Statistical Learning - Report 3: Multiple regression: model selection criteria, ridge regression and LASSO

Julia Kiczka

June 22, 2025

1) Prove that the trace of the symmetric real matrix is equal to the sum of its eigenvalues.

$$\mathrm{Tr}(A) = \mathrm{Tr}(P\Lambda P^T) = \mathrm{Tr}(P^T P\Lambda) = \mathrm{Tr}(\Lambda) = \sum_{i=1}^{n} \lambda_i$$

2) Properties of $X^T X$

Let X be a real matrix of size $n \times p$, where n is the number of rows and p is the number of columns.

(a) Proof that $X^T X$ is semipositive definite and its eigenvalues are non-negative

We need to show that $X^T X$ is positive semidefinite, i.e., for any vector $v \in R^p$,

$$v^T (X^T X) v \geq 0.$$

First, observe that:

$$v^T (X^T X) v = (Xv)^T (Xv) = \|Xv\|^2.$$

Since $\|Xv\|^2 \geq 0$ for all v, it follows that $v^T (X^T X) v \geq 0$, showing that $X^T X$ is positive semidefinite. Next, let λ be an eigenvalue of $X^T X$ with eigenvector v, i.e.,

$$X^T X v = \lambda v.$$

Multiplying both sides by v^T, we get:

$$\|Xv\|^2 = v^T X^T X v = \lambda v^T v.$$

Since $v^T v = \|v\|^2 \geq 0$, it follows that:

$$\lambda \|v\|^2 \geq 0.$$

Thus, $\lambda \geq 0$, meaning that the eigenvalues of $X^T X$ are non-negative.

(b) Proof that when $p > n$, at least one eigenvalue of $X^T X$ is zero (i.e., $X^T X$ is singular)

When $p > n$, the matrix X has more columns than rows. The rank of $X^T X$ is at most the rank of X, and the rank of X is at most n, since X has only n rows. Therefore, the rank of $X^T X$ is at most n. However, $X^T X$ is a $p \times p$ matrix. Since its rank is at most n and $p > n$, the rank of $X^T X$ is strictly less than p. This implies that $X^T X$ is rank-deficient, and hence, it must be singular. Therefore, at least one eigenvalue of $X^T X$ is zero.

3) Your data contains 10 variables. You fit 10 regression models including the first variable, the first two variables, etc. The residual sums of squares for these 10 consecutive models are equal to $(1731, 730, 49, 38.9, 32, 29, 28.5, 27.8, 27.6, 26.6)$. The sample size is equal to 100. Which of these 10 models will be selected by AIC ? And which model will be selected by BIC or RIC? Assume that the standard deviation of the error term is known; $\sigma = 1$.

We are fitting linear regression models using up to 10 predictor variables. For each model, the residual sum of squares (RSS) is recorded after including 1, 2, ..., 10 predictors. The sample size is $n = 100$, and the standard deviation of the error term is known to be $\sigma = 1$. The residual sums of squares for each model are:

$$\text{RSS}_j = (1731, 730, 49, 38.9, 32, 29, 28.5, 27.8, 27.6, 26.6)$$

Each model includes an intercept, so the total number of parameters is:

$$k = j + 1 \quad (\text{where } j = \# \text{ of predictors})$$

The model selection criteria are:

$$\begin{aligned} \text{AIC} &= \text{RSS} + 2k \\ \text{BIC} &= \text{RSS} + k \log(n) \\ \text{RIC} &= \text{RSS} + 2k \log(k) \end{aligned}$$

We now calculate each criterion for all models:

$$\log(100) \approx 4.605, \quad \log(k) \text{ is computed individually for each } k.$$

j	**k = j+1**	**RSS**	**AIC = RSS + 2k**	**BIC = RSS + klog(100)**	**RIC = RSS + 2klog(k)**
1	2	1731	1731 + 4 = 1735.00	1731 + 9.210 = 1740.21	1731 + 2(2)(0.693) = 1733.77
2	3	730	730 + 6 = 736.00	730 + 13.815 = 743.82	730 + 2(3)(1.099) = 743.59
3	4	49	49 + 8 = 57.00	49 + 18.420 = 67.42	49 + 2(4)(1.386) = 60.09
4	5	38.9	38.9 + 10 = 48.90	38.9 + 23.025 = 61.93	38.9 + 2(5)(1.609) = 55.99
5	6	32	32 + 12 = 44.00	32 + 27.630 = **59.63**	32 + 2(6)(1.792) = **53.00**
6	7	29	29 + 14 = **43.00**	29 + 32.235 = 61.24	29 + 2(7)(1.946) = 56.26
7	8	28.5	28.5 + 16 = 44.50	28.5 + 36.840 = 65.34	28.5 + 2(8)(2.079) = 61.76
8	9	27.8	27.8 + 18 = 45.80	27.8 + 41.445 = 69.24	27.8 + 2(9)(2.197) = 67.36
9	10	27.6	27.6 + 20 = 47.60	27.6 + 46.050 = 73.65	27.6 + 2(10)(2.302) = 73.64
10	11	26.6	26.6 + 22 = 48.60	26.6 + 50.655 = 77.26	26.6 + 2(11)(2.398) = 78.36

Table 1: Model selection criteria using AIC, BIC, and RIC with $k = \#\text{predictors} + 1$

Conclusion:

- **AIC** selects **Model 6**, with the lowest AIC = 43.00.
- **BIC** selects **Model 5**, with the lowest BIC = 59.63.
- **RIC** selects **Model 5** as well, with the lowest RIC = 53.00.

