Statistical and Sequential Learning for Time Series Forecasting

Regressions

Margaux Brégère

Regression framework

Linear regression

- Univariate
- Multivariate
- Generalised linear model
- Online approaches

Penalised Regression

- Ridge regression
- Lasso regression
- Regularisation parameter tuning
- Elastic Net
- Online approaches and implementation

Generalised Additives Models

- Formulation, estimation and implementation
- Online approaches

Quantile regression

Regression framework

Setting

Regression covers several statistical analysis methods used to approximate a random variable Y with a set of other random variables $X_1, X_2, \ldots, X_p$ which are correlated to it; they are called explicative variables or features and gathered in a random vector X

Assumption

The regression model links the quantity of interest $Y \in \mathbb{R}$ with the p-dimensional vector $X \in \mathbb{R}^p$ by assuming that, for any realisation $(Y_i, X_i) \overset{\text{i.i.d}}{\sim} (X, Y)$,

$$Y_i = f^\star(X_i) + \varepsilon_i$$

where $f^\star : \mathbb{R}^p \to \mathbb{R}$ is an unknown function and $\varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,\sigma^2)$

Aim:

Finding a model $\hat{f} : \mathbb{R}^p \to \mathbb{R}$ as close as possible to $f^\star$ in oder to forecast any new realisation Y_{new} of Y based on the observation of X_{new} with $\hat{Y}_{\text{new}} = \hat{f}(X_{\text{new}})$

Setting

To estimate $f^\star$, we introduce

- $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ a loss function (quadratic, etc.)
- $\mathcal{F}$ a space of functions in which the model is sought

The objective is to solve the following minimisation problem:

$$\tilde{f} \in \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(Y,X)}\left[\ell\left(Y, f(X)\right)\right]$$

To solve this minimisation problem, the expectation of the prediction error has to be approximated using a training data set

What about data?

$\mathbb{E}\left[\ell\left(Y, f(X)\right)\right]$ is approximated on the basis of a sample of observations $\left(Y_i, X_{i1}, \ldots, X_{ip}\right)_{i=1,\ldots n}$

Rating abuse:

- $Y = (Y_1, Y_2, \ldots Y_n)$ is the n-size vector of the observations of the random variable Y
- $X \in \mathscr{M}_{n\times p}(\mathbb{R})$ is the matrix of n nows and p columns which contains the n observations $X_i = (X_{i1}, X_{i2}, \ldots X_{ip})$ of the random variables $X_1, \ldots, X_p$

$\mathbb{E}\left[\ell\left(Y, f(X)\right)\right]$ is approximated with

$$\mathbb{E}\left[\ell\left(Y, f(X)\right)\right] \approx \frac{1}{n}\sum_{i=1}^{n} \ell\left(Y_i, f\left(X_{i1}, \ldots X_{ip}\right)\right)$$

Aim: find a model $\hat{f} : \mathbb{R}^p \to \mathbb{R}$ such that

$$\hat{f} \in \arg\min_{f\in\mathscr{F}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(Y_i, f\left(X_{i1}, \ldots X_{ip}\right)\right)$$

Model selection or how to choose $\mathscr{F}$?

Choosing $\mathscr{F}$ is challenging:

- it depends on the relationships between Y and X (linear, polynomial, etc.)
- it depends on the available training data (size n, representativeness, quality)

For a new observation $(Y_{\text{new}}, X_{\text{new}})$, the error of the prediction $\hat{Y}_{\text{new}}$ can be decomposed into an irreducible error due to the noise and a two-terms error:

$$Y_{\text{new}} - \hat{Y}_{\text{new}} = f^\star(X_{\text{new}}) + \varepsilon_{\text{new}} - \hat{f}(X_{\text{new}}) = \varepsilon_{\text{new}} + f^\star(X_{\text{new}}) - \tilde{f}(X_{\text{new}}) + \tilde{f}(X_{\text{new}}) - \hat{f}(X_{\text{new}})$$

- If $\mathscr{F}$ is too restive, $\hat{f}$ is biased = under-fitting / over-smoothing

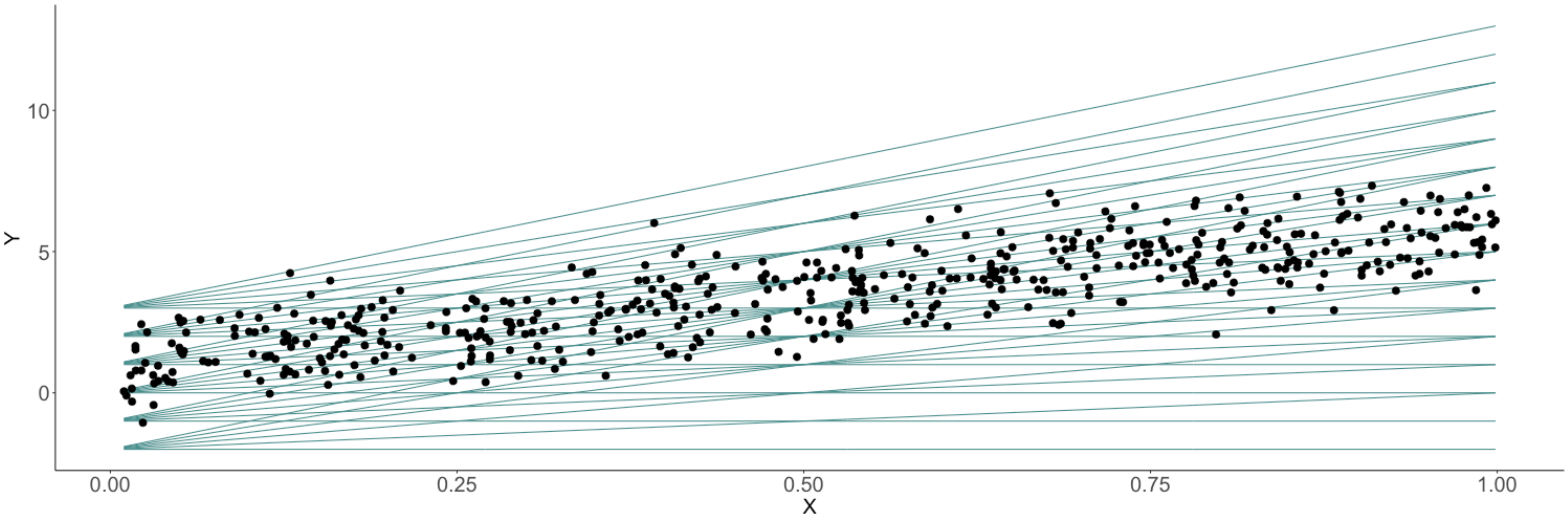
  $\hat{f}$ close to $\tilde{f}$ but $\tilde{f}$ far from $f^\star$

- If $\mathscr{F}$ is too large, $\hat{f}$ has a high variance (it is very sensitive to the training data) = over-fitting

  $\tilde{f}$ close to $f^\star$ but $\hat{f}$ far from $\tilde{f}$

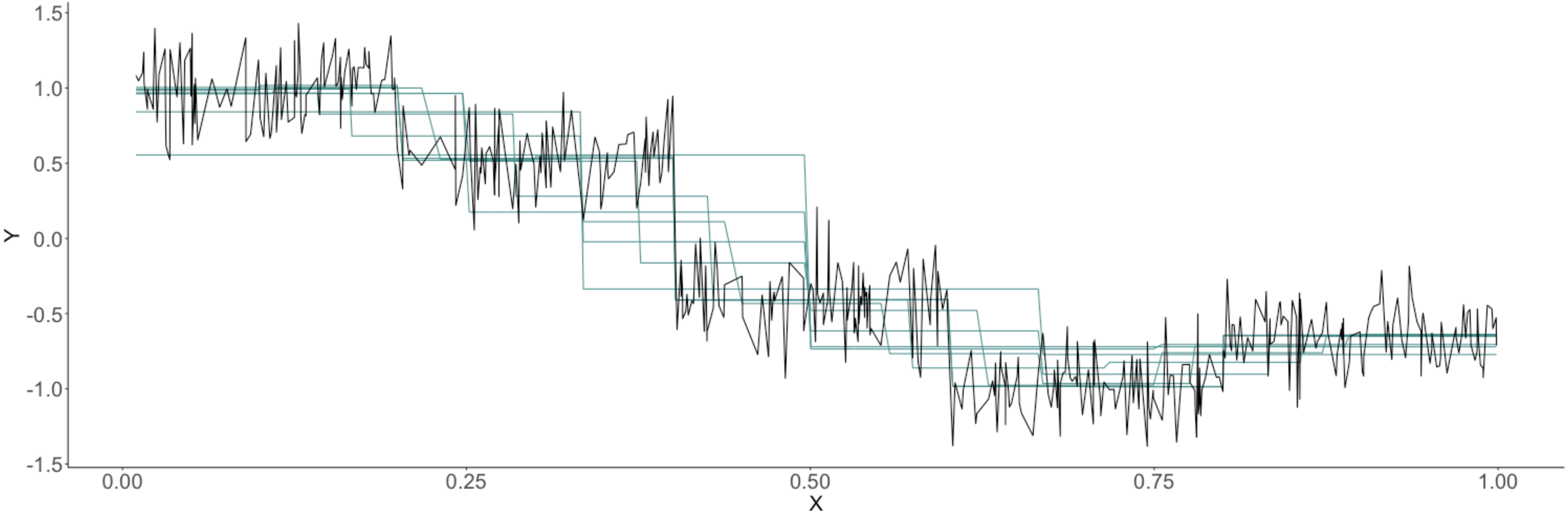
Example - univariate linear regression

$$\mathscr{F} = \{f_{\alpha,\beta} : x \mapsto \alpha + x\beta\}$$



Example - rupture detection

$$\mathcal{F} = \left\{ f_{x_0,a_0,\ldots,x_K,a_K} : x \mapsto \sum_{k=1}^{K} a_k \mathbf{1}_{x_{k-1} \leq x < x_k}(x) \right\}$$



Linear regression

Univariate linear regression

Formulation

Let $(Y_i, X_i)_{i=1,\dots,n}$ be n observations independent and identically distributed of two reals random variables Y and X

Assumptions

$Y_i = X_i\beta^\star + \varepsilon_i$ where the processus $(\varepsilon_i)_i$ is a white noise, namely $\varepsilon_i \overset{\text{i.i.d}}{\sim} \varepsilon$ with $\mathbb{E}[\varepsilon] = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$

Thus the space of models is $\mathcal{F} = \{\beta \mid \beta \in \mathbb{R}\}$

and to estimate $\beta^\star \in \mathbb{R}$, we consider the quadric loss function $\ell : \begin{array}{ccc} \mathbb{R} \times \mathbb{R} & \to & \mathbb{R}^+ \\ (y, \hat{y}) & \mapsto & (y - \hat{y})^2 \end{array}$

Ordinary Least Squares

The Ordinary Least Squares (OLS) estimator minimises the quadratic error computed over the sample $(Y_i, X_i)_{i=1,\ldots,n}$:

$$\hat{\beta}^{OLS} \in \arg\min_{\beta \in \mathbb{R}} \mathrm{Err}(\beta) \quad \text{with} \quad \mathrm{Err}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i\beta)^2$$

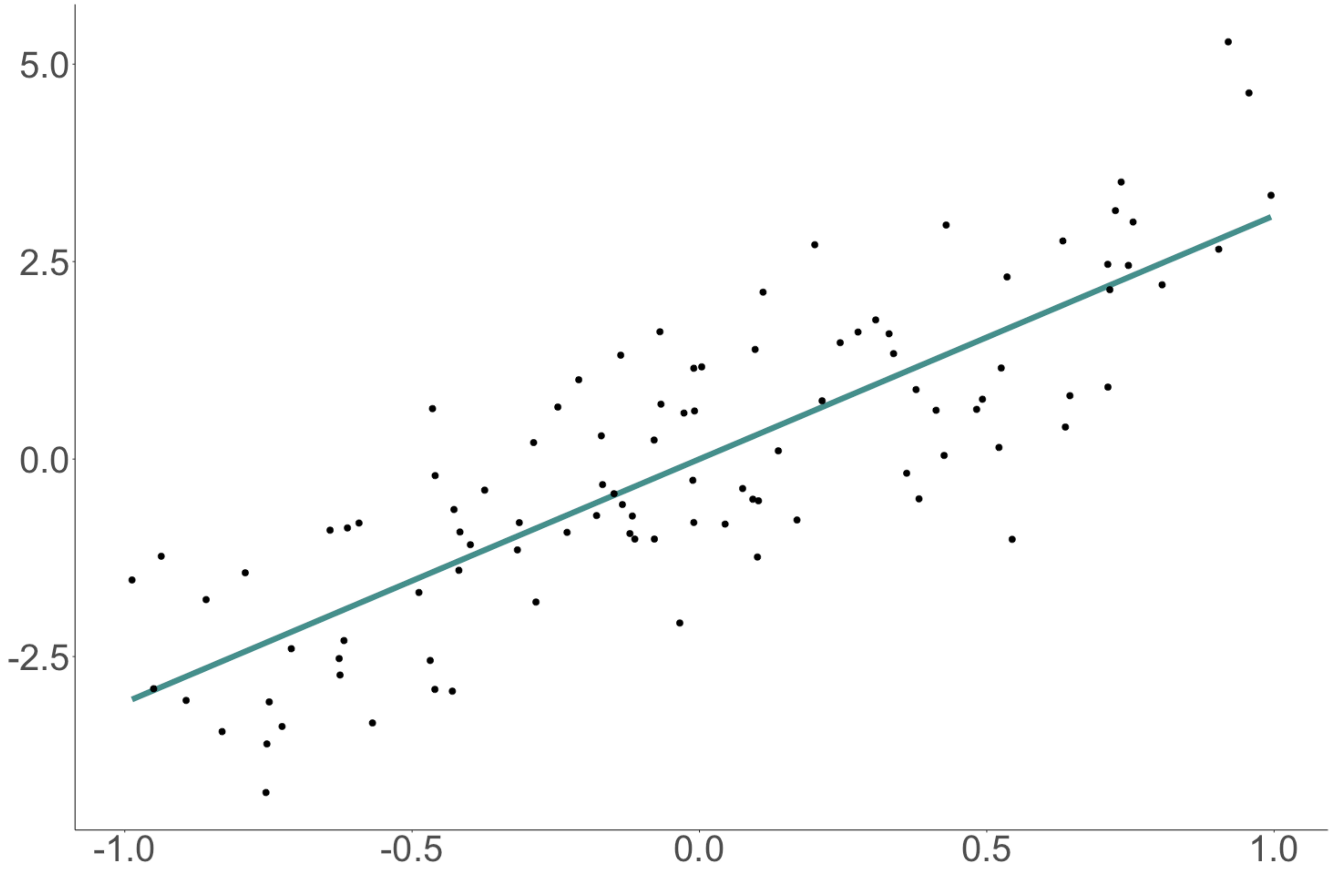
As the function Err is continuous, derivable, and convex, this minimisation problem is solved by cancelling its derivative:

$$\frac{\partial Err(\beta)}{\partial \beta} = \frac{\partial\left(\sum_{i=1}^{n}(Y_i - X_i\beta)^2\right)}{\partial \beta} = -\sum_{i=1}^{n} 2X_i(Y_i - X_i\beta) = 0$$

Therefore, the Ordinary Least Squares estimator is $\hat{\beta}^{OLS} = \dfrac{\sum_{i=1}^{n} X_i Y_i}{\sum_{i=1}^{n} X_i^2}$

Example

$X_i \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1) \qquad \varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1) \qquad \beta^\star = 3 \qquad n = 100 \qquad \hat{\beta}^{\text{OLS}} = 3.08$



Ordinary Least Squares distribution

Assumption the normality of Y: $Y_i | X_i \sim \mathcal{N}\left(X_i\beta, \sigma^2\right)$, the distribution of the ordinary least squares is

$$\hat{\beta}^{OLS} | X_1, \ldots X_n \sim \mathcal{N}\left(\beta, \frac{\sigma^2}{\sum_{i=1}^n X_i^2}\right)$$

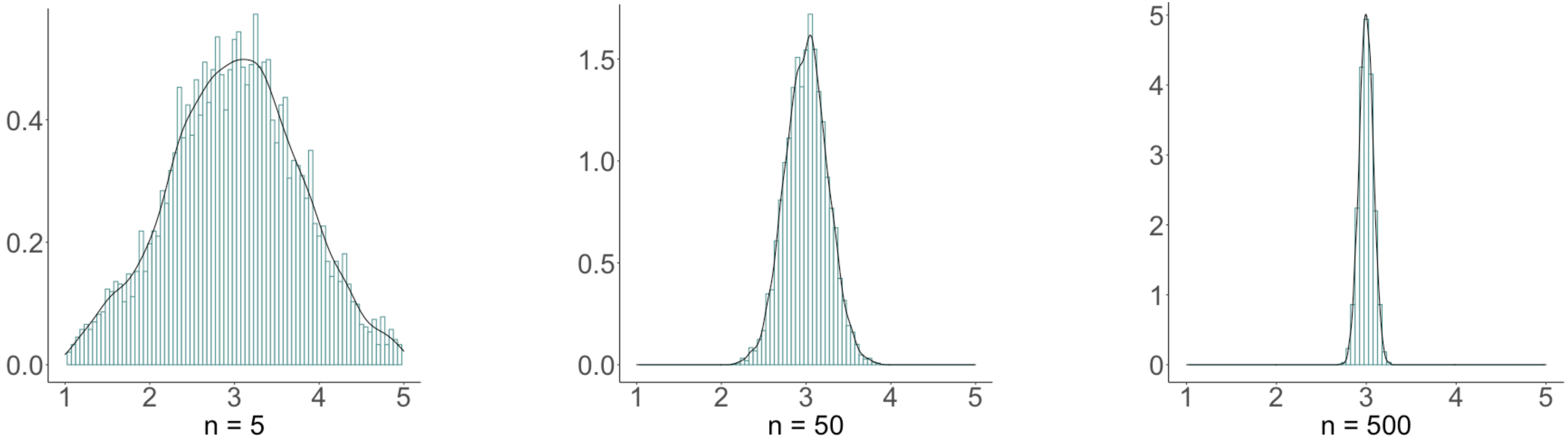
Proof:

Recalling that if $Z_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Z_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ are two independent random variables that are normally distributed then $a_1 Z_1 + a_2 Z_2 \sim \mathcal{N}(\mu_1 + \mu_2, a_1^2\sigma_1^2 + a_2^2\sigma_2^2)$, we get that

$\displaystyle\sum_{i=1}^n X_i Y_i | X_1, \ldots X_n \sim \mathcal{N}\left(\sum_{i=1}^n X_i X_i \beta,\ \sigma^2 \sum_{i=1}^n X_i^2\right)$ and thus as $\hat{\beta}^{OLS} = \dfrac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2}$,

$$\hat{\beta}^{OLS} | X_1, \ldots X_n \sim \mathcal{N}\left(\beta, \frac{\sigma^2}{\sum_{i=1}^n X_i^2}\right)$$

Ordinary Least Squares distribution



Multivariate linear regression

Formulation

Let $(Y_i, X_{i1}, \dots X_{ip})_{i=1,\dots,n}$ be n observations independent and identically distributed of $p+1$ reals random variables $Y, X_1, \dots, X_p$

Assumptions

$Y_i = X_{i,1}\beta_1^\star + X_{i,2}\beta_2^\star + \dots + X_{i,p}\beta_p^\star + \varepsilon_i$ where the processus $(\varepsilon_i)_i$ is a white noise

Using the matrix notations $Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$, $\beta^\star = \begin{bmatrix} \beta_1^\star \\ \vdots \\ \beta_p^\star \end{bmatrix}$, $\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$ and $X = \begin{bmatrix} X_{1,1} \dots X_{1,p} \\ \vdots \; X_{i,j} \; \vdots \\ X_{n,1} \dots X_{n,p} \end{bmatrix} \in \mathscr{M}_{n\times p}(\mathbb{R})$

the design matrix the assumption can be rewritten

$$Y = X\beta^\star + \varepsilon$$

The space of models is now $\mathscr{F} = \left\{\beta \,|\, \beta \in \mathbb{R}^p\right\}$ and we still consider the quadric loss function

Ordinary Least Squares

The Ordinary Least Squares (OLS) estimator minimises the quadratic error computed over the sample $(Y_i, X_i)_{i=1,\ldots,n}$:

$$\hat{\beta}^{OLS} \in \arg\min_{\beta \in \mathbb{R}} \mathrm{Err}(\beta) \quad \text{with} \quad \mathrm{Err}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i\beta)^2$$

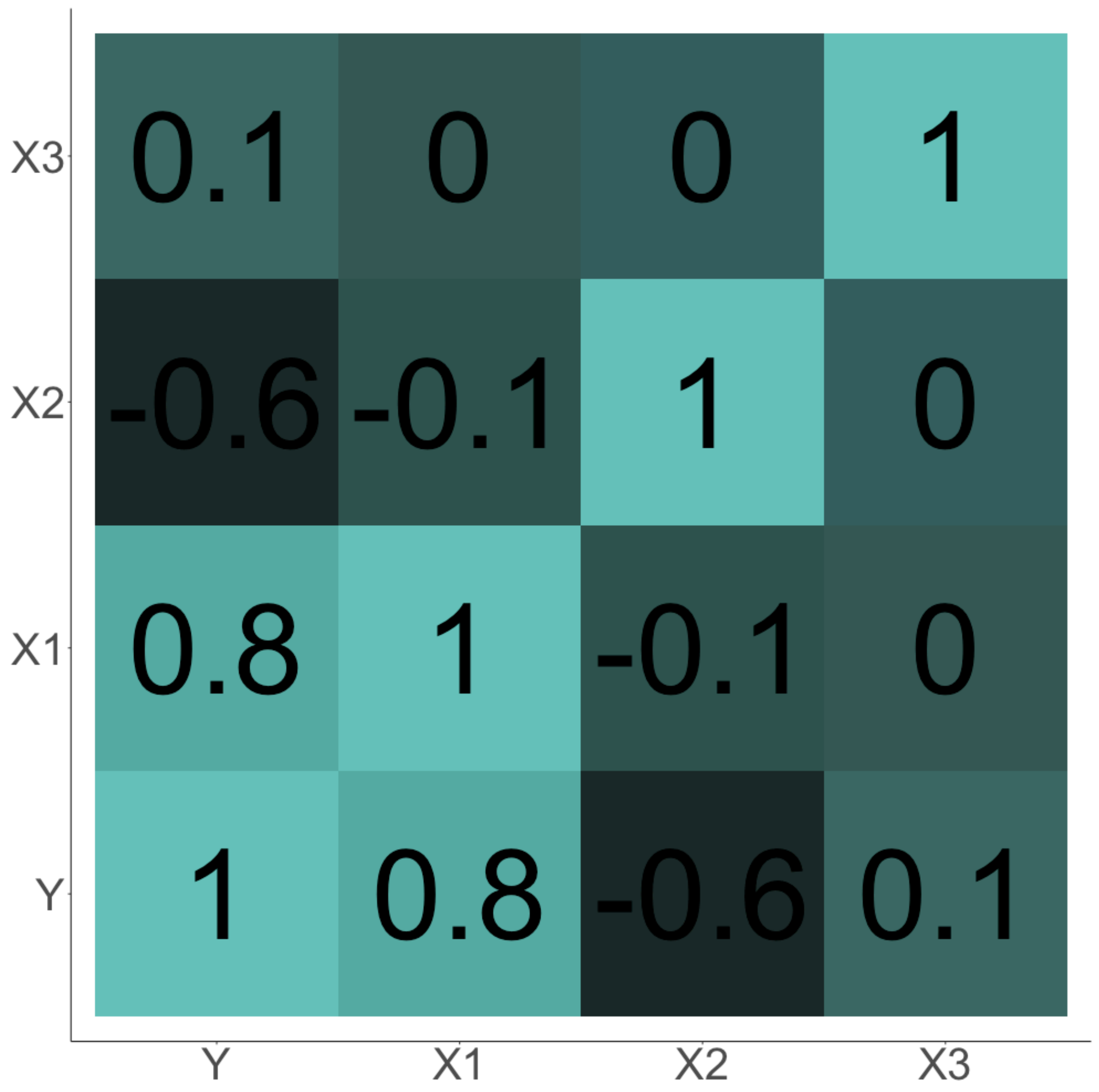
As the function Err is continuous, derivable, and convex, this minimisation problem is solved by cancelling its derivative:

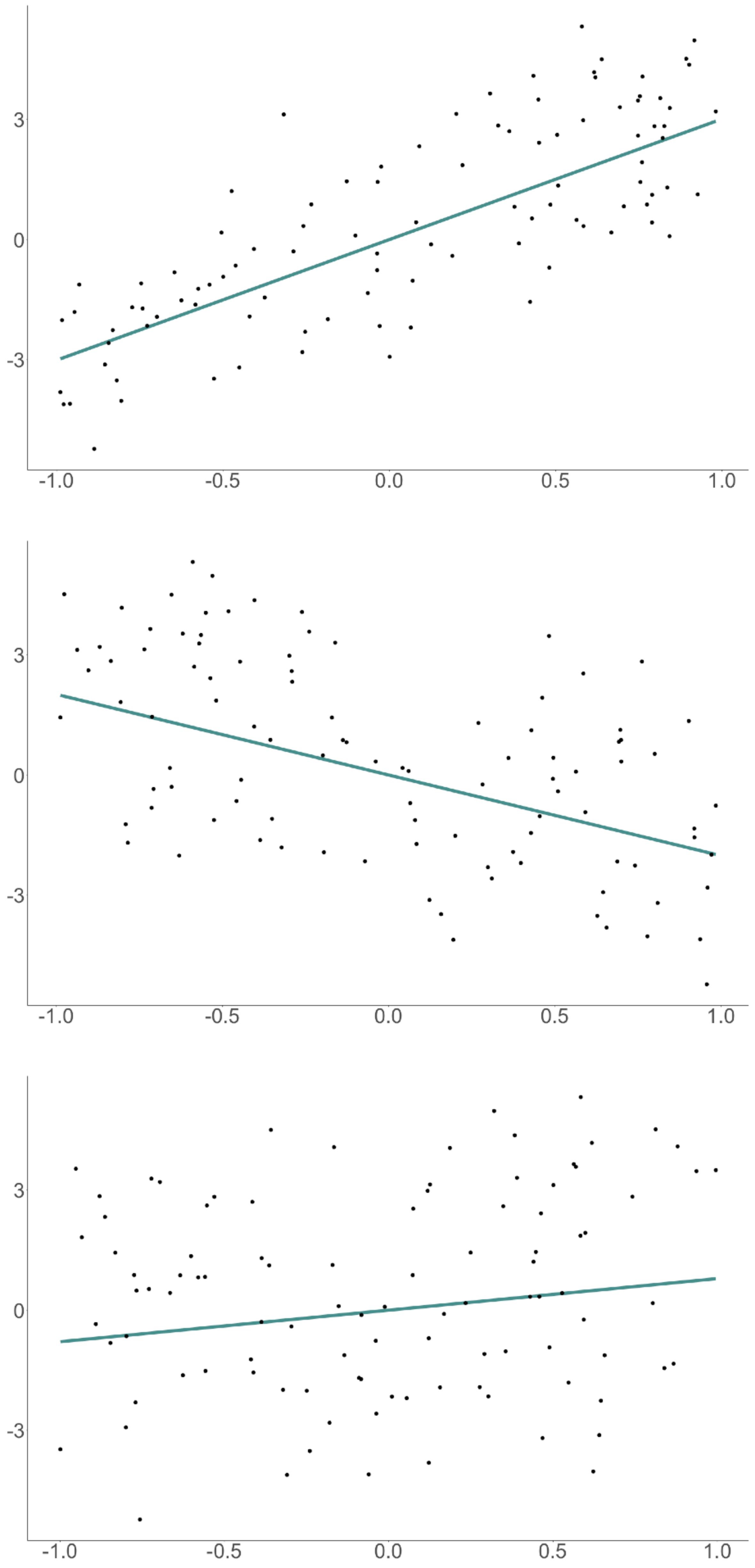
$$\frac{\partial Err(\beta)}{\partial \beta} = \frac{\partial\left(\sum_{i=1}^{n}(Y_i - X_i\beta)^2\right)}{\partial \beta} = -\sum_{i=1}^{n} 2X_i^{\mathrm{T}}(Y_i - X_i\beta) = 0$$

Therefore, the Ordinary Least Squares estimator is $\widehat{\beta}^{OLS} = \left(XX^{\mathrm{T}}\right)^{-1}X^{\mathrm{T}}Y$

Example

$$X_{i1} \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)$$
$$X_{i2} \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)$$
$$X_{i3} \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)$$
$$\varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$$
$$\beta^\star = [3, -2, 1]$$
$$n = 100$$
$$\hat{\beta}^{\text{OLS}} = [3.02, -2.15, 1.18]$$





Ordinary Least Squares distribution

Assumption the normality of Y: $Y_i \mid X_i \sim \mathcal{N}\left(X_i\beta^\star, \sigma^2\right)$, the distribution of the ordinary least squares is

$$\hat{\beta}^{OLS} \mid X \sim \mathcal{N}\left(\beta^\star,\ \left(X^\mathrm{T}X\right)^{-1}\sigma^2\right)$$

Proof:

$$\mathbb{E}\left[\widehat{\beta}^{OLS}\right] \quad = \mathbb{E}\left[\left(XX^\mathrm{T}\right)^{-1}X^\mathrm{T}Y\right] \quad = \mathbb{E}\left[\left(XX^\mathrm{T}\right)^{-1}X^\mathrm{T}X\beta^\star + \varepsilon\right] \quad = \beta^\star$$

$$\mathrm{Var}\left(\widehat{\beta}^{OLS}\right) \quad = \mathrm{Var}\left(\left(XX^\mathrm{T}\right)^{-1}X^\mathrm{T}Y\right) \quad = \left(XX^\mathrm{T}\right)^{-1}X^\mathrm{T}\mathrm{Var}(Y)X\left(X^\mathrm{T}X\right)^{-1} \quad = \left(X^\mathrm{T}X\right)^{-1}\sigma^2$$

$\square$

OLS and likelihood

The likelihood of β given n observations ($\sim$ probability of observing these observations if they are well distributed according to the model defined by β) in the case where the noise is Gaussian is

$$L(X, \beta, \sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{\|Y - X\beta\|^2}{2\sigma^2} \right)$$

Maximising the likelihood is equivalent to minimising the quatradic error $\|Y - X\beta\|^2$ so the maximum likelihood estimator equals to the ordinary least squares estimator

When the data no longer respect the hypothesis of independence or constant variance:
$Y \sim \mathcal{N}\left(X\beta^{\star}, \mathbf{V}\sigma^2\right)$ with $\mathbf{V}$ a positive definite matrix, the likelihood is

$$L(X, \beta, \sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2 |\mathbf{V}|}} \exp\left(- \frac{(Y - X\beta)^{\mathrm{T}}\mathbf{V}(Y - X\beta)}{2\sigma^2} \right)$$

and both estimators are not equal anymore

Generalised linear model

Formulation

Let $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$ be n observations independent and identically distributed of $p+1$ reals random variables $Y, X_1, \ldots, X_p$

Assumptions

There exists a link function g monotonic and regular (for example the identity or log functions) relating the expected value of Y to the predictor variables via a structure such as

$$g\left(\mathbb{E}[Y]\right) = X\beta^\star$$

Knowing X, observations follows an exponential distribution: there exist three functions a, b and c, a two parameters ϕ and θ such that the density of $Y \mid X$ is

$$f_{Y|X}(y) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right)$$

Exponential family

	$\text{Gaussian}(\mu, \sigma^2)$	$\text{Poisson}(\lambda)$	$\text{Binomiale}(n, p)$	$\text{Gamma}(\alpha, \beta)$
θ	μ	$\log \lambda$	$\log \frac{p}{1-p}$	$-\frac{\alpha}{\beta}$
ϕ	σ^2	1	1	$\frac{1}{\alpha}$
$a(\phi)$	ϕ	ϕ	ϕ	ϕ
$b(\theta)$	$\frac{\theta^2}{2}$	$\exp \theta$	$n \log(1 + \exp \theta)$	$-\log(-\theta)$
$c(y, \theta)$	$\frac{1}{2}\left(\frac{y^2}{\phi} + \log 2\pi\phi\right)$	$-\log y!$	$\log \binom{n}{y}$	$\frac{1}{\phi} \log \frac{y}{\phi} - \log\left(y\Gamma\left(\frac{1}{\phi}\right)\right)$
$f(y)$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$	$\frac{\lambda^y \exp(-y)}{y!}$	$\binom{n}{y} p^y (1-p)^{n-y}$	$\frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y)$

Use case examples:

- Modelling electrical power consumption: Gaussian
- Modelling arrivals and departures at electric vehicle charging stations: Poisson

Likelihood and IRLS

Si la variable aléatoire Y est dans la famille exponentielle alors

$$\mathbb{E}[Y] = b'(\theta) \quad \text{and} \quad \mathrm{Var}\big(Y\big) = b''(\theta)a(\phi)$$

As $g\big(\mathbb{E}[Y]\big) = X\beta$, the likelihood of β and the n observations $(Y_i, X_{i1}, \dots X_{ip})_{i=1,\dots,n}$ is

$$L(X, \beta) = \prod_{i=1}^{n} f_{a_i,b_i,c_i,\theta_i,\phi_i}(Y_i)$$

As it is then difficult to maximise the likelihood exactly, Newton's method (a numerical method with a step for calculating the gradient and the Hessian of the log-likelihood) is used to estimate iteratively β

At each iteration, we need to solve a weighted least squares problem - see Algorithm IRLS : iteratively re-weighted least square (cf. Wood) for further details

Online approaches

Online Linear Regression

Initialisation:

- $\hat{\beta}_0$ estimated with a sample $(Y_i, X_{i1}, \dots X_{ip})_{i=1,\dots,n}$

$\hat{\beta}_0 \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \|Y - X\beta\|^2 = \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{p} x_{i,j}\beta_j \right)$ and $H_1 = X^T X$

For $k = 2, \dots$

- Observe a new batch $(Y_t, X_{t1}, \dots X_{tp})_{t=t_k,\dots,t_{k+1}-1} = (\mathbf{Y}_k, \mathbf{X}_k)$
- Update the estimator $\hat{\beta}_k = \hat{\beta}_{k-1} + (H_k)^{-1} \mathbf{X}_k^T (\mathbf{Y}_k - \mathbf{X}_k \beta_{k-1})$ with $H_k = H_{k-1} + \mathbf{X}_k^T \mathbf{X}_k$

$$\begin{aligned} \hat{\beta}_k &\in \arg\min_{\beta \in \mathbb{R}^p} \sum_{l=1}^{k} \|\mathbf{Y}_k - \mathbf{X}_k \beta\|^2 \\ &\in \arg\min_{\beta \in \mathbb{R}^p} \sum_{s=1}^{t_k} (Y_i - X_i \beta)^2 \end{aligned}$$

as soon as batches have equal size

Weighted Linear Regression

How to give more « importance » to recent data ?

