Selected Topics in Mathematics of Learning

High-Dimensional Statistics

Lecturer: Marius Yamakou

Winter Semester 2024/25
Department of Data Science, FAU

November 19, 2024

Part III continued

Sparse Linear Models

Sparse linear models: A theoretical perspective

6. Sparse linear models: A theoretical perspective: Outline

Question: How can we access an estimator's performance?

- Different error metrics
  1. Prediction Error
  2. Parametric Error
  3. Variable Selection

6.1 Prediction Error

Proposition (High-Probability Bound)

Assume $y = X\beta + \varepsilon$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Let $\widehat{\beta}$ be the LASSO estimator:

$$\widehat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^q} \left(\frac{1}{2n} \|y - X\beta\|_2^2 + \lambda_n \|\beta\|_1 \right),$$

Then, for any $\tau > 0$, the prediction error satisfies:

$$\mathbb{P}\left(\frac{1}{n} \|X\widehat{\beta} - X\beta\|_2^2 > 4\sigma \sqrt{\frac{2(1+\tau)\log(p)}{n}} \|\beta\|_1 \right) \leq 2p^{-\tau}.$$

6.1 Prediction Error

Proposition (High-Probability Bound)

Assume $y = X\beta + \varepsilon$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Let $\widehat{\beta}$ be the LASSO estimator:

$$\widehat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^q} \left(\frac{1}{2n} \|y - X\beta\|_2^2 + \lambda_n \|\beta\|_1 \right),$$

Then, for any $\tau > 0$, the prediction error satisfies:

$$\mathbb{P}\left(\frac{1}{n} \|X\widehat{\beta} - X\beta\|_2^2 > 4\sigma \sqrt{\frac{2(1+\tau)\log(p)}{n}} \|\beta\|_1 \right) \leq 2p^{-\tau}.$$

Key Insight: Sparsity Matters!

- The bound depends on $\|\beta\|_1$, highlighting the importance of sparsity in β.
- Larger $\|\beta\|_1$ can increase the error bound, emphasizing the benefit of sparse solutions in high-dimensional settings.
- Notice that $\mathbb{P} \to 0$ as $p \to \infty$.

6.1 Prediction Error

Proof:

$$\frac{1}{2n}\|y - X\hat{\beta}\|_2^2 + \lambda_n\|\hat{\beta}\|_1 \leq \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda_n\|\beta\|_1, \quad \forall\beta \in \mathbb{R}^p,$$

$$\implies \tfrac{1}{2n}\big(\|y - X\hat{\beta}\|_2^2 - \|y - X\beta\|_2^2\big) \leq \lambda_n\big(\|\beta\|_1 - \|\hat{\beta}\|_1\big),$$

$$\implies \tfrac{1}{2n}\big(\|y - X\beta + X\beta - X\hat{\beta}\|_2^2 - \|y - X\beta\|_2^2\big) \leq \lambda_n\big(\|\beta\|_1 - \|\hat{\beta}\|_1\big),$$

$$\implies \tfrac{1}{2n}\big(\|y - X\beta\|_2^2 + \|X\beta - X\hat{\beta}\|_2^2 + 2\langle y - X\beta, X\beta - X\hat{\beta}\rangle - \|y - X\beta\|_2^2\big)$$
$$\leq \lambda_n\big(\|\beta\|_1 - \|\hat{\beta}\|_1\big),$$

$$\implies \tfrac{1}{2n}\big(2\langle y - X\beta, X\beta - X\hat{\beta}\rangle + \|X\beta - X\hat{\beta}\|_2^2\big) \leq \lambda_n\big(\|\beta\|_1 - \|\hat{\beta}\|_1\big),$$

With $\varepsilon = y - X\beta,$

$$\implies \tfrac{1}{n}\big(\|X\beta - X\hat{\beta}\|_2^2\big) \leq 2\lambda_n\big(\|\beta\|_1 - \|\hat{\beta}\|_1\big) + \tfrac{2}{n}\langle\varepsilon, X(\hat{\beta} - \beta)\rangle, \quad \text{(*)}$$

6.1 Prediction Error

(i) What about the inner product $\langle \varepsilon, X(\hat{\beta} - \beta)\rangle$?

Apply $\forall u, v \in \mathbb{R}^p$: $\langle u, v\rangle \leq \max_{i=1,\dots,p} |u_i| \|v\|_1$ and $\langle u, v\rangle := u^\top v = \sum_{i=1}^{p} u_i v_i$

We get:

$$\langle \varepsilon, X(\hat{\beta} - \beta)\rangle = \varepsilon^\top [X(\hat{\beta} - \beta)] = [X^\top \varepsilon]^\top (\hat{\beta} - \beta) = \langle X^\top \varepsilon, \hat{\beta} - \beta\rangle \leq \|X^\top \varepsilon\|_{\max} \|\hat{\beta} - \beta\|_1$$

6.1 Prediction Error

(i) What about the inner product $\langle \varepsilon, X(\hat{\beta} - \beta) \rangle$?

Apply $\forall u, v \in \mathbb{R}^p : \langle u, v \rangle \leq \max_{i=1,\dots,p} |u_i| \|v\|_1$ and $\langle u, v \rangle := u^\top v = \sum_{i=1}^{p} u_i v_i$

We get:

$$\langle \varepsilon, X(\hat{\beta} - \beta) \rangle = \varepsilon^\top [X(\hat{\beta} - \beta)] = [X^\top \varepsilon]^\top (\hat{\beta} - \beta) = \langle X^\top \varepsilon, \hat{\beta} - \beta \rangle \leq \|X^\top \varepsilon\|_{\max} \|\hat{\beta} - \beta\|_1$$

(ii) Continuing from Eqn (*) in previous slide:

$$\frac{1}{n} \|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + \frac{2}{n} \langle \varepsilon, X(\hat{\beta} - \beta) \rangle$$

$$\frac{1}{n} \|X\beta - X\hat{\beta}\|_2^2 \overset{(i)}{\leq} 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + \frac{2}{n} \|X^\top \varepsilon\|_{\max} \|\hat{\beta} - \beta\|_1$$

6.1 Prediction Error

(iii) Suppose we are on $\mathcal{A} = \left\{ \frac{1}{n} \|X^\top \varepsilon\|_{\max} \leq \lambda_n \right\}$, which we will show is true with high probability.

6.1 Prediction Error

(iii) Suppose we are on $\mathcal{A} = \left\{ \frac{1}{n} \|X^\top \varepsilon\|_{\max} \leq \lambda_n \right\}$, which we will show is true with high probability.

(iv) Put (i) and (iii) together, we have:

$$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1\right) + 2\lambda_n \|\hat{\beta} - \beta\|_1. \quad \text{(**)}$$

6.1 Prediction Error

- (iii) Suppose we are on $\mathcal{A} = \left\{ \frac{1}{n} \|X^\top \varepsilon\|_{\max} \leq \lambda_n \right\}$, which we will show is true with high probability.
- (iv) Put (i) and (iii) together, we have:

$$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + 2\lambda_n \|\hat{\beta} - \beta\|_1. \quad \text{(**)}$$

Using the triangle inequality: $\|\hat{\beta} - \beta\|_1 = \|\hat{\beta} + (-\beta)\|_1 \leq \|\hat{\beta}\|_1 + \| - \beta\|_1$.

