Midterm 2 for Linear Models

Group: OR Group

December 4, 2018

Consider a Linear Regression model $Y_{n1} = X_{n2}\beta_{21} + \epsilon_{n1}$ where $E(\epsilon) = 0$ and $E(\epsilon^2) = \sigma^2 I$. The covariates $\sum_{i=1}^{n}(x_{i1}) = \sum_{i=1}^{n}(x_{i2}) = 0$ and $\sum_{i=1}^{n}(x_{i1})^2 = \sum_{i=1}^{n}(x_{i2})^2 = 1$ and $\sum_{i=1}^{n}(x_{i1}x_{i2}) = r$ where r is the Pearson correlation coefficient.

1) For the two variable model specified above, derive the least squares estimator of $\hat{\beta}$ and its variance.

Sol: Estimate $\hat{\beta}$ using least square estimation:

$$Y = X\hat{\beta} + \epsilon$$

$$\begin{aligned}\epsilon^T\epsilon &= (Y - X\hat{\beta})^T(Y - X\hat{\beta}) \\ &= Y^TY - Y^TX\hat{\beta} - \hat{\beta}^TX^TY + \hat{\beta}^TX^TX\hat{\beta} \\ &= Y^TY - 2\hat{\beta}^TX^TY + \hat{\beta}^TX^TX\hat{\beta}\end{aligned} \tag{0.1}$$

$$\frac{\partial \epsilon^T\epsilon}{\partial \hat{\beta}} = 0 \tag{0.2}$$

$$\hat{\beta}X^TX = X^TY \tag{0.3}$$

$$\hat{\beta} = (X^TX)^{-1}X^TY \tag{0.4}$$

$$(X^TX) = \begin{bmatrix} x_{11}^2 + x_{21}^2 + x_{31}^2 + ...x_{n1}^2 & x_{11} * x_{12} + x_{21} * x_{22} + x_{31} * x_{32} + ...x_{n1} * x_{n2} \\ x_{11} * x_{12} + x_{21} * x_{22} + x_{31} * x_{32} + ...x_{n1} * x_{n2} & x_{11}^2 + x_{21}^2 + x_{31}^2 + ...x_{n1}^2 \end{bmatrix}$$

Therefore, $(\mathrm{X}^TX) = \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}$

$(\mathrm{X}^TX)^{-1} = \frac{1}{1-r^2}\begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix}$

Hence,

$$\hat{\beta} = (X^TX)^{-1}X^TY$$

where, $(\mathrm{X}^TX)^{-1} = \frac{1}{1-r^2}\begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix}$

Estimate $var(\hat{\beta})$:

$var(\hat{\beta})$= var$((\mathrm{X}^TX)^{-1}X^TY)$

Let $(\mathrm{X}^T X)^{-1} X^T = Z$

Then, $var(\hat{\beta})$= var(ZY)= $ZZ^T var(Y)$

$$ZZ^T = ((X^T X)^{-1} X^T)((X^T X)^{-1} X^T)^T$$

$$= (X^T X)^{-1} X^T X (X^T X)^{-1}$$
$$= (X^T X)^{-1}$$

Therefore, $var(\hat{\beta})$=

$$(X^T X)^{-1} var(Y)$$
$$= (X^T X)^{-1} var(X\beta + \epsilon)$$
$$= (X^T X)^{-1} \sigma^2$$

where,

$(\mathrm{X}^T X)^{-1} = \frac{1}{1-r^2} \begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix}$

■

2) Consider the two estimators of $\beta = (\beta_1, \beta_2)$ by Principal Components (PCR) and Ridge Regression. Using the transformation Z = XL where L'L = LL = I, transform the linear model Y = Xβ + ϵ yielding Y = Zα + ϵ where α = L'β

i) For the two variable model specified above, derive the principal components estimator of $\hat{\beta}$ and its variance.

Sol:

Equation for OLS is Y = Xβ + ϵ. Replace X with Z=XL where L'L = LL' = I and $\alpha = L'\beta$

Substituting above values, we get the linear equation in terms of Z.

$$Y = Z\hat{\beta} + \epsilon$$
$$\epsilon^T \epsilon = (Y - Z\hat{\beta})^T (Y - Z\hat{\beta})$$
$$= Y^T Y - Y^T Z\hat{\beta} - \hat{\beta}^T Z^T Y + \hat{\beta}^T Z^T Z\hat{\beta} \quad (0.5)$$
$$= Y^T Y - 2\hat{\beta}^T Z^T Y + \hat{\beta}^T Z^T Z\hat{\beta}$$
$$\frac{\partial \epsilon^T \epsilon}{\partial \hat{\beta}} = 0 \quad (0.6)$$
$$\hat{\beta} Z^T X = Z^T Y \quad (0.7)$$
$$\hat{\beta} = (Z^T Z)^{-1} Z^T Y \quad (0.8)$$