$$\hat{\beta}_t \in \arg\min_{\beta \in \mathbb{R}} \sum_{s=1}^{t} \omega_s (Y_s - X_s\beta)^2 \quad \text{with} \quad \omega_s = \mu^{t-s} \text{ and } \mu \in]0,1[\text{ or } \omega_s = \exp\big(-\eta(t-s)\big)$$

As the function to minimise is continuous, derivable, and convex, this minimisation problem is solved by cancelling its derivative:

$$\frac{\partial\left(\sum_{s=1}^{t} \omega_s (Y_s - X_s\beta)^2\right)}{\partial \beta} = -\sum_{s=1}^{t} 2\omega_s X_s^{\mathrm{T}} (Y_s - X_s\beta) = 0$$

$$\hat{\beta}_t = \left(\tilde{X}^{\mathrm{T}}\tilde{X}\right)^{-1}\tilde{X}^{\mathrm{T}}\tilde{Y} \text{ with } \tilde{X}_{sj} = \omega_s X_{sj} \text{ and } \tilde{Y}_s = \omega_s Y_s$$

→ New challenge: tuning μ

Interpretation with an example: with $\mu = 0.95, \quad \mu^{200} \approx 3.10^{-5}$ so after 200 time steps, observations can be considered as totally forgotten

Weighted Online Linear Regression

Assumption:

For time step $t_1 = 1, t_2, t_3, \ldots, t_k, \ldots,$ we get access to a sample $(Y_t, X_{t1}, \ldots X_{tp})_{t=t_k,\ldots,t_{k+1}-1} = (Y_k, X_k)$ which is big enough to ensure that $X_k^{\mathrm{T}} X_k$ is inversible

Initialisation:

- $\hat{\beta}_1 = \left(X_1 X_1^{\mathrm{T}}\right)^{-1} X_1^{\mathrm{T}} Y_1$ and $H_1 = X_1^{\mathrm{T}} X_1$

For $k = 2, \ldots$

- Observe $(Y_t, X_{t1}, \ldots X_{tp})_{t=t_k,\ldots,t_{k+1}-1} = (Y_k, X_k)$
- Update the estimator $\hat{\beta}_k = \hat{\beta}_{k-1} + \left(H_k\right)^{-1} \mathbf{X}_k^{\mathrm{T}} \left(\mathbf{Y}_k - \mathbf{X}_k \beta_{k-1}\right)$ with $H_k = \mu H_{k-1} + \mathbf{X}_k^{\mathrm{T}} \mathbf{X}_k$

$$\hat{\beta}_k \in \arg\min_{\beta \in \mathbb{R}^p} \sum_{l=1}^{k} \mu^{k-l} \| Y_k - X_k \beta \|^2$$

Penalised Regression

Bias - Variance trade-off

The ordinary least squares method allows to estimate a model $\hat{f}(X) = X\hat{\beta}$ from a sample $(Y_i, X_i)_{i=1,\dots,n}$
Under the linear model assumption $Y = X\beta^\star + \varepsilon$, the estimator $\hat{\beta}$ is unbiased with minimum variance among unbiased estimators (Gauss-Markov Theorem)

For a new set of explanatory variables X_{new} it is then possible to predict Y_{new} with $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$
The quadratic error of this prediction can be decomposed into an irreducible error σ^2, a term related to the variance of the estimator $X_{\text{new}}\text{Var}\left(\hat{\beta}\right)X_{\text{new}}$ and the squared bias of the estimator $\left(\beta^\star - \mathbb{E}(\hat{\beta})\right)^2$:

$$
\begin{aligned}
\mathbb{E}\left[\left(Y_{\text{new}} - \hat{Y}_{\text{new}}\right)^2\right] &= \mathbb{E}\left[(X_{\text{new}}\beta^\star + \varepsilon_{\text{new}} - X_{\text{new}}\hat{\beta})^2\right] \\
&= \sigma^2 + \mathbb{E}\left[\left(X_{\text{new}}(\beta^\star - \hat{\beta})\right)^2\right] \\
&= \sigma^2 + X_{\text{new}}\text{Var}\left(\hat{\beta}\right)X_{\text{new}}^{\text{T}} + \left(\beta^\star - X_{\text{new}}\mathbb{E}\left(\hat{\beta}\right)\right)^2 X_{\text{new}}^{\text{T}}
\end{aligned}
$$

Bias - Variance trade-off - Illustration

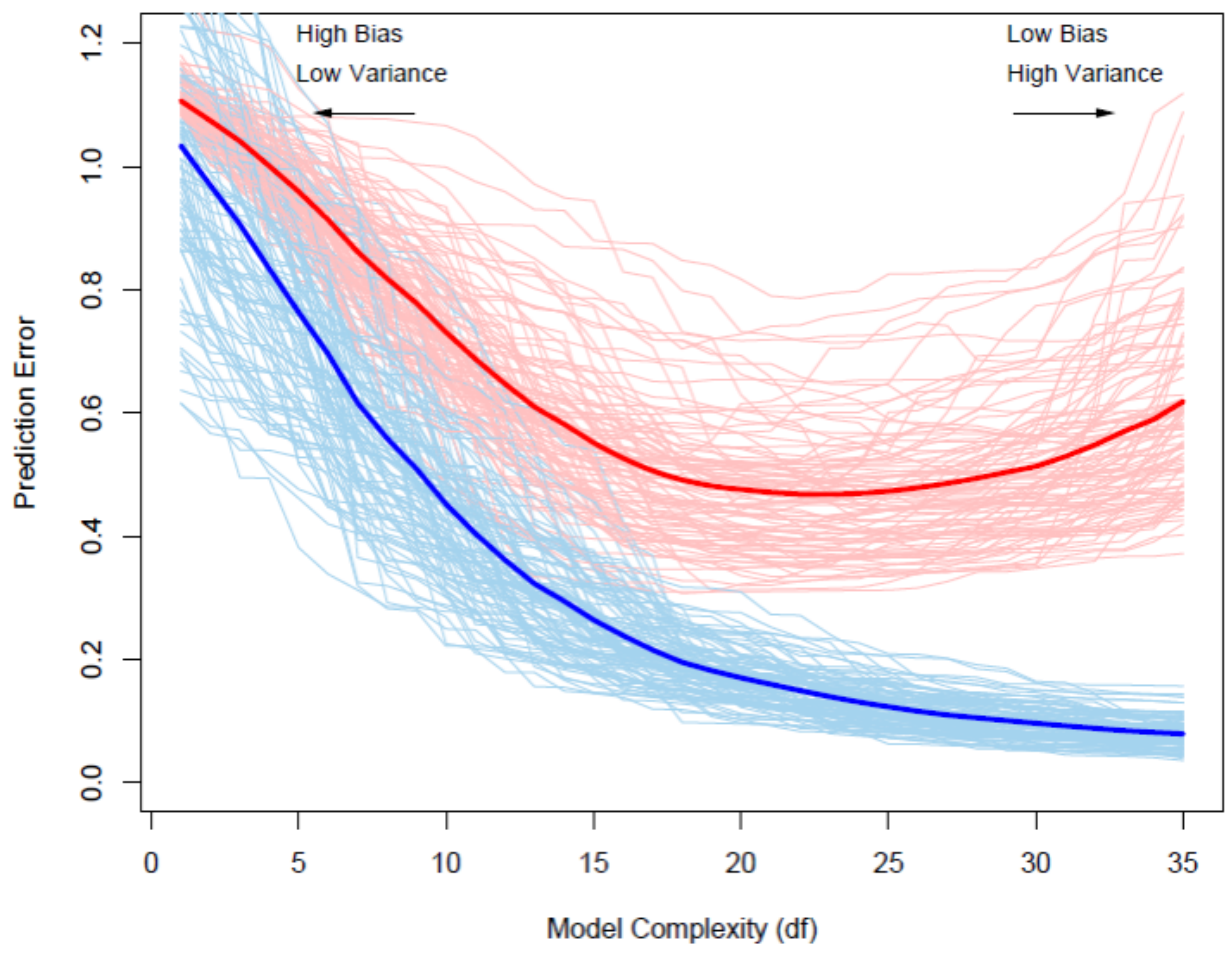


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for* 100 *training sets of size* 50 *each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\text{E}[\overline{\text{err}}]$.

Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer series in statistics - 2001

Ridge regression

Motivation

Example:

- Univariate linear model: $Y = X_1\beta_1^\star + \varepsilon$
- Adding of a second explanatory variable: $X_2 = X_1 + \text{noise}$

$$\forall a \in \mathbb{R}, \quad \beta_a = \begin{bmatrix} (a+1)\beta_1^\star \\ -a\beta_1^\star \end{bmatrix}$$ is an unbiased estimator

$$\mathbb{E}[\hat{Y}] = \mathbb{E}\left[(a+1)X_1\beta_1^\star - aX_2\beta_1^\star\right] = X_1\beta_1^\star = \mathbb{E}\left[(a+1)X_1\beta_1^\star - aX_1\beta_1^\star - aX_1\text{noise}\right] = X_1\beta_1^\star = \mathbb{E}[Y]$$

of variance

$$\text{Var}(\hat{Y}) = \mathbb{E}\left[\left((a+1)X_1\beta_1^\star + aX_1\beta_1^\star + aX_1\text{noise} - X_1\beta_1^\star\right)^2\right] = a^2\beta_1^2\text{Var}(\text{ noise })$$

Motivation

$$X_{i1} \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)$$

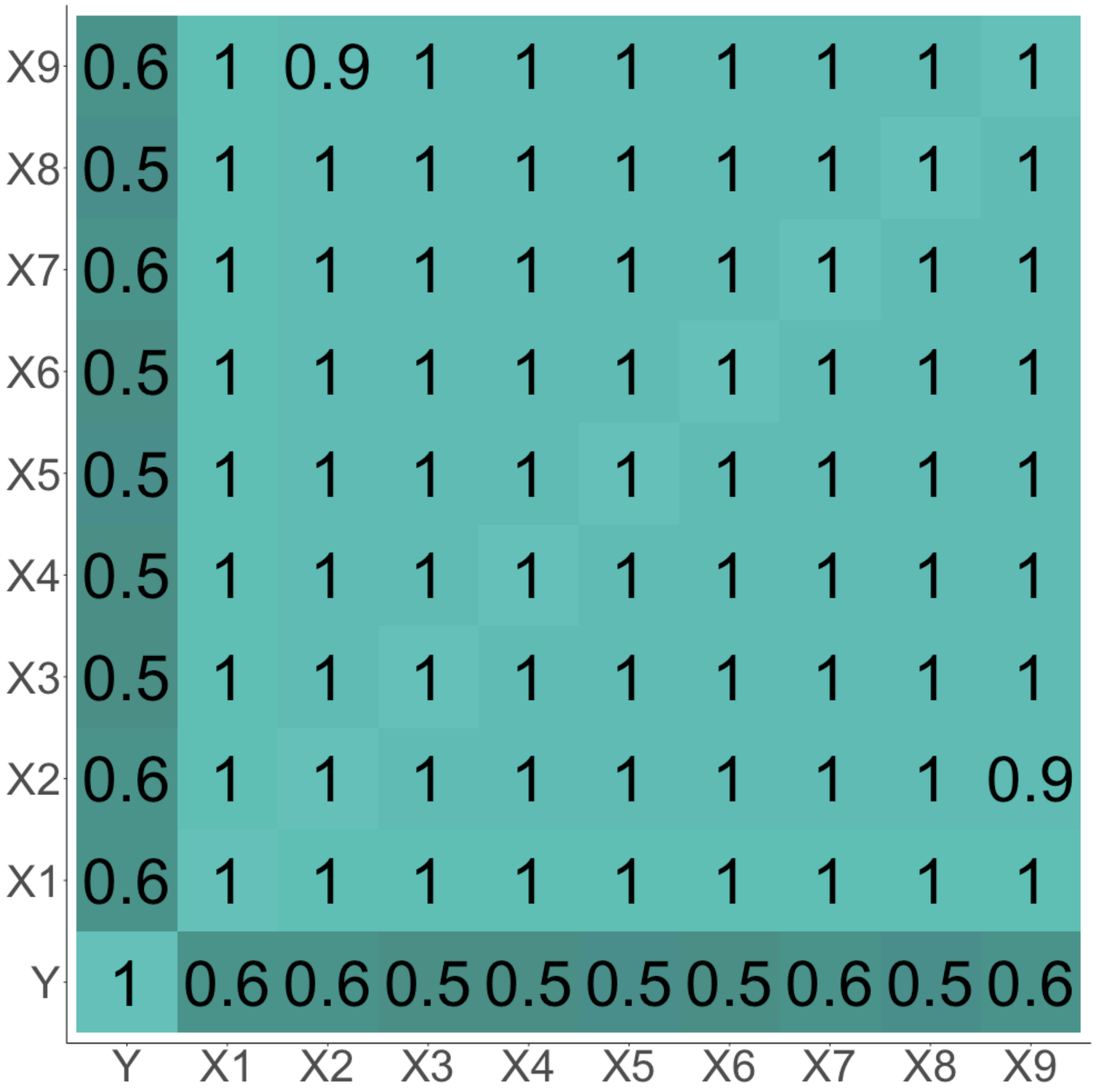
$$X_{i2} = X_1 + \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)/5$$

$$\dots$$

$$X_{i9} = X_1 + \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)/5$$

$$\varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$$

$$\beta^\star = \begin{bmatrix} -1 \\ 1 \\ -0.5 \\ 0.5 \\ -0.2 \\ 0.2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

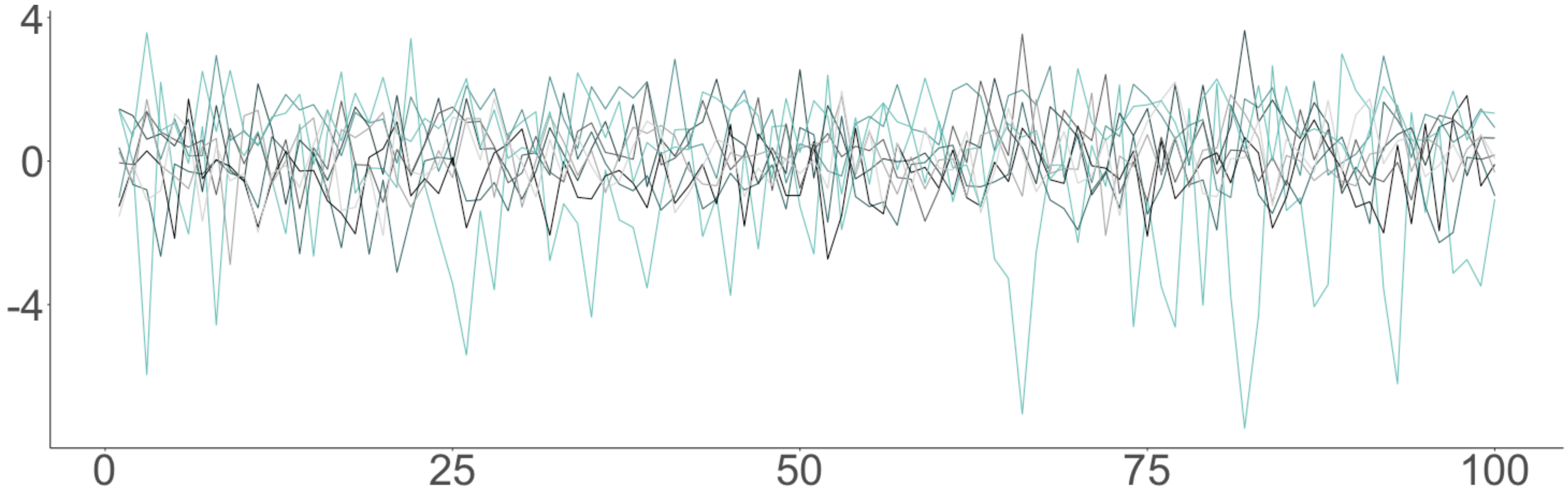


Motivation

For $k = 1,\ldots,100$

- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^{OLS,k} = \left(XX^{\mathrm{T}}\right)^{-1} X^{\mathrm{T}} Y$

$$\beta^{\star} = \begin{bmatrix} -1 \\ 1 \\ -0.5 \\ 0.5 \\ -0.2 \\ 0.2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



Penalisation

If the coefficients of the estimator β are not constraints

- they may explode
- the variance of estimator may be high

Indeed, if the explanatory variables are correlated, the unicity of the solution is not obvious (a high coefficient for a variable can be cancelled by a high negative coefficient on another correlated variable)

→ Need to impose a constraint on the value of the coefficients:

$$\arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||^2 \quad \text{with} \quad ||\beta||^2 \leq \text{constant}$$

This problem is equivalent to solve

$$\arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||^2 + \lambda ||\beta||^2 = \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{p} X_{i,j}\beta_j + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

Ridge estimator distribution

As the function $\beta \mapsto \|Y - X\beta\|^2 + \lambda\|\beta\|^2$ is continuous, derivable, and convex so the minimisation problem is solved by cancelling its derivative

$$\frac{\partial\left(\|Y - X\beta\|^2 + \lambda\|\beta\|^2\right)}{\partial\beta} = 2X^{\mathrm{T}}\left(Y - X\beta\right) + 2\lambda\beta$$

The Ridge estimator is thus

$$\hat{\beta}_\lambda = \left(X^{\mathrm{T}}X + \lambda\mathbf{I}_p\right)^{-1}X^{\mathrm{T}}Y$$

This estimator is biased

$$\mathbb{E}\left[\hat{\beta}_\lambda\right] = \mathbb{E}\left[\left(X^{\mathrm{T}}X + \lambda\mathbf{I}_p\right)^{-1}X^{\mathrm{T}}\left(X\beta^\star + \varepsilon\right)\right] = \beta^\star - \lambda\left(X^{\mathrm{T}}X + \lambda\mathbf{I}_p\right)^{-1}\beta^\star$$