The ℓ_1-norm is absolute, so $\| - \beta\|_1 = \|\beta\|_1$. Thus $\|\hat{\beta} - \beta\|_1 \leq \|\hat{\beta}\|_1 + \|\beta\|_1$

6.1 Prediction Error

(iii) Suppose we are on $\mathcal{A} = \left\{ \frac{1}{n} \|X^\top \varepsilon\|_{\max} \leq \lambda_n \right\}$, which we will show is true with high probability.

(iv) Put (i) and (iii) together, we have:

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + 2\lambda_n \|\hat{\beta} - \beta\|_1.$ (**)

Using the triangle inequality: $\|\hat{\beta} - \beta\|_1 = \|\hat{\beta} + (-\beta)\|_1 \leq \|\hat{\beta}\|_1 + \| - \beta\|_1.$

The ℓ_1-norm is absolute, so $\| - \beta\|_1 = \|\beta\|_1$. Thus $\|\hat{\beta} - \beta\|_1 \leq \|\hat{\beta}\|_1 + \|\beta\|_1$

Using this in (**), we get:

$$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + 2\lambda_n \left(\|\hat{\beta}\|_1 + \|\beta\|_1 \right)$$

6.1 Prediction Error

(iii) Suppose we are on $\mathcal{A} = \left\{ \frac{1}{n} \|X^\top \varepsilon\|_{\max} \le \lambda_n \right\}$, which we will show is true with high probability.

(iv) Put (i) and (iii) together, we have:

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \le 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + 2\lambda_n \|\hat{\beta} - \beta\|_1.$ (**)

Using the triangle inequality: $\|\hat{\beta} - \beta\|_1 = \|\hat{\beta} + (-\beta)\|_1 \le \|\hat{\beta}\|_1 + \|-\beta\|_1$.

The ℓ_1-norm is absolute, so $\|-\beta\|_1 = \|\beta\|_1$. Thus $\|\hat{\beta} - \beta\|_1 \le \|\hat{\beta}\|_1 + \|\beta\|_1$

Using this in (**), we get:

$$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \le 2\lambda_n \left(\|\beta\|_1 - \|\hat{\beta}\|_1 \right) + 2\lambda_n \left(\|\hat{\beta}\|_1 + \|\beta\|_1 \right)$$

$$\implies \frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \le 2\lambda_n \left[\|\beta\|_1 - \|\hat{\beta}\|_1 + \|\hat{\beta}\|_1 + \|\beta\|_1 \right]$$

6.1 Prediction Error

(iii) Suppose we are on $\mathcal{A} = \left\{ \frac{1}{n}\|X^\top \varepsilon\|_{\max} \leq \lambda_n \right\}$, which we will show is true with high probability.

(iv) Put (i) and (iii) together, we have:

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n\left(\|\beta\|_1 - \|\hat{\beta}\|_1\right) + 2\lambda_n\|\hat{\beta} - \beta\|_1.$ (**)

Using the triangle inequality: $\|\hat{\beta} - \beta\|_1 = \|\hat{\beta} + (-\beta)\|_1 \leq \|\hat{\beta}\|_1 + \|-\beta\|_1.$

The ℓ_1-norm is absolute, so $\|-\beta\|_1 = \|\beta\|_1$. Thus $\|\hat{\beta} - \beta\|_1 \leq \|\hat{\beta}\|_1 + \|\beta\|_1$

Using this in (**), we get:

$$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n\left(\|\beta\|_1 - \|\hat{\beta}\|_1\right) + 2\lambda_n\left(\|\hat{\beta}\|_1 + \|\beta\|_1\right)$$

$$\implies \frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 2\lambda_n\left[\|\beta\|_1 - \|\hat{\beta}\|_1 + \|\hat{\beta}\|_1 + \|\beta\|_1\right]$$

$$\implies \frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$$

6.1 Prediction Error

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

Observations:

- picking a smaller λ yields a tighter bound.

6.1 Prediction Error

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

Observations:

- picking a smaller λ yields a tighter bound.
- probability of being on $\mathcal{A} = \{\frac{1}{n}\|X'\varepsilon\|_{\max} \leq \lambda\}$ decreases as λ decreases.

6.1 Prediction Error

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

Observations:

- picking a smaller λ yields a tighter bound.
- probability of being on $\mathcal{A} = \{\frac{1}{n}\|X'\varepsilon\|_{\max} \leq \lambda\}$ decreases as λ decreases.
- There is ultimately a trade-off to be made, and we will see that it is possible to parameterize λ in a nice way to show off the various bounds.

6.1 Prediction Error

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

Observations:

- picking a smaller λ yields a tighter bound.
- probability of being on $\mathcal{A} = \{\frac{1}{n}\|X'\varepsilon\|_{\max} \leq \lambda\}$ decreases as λ decreases.
- There is ultimately a trade-off to be made, and we will see that it is possible to parameterize λ in a nice way to show off the various bounds.

$$y = X\beta + \varepsilon$$

For any vector $z = (z_1, \ldots, z_n)^\top \in \mathbb{R}^n$, the $\max$-norm is defined as $\|z\|_{\max} = \max_{i=1,\ldots,n} |z_i|$.

6.1 Prediction Error

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

Observations:

- picking a smaller λ yields a tighter bound.
- probability of being on $\mathcal{A} = \{\frac{1}{n}\|X'\varepsilon\|_{\max} \leq \lambda\}$ decreases as λ decreases.
- There is ultimately a trade-off to be made, and we will see that it is possible to parameterize λ in a nice way to show off the various bounds.

$$y = X\beta + \varepsilon$$

For any vector $z = (z_1, \ldots, z_n)^\top \in \mathbb{R}^n$, the $\max$-norm is defined as $\|z\|_{\max} = \max_{i=1,\ldots,n} |z_i|$.

