Decorrelation methods and their effects on proposed method

Xuelong Wang

2018-10-04

Contents

1 Motivation

Based on the previous simulation result, we found that the decorrelation step has a big influence on the final performance of the proposed method. More sepcifically, when the $n < p$ is happening then we known that the sample covariance matrix Σ_X is not full rank. Therefore, Σ_X^{-1}, the inverse of Σ_X, doesn't exist. So we could calculate the general inverse of the covariance matrix Σ_X. In such situation, I just adapted one of commonly used g-inverse – the moore penrose inverse Σ_X^+ during the decorrelation procedure. But the result is not very well compared with the original method. Thus, the following is trying to discuss the reason of why this is not working.

2 SVD decorrelation procedure

$$Var(X) = \Sigma_X = U\Lambda V^T,$$

- X is the random vector with dim as $p \times 1$,
- Σ_X is $p \times p$ symmtery matrix,
- $U = V$ are orthogonal matrix and each column is the eigenvector
- Λ is a diagonal matrix with each diagonal element as the eigenvalue.

2.1 Assume the Σ_X is full rank

To decorreate the X, we could just take the reciprocal of each square root of eigenvalue as following.

$$\Sigma_X^{-1/2} = U\Lambda^{-1/2}U^T,$$

where $\Lambda^{-1/2} = \begin{bmatrix} e_1^{-1/2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e_p^{-1/2} \end{bmatrix}$

So that after transformation the $\Sigma_X^{-1/2}X$ has identity covariance matrix as following,

$$Var(\Sigma_X^{-1/2}X) = \Sigma_X^{-1/2}\Sigma_X\Sigma_X^{-1/2} = U\Lambda^{-1/2}U^TU\Lambda^{-1}U^TU\Lambda^{-1/2}U^T = I_p.$$

2.2 Assume the Σ_X is not full rank

$$Var(X) = \Sigma_X = U\Lambda V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} = U_1\Lambda_1U_1^T,$$

- U_1 is a $p \times r$ matrix with r < p and in most of case r = n the sample size.

Then after applying the same procedure we get following,

$$\Sigma_X^{-1/2} = U_1\Lambda_1^{-1/2}U_1^T,$$

Note that in this case, I'm using Moore Penrose inverse.

After transformation the X we have,

$$Var(\Sigma_X^{-1/2}X) = \Sigma_X^{-1/2}\Sigma_X\Sigma_X^{-1/2} = U_1\Lambda_1^{-1/2}U_1^TU_1\Lambda_1^{-1}U_1^TU_1\Lambda_1^{-1/2}U_1^T = U_1U_1^T,$$

Note that by the property of the U we have

$$U_1U_1^T + U_2U_2^T = I_p(U_1U_1^T)^TU_1U_1^T = U_1U_1^T,$$

Besides, $U_1U_1^T$ and $U_2U_2^T$ are indempotent and $rank(U_2U_2^T) + rank(U_1U_1^T) = p$.

So if the X is not full rank we cannot decorrelation the covariance matrix to an identity matrix.

2.3 Simulation stduy

2.3.1 Simulation 1

```
# How the singular sample covariance affect the SVD decorrelation result
set.seed(123)
p <- 200
n <- 200
Sig <- matrix(rep(0.5, 200 * 200), ncol = 200)
diag(Sig) <- 1
x_total <- mvrnorm(n, numeric(p), Sigma = Sig)

x_100 <- x_total[1:100, ]
Est_sqrt_ins_cov_100 <- invsqrt(cov(x_100))
cor(x_100 %*% Est_sqrt_ins_cov_100)[1:5, 1:5] %>% round(., 4)
FALSE          [,1]    [,2]    [,3]   [,4]    [,5]
FALSE [1,]  1.0000 -0.0480 -0.0600 0.0396 -0.0232
FALSE [2,] -0.0480  1.0000 -0.0339 0.1562 -0.0701
FALSE [3,] -0.0600 -0.0339  1.0000 0.0389 -0.0570
FALSE [4,]  0.0396  0.1562  0.0389 1.0000  0.0734
FALSE [5,] -0.0232 -0.0701 -0.0570 0.0734  1.0000
cor(x_100 %*% Est_sqrt_ins_cov_100) %>% abs(.) %>% sum(.)
FALSE [1] 2480.243
cor(x_100 %*% Est_sqrt_ins_cov_100) %>% diag(.) %>% sum(.)
FALSE [1] 200
```

```r
cor(x_100 %*% Est_sqrt_ins_cov_100)[cor(x_100 %*% Est_sqrt_ins_cov_100) %>%
    lower.tri(., diag = FALSE)] %>% max()
FALSE [1] 0.2839148

x_200 <- x_total
Est_sqrt_ins_cov_200 <- invsqrt(cov(x_200))
cor(x_200 %*% Est_sqrt_ins_cov_200)[1:5, 1:5] %>% round(., 4)
FALSE         [,1]    [,2]    [,3]   [,4]    [,5]
FALSE [1,]  1.0000  0.0032  0.0024 -3e-04  0.0034
FALSE [2,]  0.0032  1.0000 -0.0049  6e-04 -0.0069
FALSE [3,]  0.0024 -0.0049  1.0000  4e-04 -0.0052
FALSE [4,] -0.0003  0.0006  0.0004  1e+00  0.0006
FALSE [5,]  0.0034 -0.0069 -0.0052  6e-04  1.0000
cor(x_200 %*% Est_sqrt_ins_cov_200) %>% abs(.) %>% sum(.)
FALSE [1] 327.6482
cor(x_200 %*% Est_sqrt_ins_cov_200) %>% diag(.) %>% sum(.)
FALSE [1] 200
cor(x_200 %*% Est_sqrt_ins_cov_200)[cor(x_200 %*% Est_sqrt_ins_cov_200) %>%
    lower.tri(., diag = FALSE)] %>% max()
FALSE [1] 0.03669759

# if we use the inverse information of x_200
cor(x_100 %*% Est_sqrt_ins_cov_200)[1:5, 1:5] %>% round(., 4)
FALSE         [,1]    [,2]    [,3]    [,4]    [,5]
FALSE [1,]  1.0000 -0.0065 -0.1252 -0.0091 -0.0865
FALSE [2,] -0.0065  1.0000 -0.0076  0.2172 -0.0999
FALSE [3,] -0.1252 -0.0076  1.0000  0.0741 -0.0986
FALSE [4,] -0.0091  0.2172  0.0741  1.0000  0.0489
FALSE [5,] -0.0865 -0.0999 -0.0986  0.0489  1.0000
cor(x_100 %*% Est_sqrt_ins_cov_200) %>% abs(.) %>% sum(.)
FALSE [1] 2479.92
cor(x_100 %*% Est_sqrt_ins_cov_200) %>% diag(.) %>% sum(.)
FALSE [1] 200
cor(x_100 %*% Est_sqrt_ins_cov_200)[cor(x_100 %*% Est_sqrt_ins_cov_200) %>%
    lower.tri(., diag = FALSE)] %>% max()
FALSE [1] 0.2662189
```

- As we expected, when $n < p$, SVD decorrelation's result is not as good as full rank case ($n \geq p$), which means off diagonal elements are not equal or closed to zero
- By just looking at the final correlation matrix,