4) Expected Number of False Discoveries under AIC, BIC, and RIC

Assume an orthogonal design ($X'X = I$) with $n = p = 10000$ and all variables being null (i.e., $p_0 = p$). We calculate the expected number of false discoveries under AIC, BIC, and RIC.

AIC

In this setup, the probability of a type I error is:

$$P(X_i \text{ is selected} \mid \beta_i = 0) = 2(1 - \Phi(\sqrt{2})) \approx 0.1573$$

Then, the expected number of false discoveries is:

$$10000 \times 0.16 \approx 1573$$

BIC

$$P(X_i \text{ is selected} \mid \beta_i = 0) = 2(1 - \Phi(\sqrt{\log 10000})) \approx 0.0024$$

$$\text{Expected number of false discoveries} \approx 10000 \times 0.0024 = 24$$

BIC is more restrictive than AIC and tends to yield fewer false discoveries on average.

RIC

$$P(X_i \text{ is selected} \mid \beta_i = 0) = 2\left(1 - \Phi\left(\sqrt{2\log p}\right)\right) = 2\left(1 - \Phi\left(\sqrt{2\log 10000}\right)\right) \approx 0.000018$$

$$\text{Expected number of false discoveries} \approx 10000 \times 0.000018 \approx 0$$

RIC is designed to control false discoveries even in large models and ensures the expected number of false positives remains below 1.

5) When to Use AIC, BIC, and RIC

AIC (Akaike Information Criterion)

AIC is best suited for predictive modeling, where the primary goal is to minimize prediction error and not necessarily to find the true model. It tends to favor more complex models due to its relatively lighter penalty for additional parameters. Use AIC when:

- The objective is prediction, not inference.
- The sample size is relatively small or moderate.
- Overfitting is not a major concern and you prefer capturing more potential signals.

BIC (Bayesian Information Criterion)

BIC is more conservative than AIC and includes a stronger penalty for model complexity. It is more appropriate when the goal is to identify the true model (if it exists) and avoid overfitting. Use BIC when:

- You aim for model selection or inference.
- The sample size is large.
- You prefer simpler models that generalize better.

RIC (Risk Inflation Criterion)

RIC is specifically designed for high-dimensional settings (i.e., when the number of variables p is large relative to n). It is very conservative and aims to control the number of false discoveries. Use RIC when:

- You're in a large-scale variable selection problem (e.g., genomics, signal processing).
- Avoiding false discoveries is critical.
- The cost of including irrelevant variables is high.

6) Bias, Variance, and MSE of Ridge Regression under Orthogonal Design

Assume the linear model $y = X\beta + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ and $X^\top X = I$.

Ridge Estimator

The ridge regression estimator is:

$$\hat{\beta}^{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top y = \frac{1}{1+\lambda} X^\top y$$

Bias

$$\text{Bias}(\hat{\beta}^{\text{ridge}}) = E[\hat{\beta}^{\text{ridge}}] - \beta = \left(\frac{1}{1+\lambda} - 1\right)\beta = -\frac{\lambda}{1+\lambda}\beta$$

Variance

$$\text{Var}(\hat{\beta}^{\text{ridge}}) = \sigma^2 (X^\top X + \lambda I)^{-1} X^\top X (X^\top X + \lambda I)^{-1} = \sigma^2 (I + \lambda I)^{-1} I (I + \lambda I)^{-1} = \frac{\sigma^2}{(1+\lambda)^2} I$$

Mean Squared Error (MSE)

$$\text{MSE}(\hat{\beta}^{\text{ridge}}) = \text{Bias}^2 + \text{Variance} = \left(\frac{\lambda}{1+\lambda}\right)^2 \|\beta\|^2 + \frac{\sigma^2}{(1+\lambda)^2} p$$

Comparison to Least Squares (OLS) - orthogonal design

For OLS, $\hat{\beta}^{\text{OLS}} = X^\top y$:

$$\begin{aligned}
\text{Bias}(\hat{\beta}^{\text{OLS}}) &= 0 \\
\text{Var}(\hat{\beta}^{\text{OLS}}) &= \sigma^2 I \\
\text{MSE}(\hat{\beta}^{\text{OLS}}) &= \sigma^2 p
\end{aligned}$$