Estimate $var(\hat{\beta})$:

$var(\hat{\beta})$= var($(\mathrm{Z}^T Z)^{-1} Z^T Y$)

Let $(\mathrm{Z}^T Z)^{-1} Z^T = P$

Then, $var(\hat{\beta})$= var(PY)= $\mathrm{PP}^T var(Y)$

$$PP^T = ((Z^T Z)^{-1} Z^T)((Z^T Z)^{-1} Z^T)^T$$

$$= (Z^T Z)^{-1} Z^T Z (Z^T Z)^{-1}$$
$$= (Z^T Z)^{-1}$$

Therefore, $var(\hat{\beta})$=

$$(Z^T Z)^{-1} var(Y)$$
$$= (Z^T Z)^{-1} var(Z\beta + \epsilon)$$
$$= (Z^T Z)^{-1} \sigma^2$$

where $(\mathrm{Z}^T Z)^{-1} = ((XL)^T XL) - 1 = (L^T X^T XL)^{-1}$

Here L are the eigen vectors of X.

■

ii) Derive the Mean Square Error of the Ridge Regression Estimator $\alpha_{\hat{ridge}}$.

Sol: var($\beta_{\hat{ridge}}$)=$\sigma^2 (X^T X + \lambda I)^{-1} X^T X (X^T X + \lambda I)^{-1} = \sigma^2 [A^{-1} - \lambda A^{-2}]$

where A=$(\mathrm{X}^T X + \lambda I)$

Bias= $(\mathrm{X}^T X + \lambda I)^{-1} X^T X \beta - \beta = (X^T X + \lambda I)^{-1} (X^T X + \lambda I - \lambda I)\beta - \beta$

=[I-$\lambda (X^T X + \lambda I)^{-1}]\beta - \beta$

=$\lambda (X^T X + \lambda I)^{-1} \beta$

=$\lambda A^{-1} \beta$

Therefore MSE = $\sigma^2 (A^{-1} - \lambda A^{-2}) + \lambda^2 A^{-1} \beta \beta^T A^{-1}$

■

iii Find the optimal value of the λ the regularization parameter by minimizing the MSE with respect to λ

Sol: To find optimum value of mse,

$$\frac{dmse}{d\lambda} = 0$$

$$\sigma^2(-A^{-2} + 2\lambda A^{-3}) + \lambda^2(-A^{-2)} \beta\beta^T A^{-1} + \lambda^2 A^{-1} \beta\beta^T (-A^{-2}) + 2\lambda A^{-1} \beta\beta^T A^{-1} = 0$$

$$\sigma^2 A^{-1}(2\lambda A^{-1} - I)A^{-1} - \lambda^2 A^{-1} A^{-1} \beta\beta^T A^{-1} - A^{-1} \lambda^2 \beta\beta^T A^{-1} A^{-1} + 2\lambda A^{-1} A^{-1} \beta\beta^T A^{-1} = 0$$

$$\sigma^2(2\lambda A^{-1} - I) - \lambda^2 A^{-1} \beta\beta^T - \lambda^2 \beta\beta^T A^{-1} + 2\lambda A^{-1} \beta\beta^T = 0$$

The above equation suggests that we cannot determine the value of λ analytically. After getting the above expression, we need the numerical optimization procedures to solve it further. In our computational assignments, we shall rather use grid search to solve the problem. Details are mentioned in computational section.

■

iv Using the two variable regression model, compose a convex combination of OLS and PCR of the regression parameter β_1 with α weight, 0 ¡= α ¡= 1 given by $\beta_{1cc} = \alpha\beta_{1OLS} + (1-\alpha)\beta_{1PCR}$

Sol: $\beta_{1cc} = \alpha\beta_{1OLS} + (1-\alpha)\beta_{1PCR}$ ■

v Derive the Bias, Variance, and MSE of β_{1cc}.

Sol: L is the Loading matrix from the Principal Component Regression.

