Decorrelation methods and their effects on proposed method

Xuelong Wang

2018-10-23

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 specifically, 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 decorrelation procedure using Eigenvalue Decomposition

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

- X is the random vector with dim as $p \times 1$,
- Σ_X is $p \times p$ symmetry and p.d. matrix,
- 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.

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

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

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

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

2.2 Assume the Σ_X is not full rank

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

- 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,

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

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

After transformation the X we have,

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

Note that by the property of the U we have

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

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  0.0092 -0.0078  0.1251  0.0462
FALSE [2,]  0.0092  1.0000 -0.0707 -0.0113  0.1233
```

```
FALSE [3,] -0.0078 -0.0707  1.0000  0.0267  0.0680
FALSE [4,]  0.1251 -0.0113  0.0267  1.0000 -0.1413
FALSE [5,]  0.0462  0.1233  0.0680 -0.1413  1.0000
cor(x_100%*%Est_sqrt_ins_cov_100) %>% abs(.) %>% sum(.)
FALSE [1] 2485.784
cov(x_100%*%Est_sqrt_ins_cov_100) %>% diag(.) %>% sum(.)
FALSE [1] 99
cor(x_100%*%Est_sqrt_ins_cov_100)[cor(x_100%*%Est_sqrt_ins_cov_100) %>%
                                    lower.tri(., diag = FALSE)] %>% max()
FALSE [1] 0.3012618

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 -0.0086 -0.0089  0.0045  0.0124
FALSE [2,] -0.0086  1.0000 -0.0045  0.0023  0.0063
FALSE [3,] -0.0089 -0.0045  1.0000  0.0024  0.0065
FALSE [4,]  0.0045  0.0023  0.0024  1.0000 -0.0033
FALSE [5,]  0.0124  0.0063  0.0065 -0.0033  1.0000
cor(x_200%*%Est_sqrt_ins_cov_200) %>% abs(.) %>% sum(.)
FALSE [1] 325.0432
cov(x_200%*%Est_sqrt_ins_cov_200) %>% diag(.) %>% sum(.)
FALSE [1] 199
cor(x_200%*%Est_sqrt_ins_cov_200)[cor(x_200%*%Est_sqrt_ins_cov_200) %>%
                                    lower.tri(., diag = FALSE)] %>% max()
FALSE [1] 0.03819551

# 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  0.0589  0.0833  0.0956  0.0781
FALSE [2,] 0.0589  1.0000 -0.0444 -0.0154  0.1082
FALSE [3,] 0.0833 -0.0444  1.0000  0.0436  0.0440
FALSE [4,] 0.0956 -0.0154  0.0436  1.0000 -0.0724
FALSE [5,] 0.0781  0.1082  0.0440 -0.0724  1.0000
cor(x_100%*%Est_sqrt_ins_cov_200) %>% abs(.) %>% sum(.)
FALSE [1] 2481.806
cov(x_100%*%Est_sqrt_ins_cov_200) %>% diag(.) %>% sum(.)
FALSE [1] 199
cor(x_100%*%Est_sqrt_ins_cov_200)[cor(x_100%*%Est_sqrt_ins_cov_200) %>%
                                    lower.tri(., diag = FALSE)] %>% max()
FALSE [1] 0.3148402
```

- 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
- The largest correlation coefficient of X_{100} is 0.28 and the for X_{200} is 0.0036
- If we use the sample variance of X_{200} to decorrelate X_{100}, it seems there is a little improvement on the max correlation coefficient

2.3.2 Simulation 2

Next, I tried to evaluate the performance of the GCTA and proposed method under the $n < p$ scenario. To keep the problem simple, I just let the covariates to be independent to each other. We want to see if the original GCTA method is affected by the singular sample covariance matrix problem. If it was able to work fine, then it verify the proposed method. More specifically, If we could find a linear transformation on covariates X to make them independent or un-correlated at least in population level (although we still have the singular sample covariance matrix problem), then we could still get a unbiased estimation of the total covariates variance.

2.3.2.1 A toy sample