Ridge regression introduces bias but reduces variance, potentially lowering MSE depending on λ and $\|\beta\|^2$.**Condition when Ridge outperforms Least Squares (LS):**

$$\text{MSE}_{\text{Ridge}} = \frac{\lambda^2 \|\beta\|^2 + p\sigma^2}{(1+\lambda)^2} < \text{MSE}_{\text{LS}} = p\sigma^2$$

Multiply both sides by $(1+\lambda)^2$:

$$\lambda^2 \|\beta\|^2 + p\sigma^2 < p\sigma^2 (1+\lambda)^2$$

Expand the right-hand side:

$$\lambda^2\|\beta\|^2 + p\sigma^2 < p\sigma^2(1 + 2\lambda + \lambda^2)$$

Subtract $p\sigma^2$ from both sides:

$$\lambda^2\|\beta\|^2 < p\sigma^2(2\lambda + \lambda^2)$$

Divide both sides by λ (assume $\lambda > 0$):

$$\lambda\|\beta\|^2 < p\sigma^2(2 + \lambda)$$

Rewriting:

$$\|\beta\|^2 < \frac{p\sigma^2(2 + \lambda)}{\lambda}$$

Or, isolating λ:

$$\lambda < \frac{2p\sigma^2}{\|\beta\|^2 - p\sigma^2}$$

7) Comparison of Prediction Error: OLS vs Ridge Regression

Given:

- Number of predictors: $p = 40$
- Residual Sum of Squares (RSS):
  - OLS: $\text{RSS}_{\text{OLS}} = 4.5$
  - Ridge: $\text{RSS}_{\text{ridge}} = 11.6$
- Effective degrees of freedom for ridge:

$$\text{df}_{\text{ridge}} = \text{tr}\left(X(X^\top X + \gamma I)^{-1}X^\top\right) = 32$$

Estimated Prediction Error (EPE) is approximated by:

$$\text{EPE} \approx \frac{\text{RSS}}{n} + \frac{2\sigma^2 \cdot \text{df}}{n}$$

Assuming same noise level σ^2 and sample size n, comparison reduces to:

$$\text{EPE}_{\text{OLS}} \propto \text{RSS}_{\text{OLS}} + 2\sigma^2 p = 4.5 + 2\sigma^2 \cdot 40$$

$$\text{EPE}_{\text{ridge}} \propto \text{RSS}_{\text{ridge}} + 2\sigma^2 \cdot 32 = 11.6 + 2\sigma^2 \cdot 32$$

Compare the constants:

$$\Delta = (\text{RSS}_{\text{ridge}} - \text{RSS}_{\text{OLS}}) - 2\sigma^2(p - \text{df}_{\text{ridge}}) = 7.1 - 2\sigma^2 \cdot 8$$

Ridge has lower estimated prediction error when:

$$\sigma^2 > \frac{7.1}{16} \approx 0.444$$

Conclusion

If the noise variance $\sigma^2 > 0.444$, ridge regression yields better prediction error than OLS. Otherwise, OLS performs better.

8) LASSO: False Discoveries and Power under Orthogonal Design

We consider the linear model:

$$y = X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I), \quad \text{with} \quad X^\top X = I.$$

In this orthogonal case, the LASSO solution simplifies to coordinate-wise **soft-thresholding**:

$$\hat{\beta}_j^{\text{lasso}} = \text{sign}(\hat{\beta}_j^{\text{OLS}}) \cdot \left(|\hat{\beta}_j^{\text{OLS}}| - \lambda \right)_+ .$$

Setup

- Let $S \subset \{1, \dots, p\}$ be the index set of true signals ($|S| = s$).
- For $j \in S$: $\beta_j \neq 0$ (true signals).
- For $j \notin S$: $\beta_j = 0$ (nulls, noise).

False Discoveries

For null variables ($\beta_j = 0$), we have:

$$\hat{\beta}_j^{\text{OLS}} \sim \mathcal{N}(0, \sigma^2), \quad P(\hat{\beta}_j^{\text{lasso}} \neq 0) = 2\Phi\left(-\frac{\lambda}{\sigma}\right).$$

So the **expected number of false discoveries** is:

$$E[\text{FD}] = (p - s) \cdot 2\Phi\left(-\frac{\lambda}{\sigma}\right).$$

Power

For true variables ($\beta_j \neq 0$):

$$\hat{\beta}_j^{\text{OLS}} \sim \mathcal{N}(\beta_j, \sigma^2), \quad P(\hat{\beta}_j^{\text{lasso}} \neq 0) = \Phi\left(\frac{\beta_j - \lambda}{\sigma}\right) + \Phi\left(\frac{-\beta_j - \lambda}{\sigma}\right).$$

9) Adaptive LASSO under Orthogonal Design

Consider the adaptive LASSO estimator defined by minimizing:

$$\hat{\beta}^{\text{adapt}} = \arg\min_{\beta} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} w_i |\beta_i| \right\}$$

where w_i are variable-specific weights.

i) Using a Standard LASSO Solver for Adaptive LASSO

To compute the adaptive LASSO estimator using a standard LASSO solver like `glmnet`, perform the following steps:

1. Scale each column X_i of the design matrix by dividing it by the corresponding weight w_i: define $\tilde{X}_i = X_i / w_i$.
2. Fit a standard LASSO using $\tilde{X}$ and response y, with penalty λ.
3. Rescale the resulting coefficients: $\hat{\beta}_i^{\text{adapt}} = \hat{\tilde{\beta}}_i / w_i$.