And its variance satisfies

$$\mathrm{Var}\left(\hat{\beta}_\lambda\right) = \sigma^2\left(X^{\mathrm{T}}X + \lambda\mathbf{I}_p\right)^{-1}X^{\mathrm{T}}X\left(X^{\mathrm{T}}X + \lambda\mathbf{I}_p\right)^{-1}$$

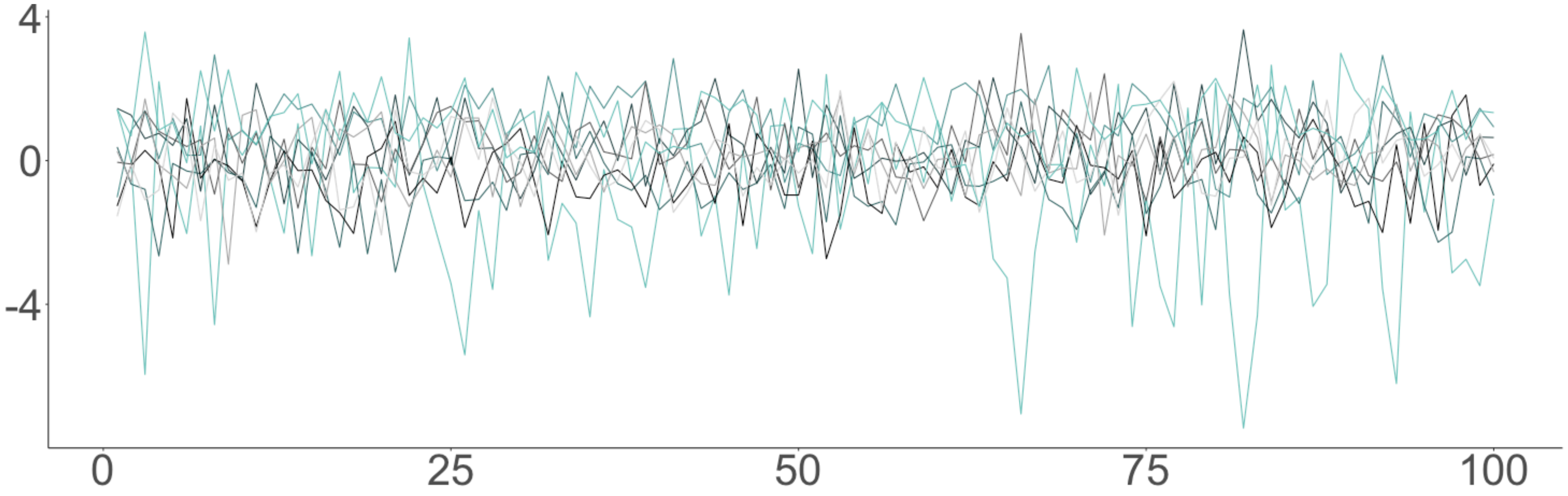
Example

For $k = 1,\ldots,100$

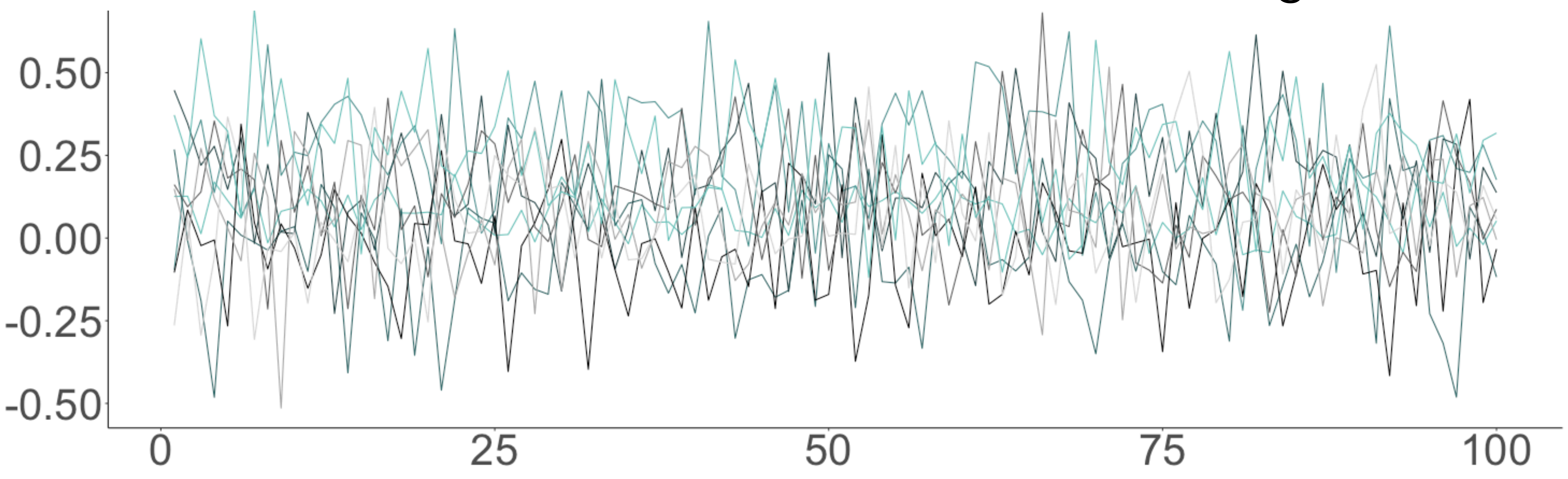
- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^{OLS,k}$ and $\widehat{\beta}^{Ridge,k}$

$$\beta^\star = \begin{bmatrix} -1 \\ 1 \\ -0.5 \\ 0.5 \\ -0.2 \\ 0.2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Ordinary Least Squares estimator



Ridge estimator

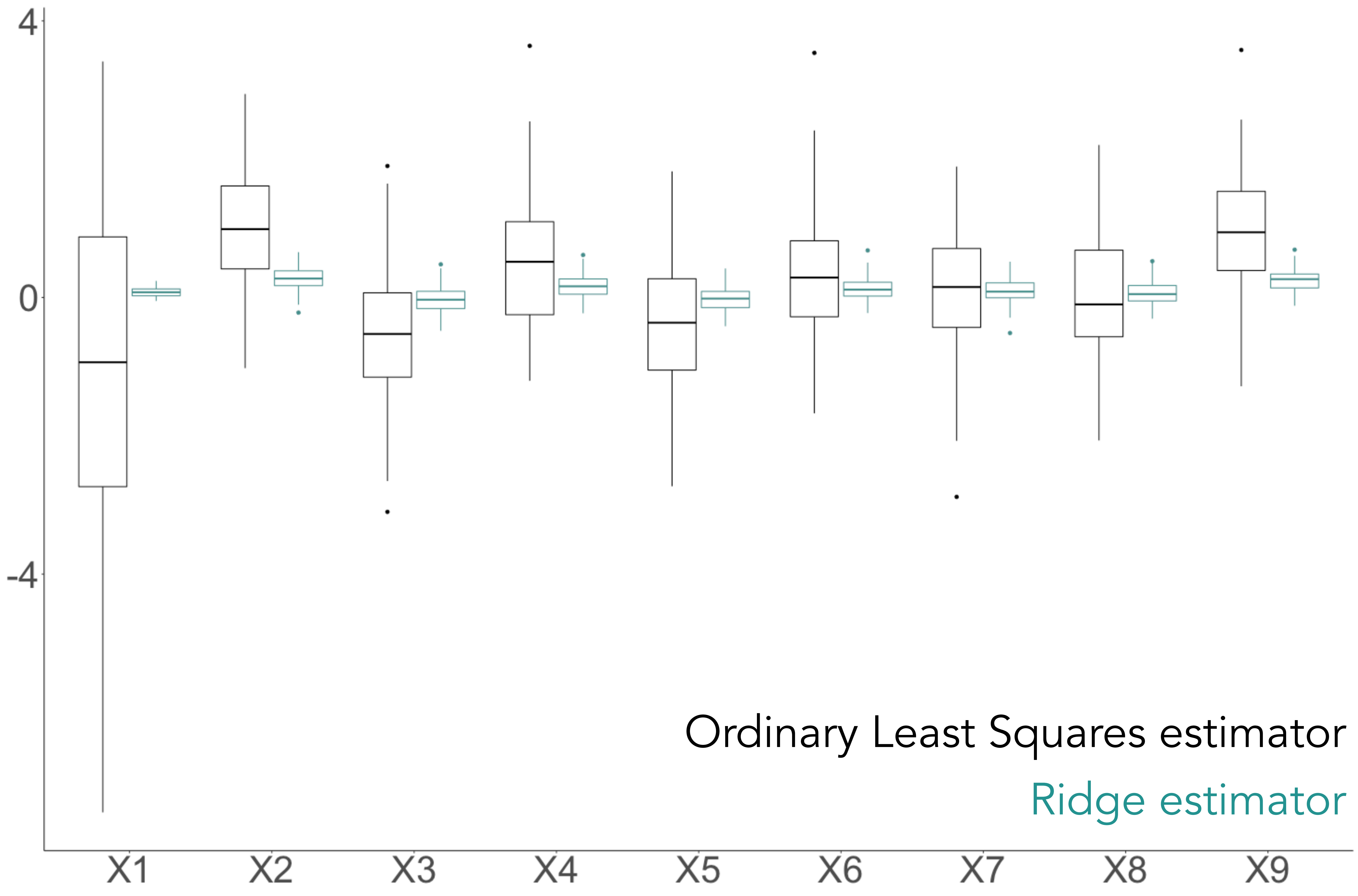


Example

For $k = 1,\ldots,100$

- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^{OLS,k}$ and $\widehat{\beta}^{Ridge,k}$

$$\beta^\star = \begin{bmatrix} -1 \\ 1 \\ -0.5 \\ 0.5 \\ -0.2 \\ 0.2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



Example

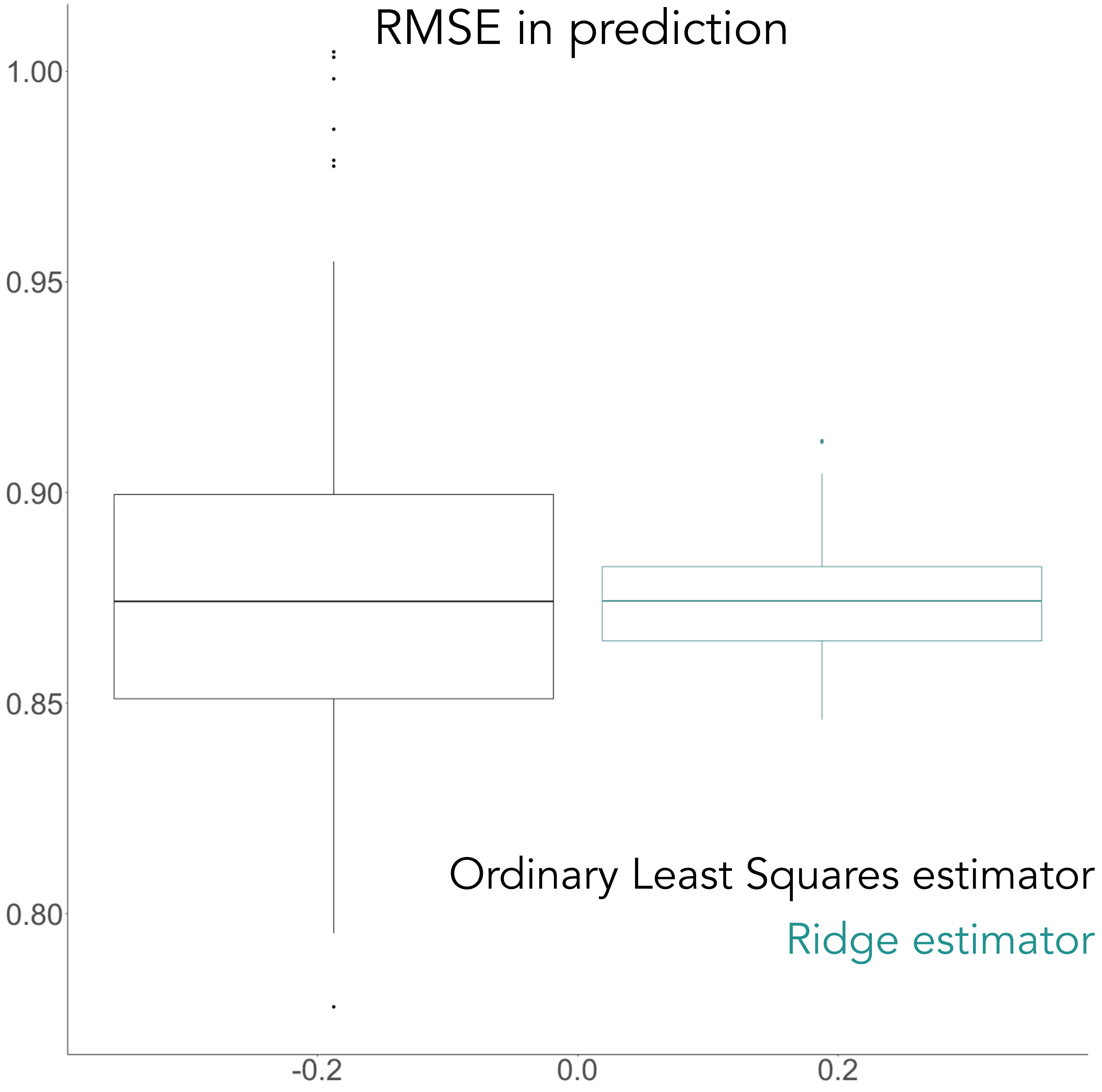
For $k = 1,\ldots,100$

- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^k = \left(XX^{\mathrm{T}}\right)^{-1} X^{\mathrm{T}} Y$

For a new sample $(Y_{\text{new},i}, X_{\text{new},i1}, \ldots X_{\text{new},ip})_{i=1,\ldots,n}$

Compute the Root Mean Squared Error (RMSE) for each $k = 1,\ldots,100$:

$$\sum_{i=1}^{n} \left(\hat{Y}^k_{\text{new,i}} - Y_{\text{new,i}}\right)^2$$



LASSO regression

Motivation and penalisation

LASSO, for Least Absolute Shrinkage and Selection Operator, regression has introduced in a variable selection perspective and under the assumption that $\beta^\star$ is a sparse vector (*i.e.*, lots of its coefficients are zero)

→ Need to impose a constraint on the number of non-zero coefficients

$$\arg\min_{\beta\in\mathbb{R}^p} \|Y - X\beta\|^2 \quad \text{with} \quad \|\beta\|_0 = \sum_{j=1}^{p} \mathbf{1}_{\beta_j\neq 0} \leq \text{constant}$$

But this norm is not continuous and, thus non sub derivative

Therefore, LASSO aims to solve

$$\arg\min_{\beta\in\mathbb{R}^p} \|Y - X\beta\|^2 \quad \text{with } \|\beta\|_1 \leq \text{constant}$$

This problem is equivalent to solve

$$\arg\min_{\beta\in\mathbb{R}^p} \|Y - X\beta\|^2 + \lambda\|\beta\|_1 = \arg\min_{\beta\in\mathbb{R}^p} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{p} X_{i,j}\beta_j + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

Ridge versus LASSO - Illustration

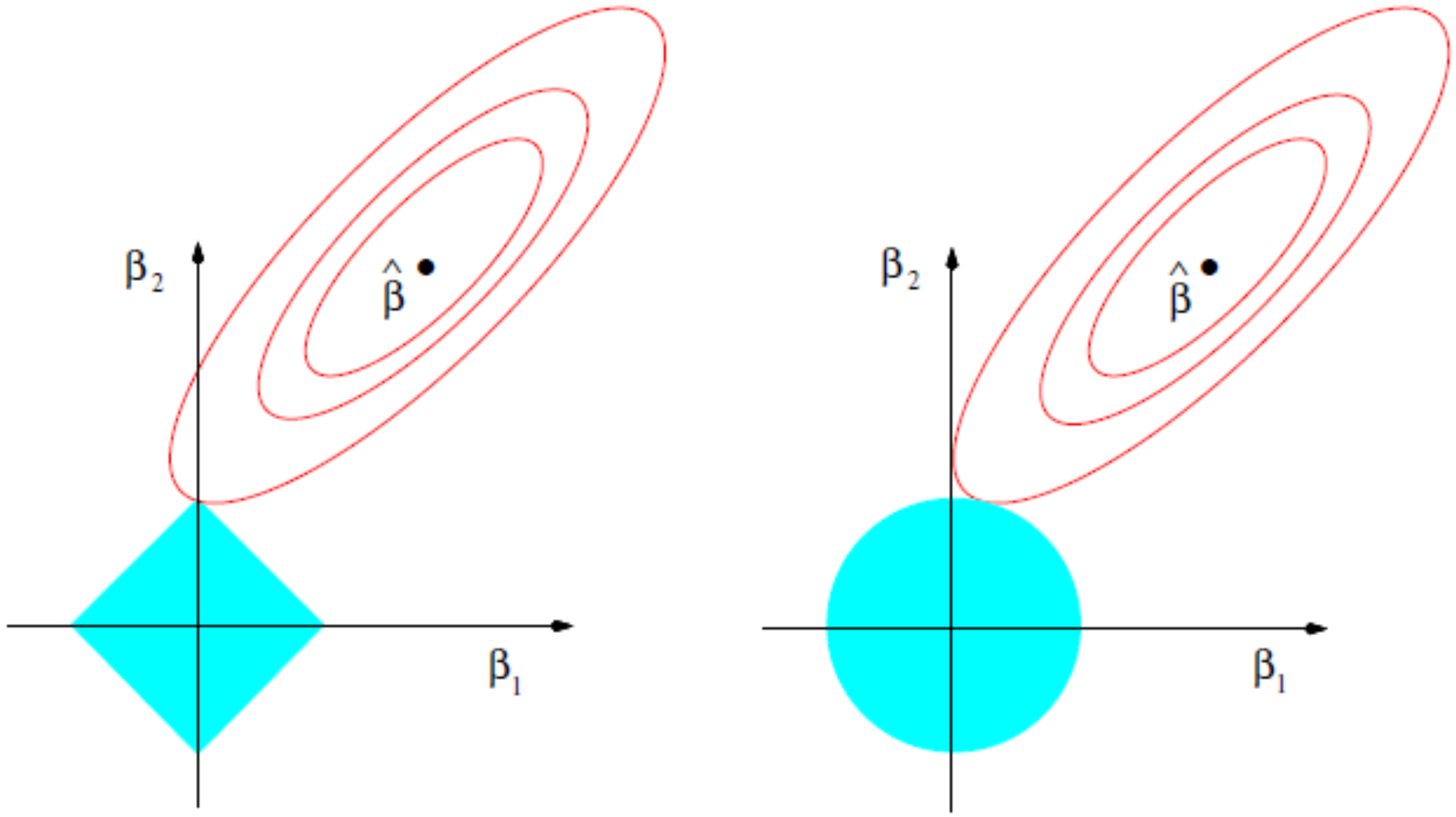


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer series in statistics - 2001