```
x <- mvrnorm(100, numeric(200), diag(200))
cor(x) %>% abs(.) %>% sum(.)
FALSE [1] 3412.224
max(cor(x)[lower.tri(cor(x))])
FALSE [1] 0.3585155
```

- this is to show that although the covariates are independent in population level, their sample correlation may still be large
- the GCTA method may be effected by the this problem or not

2.3.2.2 Simulation set up

$$X = [x_1^2, \ldots, x_p^2]^T$$

- $p = 34$
- $x_i \overset{iid}{\sim} N(0, 1)$
- So x_i^2 are iid chi-square with $df = 1$

We consider to estimate the total variance, so that the total covariates (combined main and interaction terms) is 595. Let the sample size n increase from 100 to 600 in order to see how the $p < n$ simulation affect the GCTA and proposed method's performance. Following is the simulation result.

2.3.2.3 Simulation results

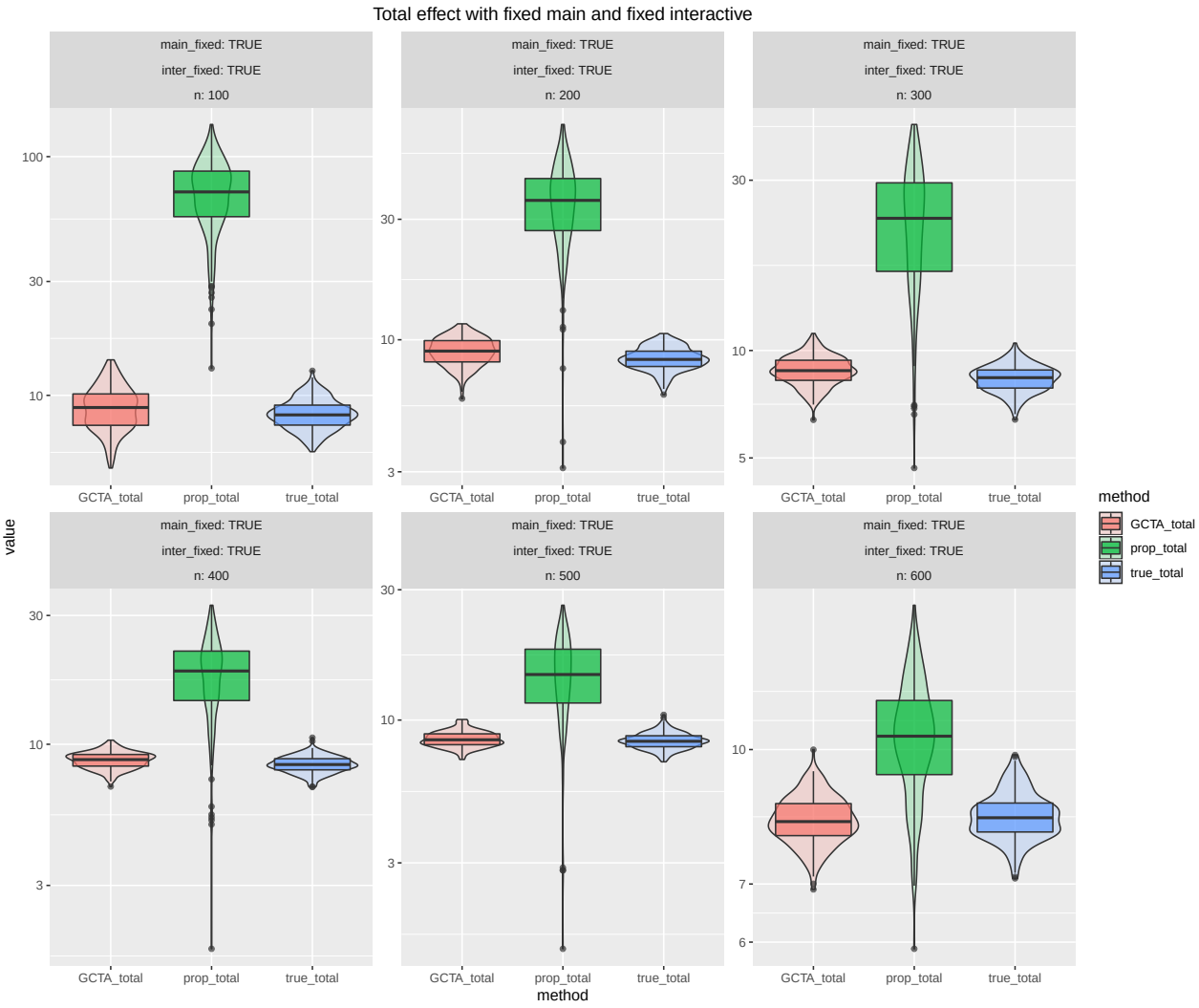

Figure 1: Fixed_Fixed total independet chi with df = 1

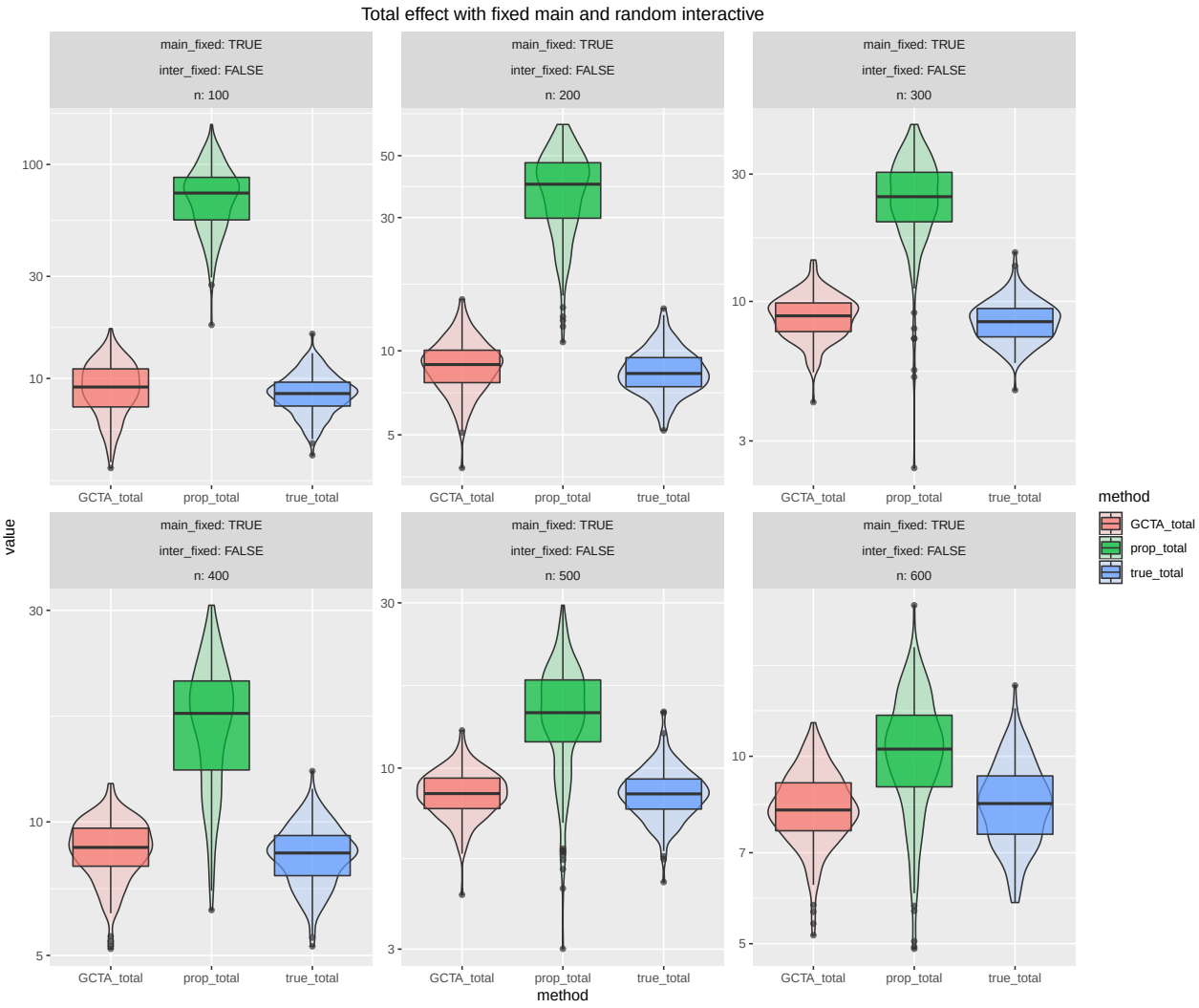