Z = XL is the transformation used to mitigate the multicollinearity in PCR.

$$\begin{aligned}
Bias(\hat{\beta_{cc}}) &= E(\hat{\beta_{cc}}) - \beta \\
&= \alpha E(\hat{\beta_{OLS}}) + (1-\alpha)E(\hat{\beta_{PCR}}) - \beta \\
&= \alpha\beta + (1-\alpha)L^T\beta - \beta \\
&= [\alpha I + (1-\alpha)L^T - I]\beta \\
&= [I - \alpha I][L^T - I]\beta \qquad (0.9) \\
Variance(\hat{\beta_{cc}}) &= \alpha^2 Var(\hat{\beta_{OLS}}) + (1-\alpha)^2 Var(\hat{\beta_{PCR}}) + 2cov(\alpha\hat{\beta_{OLS}}, (1-\alpha)\hat{\beta_{PCR}})) \\
&= \alpha^2(X^TX)^{-1}\sigma^2 + (1-\alpha)^2(Z^TZ)^{-1}\sigma^2 + 2\alpha(1-\alpha)Var(\hat{\beta_{OLS}})L \\
&= \alpha^2(X^TX)^{-1}\sigma^2 + (1-\alpha)^2(L^TX^TXL)^{-1}\sigma^2 + 2\alpha(1-\alpha)(X^TX)^{-1}\sigma^2 L \qquad (0.10) \\
MSE(\hat{\beta_{1cc}}) &= Variance(\hat{\beta_{cc}}) + Bias(\hat{\beta_{cc}})^2 \\
&= Variance(\hat{\beta_{cc}}) + [I - \alpha I][L^T - I]\beta\beta^T[L - I][I - \alpha I] \qquad (0.11)
\end{aligned}$$

■

vi Derive the optimal weight of α by minimizing the MSE.
Sol:

$$\begin{aligned}
\frac{dVar(\hat{\beta_{cc}})}{d\alpha} &= \alpha[(X^TX)^{-1}\sigma^2 + (L^TX^TXL)^{-1}\sigma^2 - 2(X^TX)^{-1}\sigma^2 L] - (L^TX^TXL)^{-1}\sigma^2 + (X^TX)^{-1}\sigma^2 L \\
\frac{dBias(\hat{\beta_{cc}})^2}{d\alpha} &= \alpha[\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T] - [\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T]
\end{aligned}$$

Setting, $\frac{d(MSE(\hat{\beta_{cc}}))}{d\alpha} = 0$, we get α as follows

$$\alpha = \frac{[\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T + (L^TX^TXL)^{-1}\sigma^2 - (X^TX)^{-1}\sigma^2 L]}{[(X^TX)^{-1}\sigma^2 + (L^TX^TXL)^{-1}\sigma^2 - 2(X^TX)^{-1}\sigma^2 L] + [\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T]} \qquad (0.12)$$

■

vii) Set the weight $\alpha = r^2$ in β_{cc} and derive the condition under which the MSE of β_{cc} is less than or equal to β_{cc}
Sol:

$$Bias(\hat{\beta_{cc}})^2 = \alpha^2[\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T] + 2\alpha[\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T] + [\beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T]$$

$Var(\hat{\beta_{cc}}) = \alpha^2[(X^TX)^{-1}\sigma^2 + (L^TX^TXL)^{-1}\sigma^2 - 2(X^TX)^{-1}\sigma^2L] + 2\alpha[(X^TX)^{-1}\sigma^2L - (L^TX^TXL)^{-1}\sigma^2] + (L^TX^TXL)^{-1}\sigma^2$

Take, $A = (X^TX)^{-1}\sigma^2$; $B = (L^TX^TXL)^{-1}\sigma^2$; $C = (X^TX)^{-1}\sigma^2L$ and $K = \beta\beta^T + L^T\beta\beta^TL - \beta\beta^TL - L^T\beta\beta^T$

Setting the condition that $MSE(\hat{\beta_{cc}})$ less than $MSE(\hat{\beta_{OLS}})$ we get the following condition,

Substitute $\alpha = r^2$

$$r^4[A + B - 2C + K] + 2r^2[C - B + K] + B + K \leq A$$
$$r^4[A + B - 2C + K] + 2r^2[C - B + K] + B + K - A \leq 0 \tag{0.13}$$

Solving the above quadratic equation gives range of values of r^2 for α where the MSE of β_{cc} is less than or equal to β_{cc} ■

Simulation questions

The results for the next 2 questions on simulations are tabulated in a separate document attached below. It has results and discussion for the results. The code for the same is attached as well.