Lemma

Suppose $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\top \in \mathbb{R}^n$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and X a fixed design matrix with columns $(x_1, \ldots, x_p)$ with $\|x_j\|_2^2 = n$. Then, for any $\gamma > 0$,

$$\mathbb{P}\left(\frac{1}{n}\|\varepsilon^\top X\|_{\max} > \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\right) \leq 2p^{-\gamma}.$$

6.1 Prediction Error

Proof:

$$
\begin{aligned}
\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) &= \mathbb{P}\left(\left\{\tfrac{1}{n}|(X^\top\varepsilon)_1| > \lambda\right\} \cup \cdots \cup \left\{\tfrac{1}{n}|(X^\top\varepsilon)_p| > \lambda\right\}\right) \\
&= \mathbb{P}\left(\bigcup_{i=1}^p \left\{\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right\}\right) \\
&\leq \sum_{i=1}^p \mathbb{P}\left(\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right), \quad \textbf{(**)}
\end{aligned}
$$

6.1 Prediction Error

Proof:

$$\begin{aligned}\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) &= \mathbb{P}\left(\left\{\tfrac{1}{n}|(X^\top \varepsilon)_1| > \lambda\right\} \cup \cdots \cup \left\{\tfrac{1}{n}|(X^\top \varepsilon)_p| > \lambda\right\}\right)\\ &= \mathbb{P}\left(\textstyle\bigcup_{i=1}^p \left\{\tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right\}\right)\\ &\leq \sum_{i=1}^p \mathbb{P}\left(\tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right), \quad \textbf{(**)}\end{aligned}$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2) \implies \frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^n \varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)$$

6.1 Prediction Error

Proof:

$$\begin{aligned}
\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) &= \mathbb{P}\left(\left\{\tfrac{1}{n}|(X^\top\varepsilon)_1| > \lambda\right\} \cup \cdots \cup \left\{\tfrac{1}{n}|(X^\top\varepsilon)_p| > \lambda\right\}\right) \\
&= \mathbb{P}\left(\textstyle\bigcup_{i=1}^p \left\{\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right\}\right) \\
&\leq \sum_{i=1}^p \mathbb{P}\left(\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right), \quad \textbf{(**)}
\end{aligned}$$

$$\varepsilon_i \sim \mathcal{N}(0,\sigma^2) \implies \frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^n \varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)$$

$$\mathbb{E}\left[\frac{1}{n}\varepsilon^\top X\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}(\varepsilon_i x_{ij}) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}(\varepsilon_i) x_{ij} = \frac{1}{n}\sum_{i=1}^n 0 \cdot x_{ij} = 0$$

6.1 Prediction Error

Proof:

$$
\begin{aligned}
\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) &= \mathbb{P}\left(\left\{\tfrac{1}{n}|(X^\top\varepsilon)_1| > \lambda\right\} \cup \cdots \cup \left\{\tfrac{1}{n}|(X^\top\varepsilon)_p| > \lambda\right\}\right) \\
&= \mathbb{P}\left(\bigcup_{i=1}^p \left\{\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right\}\right) \\
&\leq \sum_{i=1}^p \mathbb{P}\left(\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right), \quad \text{(**)}
\end{aligned}
$$

$$
\varepsilon_i \sim \mathcal{N}(0,\sigma^2) \implies \frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^n \varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)
$$

$$
\mathbb{E}\left[\frac{1}{n}\varepsilon^\top X\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}(\varepsilon_i x_{ij}) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}(\varepsilon_i)x_{ij} = \frac{1}{n}\sum_{i=1}^n 0\cdot x_{ij} = 0
$$

$$
\begin{aligned}
\mathbb{E}\left[\left(\tfrac{1}{n}\varepsilon^\top X\right)^2\right] &= \mathbb{E}\left[\left(\tfrac{1}{n}\textstyle\sum_{i=1}^n \varepsilon_i x_{ij}\right)\left(\tfrac{1}{n}\textstyle\sum_{k=1}^n \varepsilon_k x_{kj}\right)\right] \\
&= \mathbb{E}\left[\tfrac{1}{n^2}\textstyle\sum_{i,k=1}^n \varepsilon_i\varepsilon_k x_{ij}x_{kj}\right] \\
&= \tfrac{1}{n^2}\textstyle\sum_{i,k=1}^n \mathbb{E}(\varepsilon_i\varepsilon_k)x_{ij}x_{kj}
\end{aligned}
$$

6.1 Prediction Error

$$\mathbb{E}(\varepsilon_i \varepsilon_k) = \begin{cases} \sigma^2 & \text{if } i = k, \\ 0 & \text{if } i \neq k. \end{cases}$$

$$\mathbb{E}\left[\left(\frac{1}{n}\varepsilon^\top X\right)^2\right] = \frac{\sigma^2}{n^2}\sum_{i=1}^n x_{ij}^2 = \frac{\sigma^2}{n^2}\cdot \|x_j\|_2^2 = \frac{\sigma^2}{n^2}\cdot n = \frac{\sigma^2}{n}$$

Hence, $\frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^n \varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)$

6.1 Prediction Error

$$\mathbb{E}(\varepsilon_i \varepsilon_k) = \begin{cases} \sigma^2 & \text{if } i = k, \\ 0 & \text{if } i \neq k. \end{cases}$$

$$\mathbb{E}\left[\left(\frac{1}{n}\varepsilon^\top X\right)^2\right] = \frac{\sigma^2}{n^2}\sum_{i=1}^{n} x_{ij}^2 = \frac{\sigma^2}{n^2} \cdot \|x_j\|_2^2 = \frac{\sigma^2}{n^2} \cdot n = \frac{\sigma^2}{n}$$

Hence, $\frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)$

We note that both expressions, $\frac{1}{n}\varepsilon^\top X$ and $\frac{1}{n}X^\top \varepsilon$, correspond to the same sum and hence have the same Gaussian distribution.

6.1 Prediction Error

$$\mathbb{E}(\varepsilon_i \varepsilon_k) = \begin{cases} \sigma^2 & \text{if } i = k, \\ 0 & \text{if } i \neq k. \end{cases}$$

$$\mathbb{E}\left[\left(\frac{1}{n}\varepsilon^\top X\right)^2\right] = \frac{\sigma^2}{n^2}\sum_{i=1}^{n} x_{ij}^2 = \frac{\sigma^2}{n^2}\cdot \|x_j\|_2^2 = \frac{\sigma^2}{n^2}\cdot n = \frac{\sigma^2}{n}$$

Hence, $\frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)$

We note that both expressions, $\frac{1}{n}\varepsilon^\top X$ and $\frac{1}{n}X^\top\varepsilon$, correspond to the same sum and hence have the same Gaussian distribution.

$$\text{Recall: } \frac{1}{n}(X^\top\varepsilon)_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right) \implies \frac{1}{n}(X^\top\varepsilon)_i \text{ is sub-Gaussian}$$

6.1 Prediction Error

$$\mathbb{E}(\varepsilon_i \varepsilon_k) = \begin{cases} \sigma^2 & \text{if } i = k, \\ 0 & \text{if } i \neq k. \end{cases}$$

$$\mathbb{E}\left[\left(\frac{1}{n}\varepsilon^\top X\right)^2\right] = \frac{\sigma^2}{n^2}\sum_{i=1}^{n} x_{ij}^2 = \frac{\sigma^2}{n^2} \cdot \|x_j\|_2^2 = \frac{\sigma^2}{n^2} \cdot n = \frac{\sigma^2}{n}$$

Hence, $\frac{1}{n}\varepsilon^\top X = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i x_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right)$

We note that both expressions, $\frac{1}{n}\varepsilon^\top X$ and $\frac{1}{n}X^\top \varepsilon$, correspond to the same sum and hence have the same Gaussian distribution.

$$\text{Recall: } \frac{1}{n}(X^\top \varepsilon)_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{n}\right) \implies \frac{1}{n}(X^\top \varepsilon)_i \text{ is sub-Gaussian}$$

$$\implies P\left(\left|\frac{1}{n}(X^\top \varepsilon)_i\right| > \lambda\right) \leq 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

