, 10)
```

```
set.seed(12345)
X0 <- MASS::mvrnorm(n=120, mu = r, Sigma = V)
```

```
colMeans(X0)
```

```
##  [1]  0.06049362 -0.11921312 -0.22629882  0.16626835  0.04674217  0.01787684
##  [7]  0.06585541  0.07328721  0.03908269  0.23791583
```

```
r_mean=colMeans(X0)
z=X0-r_mean
```

```
res=0
for (r in 1:nrow(z))
  res=res+z[r,]%*% t(z[r,])
  res=res/nrow(z)
V2=res
```

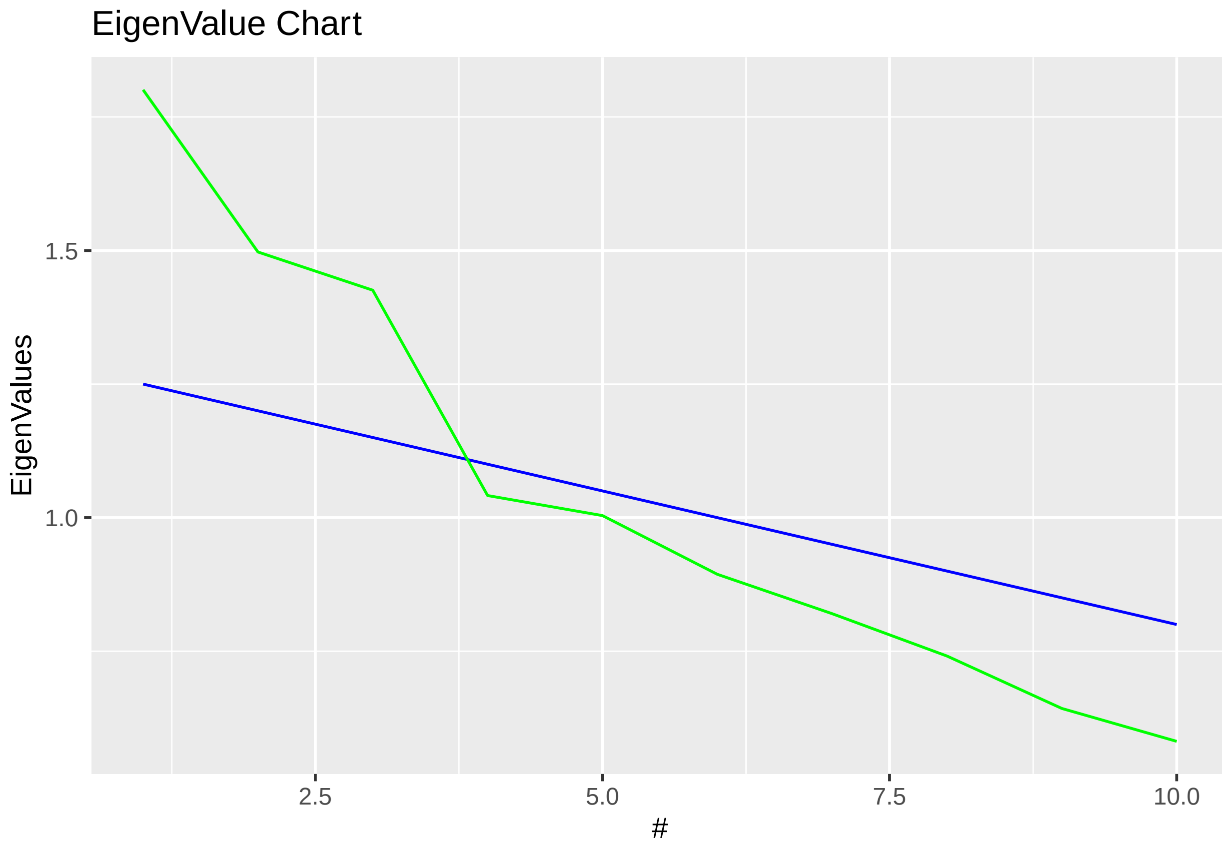
```
e <- eigen(V)
e$values
```

```
##  [1] 1.25 1.20 1.15 1.10 1.05 1.00 0.95 0.90 0.85 0.80
```

```
e2 <- eigen(V2)
e2$values
```

```
##  [1] 1.8008256 1.4972401 1.4256470 1.0413507 1.0038687 0.8939928 0.8202445
##  [8] 0.7409669 0.6428633 0.5812950
```