Extra Credit) Using the convex combination of Ridge and OLS, derive the optimal estimator of the weight α by minimizing the MSE. And perform simulations as in Q3. *Sol:*

$$\hat{\beta_{cc}} = \alpha(X^TX)^{-1}X^TY + (1-\alpha)(X^TX + \lambda I)^{-1}X^TY$$

$$\begin{aligned}
Bias(\hat{\beta_{cc}}) &= E(\hat{\beta_{cc}}) - \beta \\
&= \alpha E(\hat{\beta_{OLS}}) + (1-\alpha)E(\hat{\beta_{Ridge}}) - \beta \\
&= \alpha\beta + (1-\alpha)(X^TX + \lambda I)^{-1}X^TX\beta - \beta \\
&= [\alpha + (1-\alpha)(X^TX + \lambda I)^{-1}X^TX - 1]\beta \\
&= [1-\alpha][(X^TX + \lambda I)^{-1}X^TX - I]\beta \\
Var(\hat{\beta_{cc}}) &= \alpha^2 Var(\hat{\beta_{OLS}}) + (1-\alpha)^2 Var(\hat{\beta_{PCR}}) + 2cov(\alpha\hat{\beta_{OLS}}, (1-\alpha)\hat{\beta_{Ridge}}) \\
&= \alpha^2(X^TX)^{-1}\sigma^2 + (1-\alpha)^2(X^TX + \lambda I)^{-1}X^TX(X^TX + \lambda I)^{-1}\sigma^2 + 2\alpha(1-\alpha)(X^TX)^{-1}X^T(X^TX + \lambda I)^{-1}X \\
MSE(\hat{\beta_{1cc}}) &= Variance(\hat{\beta_{cc}}) + Bias(\hat{\beta_{cc}})^2 \\
&= Variance(\hat{\beta_{cc}}) + [1-\alpha^2][(X^TX + \lambda I)^{-1} - I]\beta\beta^T[(X^TX + \lambda I)^{-1} - I]
\end{aligned}$$

Now take

$$A = (X^TX + \lambda I)^{-1}$$

$$\frac{dMSE(\hat{\beta_{cc}})}{d\alpha} = \alpha[(X^TX)^{-1}\sigma^2 + (AX^TXA)\sigma^2 - 2(X^TX)^{-1}(X^TAX^T)\sigma^2 + (AX^TX - I)\beta\beta^T(AX^TX - I)]$$
$$+[(X^TX)^{-1}(X^TAX^T)\sigma^2 - (AX^TXA)\sigma^2 - (AX^TX - I)\beta\beta^T(AX^TX - I)]$$

Setting, $\frac{d(MSE(\hat{\beta_{cc}}))}{d\alpha} = 0$, we get α as follows:

$$\alpha = \frac{[(AX^TX - I)\beta\beta^T(AX^TX - I) + (AX^TXA)\sigma^2 - (X^TX)^{-1}(X^TAX^T)\sigma^2]}{[(X^TX)^{-1}\sigma^2 + (AX^TXA)\sigma^2 - 2(X^TX)^{-1}(X^TAX^T)\sigma^2 + (AX^TX - I)\beta\beta^T(AX^TX - I)]}$$

■

Results and Discussions:

In the below section we present the summary of computations. We show that in case of multicollinearity OLS estimates have very high variance. We further show that this problem can be resolved using PCR and Ridge estimators. But these estimators are biased. We have also tabulated results of convex combination of these estimators to understand bias variance trade-off. So OLS estimators increase variance and PCR, ridge estimators push bias.

As optimal value for penalty term cannot be calculated for ridge estimator analytically (We have shown this in proof section. We need numerical computation procedures for the same), we conduct a grid search to decide a good choice of penalty parameter.

For convex combination we choose different values of α. It is also shown that for convex combination terms, we cannot have a general rule of α. The results indicate for higher and lower correlation terms, the value changes.

We have also solved the extra credit problem based on the optimal α that Professor suggested us to use (the expression in his notes). However we have left one subsection. Where we needed to use optimal α for convex combination of PCR and OLS. The reason is that when we derived the term for optimal λ (for the convex combination of PCR and ridge), the final matrix did not turn out to be a diagonal matrix as was expected. This error is probably due to error propagation that takes during matrix inversion procedures, cholesky's decomposition etc. (Although we did properly derive the expression). We were not sure how to pick right terms from that matrix.

$$\lambda = \frac{(a^2 - r^2)(1 - r^2)}{(a-1)(a+r^2)} \frac{D^2}{\sigma^2}$$

Where D = $\frac{a(1-a)}{a^2-r^2}\beta_1 + \frac{r(1-a)}{a^2-r^2}\beta_2$

We conducted a total of 48 different computational experiments.

- 12 experiments (of different co-relations) for each of OLS, PCR and Ridge estimators.
- 16 experiments (of different co-relations) for convex combination of OLS and PCR with 4 different α.
- 16 experiments (of different co-relations) for convex combination of OLS and Ridge with 4 different α.

We did 2500 simulations in each experiment and each simulation had a dataset with 30 rows and 2 columns.

Case 1: Simulation for OLS estimates for different correlation values.

The computational results match our expected values. In cases when the correlation values were high, the Bias values were close to zero but the variance value did shoot up. This can be attributed to the presence of collinearity between two features. As correlation value decreased the variance decreased as well. Therefore, we can infer that OLS estimates are best linear unbiased estimators assuming the features are independent.