This approach works because penalizing $w_i |\beta_i|$ is equivalent to penalizing $|\tilde{\beta}_i|$ where $\tilde{\beta}_i = w_i \beta_i$.

ii) Adaptive LASSO in the Orthogonal Case

In the orthogonal case $X'X = I$, the adaptive LASSO estimator reduces to a weighted soft-thresholding rule:

$$\hat{\beta}_i^{\text{adapt}} = \text{sign}(z_i) \cdot \max\left(|z_i| - \lambda w_i, 0\right)$$

where $z_i = X_i^T y$ is the least squares estimator for β_i.

iii) Numerical Example

Suppose the OLS estimator for β_1 is:

$$z_1 = \hat{\beta}_1^{\text{OLS}} = 3$$

and the LASSO estimator is:

$$\hat{\beta}_1^{\text{LASSO}} = 2$$

Using the LASSO soft-thresholding formula:

$$2 = \text{sign}(3) \cdot \max(3 - \lambda, 0) \Rightarrow \lambda = 1$$

Now compute the adaptive LASSO estimator with weight $w_1 = \frac{1}{4}$:

$$\hat{\beta}_1^{\text{adapt}} = \text{sign}(3) \cdot \max(3 - 1 \cdot \tfrac{1}{4}, 0) = \max(2.75, 0) = 2.75$$

Project 1 - James Stein estimator and prediction error for multiple regression

Given a gene expression matrix $X \in R^{n \times p}$, we standardize each gene j as follows:

$$\tilde{X}_{ij} = \mu_j + \frac{X_{ij} - \mu_j}{\sigma_j}$$

where

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}, \quad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{ij} - \mu_j)^2}$$

b) Centering Around Zero

Assuming the average gene expression is close to 10, we subtract 10:

$$Z_{ij} = \tilde{X}_{ij} - 10$$

c) Estimators: MLE & James Stein

Maximum Likelihood Estimator (MLE):

$$\hat{\mu}_j^{\text{MLE}} = \frac{1}{5}\sum_{i=1}^{5} Z_{ij}$$

James-Stein Estimator (shrinkage toward 0):

$$\hat{\mu}^{\text{JS}} = \left(1 - \frac{(p-2)\sigma^2}{\|\hat{\mu}^{\text{MLE}}\|^2}\right)\hat{\mu}^{\text{MLE}}$$

James-Stein Estimator (shrinkage toward common mean):