Figure 2: fixed_random total independet chi with df = 1

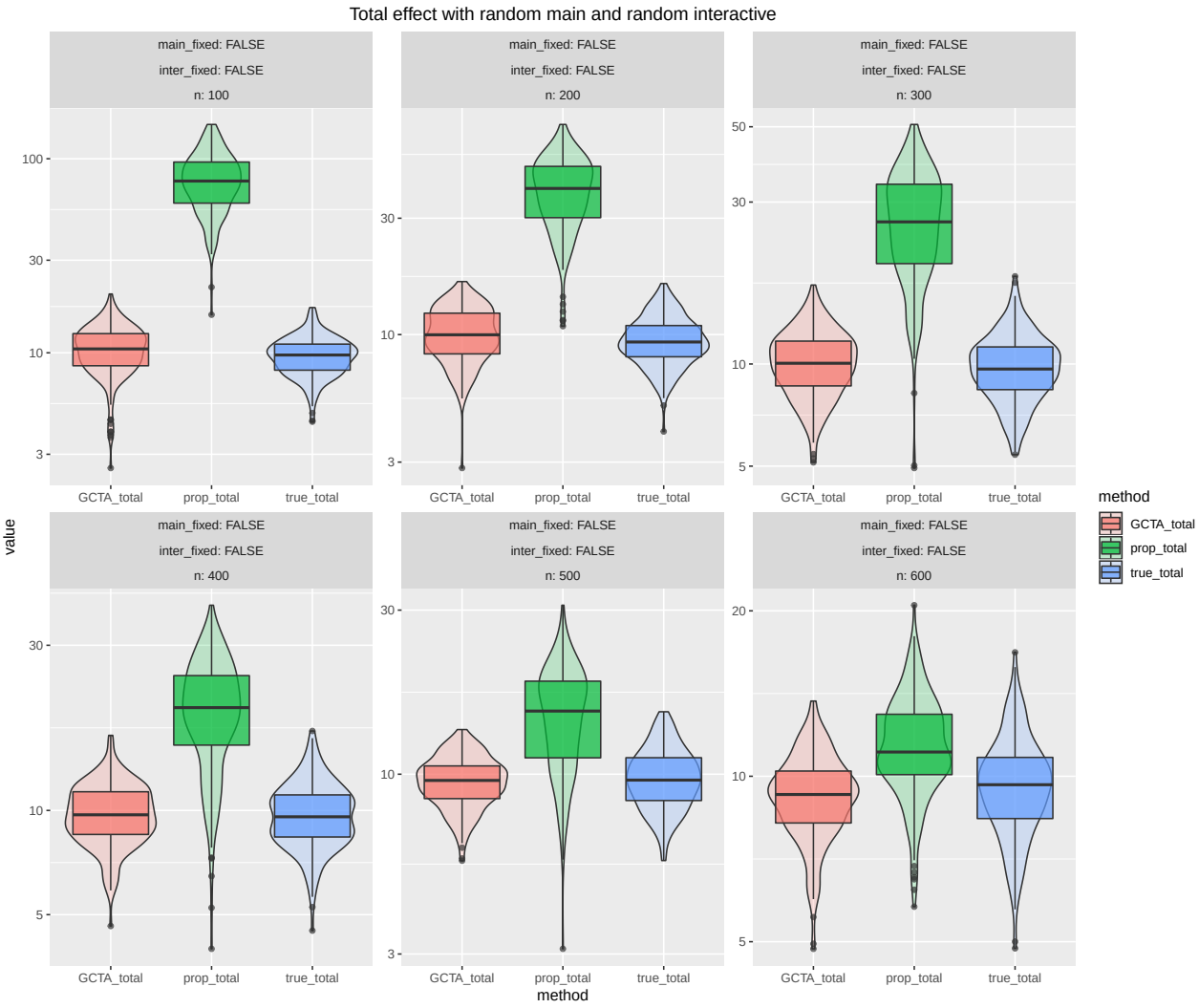

Figure 3: random_random total independet chi with df = 1

- The original GCTA method is not affected by the $p < n$ condition if the covariates are independent in the population level
- note that actual the covariates are not perfectly independent because of the interaction terms
- the proposed method is affected a lot by the $p < n$ situation, but it getting better when sample size is increasing

3 Lasso regression decorrelation procedure

3.1 Motivation

Based on the previous result, we find that the original GCTA is not sensitive to the $p < n$ situation. So the next question is to find a way to decorrelating the covariates so that they become uncorrelated in the population level. The SVD method seems not work well in that situation as previous simulations shows. Therefore, we look for the lasso regression method.