Example

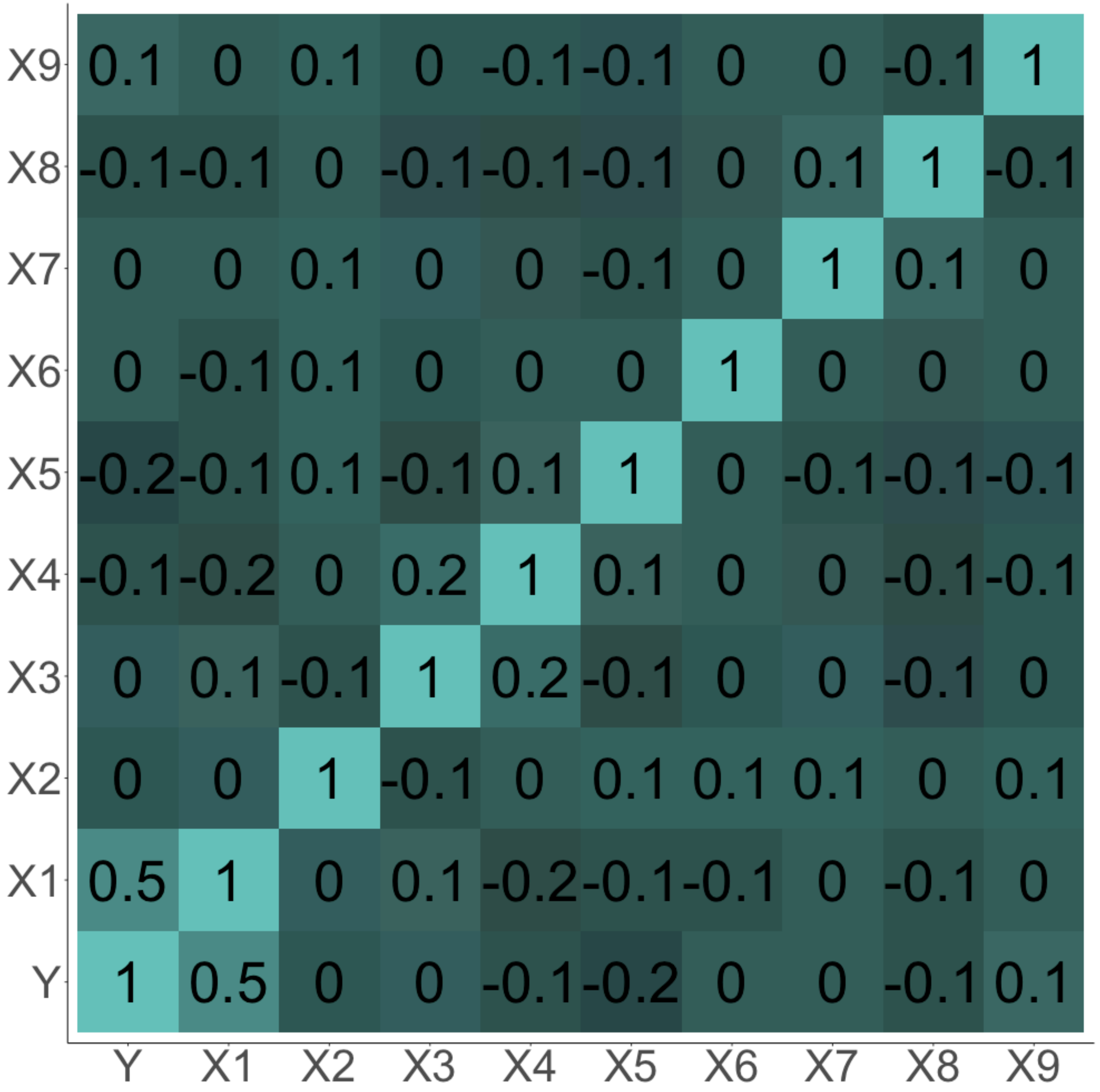
$$X_{i1} \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)$$

$$\dots$$

$$X_{i9} \overset{\text{i.i.d}}{\sim} \mathcal{U}(-1,1)$$

$$\varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$$

$$\beta^\star = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



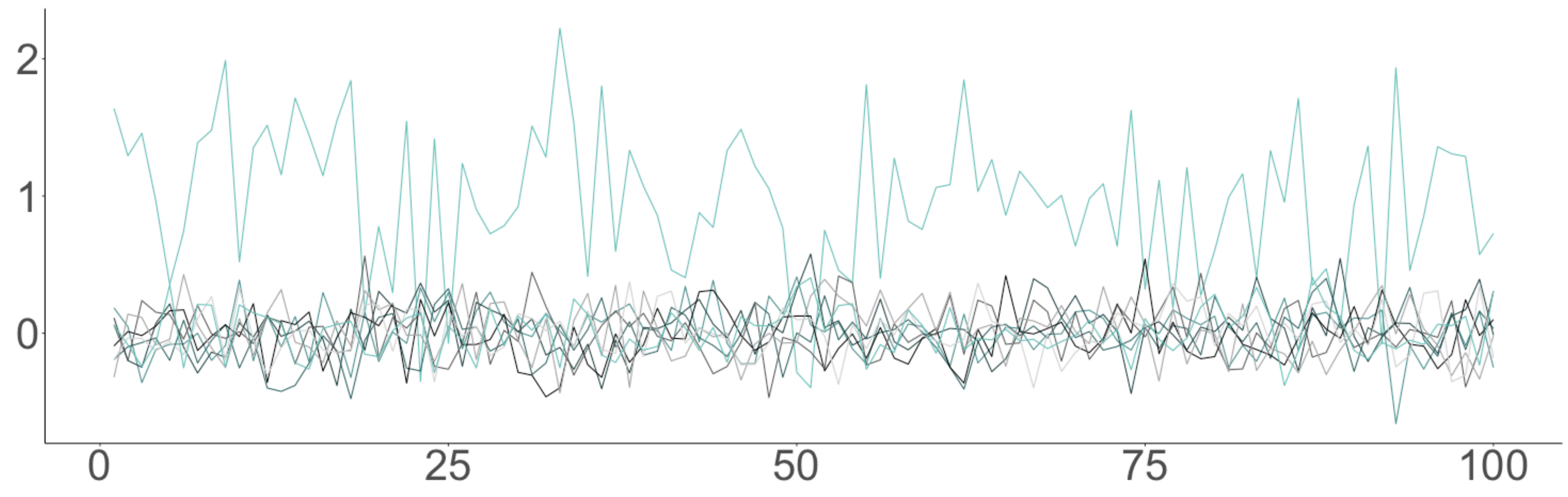
Example

For $k = 1,\ldots,100$

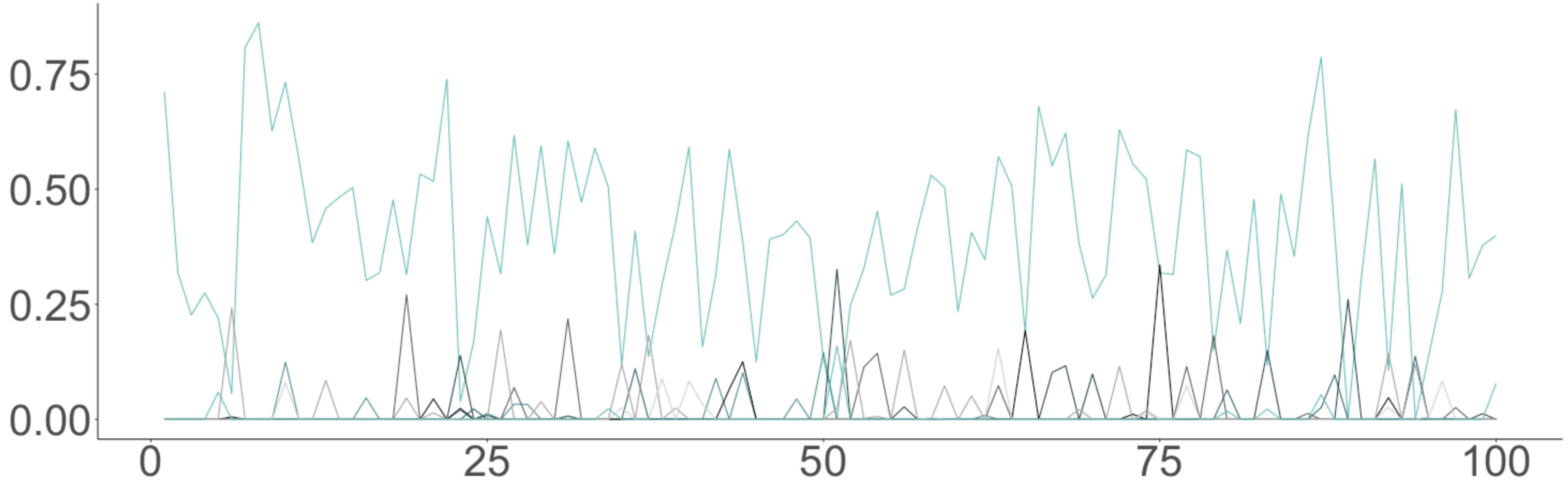
- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^{OLS,k}$ and $\widehat{\beta}^{LASSO,k}$

$$\beta^\star = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Ordinary Least Squares estimator



LASSO estimator

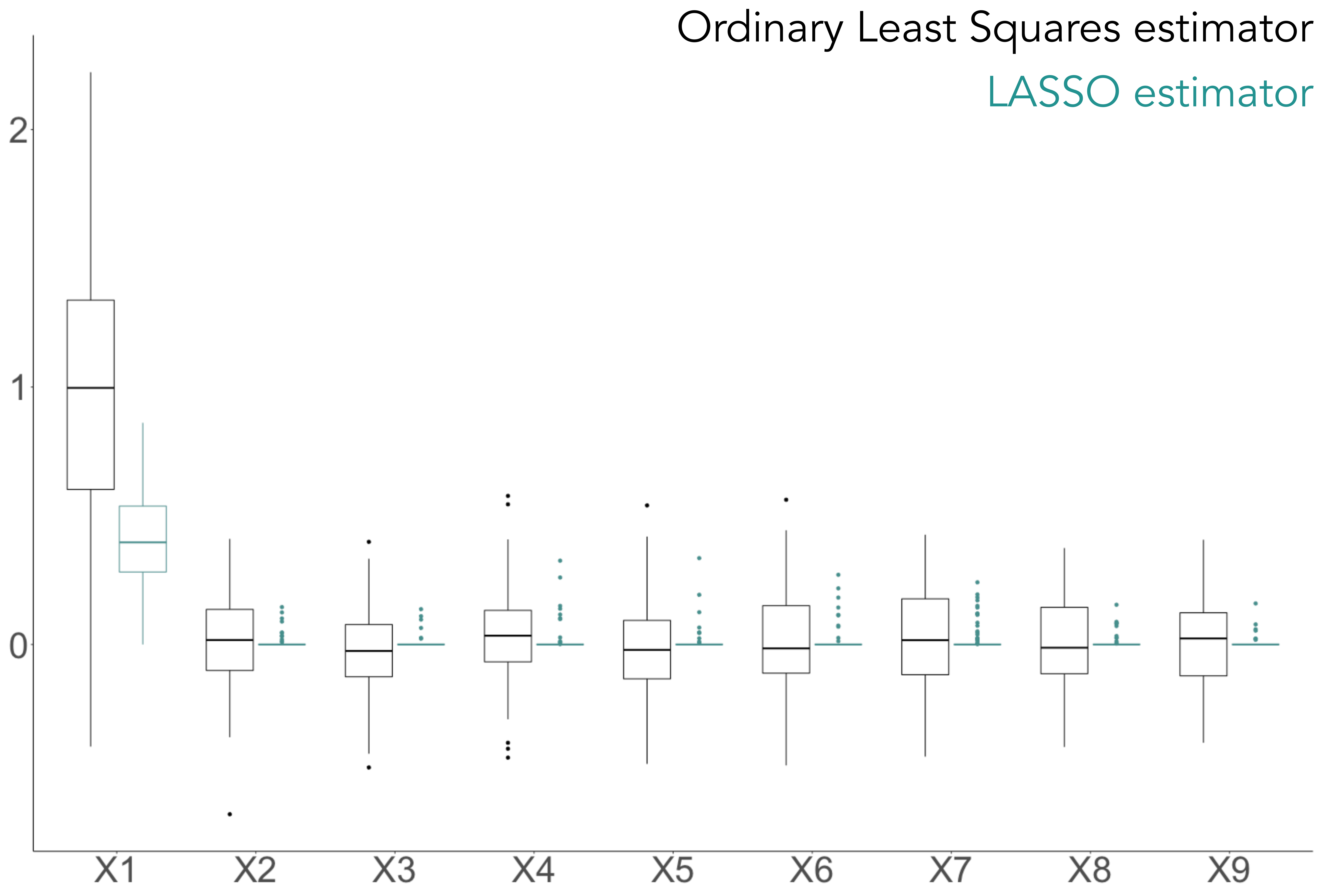


Example

For $k = 1,\ldots,100$

- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^{OLS,k}$ and $\widehat{\beta}^{LASSO,k}$

$$\beta^\star = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Example

For $k = 1,\ldots,100$

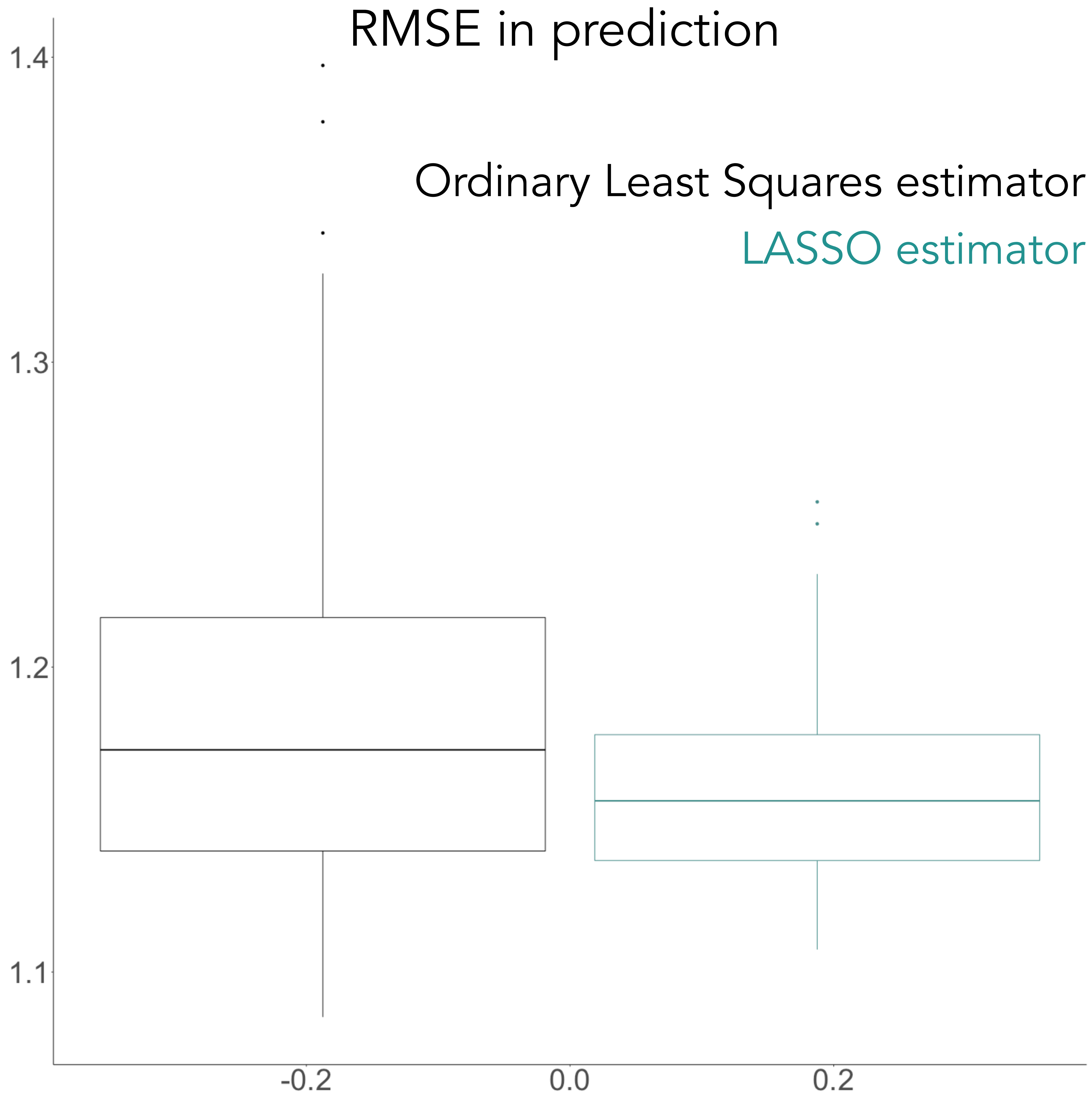
- Sample $(Y_i, X_{i1}, \ldots X_{ip})_{i=1,\ldots,n}$
- Estimate $\widehat{\beta}^k = \left(XX^{\mathrm{T}}\right)^{-1} X^{\mathrm{T}} Y$