6.1 Prediction Error

From Eqn (**) we have:

$$\mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} \mathbb{P}\left(\frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

6.1 Prediction Error

From Eqn (**) we have:

$$\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} \mathbb{P}\left(\tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq p \cdot 2\exp\left(-\frac{n\lambda^2}{2\sigma^2}\right)$$

6.1 Prediction Error

From Eqn (**) we have:

$$\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} \mathbb{P}\left(\tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq p \cdot 2\exp\left(-\frac{n\lambda^2}{2\sigma^2}\right)$$

We choose $\lambda = \sqrt{\frac{\sigma^2 2(1+\gamma)\log(p)}{n}}$

6.1 Prediction Error

From Eqn (**) we have:

$$\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \le \sum_{i=1}^{p} \mathbb{P}\left(\tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \le \sum_{i=1}^{p} 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \le p \cdot 2\exp\left(-\frac{n\lambda^2}{2\sigma^2}\right)$$

We choose $\lambda = \sqrt{\frac{\sigma^2 2(1+\gamma)\log(p)}{n}}$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \le p \cdot 2\exp\left[-\frac{\sigma^2 2(1+\gamma)\log(p)}{n} \cdot \frac{n}{2\sigma^2}\right]$$

6.1 Prediction Error

From Eqn (**) we have:

$$\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} \mathbb{P}\left(\tfrac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) \leq p \cdot 2\exp\left(-\frac{n\lambda^2}{2\sigma^2}\right)$$

We choose $\lambda = \sqrt{\frac{\sigma^2 2(1+\gamma)\log(p)}{n}}$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top\varepsilon)_i| > \lambda\right) \leq p \cdot 2\exp\left[-\frac{\sigma^2 2(1+\gamma)\log(p)}{n} \cdot \frac{n}{2\sigma^2}\right]$$

Note: $p \cdot 2\exp\left[-(1+\gamma)\log(p)\right] = 2p \cdot p^{-(1+\gamma)} = 2p^{-\gamma}.$

6.1 Prediction Error

From Eqn (**) we have:

$$\mathbb{P}\left(\max_i \tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} \mathbb{P}\left(\tfrac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq \sum_{i=1}^{p} 2\exp\left(-\frac{\lambda^2}{2\frac{\sigma^2}{n}}\right)$$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq p \cdot 2\exp\left(-\frac{n\lambda^2}{2\sigma^2}\right)$$

We choose $\lambda = \sqrt{\frac{\sigma^2 2(1+\gamma)\log(p)}{n}}$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \lambda\right) \leq p \cdot 2\exp\left[-\frac{\sigma^2 2(1+\gamma)\log(p)}{n} \cdot \frac{n}{2\sigma^2}\right]$$

Note: $\quad p \cdot 2\exp\left[-(1+\gamma)\log(p)\right] = 2p \cdot p^{-(1+\gamma)} = 2p^{-\gamma}.$

$$\implies \mathbb{P}\left(\max_i \frac{1}{n}|(X^\top \varepsilon)_i| > \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\Bigg)\right) \leq 2p^{-\gamma}.$$

6.1 Prediction Error

Finally, scaling back to $\|\varepsilon^\top X\|_{\max}$, we find:

$$\mathbb{P}\left(\frac{1}{n}\|\varepsilon^\top X\|_{\max} > \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\right) \leq 2p^{-\gamma}.$$

6.1 Prediction Error

Finally, scaling back to $\|\varepsilon^\top X\|_{\max}$, we find:

$$\mathbb{P}\left(\frac{1}{n}\|\varepsilon^\top X\|_{\max} > \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\right) \leq 2p^{-\gamma}.$$

- From (iv) (5 slides above), we got $\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

6.1 Prediction Error

Finally, scaling back to $\|\varepsilon^\top X\|_{\max}$, we find:

$$\mathbb{P}\left(\frac{1}{n}\|\varepsilon^\top X\|_{\max} > \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\right) \leq 2p^{-\gamma}.$$

- From (iv) (5 slides above), we got $\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

We choose $\lambda_n = \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}$

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\|\beta\|_1$

6.1 Prediction Error

Finally, scaling back to $\|\varepsilon^\top X\|_{\max}$, we find:

$$\mathbb{P}\left(\frac{1}{n}\|\varepsilon^\top X\|_{\max} > \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\right) \leq 2p^{-\gamma}.$$

- From (iv) (5 slides above), we got $\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\lambda_n\|\beta\|_1$

We choose $\lambda_n = \sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}$

$\frac{1}{n}\|X\beta - X\hat{\beta}\|_2^2 \leq 4\sigma\sqrt{\frac{2(1+\gamma)\log(p)}{n}}\|\beta\|_1$

Using Sub-Gaussian tail bounds, we have:

$$\mathbb{P}\left(\frac{1}{n}\|X\hat{\beta} - X\beta\|_2^2 > t\right) \leq \delta,$$

it suffices to choose $t = 4\sigma\sqrt{\frac{2(1+\tau)\log(p)}{n}}\|\beta\|_1$, $\delta = 2p^{-\tau}$, with $\tau > 0$ □

6.1 Prediction Error: Key-take away

1. **Prediction Error:** $\|X(\widehat{\beta} - \beta)\|_F^2$
   - **Definition:**
     - Measures how well the estimated coefficients $\widehat{\beta}$ predict outcomes within the sample.

6.1 Prediction Error: Key-take away

1. **Prediction Error:** $\|X(\widehat{\beta} - \beta)\|_F^2$
   - **Definition:**
     - Measures how well the estimated coefficients $\widehat{\beta}$ predict outcomes within the sample.
   - **Expression:**

$$\|X(\widehat{\beta} - \beta)\|_F^2$$

   where:

   - X: Design matrix, containing observed predictor values.
   - $\widehat{\beta}$: Estimated coefficients from LASSO.
   - β: True coefficients.
   - $\| \cdot \|_F^2$: Squared Frobenius norm, summarizing total prediction error over all dimensions of X.

6.1 Prediction Error: Key-take away

1. **Prediction Error:** $\|X(\widehat{\beta} - \beta)\|_F^2$
   - **Definition:**
     - Measures how well the estimated coefficients $\widehat{\beta}$ predict outcomes within the sample.
   - **Expression:**

$$\|X(\widehat{\beta} - \beta)\|_F^2$$

   where:
     - X: Design matrix, containing observed predictor values.
     - $\widehat{\beta}$: Estimated coefficients from LASSO.
     - β: True coefficients.
     - $\| \cdot \|_F^2$: Squared Frobenius norm, summarizing total prediction error over all dimensions of X.
   - **Interpretation:**
     - Smaller prediction error implies better alignment of $\widehat{\beta}$ with β, leading to accurate predictions.

6.1 Prediction Error: Key-take away

2. Components of Prediction Error

- Design Matrix (X):
  - Represents observed values of predictors in the dataset.
  - Determines how errors in $\widehat{\beta}$ propagate through the predictions.

