Problem 2.3

Chen Bo Calvin Zhang

13/10/2020

```r
# libraries needed for this document
library("mnormt")
library("corpcor")
```

True model

Create a unit diagonal covariance matrix of dimension $d = 100$.

```r
# mean = 0
mu = numeric(100)
# covariance matrix is an identity matrix
Sigma = diag(100)
```

Compute the true eigenvalues.

```r
# compute eigenvalues
ev = eigen(Sigma)
evalues = ev$values
print(evalues)
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Simulate data sets of sample size $n = 20$, $n = 50$ and $n = 200$.

```r
# simulate data sets of sizes n = 20, n = 50, n = 200
sim_data1 = rmnorm(n = 20, mu, Sigma)
sim_data2 = rmnorm(n = 50, mu, Sigma)
sim_data3 = rmnorm(n = 200, mu, Sigma)
```

Model 1 (empirical covariance)

Compute the empirical covariance matrices ($S1.emp$, $S2.emp$, $S3.emp$).

```r
# compute the empirical covariance matrices
S1.emp = cov(sim_data1)
S2.emp = cov(sim_data2)
S3.emp = cov(sim_data3)
```

Compute the corresponding eigenvalues $S1.emp$, $S2.emp$ and $S3.emp$.

```r
# compute the eigenvalues
evalues1 = eigen(S1.emp)$values
evalues2 = eigen(S2.emp)$values
evalues3 = eigen(S3.emp)$values
```

Check the rank and condition.

```r
# check the ranks and conditions
rank.condition(S1.emp)
```

```
## $rank
## [1] 19
##
## $condition
## [1] Inf
##
## $tol
## [1] 2.381444e-13
```

```r
rank.condition(S2.emp)
```

```
## $rank
## [1] 49
##
## $condition
## [1] Inf
##
## $tol
## [1] 1.148441e-13
```

```r
rank.condition(S3.emp)
```

```
## $rank
## [1] 100
##
## $condition
## [1] 34.83065
##
## $tol
## [1] 6.29443e-14
```

Compute the total squared error of the estimated eigenvalues.

```r
# compute total squared errors
mean((evalues1 - evalues)^2)
```

```
## [1] 5.640612
```

```r
mean((evalues2 - evalues)^2)
```

```
## [1] 1.982734
```

```r
mean((evalues3 - evalues)^2)
```

```
## [1] 0.5087097
```

Model 2 (regularised estimator)

compute regularised estimates.

```r
# compute the regularised estimates
S1.reg = cov.shrink(sim_data1, verbose = FALSE)
S2.reg = cov.shrink(sim_data2, verbose = FALSE)
S3.reg = cov.shrink(sim_data3, verbose = FALSE)
```

Compute the corresponding eigenvalues.

```r
# compute the eigenvalues
reg_evalues1 = eigen(S1.reg)$values
reg_evalues2 = eigen(S2.reg)$values
reg_evalues3 = eigen(S3.reg)$values
```

Check the rank and condition.

```r
# check the ranks and conditions
rank.condition(S1.reg)
```

```
## $rank
## [1] 100
##
## $condition
## [1] 1.935267
##
## $tol
## [1] 3.500822e-14
```

```r
rank.condition(S2.reg)
```

```
## $rank
## [1] 100
##
```

```
## $condition
## [1] 1.015308
##
## $tol
## [1] 2.210554e-14
```

```r
rank.condition(S3.reg)
```

```
## $rank
## [1] 100
##
## $condition
## [1] 1.056736
##
## $tol
## [1] 2.29092e-14
```

Compute the total squared error of the estimated eigenvalues.

```r
# compute total squared errors
mean((reg_evalues1 - evalues)^2)
```

```
## [1] 0.02409171
```

```r
mean((reg_evalues2 - evalues)^2)
```

```
## [1] 0.0002908906
```

```r
mean((reg_evalues3 - evalues)^2)
```

```
## [1] 0.0002380065
```

Conclusions

Regularised estimates are better than the empirical estiamtes when working with high dimensions and small sample sizes.

From the values of the ranks and the conditions we can also see that the shrinkage covariance is non-singular even for $n < d$.