	Beta 1 OLS			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	4.016	3.996	3.995	3.996
Bias	0.016	-0.004	-0.005	-0.004
Variance	1.924	0.394	0.059	0.044
MSE	1.924	0.394	0.059	0.044

Case 2: Simulation for PCR estimates for different correlation values.

In PCR based regression, the collinearity problem was mitigated and thus the variance values did go down. But since the estimator was biased, the MSE value increased significantly.

	Beta 1 PCR			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	7.781	7.776	7.769	7.642
Bias	3.781	3.776	3.769	3.642
Variance	0.018	0.019	0.059	1.34
MSE	14.314	14.277	14.264	14.604

Case 3: Simulation for Ridge estimates for different correlation values

We were not able to find the optimal penalty value for ridge as that requires numerical computation procedures. We have instead conducted a grid search over a symmetrical scale of penalty terms. The results obtained from different penalty terms (0.01, 0.1, 1, 10, 100) are tabulated as below. We can see that optimal penalty parameter is somewhere close in the order of 10.

All penalty parameters have controlled the variance values in a descent way but some are much more biased than others. It is using this tabular presentation of the results, we can choose an optimal penalty.

	Beta 1 Ridge, penalty = 0.01			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	7.776	7.781	7.777	7.629
Bias	3.776	3.781	3.777	3.629
Variance	0.019	0.02	0.022	1.501
MSE	14.277	14.316	14.288	14.671

	Beta 1 Ridge, penalty = 0.1			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	7.765	7.771	7.753	7.663
Bias	3.765	3.771	3.753	3.663
Variance	0.018	0.019	0.061	0.977
MSE	14.193	14.239	14.146	14.395

	Beta 1 Ridge, penalty = 1			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	7.635	7.64	7.597	7.442
Bias	3.635	3.64	3.597	3.442
Variance	0.019	0.019	0.024	1.346
MSE	13.232	13.269	12.962	13.193

	Beta 1 Ridge, penalty = 10			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	6.608	6.582	6.346	6.082
Bias	2.608	2.582	2.346	2.082
Variance	0.085	0.086	0.144	1.019
MSE	6.887	6.753	5.648	5.354

	Beta 1 Ridge, penalty = 100			
	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	2.863	2.828	2.487	2.23
Bias	-1.137	-1.172	-1.513	-1.777
Variance	0.215	0.214	0.186	0.285
MSE	1.508	1.588	2.475	3.443

Case 4: Simulation for convex combination of OLS and PCR.

When we take a convex combination of OLS and PCR estimates, what we are basically doing is a bias variance tradeoff. We choose different values of the parameter λ and see how variance and bias vary. In some choices of λ we see that there is improvement in both the metrics and that serves as a natural choice for selection. It is also evident that the choice of λ also depends on the correlation values between the parameters. Some λ choices give better result for higher correlations.

It can further be seen that higher r means 1 - r^2 goes to zero and so we give higher preference to OLS estimates. This reflects in results as well. At higher correlation values, we have more variance due to higher weight being assigned to OLS estimates and this changes as correlation gradually goes down.

Case when α is equal to r^2

	Beta 1 CC			
Alpha = r^2	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	3.974	4.002	4.001	4
Bias	-0.026	0.002	0.001	0
Variance	1.889	0.39	0.06	0.044
MSE	1.89	0.39	0.06	0.044

Case when α is equal to |r|

	Beta 1 CC			
Alpha = |r|	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	3.993	4.005	3.998	3.997
Bias	-0.007	0.005	-0.002	-0.003
Variance	1.833	0.391	0.058	0.043
MSE	1.833	0.391	0.058	0.043

Case when α is equal to 1/(1 + r^2)

	Beta 1 CC			
Alpha = 1/(1+ r^2)	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	5.864	5.878	5.888	5.814
Bias	1.864	1.878	1.888	1.814
Variance	0.501	0.107	0.029	0.388
MSE	3.975	3.634	3.594	3.679

Case 4: Simulation for convex combination of OLS and Ridge.

We do the same procedure as above but compromise different term in the convex combination.