3.2 Main Idea and step

The main idea of lasso regression is to find a procedure of decorrelation. we could consider each covaraite as the dependent variable and select several other covariates as the independent variables. Then, we could preformance a lasso regression and use the residual as the new covariates. After doing that the residuals should be uncorrelated to each other.

$$X = [X_1, \dots X_p]$$

- X is a $n \times p$ observed covariates matrix
- $X_i, \quad i = 1, \dots, p$ are the columns of X

1. For each X_i, select a group $\mathcal{A}_t$ of variable as the independent variable
2. Conduct lasso regression and decide the final group of active variables as the independent variables $\mathcal{A}_f$, note that $\mathcal{A}_f \subset \mathcal{A}_t$
3. Conduct a linear regression with $Y = X_i \;\; X = X_j, j \in \mathcal{A}_f$ and use the residual as the new covariate $Z_i = X_i - \hat{X}_i$

4 Sigularity of the sample covariance matrix

Based on the simulation study, the results suggest that the proposed method has a bias in total (ultimate) effect estimation when $n < p$ situation. One of the most relative reason is that the way we used to estimate the sample covariance matrix. We suspect that the singularity of the sample covariance (Note its $rank \leq p-1$, because it using sample mean to calculate the sample covariance) affects the result the proposed method.

4.1 Solution 1, using SVD to reduce the total covariate matrix into a $n \times n$ matrix

4.1.1 Main idea

$$X = U\Sigma U^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 & U_2 \\ U_3 & U_4 \end{bmatrix}^T = \begin{bmatrix} U_1\Lambda_1 & 0 \end{bmatrix} \begin{bmatrix} U_1^T & U_3^T \\ U_2^T & U_4^T \end{bmatrix} = \begin{bmatrix} U_1\Lambda_1 U_1^T & U_1\Lambda_1 U_3^T \end{bmatrix}$$

Ignore the U_3 , then we have the X_r as following

$$X_r = U_1\Lambda_1 U_1^T.$$