For a new sample $(Y_{\text{new},i}, X_{\text{new},i1}, \ldots X_{\text{new},ip})_{i=1,\ldots,n}$

Compute the Root Mean Squared Error (RMSE)

for each $k = 1,\ldots,100$:

$$\sum_{i=1}^{n} \left(\hat{Y}^k_{\text{new,i}} - Y_{\text{new,i}}\right)^2$$



Regularisation parameter tuning

λ manages the bias variance trade-off

Ridge and LASSO estimators strongly depend on λ

- Chaque λ donne une unique solution
- λ is the regularisation - or penalisation - parameter

Extreme behaviours:

- $\lambda = 0$: $\hat{\beta}_\lambda^{Ridge} = \hat{\beta}_\lambda^{Lasso} = \hat{\beta}^{OLS}$
- $\lambda \to \infty$: $\hat{\beta}_\lambda^{Ridge} = \hat{\beta}_\lambda^{Lasso} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

The parameter λ deals with the bias-variance trade-off:

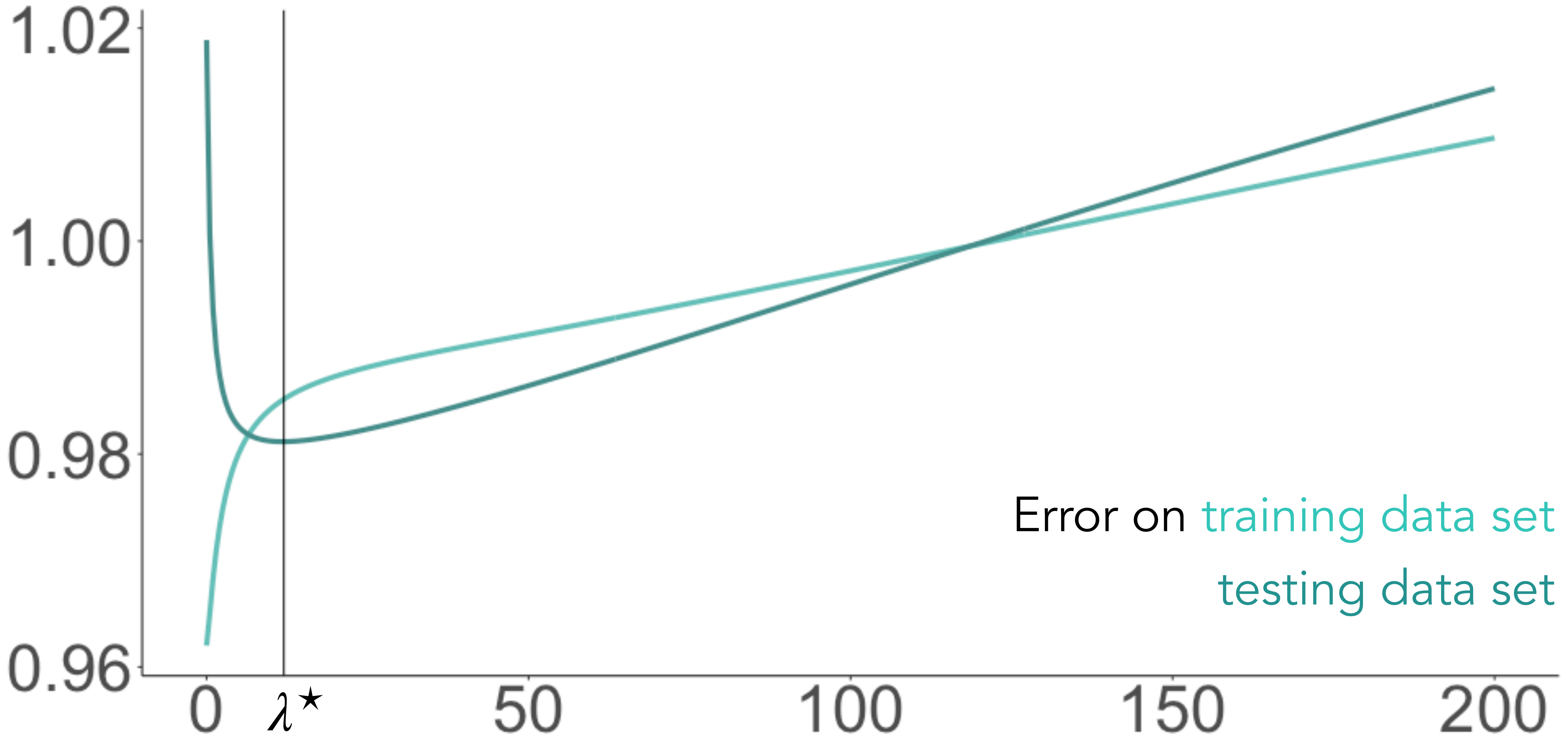
- $\lambda = 0$: $\mathbb{E}\left[\hat{\beta}_\lambda^{Ridge}\right] = \mathbb{E}\left[\hat{\beta}_\lambda^{Lasso}\right] = \mathbb{E}\left[\hat{\beta}^{OLS}\right] = \beta^\star$ but their variances may explode
- $\lambda \to \infty$: $\mathrm{Var}\left(\hat{\beta}_\lambda^{Ridge}\right) = \mathrm{Var}\left(\hat{\beta}_\lambda^{Lasso}\right) = \begin{bmatrix} 0 \ldots 0 \\ \ddots \\ 0 \ldots 0 \end{bmatrix}$ but their bias are equal to $-\beta^\star$

Tuning

Tuning the regularisation parameter to get the best prediction error is a « selection model » issue:

$$\lambda^\star \in \arg\min_{\lambda\in\mathbb{R}^+} \mathbb{E}_{(Y,X)}\left[\left(Y - X\hat{\beta}_\lambda\right)^2\right] \text{ with } \hat{\beta}_\lambda = \left(X^\mathrm{T}X + \lambda\mathbf{I}_p\right)^{-1}X^\mathrm{T}Y$$

→ λ-path: need of a training and a testing data sets, time and computational ressource consuming



→ Cross-validation criteria

Cross-validation criteria

$\forall i = 1, \ldots, n$

- Remove the observation (Y_i, X_i) for the training data set
- Estimate $\hat{\beta}_\lambda^{-i} = \left(X_{-i}^{\mathrm{T}} X_{-i} + \lambda I_p\right)^{-1} X_{-i}^{\mathrm{T}} Y_{-i}$
- Compute the prediction error $\left(Y_i - \hat{\beta}_\lambda^{-i} X_i\right)^2$

The cross-validation criteria is defined as

$$\mathrm{CV}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - X_i \hat{\beta}_\lambda^{-i}\right)^2$$

→ n estimators to compute!

But for the Ridge regression, it is possible to prove that

$$\mathrm{CV}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - X_i \hat{\beta}_\lambda^{-i}\right)^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(Y_i - X_i \hat{\beta}_\lambda\right)^2}{\left(1 - \mathbf{A}_{\lambda_{i,i}}\right)^2} \text{ with } A_\lambda = X\left(X^{\mathrm{T}} X + \lambda I_p\right)^{-1} X^{\mathrm{T}}$$

→ the single Ridge estimator is enough!

Influence matrix and degree of freedom

The influence matrix A is the matrix such as $\hat{Y} = AY$

- OLS: $A^{OLS} = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$

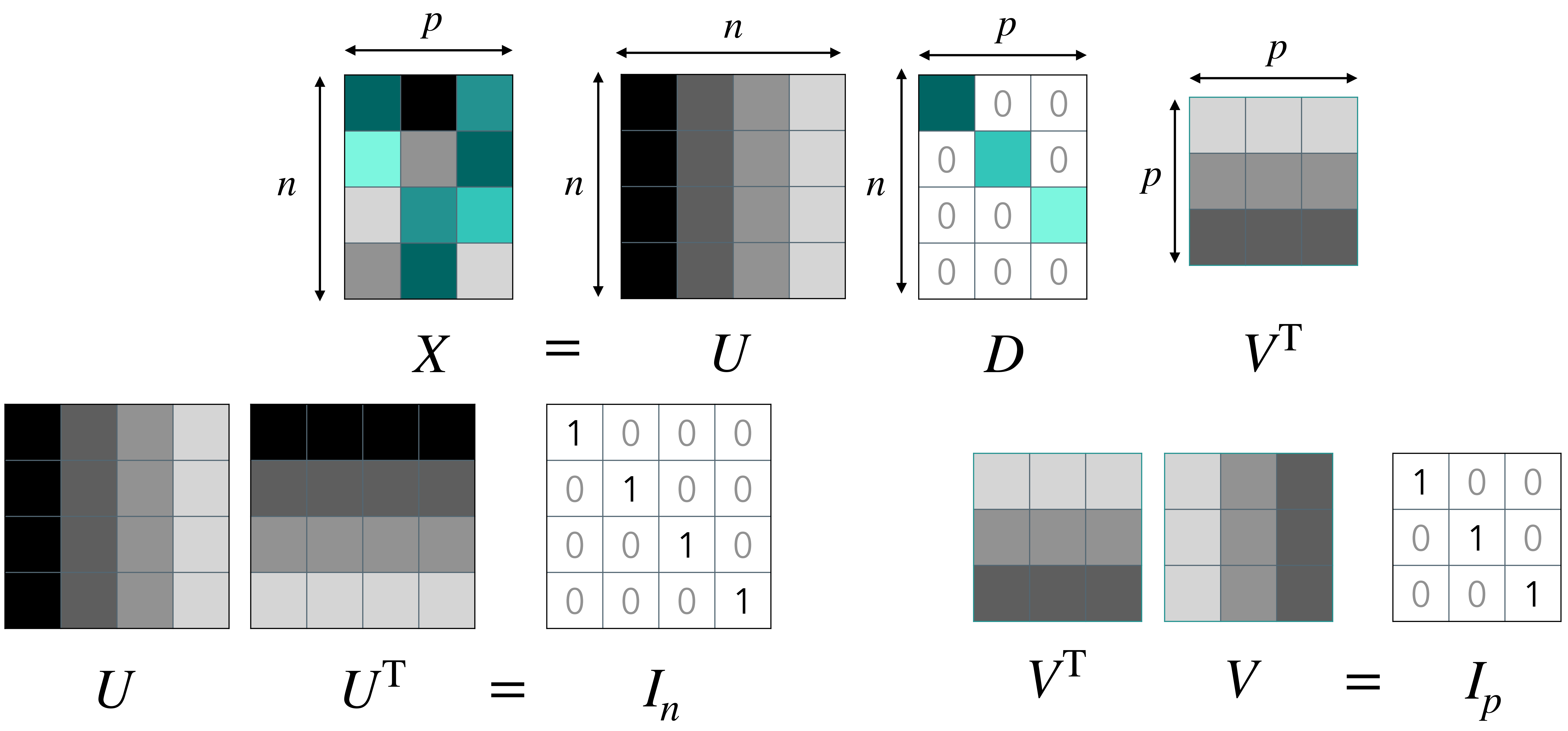
The trace $\mathrm{Tr}(\mathbf{A}^{OLS}) = \mathrm{Tr}\left(X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\right) = \mathrm{Tr}\left(X^{\mathrm{T}}X(X^{\mathrm{T}}X)^{-1}\right) = \mathrm{Tr}(I_p) = p$ equals to the number of parameters /coefficients of β to estimate and is called the degree of freedom

By analogy, for any model, the degree of freedom is the trace of its influence matrix A: $\mathrm{df}(A) = \mathrm{Tr}(A)$

- Ridge: $A_\lambda^{Ridge} = X(X^{\mathrm{T}}X + \lambda I_p)^{-1}X^{\mathrm{T}}$ and $\mathrm{df}(A_\lambda^{Ridge}) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$, with d_j the singular values of X

Singular value decomposition

The singular value decomposition (SVD) is a factorisation of a real $n \times p$ matrix X of the form UDV^{T} where U and V are $n \times n$ and $p \times p$ orthogonal matrices and the only non-zero coefficients of the $n \times p$ matrix D are the diagonal coefficients $d_j = D_{jj}$, called singular values



Generalised cross-validation criteria

We recall that for the Ridge regression

$$\mathrm{CV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - X_i\hat{\beta}_\lambda^{-i}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(Y_i - X_i\hat{\beta}_\lambda\right)^2}{\left(1 - A_{\lambda_{i,i}}^{\mathrm{Ridge}}\right)^2}$$

With the approximation $A_{\lambda_{i,i}} \approx \frac{\mathrm{Tr}\left(A_\lambda\right)}{n}$, we define a generalised cross-validation criteria generally used in the software packages as

$$\mathrm{GCV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(Y_i - X_i\hat{\beta}_\lambda\right)^2}{\left(1 - \frac{\mathrm{df}(\mathbf{A}_\lambda)}{n}\right)^2}$$

Elastic net regression

Elastic net regression

Elastic net linear regression uses the regularisations from both the LASSO and Ridge regression

It eliminates the following LASSO limitation:

when $n < p$, $\hat{\beta}^{LASSO}$ can not have more than n non-zero coefficients (saturation)

$$\hat{\beta}^{\text{Elastic.net}} \in \arg\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$$

or equally, with $0 \leq \alpha \leq 1$

$$\hat{\beta}^{\text{Elastic.net}} \in \arg\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|^2 + \lambda \left(\alpha \|\beta\|_1 + (1 - \alpha) \|\beta\|_2^2 \right)$$

Online approaches

Ridge Regression: Recursive ridge regression using second-order stochastic algorithms. Antoine Godichon-Baggioni, Bruno Portier, Wei Lu. *Computational Statistics & Data Analysis* (2023)

LASSO Regression: An homotopy algorithm for the Lasso with online observations. Pierre Garrigues and Laurent Ghaoui. *Advances in neural information precessing systems 21* (2008)

Implementation

```r
beta_ols <- lm(Y~ X-1)$coefficients
library(glmnet)
beta_ridge <- glmnet(X, Y, alpha = 0, lambda = Lambda)$beta
beta_lasso <- glmnet(X, Y, alpha = 1, lambda = Lambda)$beta
beta_elasticnet <- glmnet(X, Y, alpha = alpha, lambda = Lambda)$beta
```

⚠ λ = alpha, α = l1_ratio

```python
from sklearn.linear_model import LinearRegression
beta_ols = LinearRegression().fit(X,Y).coef_
from sklearn.linear_model import Ridge, Lasso, ElasticNet
beta_ridge = Ridge(alpha = lambda).fit(X,Y).coef_
beta_lasso = Lasso(alpha = lambda).fit(X,Y).coef_
beta_elasticnet = ElasticNet(alpha = lambda, l1_ratio = alpha).fit(X,Y).coef_
```

Generalised additive models

Formulation, estimation and implementation

Formulation

A generalised additive model (GAM) relates a random variable Y to some explanatory variables $X_1, X_2, \ldots$ via a link function g and a structure such as

$$g\left(\mathbb{E}[Y]\right) = f_1(X_1) + f_2(X_2) + f_3(X_1, X_3) + \ldots = \sum_k f_k(X_{k_1}, X_{k_2}, \ldots)$$

Assumptions:

- An exponential family distribution is specified for Y
- The unknown functions $f_1, f_2, \ldots$ are smooth