Case when α is equal to r^2

	Beta 1 CC			
Alpha = r^2	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	3.972	4.004	4.004	3.999
Bias	-0.028	0.004	0.004	-0.001
Variance	1.846	0.354	0.058	0.046
MSE	1.847	0.354	0.058	0.044

Case when α is equal to |r|

	Beta 1 CC			
Alpha = r	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	4.034	4.01	4.004	4.005
Bias	0.034	0.01	0.004	0.005
Variance	1.868	0.377	0.056	0.045
MSE	1.869	0.377	0.056	0.045

Case when α is equal to 1/(1 + r^2)

	Beta 1 CC			
Alpha=1/(1+ r^2)	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	5.294	5.284	5.178	5.028
Bias	1.294	1.284	1.178	1.028
Variance	0.477	0.119	0.052	0.32
MSE	2.151	1.768	1.44	1.377

Case 5: Simulation for convex combination of OLS and Ridge (with optimal λ as given by Professor).

Since we did not find optimal ridge parameter as it needed numerical computation. Here we choose the value as 1. Then we use the expression provided by Professor for optimal λ value.

Results with optimal values of λ

	Beta 1 CC optimal			
λ optimal	correlation = 0.99	correlation = 0.95	correlation = 0.60	correlation = 0.40
Mean	7.641	7.632	7.599	7.455
Bias	3.641	3.632	3.599	3.455

Variance	0.02	0.021	0.058	1.198
MSE	13.277	13.212	13.011	13.135

These are the calculations for OLS

Let B1 = 4, B2 = 7

In [1]:

```
%matplotlib inline

import pandas as pd
import numpy as np
from scipy.linalg import cholesky
from numpy.linalg import inv
```

Using cholesky decomposition to generate co-related matrix

In [15]:

```
simulation_count = 2500
correlation_value = 0.40
n_rows = 30

b1= 4
b2 = 7

mean = 0
deviation = 1
```

In [16]:

```
def matrix_generator(n, size):
    correlation_matrix = np.array([[1.00,n], [n,1.00]])
    A = pd.DataFrame(np.random.randn(size,2))
    x = pd.DataFrame(np.matmul(np.linalg.cholesky(correlation_matrix), np.tran
spose(A))).transpose()
    return x
```

In [17]:

```
b1_hat_mean = []
b2_hat_mean = []

for i in range(simulation_count):

    X = matrix_generator(correlation_value,n_rows)
    error = np.random.randn(n_rows)

    Y = b1*X[0] + b2*X[1] + error

    beta_vector = np.matmul(inv(np.matmul(X.transpose(),X)),np.matmul(X.transp
ose(),Y))

    b1_hat_mean.append(beta_vector[0])
    b2_hat_mean.append(beta_vector[1])
```

In [18]:

```
# Calculation for Expected value of B1

mean = np.mean(b1_hat_mean).round(3)
print("The expected value of Beta 1 hat for OLS is:\t\t", mean, "\n")

#Calculation of bias

bias = (mean - b1).round(3)
print("The Bias of Beta_1 for OLS is:\t\t\t",(mean - b1).round(3), "\n")

#Calculating the variance

var = np.var(b1_hat_mean).round(3)
print("The variance of Beta 1 for OLS is:\t\t",var, "\n")

#Calculate the MSE
MSE = var + bias**2
print("The MSE of Beta 1 for OLS is:\t\t\t",MSE.round(3))
```

```
The expected value of Beta 1 hat for OLS is:            3.996

The Bias of Beta_1 for OLS is:                          -0.004

The variance of Beta 1 for OLS is:                      0.044

The MSE of Beta 1 for OLS is:                           0.044
```

These are the calculations for PCR

Let B1 = 4, B2 = 7

In [17]:

```
%matplotlib inline

import pandas as pd
from numpy import linalg as LA
import numpy as np
from scipy.linalg import cholesky
from numpy.linalg import inv
```

Using cholesky decomposition to generate co-related matrix

In [34]:

```
simulation_count = 2500
correlation_value = 0.40
n_rows = 30

b1= 4
b2 = 7


mean = 0
deviation = 1
```

In [35]:

```
def matrix_generator(n, size):
    correlation_matrix = np.array([[1.00,n], [n,1.00]])
    A = pd.DataFrame(np.random.randn(size,2))
    x = pd.DataFrame(np.matmul(np.linalg.cholesky(correlation_matrix), np.tran
spose(A))).transpose()
    return x
```

In [70]:

```
x = matrix_generator(0.99,3000)
```

In [71]:

```
x.corr()
```

Out[71]:

	0	1
0	1.000000	0.990144
1	0.990144	1.000000

In [72]:

```
inv(np.matmul(x.transpose(),x))
```

Out[72]:

```
array([[ 0.01656735, -0.01639229],
       [-0.01639229,  0.01654368]])
```