6.1 Prediction Error: Key-take away

2. Components of Prediction Error

- Design Matrix (X):
  - Represents observed values of predictors in the dataset.
  - Determines how errors in $\widehat{\beta}$ propagate through the predictions.
- Coefficient Error ($\widehat{\beta} - \beta$):
  - Captures the deviation of the estimated coefficients from the true coefficients.
  - Larger deviations directly increase prediction error.

6.1 Prediction Error: Key-take away

2. Components of Prediction Error

- Design Matrix (X):
  - Represents observed values of predictors in the dataset.
  - Determines how errors in $\widehat{\beta}$ propagate through the predictions.
- Coefficient Error ($\widehat{\beta} - \beta$):
  - Captures the deviation of the estimated coefficients from the true coefficients.
  - Larger deviations directly increase prediction error.
- Frobenius Norm ($\| \cdot \|_F^2$):
  - Aggregates errors over all dimensions of X and the dataset.
  - Reflects the total prediction error across samples and predictors.

6.1 Prediction Error: Key-take away

3. Properties of Prediction Error

- *“Weakest” Error Measure:*
    - Focuses on prediction accuracy rather than coefficient recovery or variable selection.

6.1 Prediction Error: Key-take away

3. Properties of Prediction Error

- *"Weakest" Error Measure:*
  - Focuses on prediction accuracy rather than coefficient recovery or variable selection.
- *Appropriate When β Is Not the Primary Interest:*
  - Useful in prediction-focused applications where interpretability is less important.

6.1 Prediction Error: Key-take away

3. Properties of Prediction Error

- *"Weakest" Error Measure:*
  - Focuses on prediction accuracy rather than coefficient recovery or variable selection.
- *Appropriate When β Is Not the Primary Interest:*
  - Useful in prediction-focused applications where interpretability is less important.
- *No Restrictive Assumptions on X:*
  - Does not require strong conditions like restricted eigenvalue assumptions on X.

6.1 Prediction Error: Key-take away

3. Properties of Prediction Error

- *"Weakest" Error Measure:*
  - Focuses on prediction accuracy rather than coefficient recovery or variable selection.
- *Appropriate When β Is Not the Primary Interest:*
  - Useful in prediction-focused applications where interpretability is less important.
- *No Restrictive Assumptions on X:*
  - Does not require strong conditions like restricted eigenvalue assumptions on X.
- *Proof Technique:*
  - Relies on basic inequalities and probabilistic tools (e.g., sub-Gaussian bounds).
  - Results are widely applicable across different settings.

6.2 Parametric error

Under additional assumptions, similar statements can be made about the parametric error. This result is more involved so we state without Proof.

Proposition (Parametric error)

Under some technical assumptions, there is a constant $c > 0$ such that for any $\tau > 0$

$$\mathbb{P}\left(\|\widehat{\beta} - \beta\|_2 > c\sigma\sqrt{\frac{2(1+\tau)\log(p)}{n}s}\right) \leq 2p^{-\tau}.$$

where s is the sparsity of β.

6.2 Parametric error: Key-take away

2. **Parametric Error:** $\|\widehat{\beta} - \beta\|_F^2$

   - **Definition:** Measures the accuracy of $\widehat{\beta}$ in recovering the true coefficients β.

6.2 Parametric error: Key-take away

2. Parametric Error: $\|\widehat{\beta} - \beta\|_F^2$

- **Definition:** Measures the accuracy of $\widehat{\beta}$ in recovering the true coefficients β.
- **Components:**
  - $\widehat{\beta} - \beta$: The error in estimating the true coefficients.
  - $\| \cdot \|_F^2$: The squared Frobenius norm, representing the total error.

6.2 Parametric error: Key-take away

2. **Parametric Error:** $\|\widehat{\beta} - \beta\|_F^2$

- **Definition:** Measures the accuracy of $\widehat{\beta}$ in recovering the true coefficients β.

- **Components:**
  - $\widehat{\beta} - \beta$: The error in estimating the true coefficients.
  - $\| \cdot \|_F^2$: The squared Frobenius norm, representing the total error.

- **Properties:**
  - *Appropriate for recovery problems:* Useful when the primary goal is to estimate the true coefficients β.
  - *Requires additional assumptions on X:* Conditions like restricted eigenvalue assumptions are often necessary.
  - *Variable selection not guaranteed:* Does not ensure the structure of $\widehat{\beta}$ matches β.
  - *Proof technique:* Relies on inequalities for analysis.

6.3 Variable selection

Under additional assumptions, similar statements can be made about variable selection. The result is more involved, so we state without Proof.

Proposition (Variable selection)

Under many technical assumptions and assuming that $\lambda > \frac{2\sigma^2 \log(p)}{n}$ and in addition the minimum value of the regression vector on its support is bounded below as $\beta_{\min} > g(\lambda)$ for some function g, then,

$$\mathbb{P}\Big(\operatorname{supp}_{+-}(\widehat{\beta}) = \operatorname{supp}_{+-}(\beta)\Big) \to 1 \quad \text{as} \quad p \to \infty$$

In other words, we get support and sign consistency.

6.3 Variable Selection: Key-take away

3. **Variable Selection:** $\mathrm{supp}(\widehat{\beta}) = \mathrm{supp}(\beta)$
   - **Definition:** Assesses whether the estimated support of $\widehat{\beta}$ matches the true support of β.

6.3 Variable Selection: Key-take away

3. Variable Selection: $\mathrm{supp}(\widehat{\beta}) = \mathrm{supp}(\beta)$

- **Definition:** Assesses whether the estimated support of $\widehat{\beta}$ matches the true support of β.
- **Components:**
  - $\mathrm{supp}(\widehat{\beta})$: The indices where $\widehat{\beta}$ is non-zero.
  - $\mathrm{supp}(\beta)$: The indices where β is non-zero.

6.3 Variable Selection: Key-take away

3. Variable Selection: $\operatorname{supp}(\widehat{\beta}) = \operatorname{supp}(\beta)$

- **Definition:** Assesses whether the estimated support of $\widehat{\beta}$ matches the true support of β.
- **Components:**
  - $\operatorname{supp}(\widehat{\beta})$: The indices where $\widehat{\beta}$ is non-zero.
  - $\operatorname{supp}(\beta)$: The indices where β is non-zero.
- **Properties:**
  - *Appropriate for scientific interest in non-zero locations:* Relevant when identifying the structure of β is crucial.
  - *Most stringent of all three criteria:* Exact recovery of support is the hardest to achieve.
  - *Requires multiple conditions on X:* Stronger assumptions, such as incoherence or sparsity, are needed.
  - *Proof technique:* Uses primal-dual witness conditions, which are non-trivial.

6.4 Summary: Error Metrics: Prediction and Parametric Error

What are the different error metrics to assess estimators' performances?