$$\bar{Z} = \frac{1}{p}\sum_{j=1}^{p} \hat{\mu}_j^{\text{MLE}}$$

$$d = \frac{p-3}{p-1}\frac{\sigma^2}{Var(\hat{\mu}^{\text{MLE}})}$$

$$\hat{\mu}^{\text{JS-mean}} = (1-d)\hat{\mu}^{\text{MLE}} + d\bar{Z}$$

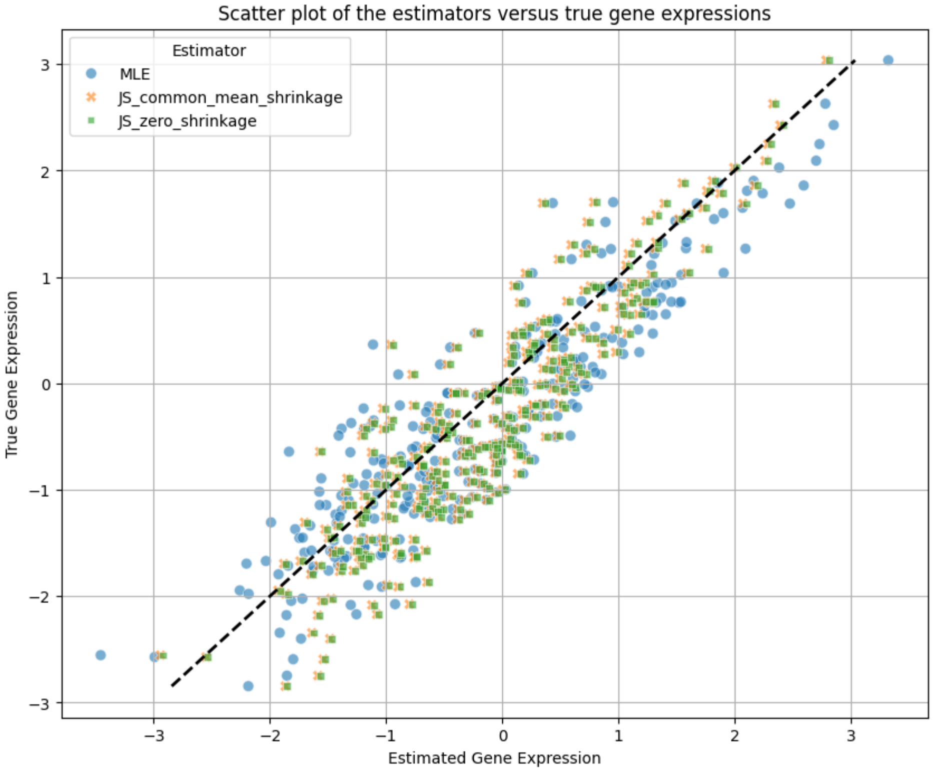


Figure 1: Estimators and true gene expressions

Estimator	Squared Error
MLE	83.619665
James-Stein (shrinkage toward 0)	77.664063
James-Stein (shrinkage toward common mean)	75.148937

Table 2: Comparison of squared error for different estimators

The variance $\sigma^2 = 0.2$ used in the estimators reflects the distribution $N(0, \sqrt{1/5})$, consistent with a sample size of $n = 5$. The results presented in Table 2 demonstrate the well-established advantage of the James-Stein estimator over the MLE in terms of squared error, especially in high-dimensional settings. Both versions of the James-Stein estimator—shrinking toward zero and toward the overall mean—achieve lower estimation error than the MLE. Notably, shrinkage toward the common mean yields the best performance, which is expected given the assumption that the true gene expression levels are approximately centered around a shared value (after adjusting for the baseline of 10).

Project 1 - Prediction Error

We will simulate the design matrix from the normal distribution:

$$X_{1000\times 950} \sim \mathcal{N}\left(0, \sigma = \frac{1}{\sqrt{1000}}\right)$$

i.i.d. The real vector of coefficients have five signals of strength three and different number $(q-5)$ of zero entries:

$$\beta = (3, 3, 3, 3, 3, 0, \ldots, 0)^T$$

Random noise will be added to the data:

$$\varepsilon \sim \mathcal{N}(0, I)$$

The length of the vector of coefficients:

- $q = 2, 5, 10, 100, 500, 950$

The number of repetitions of the experiment: 100.

Prediction Error

Relying solely on the model's fit can result in overfitting. To properly assess model performance on unseen data, it is crucial to evaluate how well it generalizes. One effective way to do this is by examining the prediction error, which reflects the expected discrepancy between predicted responses and the actual values generated using the same design matrix and new random noise. Ideally, the mean squared error (MSE) of the estimators would remain consistent between the training data and fresh data (differing only by a new error term). In this task, we will analyze the prediction error and how its estimators behave for various values of q.

Prediction error:

$$PE = E\|X(\beta - \hat{\beta}) + \varepsilon^*\|^2,$$

where $\varepsilon^* \sim \mathcal{N}(0, I)$ is a new noise vector.

Estimator of the PE (in orthogonal design):

- σ known:

$$\widehat{PE}_1 = RSS + 2\sigma^2 p$$

- σ unknown:

$$\widehat{PE}_2 = RSS + 2\hat{\sigma}^2 p = \frac{(n+p)RSS}{n-p}$$

Leave-one-out CV PE estimator:

$$\widehat{PE}_3 = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i}{1 - H_{i,i}} \right)^2,$$

where $H = X(X^T X)^{-1} X^T$ is the projection matrix.

Model Size	RSS	PE (True)	PE (σ known)	PE (σ unknown)	PE (LOOCV)
2	1015.95	1003.25	1019.95	1020.02	1020.02
5	993.88	1004.60	1003.88	1003.87	1003.98
10	999.15	1009.84	1019.15	1019.34	1019.37
100	895.55	1099.53	1095.55	1094.56	1105.60
500	509.38	1498.15	1509.38	1528.14	2043.01
950	51.89	1962.73	1951.89	2023.88	21485.92

Table 3: Prediction error and its estimates for different model sizes

Prediction Error and Overfitting Behavior

The table summarizes the prediction error (PE) and its estimates across different model sizes. Several trends emerge as we vary the number of variables k used in the model:

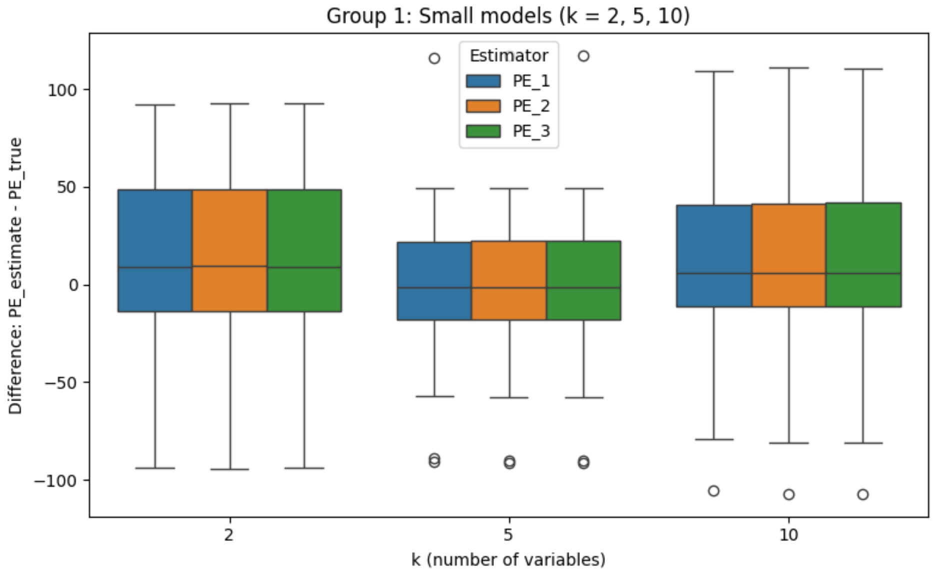
Small Models (e.g., $k = 2, 5, 10$): The true prediction error is close to all three estimates (RSS-based with known and unknown σ, and LOOCV). All estimators are relatively accurate in this regime. In particular, model size $k = 5$ appears nearly optimal—it yields the lowest true prediction error and exhibits excellent agreement between the different estimation methods.

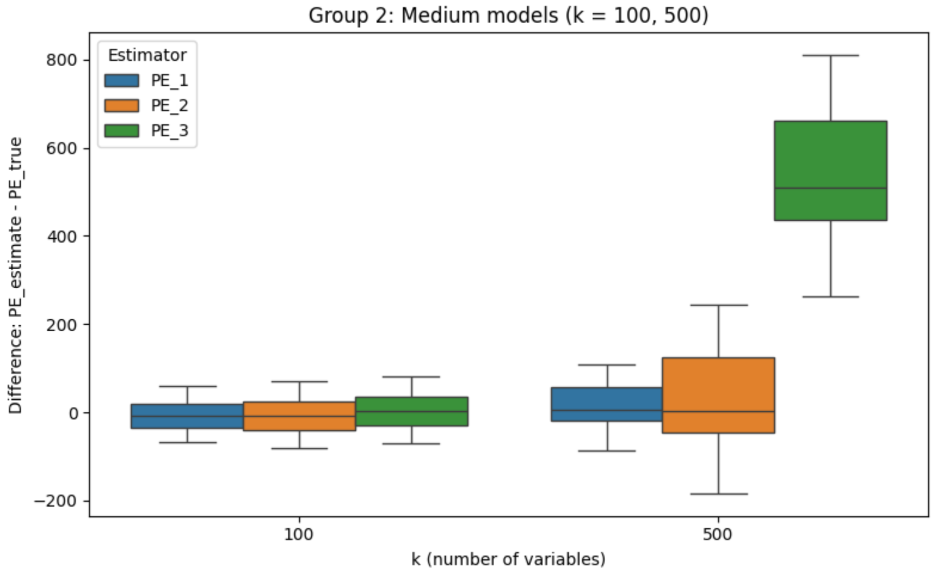
Medium Models ($k = 100, 500$): As the model size increases, the estimators begin to diverge. The LOOCV estimate starts to slightly overestimate the true prediction error, while the RSS-based methods still perform reasonably but show increasing estimation error. This reflects a growing sensitivity to variance as more variables are added.