In [36]:

```
b1_hat_mean = []
b2_hat_mean = []

for i in range(simulation_count):

    X = matrix_generator(correlation_value,n_rows)
    error = np.random.randn(n_rows)

    Y = b1*X[0] + b2*X[1] + error

    eigen_values, eigen_vectors = LA.eig(X.corr())
    Z = np.matmul(X,eigen_vectors)

    X =Z

    beta_vector = np.matmul(inv(np.matmul(X.transpose(),X)),np.matmul(X.transp
ose(),Y))

    b1_hat_mean.append(beta_vector[0])
    b2_hat_mean.append(beta_vector[1])
```

In [37]:

```
# Calculation for Expected value of B1

mean = np.mean(b1_hat_mean).round(3)
print("The expected value of Beta 1 hat for OLS is:\t\t", mean, "\n")

#Calculation of bias

bias = (mean - b1).round(3)
print("The Bias of Beta_1 for PCR is:\t\t\t",(mean - b1).round(3), "\n")

#Calculating the variance

var = np.var(b1_hat_mean).round(3)
print("The variance of Beta 1 for PCR is:\t\t",var, "\n")

#Calculate the MSE
MSE = var + bias**2
print("The MSE of Beta 1 for PCR is:\t\t\t",MSE.round(3))
```

```
The expected value of Beta 1 hat for OLS is:            7.642

The Bias of Beta_1 for OLS is:                          3.642

The variance of Beta 1 for OLS is:                      1.351

The MSE of Beta 1 for OLS is:                           14.615
```

These are the calculations for convex combination of PCR and OLS

Let B1 = 4, B2 = 7

In [49]:

```
%matplotlib inline

import pandas as pd
from numpy import linalg as LA
import numpy as np
from scipy.linalg import cholesky
from numpy.linalg import inv
```

Using cholesky decomposition to generate co-related matrix

In [66]:

```
simulation_count = 2500
correlation_value = 0.40

n_rows = 30

b1= 4
b2 = 7

mean = 0
deviation = 1
```

In [67]:

```
def matrix_generator(n, size):
    correlation_matrix = np.array([[1.00,n], [n,1.00]])
    A = pd.DataFrame(np.random.randn(size,2))
    x = pd.DataFrame(np.matmul(np.linalg.cholesky(correlation_matrix), np.tran
spose(A))).transpose()
    return x
```

In [68]:

```
b1_hat_mean = []
b2_hat_mean = []

for i in range(simulation_count):

    X = matrix_generator(correlation_value,n_rows)
    alpha = X.corr().iloc[1,1]

    error = np.random.randn(n_rows)

    Y = b1*X[0] + b2*X[1] + error

    beta_vector_OLS = np.matmul(inv(np.matmul(X.transpose(),X)),np.matmul(X.tr
anspose(),Y))

    eigen_values, eigen_vectors = LA.eig(X.corr())
    Z = np.matmul(X,eigen_vectors)

    X = Z

    beta_vector_PCR = np.matmul(inv(np.matmul(X.transpose(),X)),np.matmul(X.tr
anspose(),Y))

    beta_CC = (1/(1+alpha**2))*(beta_vector_OLS[0]) + (1 - (1/(1+alpha**2)))*(
beta_vector_PCR[0])

    b1_hat_mean.append(beta_CC)
    #b2_hat_mean.append(beta_CC[1])
```

In [69]:

```
# Calculation for Expected value of B1

mean = np.mean(b1_hat_mean).round(3)
print("The expected value of Beta 1 hat for OLS is:\t\t", mean, "\n")

#Calculation of bias

bias = (mean - b1).round(3)
print("The Bias of Beta_1 for OLS is:\t\t\t\t",(mean - b1).round(3), "\n")

#Calculating the variance

var = np.var(b1_hat_mean).round(3)
print("The variance of Beta 1 for OLS is:\t\t\t",var, "\n")

#Calculate the MSE
MSE = var + bias**2
print("The MSE of Beta 1 for OLS is:\t\t\t\t",MSE.round(3))
```

```
The expected value of Beta 1 hat for OLS is:            5.814

The Bias of Beta_1 for OLS is:                          1.814

The variance of Beta 1 for OLS is:                      0.388

The MSE of Beta 1 for OLS is:                           3.679
```

These are the calculations for Ridge

Let B1 = 4, B2 = 7

In [10]:

```
%matplotlib inline

import pandas as pd
from numpy import linalg as LA
import numpy as np
from scipy.linalg import cholesky
from numpy.linalg import inv
```

Using cholesky decomposition to generate co-related matrix

In [88]:

```
simulation_count = 2500
correlation_value = 0.4
n_rows = 30

lamda = 0.1

b1= 4
b2 = 7

mean = 0
deviation = 1
```

In [89]:

```
def matrix_generator(n, size):
    correlation_matrix = np.array([[1.00,n], [n,1.00]])
    A = pd.DataFrame(np.random.randn(size,2))
    x = pd.DataFrame(np.matmul(np.linalg.cholesky(correlation_matrix), np.tran
spose(A))).transpose()
    return x
```