→ To estimate $f_1, f_2, \ldots$, parametric forms may be specified

A basic univariate model

We consider a simple model

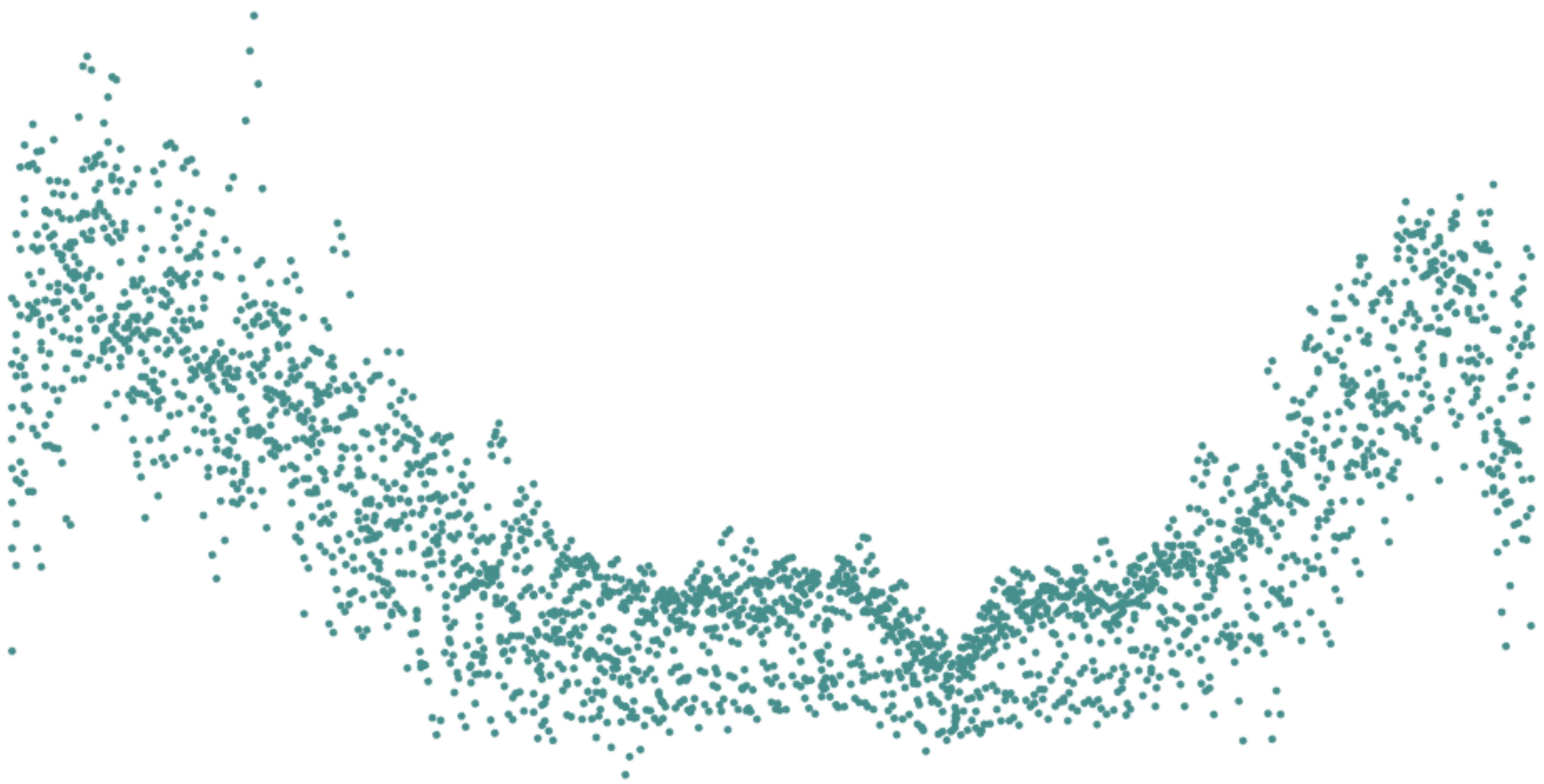
$$Y_i = f^\star(X_i) + \varepsilon_i \text{ , for } i = 1,\ldots n$$

where $f^\star : \mathbb{R} \to \mathbb{R}$ is an unknown function and $\varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,\sigma^2)$

Linear regression is not suitable!

Other solutions:

- Data transformation
- Kernel methods
- k-nearest neighbours
- Regression on a basis of functions
  - Fourier functions (for periodic functions)
  - Wavelets
  - Splines



A basic univariate model

We introduce a basis of functions $b_1, , \dots b_q$ and assume that

$$f^\star \in \left\{ f : x \mapsto \sum_{j=1}^{p} \beta_j b_j(x) \right\}$$

With $Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \\ \vdots \\ Y_n \end{bmatrix}$, $X = \begin{bmatrix} b_1(X_1) & \cdots & b_p(X_1) \\ \vdots & & \vdots \\ b_1(X_i) & \cdots & b_p(X_i) \\ \vdots & & \vdots \\ b_1(X_n) & \cdots & b_p(X_n) \end{bmatrix}$, $\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$ and $\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_i \\ \vdots \\ \varepsilon_n \end{bmatrix}$, we obtain the linear

regression model formulation $Y = X\beta + \varepsilon$

Example: B-splines (De Boor, 1978)

Splines are functions defined piecewise by polynomials

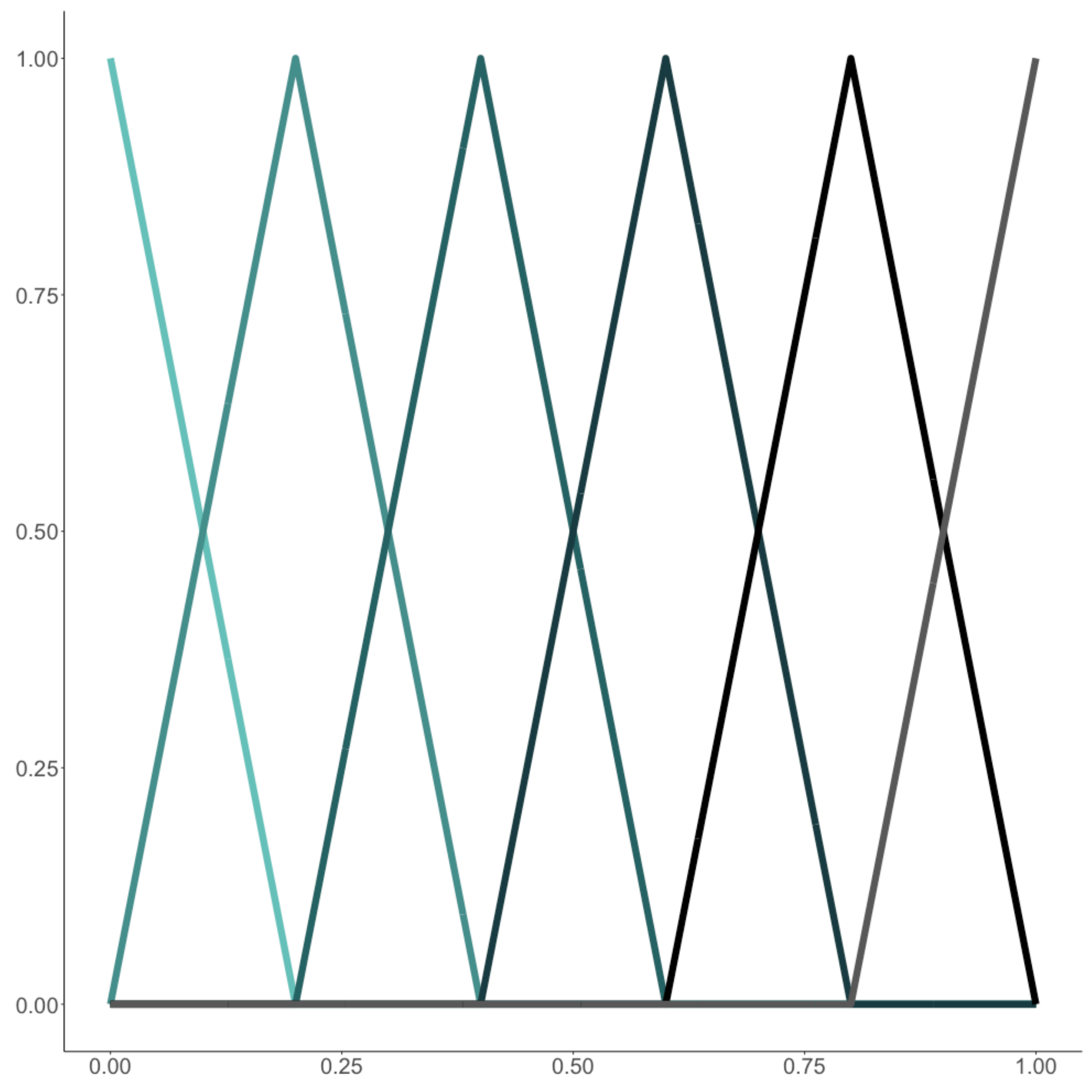
With $q+1$ knots $0 = x_0 < x_1 < x_2 < \ldots < x_q = 1$, B-splines are defined on $[0,1]$ by induction:

$$\forall j = 1,\ldots,q : \quad b_{j,0}(x) = \begin{cases} 1 & \text{if} \quad x_{j-1} < x < x_j \\ 0 & \text{else} \end{cases}$$

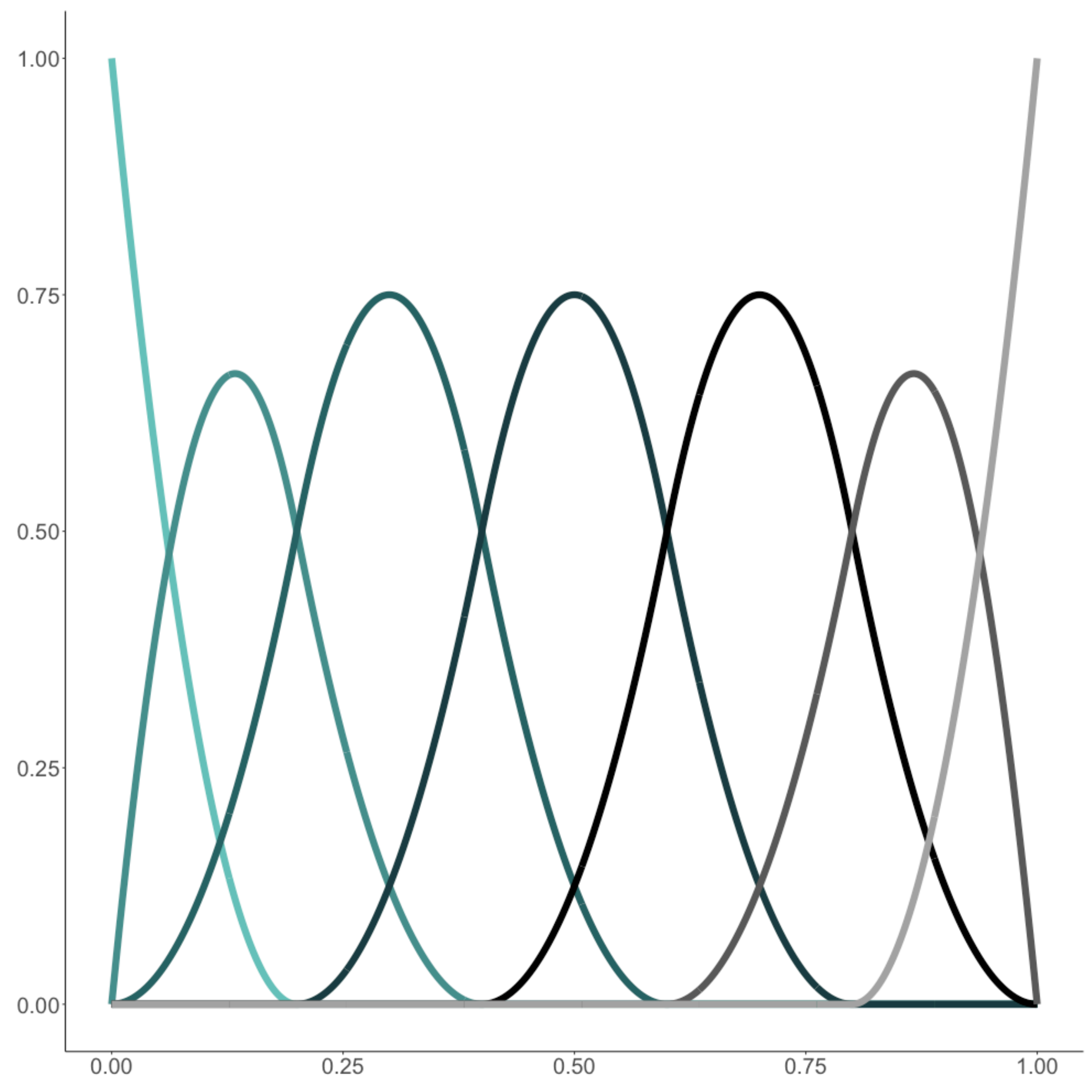
For $d = 1,\ldots$

$$b_{j,d}(x) = \frac{x - x_{j-1}}{x_{j-1+p} - x_{j-1}} b_{j-1,d-1}(x) + \frac{x_{j+p} - x}{x_{j+p} - x_j} b_{j,d-1}(x)$$

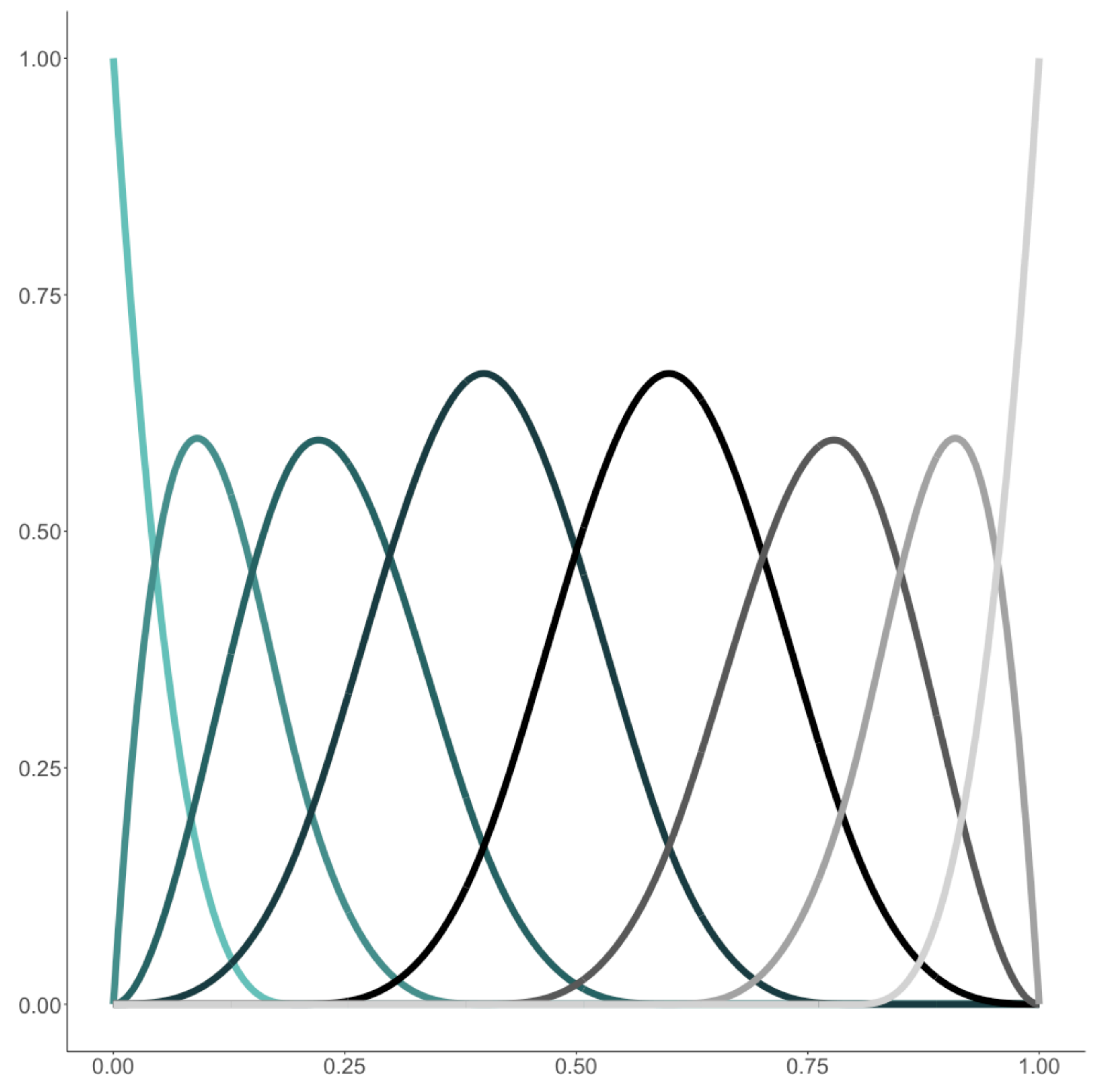
Example: B-splines (De Boor, 1978)