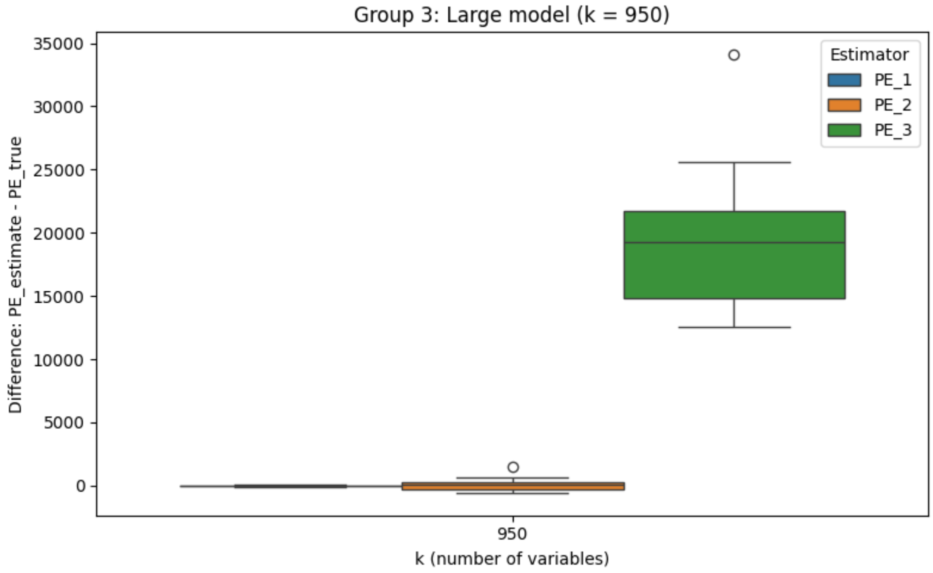
Large Model ($k = 950$): Despite a very low RSS value (only 51.89), the true prediction error jumps to 1962.73, and the LOOCV estimate rises dramatically to 21,485.92. This is a textbook example of *overfitting*—the model captures noise rather than signal. While the training fit improves (as seen by decreasing RSS), the model generalizes poorly to new data. The RSS-based estimators severely underestimate the generalization error, falsely suggesting a high-performing model. LOOCV more accurately detects the rise in prediction error but becomes unstable and overly pessimistic in this high-dimensional regime.

- Underfitting occurs in very small models (e.g., $k = 2$), where the model is too simple, leading to large residuals and unnecessarily high prediction error.
- Optimal prediction accuracy is achieved around $k = 5$ to 10, where the model is complex enough to capture the signal but not so complex as to overfit.
- Overfitting becomes increasingly problematic for $k > 100$, particularly when k approaches n (sample size). The model fits the training data too closely, losing predictive power.
- LOOCV is a powerful, data-driven estimator of prediction error that adapts well to different model sizes. However, it may become unstable in high-dimensional scenarios.
- RSS-based estimators rely on strong assumptions and begin to fail as model complexity increases and the design matrix becomes ill-conditioned.

It is important to note that increasing the dimension of the design matrix (i.e., the number of variables or columns p) typically improves the in-sample fit, as reflected by a lower RSS. However, this does not guarantee better predictive performance. Adding variables that do not contribute meaningful information can worsen prediction accuracy, despite the improved fit. This phenomenon is known as *overfitting.*


Group 1: Small models (k = 2, 5, 10)
Estimator
PE_1
PE_2
PE_3
100
50
0
−50
−100
Difference: PE_estimate - PE_true
2
5
10
k (number of variables)


Group 2: Medium models (k = 100, 500)
Estimator
PE_1
PE_2
PE_3
800
600
400
200
0
−200
Difference: PE_estimate - PE_true
100
500
k (number of variables)


Group 3: Large model (k = 950)
Estimator
PE_1
PE_2
PE_3
35000
30000
25000
20000
15000
10000
5000
0
Difference: PE_estimate - PE_true
950
k (number of variables)

Project 2 - Multiple regression - model selection and regularization

Let the design matrix $X \in R^{1000\times 950}$ be such that its elements are independent and identically distributed (i.i.d.) random variables from the normal distribution:

$$X_{ij} \sim \mathcal{N}(0, \sigma^2), \quad \text{with } \sigma^2 = \frac{1}{\sqrt{1000}}.$$

Define the true coefficient vector $\beta \in R^{950}$ such that:

$$\beta_j = \begin{cases} 6, & \text{for } j = 1, 2, \dots, 20, \\ 0, & \text{for } j = 21, 22, \dots, 950. \end{cases}$$

Then, the response vector $Y \in R^{1000}$ is generated according to the linear model, where $\varepsilon \sim \mathcal{N}(0, I)$:

$$Y = X\beta + \varepsilon.$$

Estimation Error (SE & V1)

In the following, we evaluate two key measures of estimation error:

- **Squared ℓ_2 error (SE):**
$$\text{SE} = \|\widehat{\beta} - \beta\|_2^2$$
- **Prediction error (V1):**
$$\text{V1} = \|X(\widehat{\beta} - \beta)\|_2^2$$

Estimation Error (SE, V1)

- **mBIC2:** SE = 21.27, V1 = 21.14 — Extremely low estimation error; nearly perfect selection, no shrinkage bias.
- **Ridge Regression:** SE = 465.27, V1 = 309.79 — Very high error; Ridge retains all variables, unsuitable for sparse models.
- **LASSO (lambda.min):** SE = 88.54, V1 = 83.39 — Reasonable but harmed by many false positives.
- **LASSO (lambda.1se):** SE = 147.92, V1 = 145.23 — More conservative regularization, worse estimation error than lambda.min.
- **LASSO (lambda.min) + OLS refit:** SE = 187.93, V1 = 192.97 — Poor performance: aggressive selection introduces noise that OLS amplifies.
- **LASSO (lambda.1se) + OLS refit:** SE = 71.46, V1 = 71.04 — Substantial improvement; conservative selection and OLS correction balance well.
- **Fixed LASSO:** SE = 1527.69, V1 = 614.07 — Severe shrinkage; coefficients close to zero; bad performance.
- **Fixed LASSO + OLS refit:** SE = 3851.95, V1 = 872.68 — Catastrophic: large model size, huge estimation error.
- **SLOPE:** SE = 202.33, V1 = 200.21 — Reasonable balance between selection and shrinkage.
- **SLOPE + OLS refit:** SE = 52.17, V1 = 52.20 — Excellent performance; removes shrinkage bias effectively.