```
library(ggplot2)
library(RColorBrewer)
cbPalette <- brewer.pal(10, name = "Paired")
ggplot() +
    geom_line(mapping = aes(x = seq(1,10,1), y = eigen(V)$values), color = "blue") +
    geom_line(mapping = aes(x = seq(1,10,1), y = eigen(V2)$values), color = "green") +
    labs(x = "#", y = "EigenValues", title = "EigenValue Chart")
```



The eigenvalue is distorted when using only a finite set of observations

1/a/ii

```
library(CVXR)
```

```
##
## Attaching package: 'CVXR'

## The following object is masked from 'package:stats':
##
##     power
```

```
NumAssets=10
w <- Variable(NumAssets)    # decision variables
risk <- quad_form(w, V2)  # This is w' Sample_Cov w
constraints <- list(w >= 0, sum(w) == 1)
prob <- Problem(Minimize(risk), constraints)
result <- solve(prob)
MinVar <- result$getValue(risk)
w_MinVar <- result$getValue(w)
return_Min <- t(r_mean) %*% w_MinVar
```

```
expected=t(w_MinVar) %*%V2 %*%w_MinVar
actual=t(w_MinVar) %*%V %*%w_MinVar
```

```r
library(CVXR)

NumAssets=10
w <- Variable(NumAssets)   # decision variables
risk <- quad_form(w, V)  # This is w' Sample_Cov w
constraints <- list(w >= 0, sum(w) == 1)
prob <- Problem(Minimize(risk), constraints)
result <- solve(prob)
MinVar <- result$getValue(risk)
w_MinVar2 <- result$getValue(w)
return_Min <- t(r_mean) %*% w_MinVar
```

```r
true=t(w_MinVar2) %*%V %*%w_MinVar2
```

```r
expected
```

```
##             [,1]
## [1,] 0.1251664
```

```r
actual
```

```
## 1 x 1 Matrix of class "dgeMatrix"
##            [,1]
## [1,] 0.107874
```

```r
true
```

```
## 1 x 1 Matrix of class "dgeMatrix"
##             [,1]
## [1,] 0.1004562
```

iii. Actual is always larger than the true value. For the estimated value, it is really unstable and there is not an obvious trend

b.

c.

```r
lamda=eigen(V2)$values
lamda
```

```
##  [1] 1.8008256 1.4972401 1.4256470 1.0413507 1.0038687 0.8939928 0.8202445
##  [8] 0.7409669 0.6428633 0.5812950
```

```r
lamda_mean=mean(lamda)
```

```r
C=lamda_mean*diag(10)
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:CVXR':
##
##     logistic