- **Prediction Error:**
  - Measures the accuracy of predictions:

$$\|X(\widehat{\beta} - \beta)\|_2^2$$

  - Focuses on prediction performance within the observed sample.
  - Key in applications where the outcome is of primary interest (e.g., forecasting).
- **Parametric Error:**
  - Measures the difference between the estimated and true coefficients:

$$\|\widehat{\beta} - \beta\|_2$$

  - Focuses on recovering the true values of coefficients.
  - Key in settings where coefficient values have interpretative importance.

6.4 Summary: Error Metrics: Variable Selection

What are the different error metrics to assess estimators' performances? (Continued.)

- **Variable Selection:**
  - Evaluates the ability to identify the true set of relevant predictors:
  $$\operatorname{supp}(\widehat{\beta}) = \operatorname{supp}(\beta)$$
  - Focuses on correctly identifying nonzero coefficients in β.
  - Important in feature selection applications, especially in high dimensions.

- **Parameter Consistency vs. Support Recovery:**
  - Parameter consistency ensures that the estimated coefficients converge to the true coefficients in some norm (e.g., $\|\hat{\beta} - \beta\|_2 \to 0$).
  $$\|\widehat{\beta} - \beta\|_2 \to 0$$
  - Support recovery consistency ensures that the support of $\hat{\beta}$ (nonzero entries) matches the true support of β (with correct signs).
  $$\operatorname{supp}(\widehat{\beta}) = \operatorname{supp}(\beta)$$
  - Difference:
    - Parameter consistency does not guarantee correct support recovery.
    - Support recovery is stricter and ensures identification of nonzero predictors.

6.4 Summary: Important Facts: Role of λ

- Regularization parameter λ is critical in LASSO.
- Choosing $\lambda > \sqrt{\frac{\log(p)}{n}}$:
  - Ensures sufficient regularization to suppress noise.
  - Helps reduce error metrics (prediction error, parametric error, or variable selection error) with high probability.
  - Balances between bias (over-regularization) and variance (under-regularization).

6.4 Summary: Important Facts: High-Dimensional Trade-Offs

- $\sqrt{\frac{\log(p)}{n}}$ reflects the trade-off between:
  - Dimensionality (p): Larger p increases $\log(p)$, loosening error bounds.
  - Sample size (n): Larger n reduces $\sqrt{\frac{\log(p)}{n}}$, tightening bounds.
- Sparsity in β (s) helps mitigate high-dimensional challenges.
- Proper tuning of λ is critical for performance across different metrics.

7. Sparse linear models: A practical perspective: Outline

1. How to choose the regularization λ in practice?
2. Cross-validation
    1. informal
    2. formal
3. The one standard error rule

7.1 How to choose λ in practice?

Importance of λ

- As we have seen, the penalty parameter λ is of crucial importance in penalized regression.

7.1 How to choose λ in practice?

Importance of λ

- As we have seen, the penalty parameter λ is of crucial importance in penalized regression.
- For $\lambda = 0$ we essentially just get the LS estimates of the full model.

7.1 How to choose λ in practice?

Importance of λ

- As we have seen, the penalty parameter λ is of crucial importance in penalized regression.
- For $\lambda = 0$ we essentially just get the LS estimates of the full model.
- For very large λ ridge estimates become extremely small, while LASSO estimates are exactly zero!

7.1 How to choose λ in practice?

Importance of λ

- As we have seen, the penalty parameter λ is of crucial importance in penalized regression.
- For $\lambda = 0$ we essentially just get the LS estimates of the full model.
- For very large λ ridge estimates become extremely small, while LASSO estimates are exactly zero!
- Recall the bias-variance trade-off!

7.1 How to choose λ in practice?

Importance of λ

- As we have seen, the penalty parameter λ is of crucial importance in penalized regression.
- For $\lambda = 0$ we essentially just get the LS estimates of the full model.
- For very large λ ridge estimates become extremely small, while LASSO estimates are exactly zero!
- Recall the bias-variance trade-off!

We require a principled way to fine-tune λ in order to get optimal results.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.
2. **Split the data:** Randomly split the dataset into K folds. Use $K-1$ folds for training and 1 fold for testing in each iteration.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.
2. **Split the data:** Randomly split the dataset into K folds. Use $K - 1$ folds for training and 1 fold for testing in each iteration.
3. **Define a grid of λ:** Specify the range of λ values to evaluate.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.
2. **Split the data:** Randomly split the dataset into K folds. Use $K-1$ folds for training and 1 fold for testing in each iteration.
3. **Define a grid of λ:** Specify the range of λ values to evaluate.
4. **Compute validation MSE:** For each value of λ, calculate the Mean Squared Error (MSE) on the validation set in each fold.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.
2. **Split the data:** Randomly split the dataset into K folds. Use $K-1$ folds for training and 1 fold for testing in each iteration.
3. **Define a grid of λ:** Specify the range of λ values to evaluate.
4. **Compute validation MSE:** For each value of λ, calculate the Mean Squared Error (MSE) on the validation set in each fold.
5. **Aggregate cross-validation MSE:** Compute the average MSE across all folds for each λ.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.
2. **Split the data:** Randomly split the dataset into K folds. Use $K-1$ folds for training and 1 fold for testing in each iteration.
3. **Define a grid of λ:** Specify the range of λ values to evaluate.
4. **Compute validation MSE:** For each value of λ, calculate the Mean Squared Error (MSE) on the validation set in each fold.
5. **Aggregate cross-validation MSE:** Compute the average MSE across all folds for each λ.
6. **Select optimal λ:** Identify the λ value that minimizes the cross-validation MSE.

7.2 Cross-Validation: Informal Description

A viable strategy: K-fold Cross-Validation with the following steps:

1. **Choose the number of folds (K):** Decide how many groups to divide the data into.
2. **Split the data:** Randomly split the dataset into K folds. Use $K-1$ folds for training and 1 fold for testing in each iteration.
3. **Define a grid of λ:** Specify the range of λ values to evaluate.
4. **Compute validation MSE:** For each value of λ, calculate the Mean Squared Error (MSE) on the validation set in each fold.
5. **Aggregate cross-validation MSE:** Compute the average MSE across all folds for each λ.
6. **Select optimal λ:** Identify the λ value that minimizes the cross-validation MSE.

Why K-fold Cross-Validation? Ensures robust model performance by leveraging all data for both training and validation.

7.2 Cross-Validation: Formal Description

Objective: We formalize cross-validation for the LASSO. The same procedure can be adapted for ridge regression by adjusting the objective function.

7.2 Cross-Validation: Formal Description

Objective: We formalize cross-validation for the LASSO. The same procedure can be adapted for ridge regression by adjusting the objective function.

Linear Model: $y = X\beta + \varepsilon$ where:

- $y = (y_1, \ldots, y_n)^\top$ is the response vector,
- $X = (X_1, \ldots, X_n)^\top$ is the design matrix.

7.2 Cross-Validation: Formal Description

Objective: We formalize cross-validation for the LASSO. The same procedure can be adapted for ridge regression by adjusting the objective function.

Linear Model: $y = X\beta + \varepsilon$ where:

- $y = (y_1, \ldots, y_n)^\top$ is the response vector,
- $X = (X_1, \ldots, X_n)^\top$ is the design matrix.