False Discovery Proportion (FDP)

- **mBIC2:** FDP = 0.000 — Perfect variable selection (no false positives).
- **Ridge Regression:** FDP = 0.973 — Keeps almost all variables; no sparsity.
- **LASSO (lambda.min):** FDP = 0.667 — Many false positives due to aggressive penalty choice.
- **LASSO (lambda.1se):** FDP = 0.259 — Strong improvement; more conservative selection.
- **Fixed LASSO:** FDP = 0.973 — Poor: selects almost all variables.
- **SLOPE:** FDP = 0.167 — Best false positive control among regularization methods.

True Discovery Proportion (TDP)

- **Most methods** achieve TDP ≈ 1 — they recover nearly all true positives.
- **SLOPE** slightly less than perfect due to more conservative behavior.

Simulation Results (10 Repetitions)

Method	**SE (mean ± std)**	**V1 (mean ± std)**	**Comment**
mBIC2	42.73 ± 28.23	43.23 ± 27.72	Best overall; no shrinkage bias.
Ridge Regression	478.23 ± 9.78	310.20 ± 6.80	Retains all variables; no sparsity.
LASSO (lambda.min)	108.95 ± 24.22	98.23 ± 21.18	Moderate error, harmed by false positives.
LASSO (lambda.1se)	150.29 ± 28.39	148.71 ± 27.91	Higher bias, fewer false positives.
Fixed LASSO	1456.12 ± 95.64	610.12 ± 38.94	Severe shrinkage; terrible estimation.
SLOPE	235.22 ± 36.57	228.51 ± 34.12	Good balance, but estimation bias exists.
LASSO (lambda.min) + OLS refit	287.24 ± 73.83	292.32 ± 76.15	Worse due to many false positives.
LASSO (lambda.1se) + OLS refit	118.86 ± 49.01	116.51 ± 48.25	Big improvement with conservative selection.
Fixed LASSO + OLS refit	3500.41 ± 299.80	878.92 ± 58.74	Catastrophic: huge variance after refitting.
SLOPE + OLS refit	65.77 ± 28.18	65.81 ± 27.83	Outstanding: low bias after clean selection.

Table 4: Simulation results for 10 repetitions. The table shows the mean and standard deviation for the squared error (SE) and prediction error (V1) for different methods.

Method	**FDP (mean ± std)**	**TDP (mean ± std)**
mBIC2	0.036 ± 0.056	0.980 ± 0.035
Ridge Regression	-	-
LASSO (lambda.min)	0.761 ± 0.060	1.000 ± 0.000
LASSO (lambda.1se)	0.411 ± 0.104	1.000 ± 0.000
Fixed LASSO	0.973 ± 0.001	1.000 ± 0.000
SLOPE	0.201 ± 0.076	0.995 ± 0.016
LASSO (lambda.min) + OLS refit	0.761 ± 0.060	1.000 ± 0.000
LASSO (lambda.1se) + OLS refit	0.411 ± 0.104	1.000 ± 0.000
Fixed LASSO + OLS refit	0.973 ± 0.001	1.000 ± 0.000
SLOPE + OLS refit	0.201 ± 0.076	0.995 ± 0.016

Table 5: False Discovery Proportion (FDP) and True Discovery Proportion (TDP) for all methods. OLS refitting affects only SE/V1, not the selection quality of methods.

Global Conclusions

- **mBIC2** remains the best method overall: perfect or near-perfect selection, very low estimation error (both SE and V1), and no shrinkage bias. No OLS refitting needed.

- **SLOPE** also performs very well — especially after **OLS refitting** — providing clean variable selection (low FDP) and extremely low estimation error.
- **LASSO (lambda.min)** is too aggressive: it has many false positives, which cause poor performance after refitting.
- **LASSO (lambda.1se) + OLS refitting** greatly improves results: conservative selection followed by bias removal achieves strong performance.
- **Fixed LASSO** fails: strong shrinkage and poor selection quality create disastrous results, both before and after OLS.
- **OLS refitting** is very effective only if variable selection is clean (few false positives); otherwise, it amplifies errors.
- **V1 and SE results are very consistent:** when V1 is close to SE, the estimator is stable; when they differ greatly, refitting or shrinkage matters critically.