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

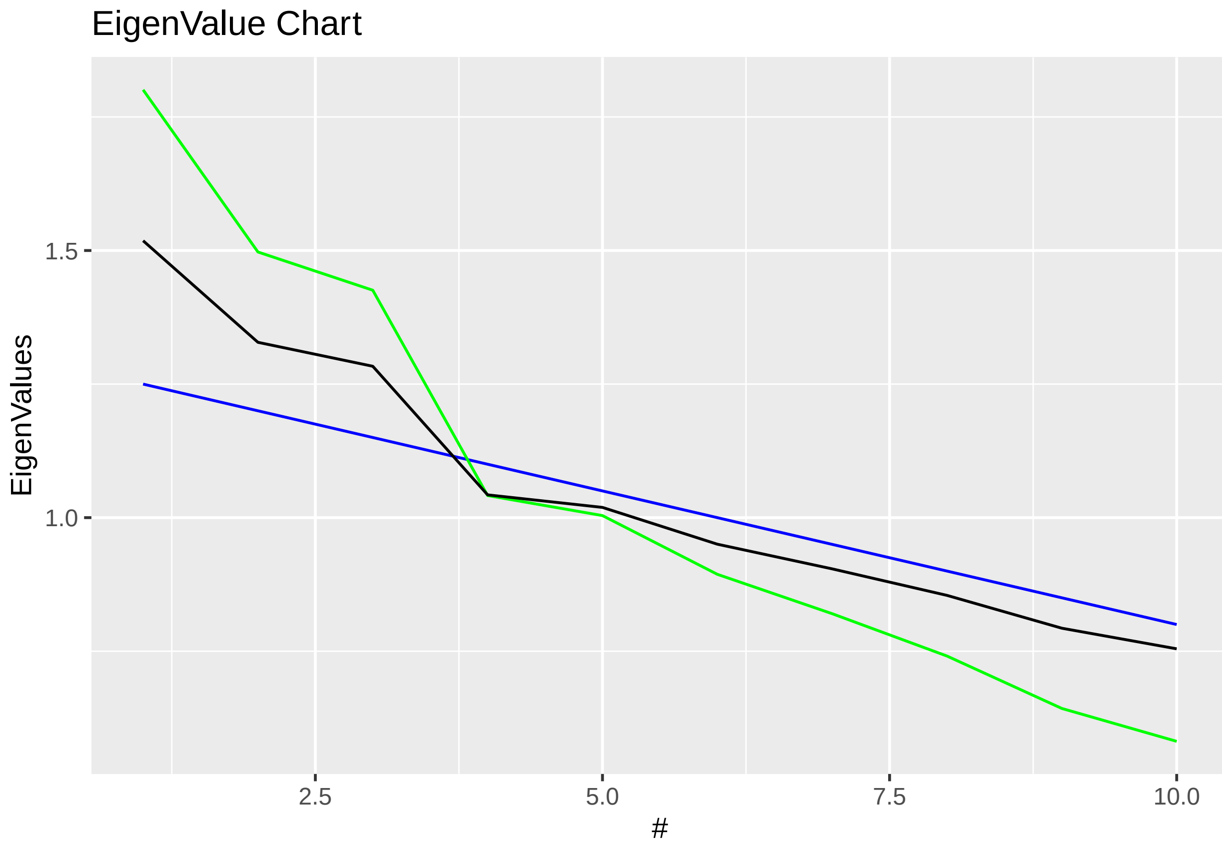
```r
res=0
for (r in 1:nrow(z))
  res=res+tr((z[r,]%*% t(z[r,])-V2)^2)
  res=res/nrow(z)
alpha=min((res/tr((V2-C)^2))/nrow(z),1)
```

```r
V3=(1-alpha)*V2+alpha*C
```

```r
eigen(V2)$values
```

```
##  [1] 1.8008256 1.4972401 1.4256470 1.0413507 1.0038687 0.8939928 0.8202445
##  [8] 0.7409669 0.6428633 0.5812950
```

```r
library(RColorBrewer)
cbPalette <- brewer.pal(10, name = "Paired")
ggplot() +
    geom_line(mapping = aes(x = seq(1,10,1), y = eigen(V)$values), color = "blue") +
    geom_line(mapping = aes(x = seq(1,10,1), y = eigen(V2)$values), color = "green") +
    geom_line(mapping = aes(x = seq(1,10,1), y = eigen(V3)$values), color = "black") +
    labs(x = "#", y = "EigenValues", title = "EigenValue Chart")
```



The eigenvalue with shrinkage method, compared with the one only using sample data, is much closer to the actual one

ii.

```
library(CVXR)

NumAssets=10
w <- Variable(NumAssets)   # decision variables
risk <- quad_form(w, V3)  # This is w' Sample_Cov w
constraints <- list(w >= 0, sum(w) == 1)
prob <- Problem(Minimize(risk), constraints)
result <- solve(prob)
MinVar <- result$getValue(risk)
w_MinVar3 <- result$getValue(w)
return_Min <- t(r_mean) %*% w_MinVar
```

```
expected=t(w_MinVar3) %*%V3 %*%w_MinVar3
actual=t(w_MinVar3) %*%V %*%w_MinVar3
```

```
library(CVXR)

NumAssets=10
w <- Variable(NumAssets)   # decision variables
risk <- quad_form(w, V)  # This is w' Sample_Cov w
```

```r
constraints <- list(w >= 0, sum(w) == 1)
prob <- Problem(Minimize(risk), constraints)
result <- solve(prob)
MinVar <- result$getValue(risk)
w_MinVar2 <- result$getValue(w)
return_Min <- t(r_mean) %*% w_MinVar
```

```r
true=t(w_MinVar2) %*%V %*%w_MinVar2
```

```r
expected
```

```
##           [,1]
## [1,] 0.1204813
```

```r
actual
```

```
## 1 x 1 Matrix of class "dgeMatrix"
##           [,1]
## [1,] 0.1026969
```

```r
true
```

```
## 1 x 1 Matrix of class "dgeMatrix"
##           [,1]
## [1,] 0.1004562
```

iii. Actual is much closer to the true value and expected is not that unstable