Recall the LASSO Objective:

$$\widehat{\beta}(\lambda) = \underset{\beta}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - X_i^\top \beta)^2 + \lambda \|\beta\|_1$$

- The penalty term $\lambda\|\beta\|_1$ induces sparsity in $\widehat{\beta}(\lambda)$.
- Cross-validation evaluates model performance across different values of λ.

7.2 Cross-Validation: Formal Description

Procedure:

- We have the dataset $\{(X_i, y_i)\}_{i=1,\ldots,n}$.
- Split the index set $\{1, \ldots, n\}$ into K subsets (i.e., into folds) of roughly equal size, denoted $F_1, \ldots, F_K$.

7.2 Cross-Validation: Formal Description

Procedure:

- We have the dataset $\{(X_i, y_i)\}_{i=1,\dots,n}$.
- Split the index set $\{1, \dots, n\}$ into K subsets (i.e., into folds) of roughly equal size, denoted $F_1, \dots, F_K$.
- **For each fold** $k = 1, \dots, K$:
    - Use all data points with indices $i \notin F_k$ as the training set.
    - Use all data points with indices $i \in F_k$ as the validation set.
    - For each tuning parameter $\lambda \in \{\lambda_1, \dots, \lambda_M\}$:

7.2 Cross-Validation: Formal Description

Procedure:

- We have the dataset $\{(X_i, y_i)\}_{i=1,\ldots,n}$.
- Split the index set $\{1, \ldots, n\}$ into K subsets (i.e., into folds) of roughly equal size, denoted $F_1, \ldots, F_K$.
- **For each fold** $k = 1, \ldots, K$:
  - Use all data points with indices $i \notin F_k$ as the training set.
  - Use all data points with indices $i \in F_k$ as the validation set.
  - For each tuning parameter $\lambda \in \{\lambda_1, \ldots, \lambda_M\}$:
    - Compute the LASSO estimate on the training set:

$$\widehat{\beta}_{-k}(\lambda) = \underset{\beta}{\text{argmin}} \frac{1}{n - n_k} \sum_{i \notin F_k} (y_i - X_i^\top \beta)^2 + \lambda \|\beta\|_1,$$

where $n_k = |F_k|$ is the size of the k-th fold.

7.2 Cross-Validation: Formal Description

Procedure:

- We have the dataset $\{(X_i, y_i)\}_{i=1,\dots,n}$.
- Split the index set $\{1, \dots, n\}$ into K subsets (i.e., into folds) of roughly equal size, denoted $F_1, \dots, F_K$.
- **For each fold** $k = 1, \dots, K$:
  - Use all data points with indices $i \notin F_k$ as the training set.
  - Use all data points with indices $i \in F_k$ as the validation set.
  - For each tuning parameter $\lambda \in \{\lambda_1, \dots, \lambda_M\}$:
    - Compute the LASSO estimate on the training set:

      $$\widehat{\beta}_{-k}(\lambda) = \underset{\beta}{\text{argmin}} \frac{1}{n - n_k} \sum_{i \notin F_k} (y_i - X_i^\top \beta)^2 + \lambda \|\beta\|_1,$$

      where $n_k = |F_k|$ is the size of the k-th fold.
    - Compute the total error on the validation set:

      $$e_k(\lambda) = \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2.$$

7.2 Cross-Validation: Formal Description

- **Compute the average cross-validation error for each λ and Select the optimal λ:**

$$\widehat{\lambda}_{CV} = \underset{\lambda \in \{\lambda_1, \ldots, \lambda_M\}}{\text{argmin}} \frac{1}{n} \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2 = \underset{\lambda \in \{\lambda_1, \ldots, \lambda_M\}}{\text{argmin}} \frac{1}{n} \sum_{k=1}^{K} e_k(\lambda)$$

7.2 Cross-Validation: Formal Description

- **Compute the average cross-validation error for each λ and Select the optimal λ:**

$$\widehat{\lambda}_{CV} = \underset{\lambda \in \{\lambda_1, \dots, \lambda_M\}}{\text{argmin}} \frac{1}{n} \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2 = \underset{\lambda \in \{\lambda_1, \dots, \lambda_M\}}{\text{argmin}} \frac{1}{n} \sum_{k=1}^{K} e_k(\lambda)$$

Key Insight: The selected $\widehat{\lambda}_{CV}$ minimizes the average cross-validation error, balancing model fit and regularization.

7.2 Example: Simulation Study

Part I

Setup: Simulation with $n = 50$ and $p = 30$. The entries of the predictor matrix $X \in \mathbb{R}^{50 \times 30}$ are all i.i.d. $\mathcal{N}(0, 1)$.

Response Generation: The response vector $y \in \mathbb{R}^{50}$ is drawn from the model:

$$y = X\beta + \varepsilon,$$

where:

- β is the coefficient vector.
- $\varepsilon \in \mathbb{R}^{50}$ represents noise, with entries i.i.d. $\mathcal{N}(0, 1)$.

7.2 Example: Simulation Study

Part I

Setup: Simulation with $n = 50$ and $p = 30$. The entries of the predictor matrix $X \in \mathbb{R}^{50\times 30}$ are all i.i.d. $\mathcal{N}(0, 1)$.

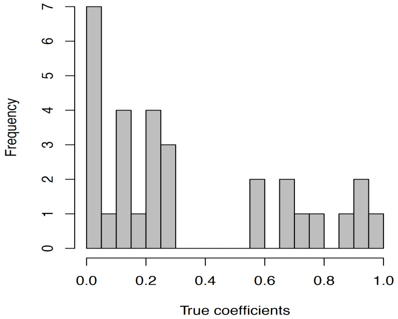
Response Generation: The response vector $y \in \mathbb{R}^{50}$ is drawn from the model:

$$y = X\beta + \varepsilon,$$

where:

- β is the coefficient vector.
- $\varepsilon \in \mathbb{R}^{50}$ represents noise, with entries i.i.d. $\mathcal{N}(0, 1)$.

Visualization:

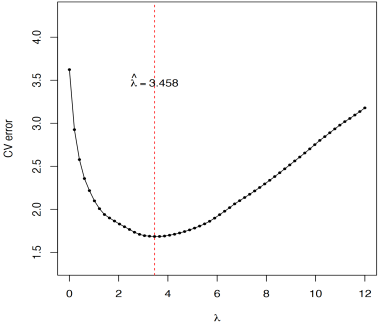


Histogram of the true regression coefficients $\beta \in \mathbb{R}^{50}$. Here 10 coefficients are large (between 0.5 and 1) and 20 coefficients are small (between 0 and 0.3)

7.2 Example: Simulation Study

Part II

The cross-validation error curve from our LASSO example.



$$\widehat{\lambda}_{CV} = \underset{\lambda \in \{\lambda_1, \ldots, \lambda_M\}}{\text{argmin}} \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} \left(y_i - X_i^\top \widehat{\beta}_{-k}(\lambda)\right)^2$$

7. 2. Standard errors for cross-validation

We can estimate the standard deviation of

$$CV(\lambda) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - X_i' \widehat{\beta}_{-k}(\lambda))^2$$

at each $\lambda \in \{\lambda_1, \ldots, \lambda_M\}$.