We use X_r as the new covariates to the proposed methd. Therefore, we reduce the dimension from p to n.

4.1.2 Simulation setup

$$X = [X_1 \dots, X_p], \quad X_i \sim \chi^2_{(1)}, \quad cov(X_i, X_j) = 0.25, \quad \forall i \neq j$$

- The sample size n is from 100 to 700
- The number of main effect is 34 (p = 34)
- The number of interaction effect is 561
- The number of totoal effect is 595

Therefore, when $n \leq 591$, we using the svd_dimension reduction method. When $n > 595$, we don't reduce the dimension of the total covariates

4.1.3 Simulation result

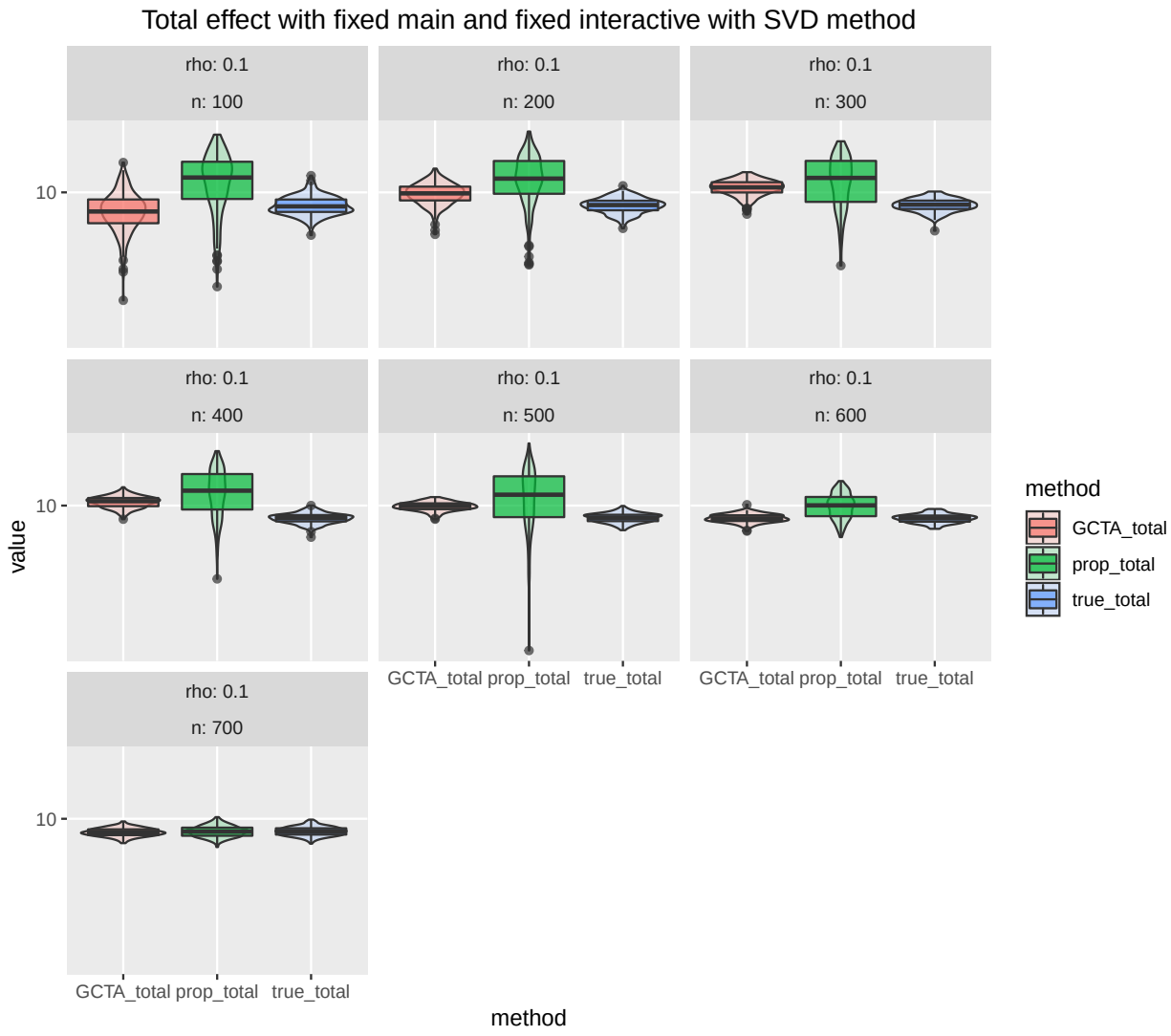

Figure 4: SVD_fixed_fixed

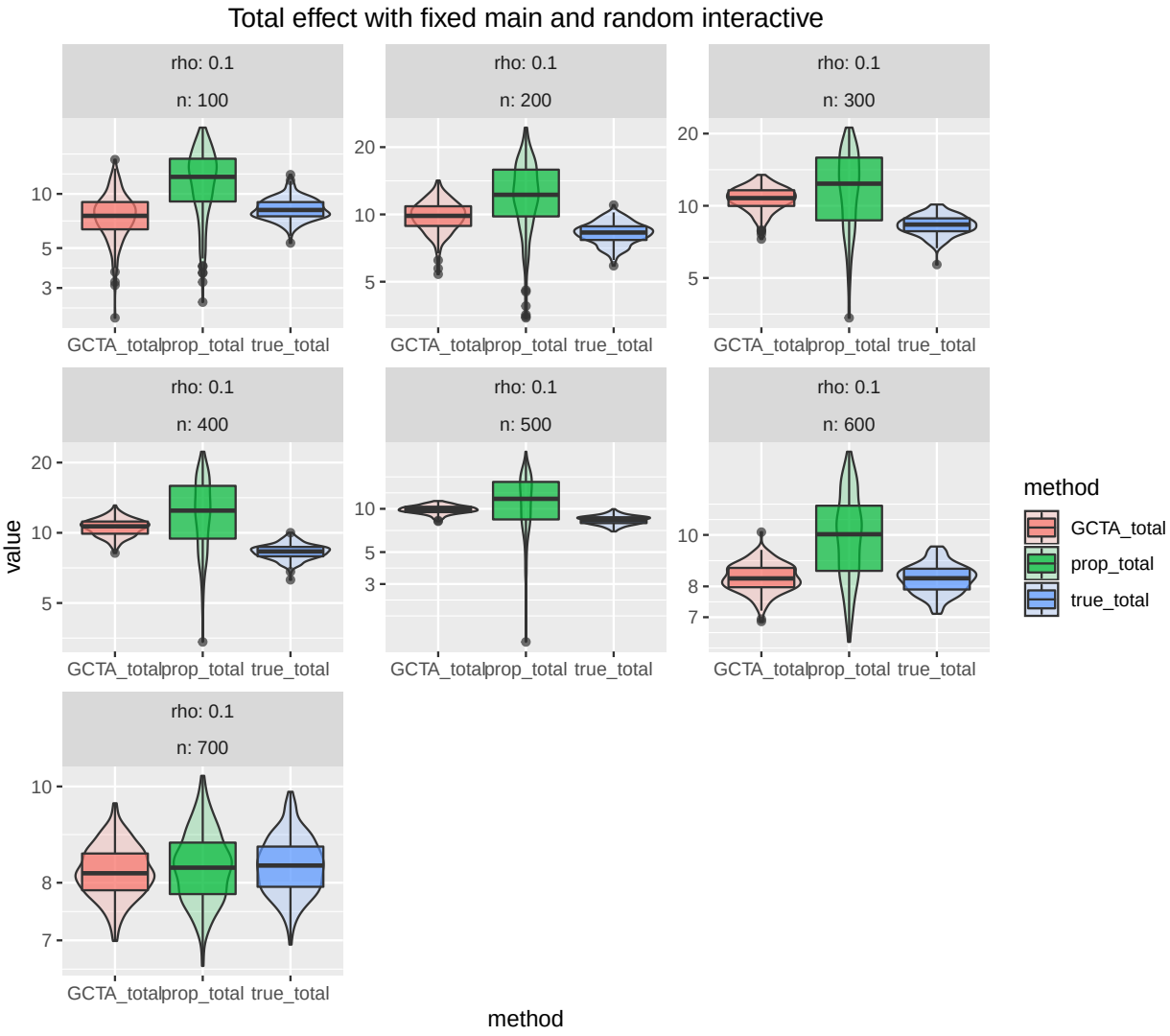

Figure 5: SVD_fixed_random

4.2 Solution 2, using PCA Regression (PCR) method

4.2.1 Motivation

Since we only care about the total effect's variance $Var(X_{total}^T \beta_{total})$, it may be possible to use PCA method to pre-screenning the data and reduce the dimension according to the sample variance. After the PCA transformation, we have $T = XW$, then we could use T as the covairate matrix to do the analysis.

4.2.2 Main idea of PCA

4.2.2.1 Singular Value Decomposition

$$X_s = UDV^T, \text{ where } x_{ij,s} = \frac{x_{ij} - \bar{x}}{s_j}$$

- $U = (u_1, \ldots, u_r)$ is a n by r orthogonal matrix
- $D = diag(d_1, \ldots, d_r)$ is a r by r diagonal matrix
- $V = (v_1, \ldots, v_r)$ is a p by r orthogonal matrix

4.2.2.2 Principle Component and Loading

$$X_s = \underbrace{\left[d_1 u_1 \ldots d_r u_r\right]}_{\text{PCs}} \underbrace{\begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}}_{\text{Loading}}$$

- $PC_j = d_j \boldsymbol{u_j} = X\boldsymbol{v_j}$ is the jth principle component
- The sample variance of PC_j is d_j^2/n

4.2.2.3 Using PCA to reduce the X's dimension

$$X_{s,k} = \sum_{j=1}^{k} d_j \boldsymbol{u_j v_k} = U_k D_k V_k^T, \quad \text{Its Variation} \quad \sum_{j=1}^{k} d_j^2/n.$$