In [90]:

```
b1_hat_mean = []
b2_hat_mean = []

for i in range(simulation_count):

    X = matrix_generator(correlation_value,n_rows)
    error = np.random.randn(n_rows)

    Y = b1*X[0] + b2*X[1] + error

    eigen_values, eigen_vectors = LA.eig(X.corr())
    Z = np.matmul(X,eigen_vectors)

    X =Z

    beta_vector = np.matmul(inv(np.matmul(X.transpose(),X) + np.diag([lamda,la
mda])),np.matmul(X.transpose(),Y))

    b1_hat_mean.append(beta_vector[0])
    b2_hat_mean.append(beta_vector[1])
```

In [91]:

```
# Calculation for Expected value of B1

mean = np.mean(b1_hat_mean).round(3)
print("The expected value of Beta 1 hat for Ridge is:\t\t", mean, "\n")

#Calculation of bias

bias = (mean - b1).round(3)
print("The Bias of Beta_1 for Ridge is:\t\t\t",(mean - b1).round(3), "\n")

#Calculating the variance

var = np.var(b1_hat_mean).round(3)
print("The variance of Beta 1 for Ridge is:\t\t\t",var, "\n")

#Calculate the MSE
MSE = var + bias**2
print("The MSE of Beta 1 for Ridge is:\t\t\t\t",MSE.round(3))
```

```
The expected value of Beta 1 hat for Ridge is:          7.663

The Bias of Beta_1 for Ridge is:                        3.663

The variance of Beta 1 for Ridge is:                    0.977

The MSE of Beta 1 for Ridge is:                         14.395
```

These are the calculations for convex combination of Ridge and OLS

Let B1 = 4, B2 = 7

In [64]:

```
%matplotlib inline

import pandas as pd
from numpy import linalg as LA
import numpy as np
from scipy.linalg import cholesky
from numpy.linalg import inv
```

Using cholesky decomposition to generate co-related matrix

In [94]:

```
simulation_count = 2500
correlation_value = 0.40

lamda = 1

n_rows = 30

b1= 4
b2 = 7

mean = 0
deviation = 1
```

In [95]:

```
def matrix_generator(n, size):
    correlation_matrix = np.array([[1.00,n], [n,1.00]])
    A = pd.DataFrame(np.random.randn(size,2))
    x = pd.DataFrame(np.matmul(np.linalg.cholesky(correlation_matrix), np.tran
spose(A))).transpose()
    return x
```

In [96]:

```
b1_hat_mean = []
b2_hat_mean = []

for i in range(simulation_count):

    X = matrix_generator(correlation_value,n_rows)

    r = X.corr().iloc[1,1]

    alpha = (((4 - r**2)*(1 - r**2))/(2 + r**2))*((8 - 7*r)/(4 - r**2))**2



    error = np.random.randn(n_rows)

    Y = b1*X[0] + b2*X[1] + error

    beta_vector_OLS = np.matmul(inv(np.matmul(X.transpose(),X)),np.matmul(X.tr
anspose(),Y))

    eigen_values, eigen_vectors = LA.eig(X.corr())
    Z = np.matmul(X,eigen_vectors)

    X = Z

    beta_vector_Ridge = np.matmul(inv(np.matmul(X.transpose(),X) + np.diag([la
mda,lamda])),np.matmul(X.transpose(),Y))

    beta_CC = alpha*(beta_vector_OLS[0]) + (1 - alpha)*(beta_vector_Ridge[0])

    b1_hat_mean.append(beta_CC)
    #b2_hat_mean.append(beta_CC[1])
```

In [97]:

```
# Calculation for Expected value of B1

mean = np.mean(b1_hat_mean).round(3)
print("The expected value of Beta 1 hat for Ridge_CC is:\t\t", mean, "\n")

#Calculation of bias

bias = (mean - b1).round(3)
print("The Bias of Beta_1 for Ridge_CC is:\t\t\t\t",(mean - b1).round(3), "\n"
)

#Calculating the variance

var = np.var(b1_hat_mean).round(3)
print("The variance of Beta 1 for Ridge_CC is:\t\t\t",var, "\n")

#Calculate the MSE
MSE = var + bias**2
print("The MSE of Beta 1 for Ridge_CC is:\t\t\t\t",MSE.round(3))
```

```
The expected value of Beta 1 hat for Ridge_CC is:                7
.455

The Bias of Beta_1 for Ridge_CC is:                              3
.455

The variance of Beta 1 for Ridge_CC is:                      1.198

The MSE of Beta 1 for Ridge_CC is:                               1
3.135
```