7. 2. Standard errors for cross-validation

We can estimate the standard deviation of

$$CV(\lambda) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - X_i' \widehat{\beta}_{-k}(\lambda))^2$$

at each $\lambda \in \{\lambda_1, \ldots, \lambda_M\}$. First, we average the validation errors in each fold:

$$CV_k(\lambda) = \frac{1}{n_k} \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2,$$

where n_k is the number of points in the kth fold.

7. 2. Standard errors for cross-validation

We can estimate the standard deviation of

$$CV(\lambda) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} \left(y_i - X_i' \widehat{\beta}_{-k}(\lambda)\right)^2$$

at each $\lambda \in \{\lambda_1, \ldots, \lambda_M\}$. First, we average the validation errors in each fold:

$$CV_k(\lambda) = \frac{1}{n_k} \sum_{i \in F_k} \left(y_i - X_i^\top \widehat{\beta}_{-k}(\lambda)\right)^2,$$

where n_k is the number of points in the kth fold. We then compute the sample standard deviation of $CV_1(\lambda), \ldots, CV_K(\lambda)$,

$$SD(\lambda) = \sqrt{Var(CV_1(\lambda), \ldots, CV_K(\lambda))} = \sqrt{\frac{1}{k-1} \sum_{k=1}^{K} \left(CV_k(\lambda) - \frac{1}{k} \sum_{k=1}^{K} CV_k(\lambda)\right)^2}$$

7. 2. Standard errors for cross-validation

We can estimate the standard deviation of

$$CV(\lambda) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - X_i' \widehat{\beta}_{-k}(\lambda))^2$$

at each $\lambda \in \{\lambda_1, \ldots, \lambda_M\}$. First, we average the validation errors in each fold:

$$CV_k(\lambda) = \frac{1}{n_k} \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2,$$

where n_k is the number of points in the kth fold. We then compute the sample standard deviation of $CV_1(\lambda), \ldots, CV_K(\lambda)$,

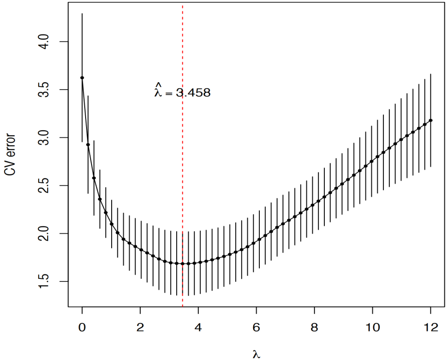
$$SD(\lambda) = \sqrt{Var(CV_1(\lambda), \ldots, CV_K(\lambda))} = \sqrt{\frac{1}{k-1} \sum_{k=1}^{K} \left(CV_k(\lambda) - \frac{1}{k} \sum_{k=1}^{K} CV_k(\lambda) \right)^2}$$

Finally we estimate the standard deviation of $SE(\lambda) = SD(\lambda)/\sqrt{K}$ called the standard error of $CV(\lambda)$.

7.2 Example: Simulation Study

Part III

The cross-validation error curve from our lasso example, with $+-$ standard errors (i.e., the range around an estimate, determined by adding and subtracting the standard error $SE(\lambda)$ of the estimate.): $SE(\lambda) = \frac{SD(\lambda)}{\sqrt{k}}$



7.2 The one standard error rule

The one standard error rule is an alternative rule for choosing the value of the tuning parameter, as opposed to the usual rule

$$\widehat{\lambda}_{CV} =_{\lambda \in \{\lambda_1, \ldots, \lambda_M\}} \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2.$$

7.2 The one standard error rule

The one standard error rule is an alternative rule for choosing the value of the tuning parameter, as opposed to the usual rule

$$\widehat{\lambda}_{CV} =_{\lambda\in\{\lambda_1,\ldots,\lambda_M\}} \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in F_k}(y_i - X_i^{\top}\widehat{\beta}_{-k}(\lambda))^2.$$

We first find the usual minimizer $\widehat{\lambda}_{CV}$ as above, and then move λ in the direction of increasing regularization as much as we can, such that the cross-validation error curve is still within one standard error of $CV(\widehat{\lambda})$. In other words, we maintain

$$CV(\lambda) \leq CV(\widehat{\lambda}_{CV}) + SE(\widehat{\lambda}_{CV})$$

7.2 The one standard error rule

The one standard error rule is an alternative rule for choosing the value of the tuning parameter, as opposed to the usual rule

$$\widehat{\lambda}_{CV} =_{\lambda \in \{\lambda_1, \ldots, \lambda_M\}} \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - X_i^\top \widehat{\beta}_{-k}(\lambda))^2.$$

We first find the usual minimizer $\widehat{\lambda}_{CV}$ as above, and then move λ in the direction of increasing regularization as much as we can, such that the cross-validation error curve is still within one standard error of $CV(\widehat{\lambda})$. In other words, we maintain

$$CV(\lambda) \leq CV(\widehat{\lambda}_{CV}) + SE(\widehat{\lambda}_{CV})$$

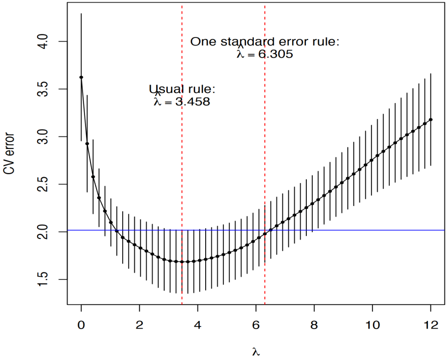
Idea: We go for the simpler (more regularized) model. The popular "One

Standard Error Rule" (1se rule) used with cross-validation (CV) is to select the most parsimonious model whose prediction error is not much worse than the minimum CV error.

7.2 Example: Simulation Study

Part III

The cross-validation error curve from our LASSO example, with $+-$ standard errors and One standard error rule: $CV(\lambda) \leq CV(\widehat{\lambda}_{CV}) + SE(\widehat{\lambda}_{CV})$.



Summary

Why is it important to choose λ?

- Bias-Variance trade-off.
- to select only relevant features.

Summary

Why is it important to choose λ?

- Bias-Variance trade-off.
- to select only relevant features.

Cross-Validation:

- What are the major steps?
- regular cross-validation vs. one standard error rule
  - regular CV:
  - 1SE rule:

Summary

Why is it important to choose λ?

- Bias-Variance trade-off.
- to select only relevant features.

Cross-Validation:

- What are the major steps?
- regular cross-validation vs. one standard error rule
  - regular CV:
  - 1SE rule:
- How can one adjust the procedure to apply it to ridge regression?

$$\beta_{-k}(\lambda) = \arg\min_{\beta} \frac{1}{n - n_k} \sum_{i \in F_k} \left(y_i - X_i^\top \beta \right)^2 + \lambda \|\beta\|_2$$