$d = 1$

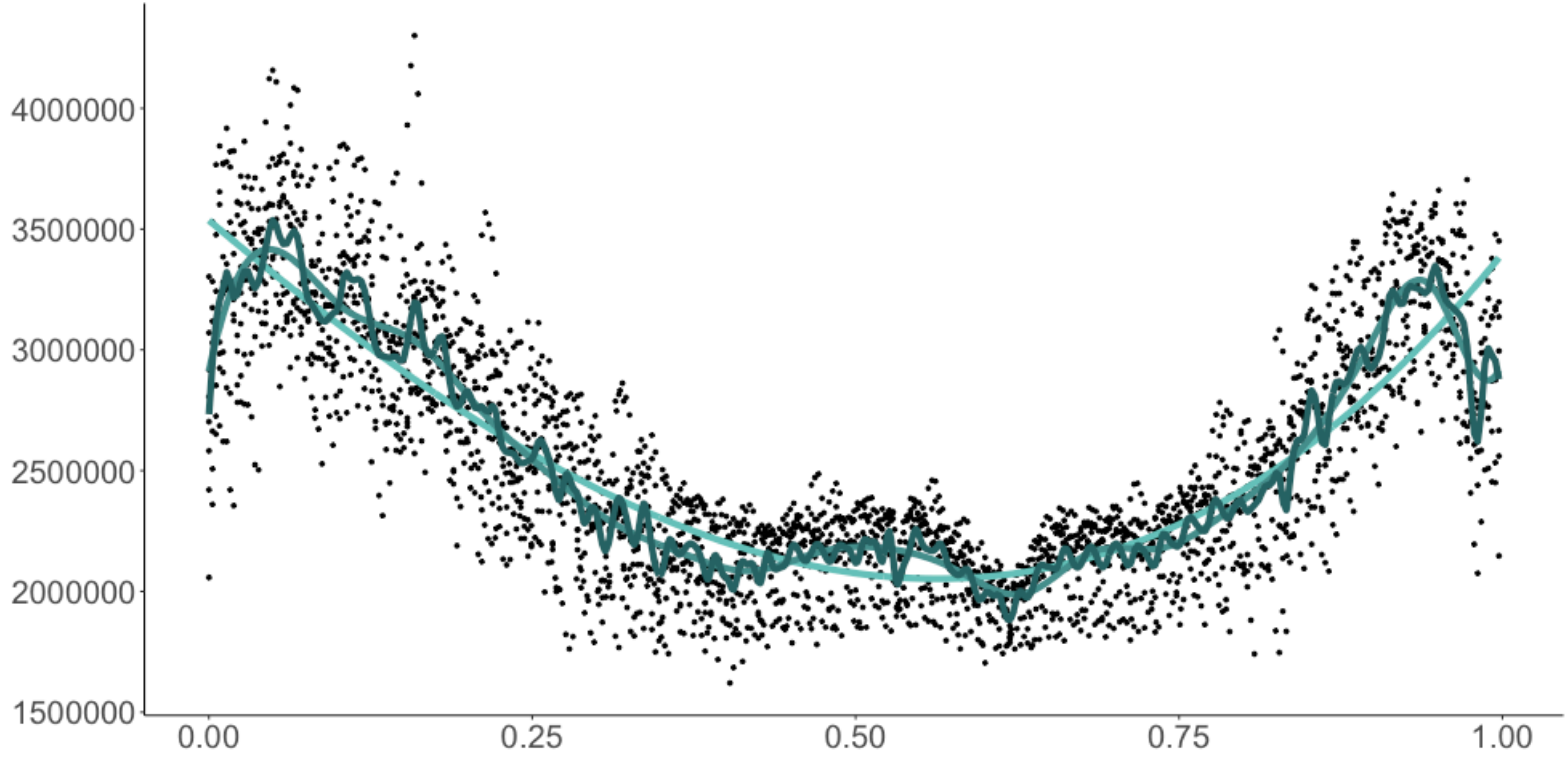


$d = 2$

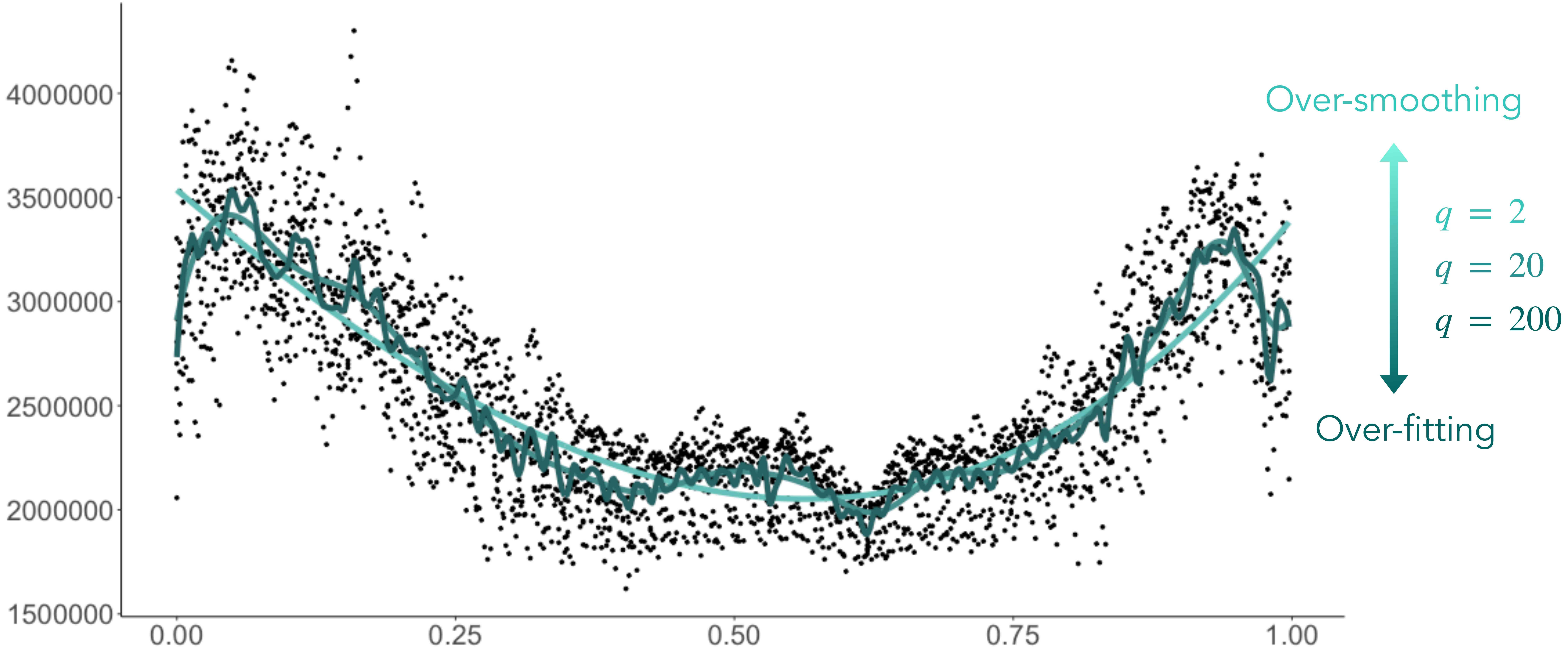


$d = 3$

Knot position and number



Knot position and number



Regression on spline basis - Penalisation

→ Need to impose a constraint on the smoothness:

$$\arg\min_{\beta\in\mathbb{R}^p} \|Y - f(X)\|^2 \quad \text{with} \quad \int_{\mathbb{R}} f''(x)^2 \mathrm{d}x \leq \text{constant}$$

As $f(x) = \sum_{j=1}^{p} \beta_j b_j(x)$, by linearity of the differentiation $f''(x) = \sum_{j=1}^{p} \beta_j b_j''(x)$

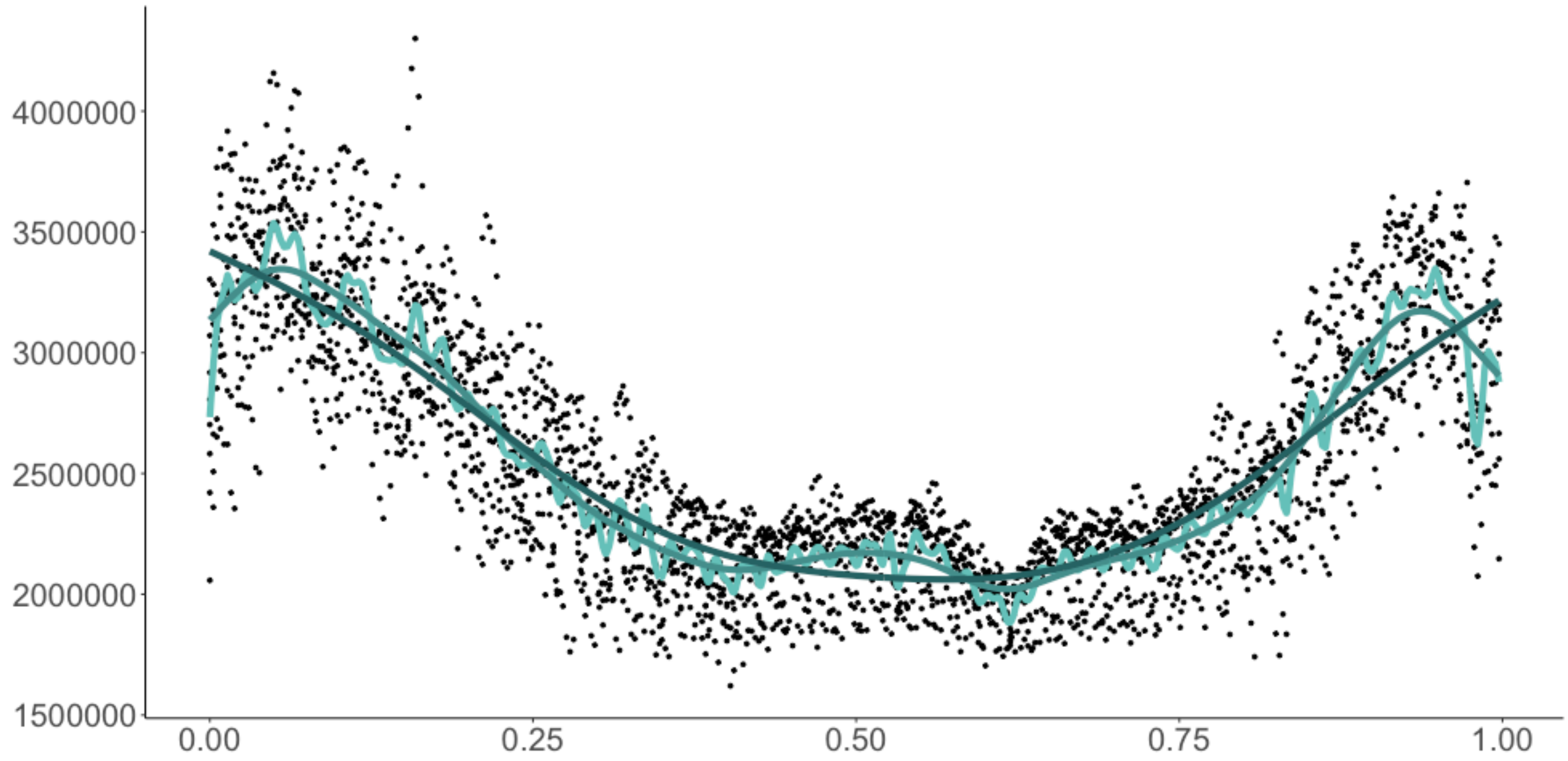
Therefore, $\int_{\mathbb{R}} f''(x)^2 \mathrm{d}x = \beta^{\mathrm{T}} \int_{\mathbb{R}} d(x) d(x)^{\mathrm{T}} \mathrm{d}x \, \beta$ where $d(x) = \begin{bmatrix} b_1''(x) \\ \vdots \\ b_p''(x) \end{bmatrix}$

With S the $p \times p$-matrix such as $S_{jj'} = \int_{\mathbb{R}} b_j''(x) b_{j'}''(x) \mathrm{d}x$, we get that $\int_{\mathbb{R}} f''(x)^2 \mathrm{d}x = \beta^{\mathrm{T}} S \beta$ and the problem is equivalent to solve, for a regularisation parameter $\lambda > 0$

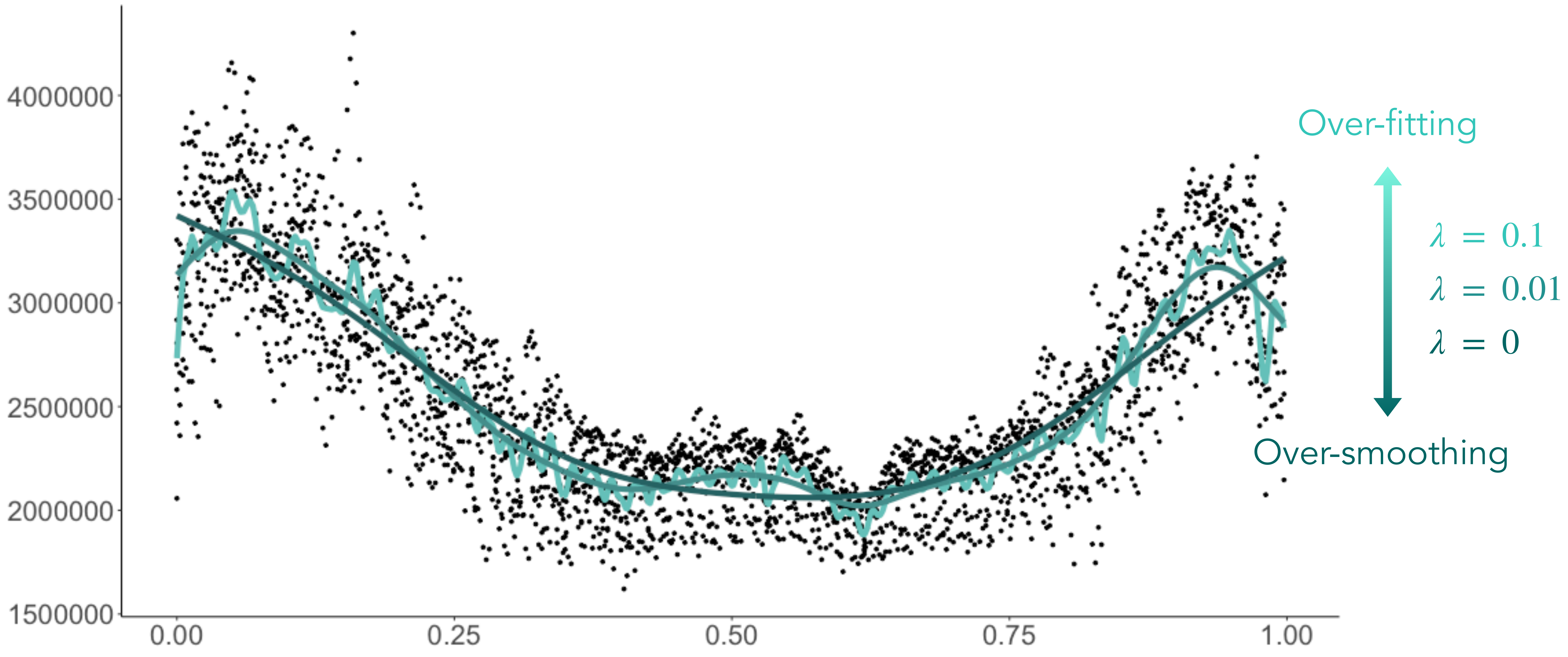
$$\arg\min_{\beta\in\mathbb{R}^p} \|Y - X\beta\|^2 + \lambda \beta^{\mathrm{T}} S \beta$$

→ $\hat{\beta}_\lambda = \left(X^{\mathrm{T}} X + \lambda S\right)^{-1} X^{\mathrm{T}} Y$

Regularisation parameter



Regularisation parameter



Generalised cross-validation criteria

With $A_\lambda = X\left(X^\mathrm{T}X + \lambda S\right)^{-1}X^\mathrm{T}$ and $\hat{\beta}_\lambda = \left(X^\mathrm{T}X + \lambda S\right)^{-1}X^\mathrm{T}Y$,

The regularisation parameter is chosen by minimising the generalised cross-validation criteria

$$\mathrm{GCV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(Y_i - \hat{\beta}_\lambda X_i\right)^2}{\left(1 - \frac{\mathrm{Tr}(\mathbf{A}_\lambda)}{n}\right)^2}$$

From GAM to linear regression

We recall the formulation

$$g\left(\mathbb{E}[Y]\right) = f_1(X_1) + f_2(X_2) + f_3(X_1, X_3) + \ldots = \sum_k f_k\left(X_{k_1}, X_{k_2}, \ldots\right)$$

For each k

A spline basis and a penalisation are specified

For bi/multi-variate functions:

Bivariate function basis (thin plates)

Tensor product $f(x_1, x_2) = \sum_{j=1}^{p} \sum_{j'=1}^{p'} \beta_j^1 \beta_{j'}^2 b_j^1(x_1) b_{j'}^2(x_2)$

A constraint is added - $\int f_k(x)\mathrm{d}x = 0$, e.g. - to ensure the identifiability of the model

→ We obtain a linear formulation $f_k\left(X_{k_1}, X_{k_2}, \ldots\right) = \mathbf{X}_k \beta_k$ and a penalisation $\lambda_k \beta_k^{\mathrm{T}} S_k \beta_k$

From GAM to linear regression

With $\mathbf{X} = \left[\mathbf{X}_1 \,|\, \ldots \,|\, \mathbf{X}_k \,|\, \ldots\right]$ and $\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \\ \vdots \end{bmatrix}$, we obtain an over-parametrised linear model formulation

$$Y = \mathbf{X}\beta + \varepsilon$$

The penalisation terms are gathered into $\beta^{\mathrm{T}}\mathbf{S}_\lambda\beta$ where $\mathbf{S}_\lambda = \sum_k \lambda_k \begin{bmatrix} 0 & 0 & 0 \\ 0 & S_k & 0 \\ 0 & 0 & 0 \end{bmatrix}$, so we aim to solve

$$\arg\min_\beta ||Y - \mathbf{X}\beta||^2 + \beta^{\mathrm{T}}\mathbf{S}_\lambda\beta$$

→ $\hat{\beta}_\lambda = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \mathbf{S}_\lambda\right)^{-1}\mathbf{X}^{\mathrm{T}}Y$ and the vector λ is chosen to minimise the GCV criteria

Implementation

```r
library(mgcv)
eq  <- y ~ s(x1, bs = 'cr', k = 10, by = x2) +
              s(x3, bs = 'cc', k = 10) +
              as.factor(x4) + te(x5,x6)
mod <- gam(formula = eq, data = data_train)
summary(mod)
hat_y <- predict(mod, newdata = data_test)
```

⚠ not as mature as mgcv

```python
import statsmodels.api as sm
from stats models.gam.api import GLMGam, BSplines
mod = GLMGam.from_formula(y ~ x1, data = data_train,
   smoother = BSplines(data_train[[‘x2’,’x3’,’x3’]],
   df = [10,10,10], degree = [3,3,3]), alpha = alpha).fit()
```

Online approaches

Online Generalised Additive Models

First idea: retrain all the model at each time step and eventually weight the observations

$$\arg\min_{f_k} \sum_{s=1}^{t} \omega_s \Big(Y_s - \sum_k f_k(X_{s,k_1}, X_{s,k_2}, \ldots) \Big)^2$$

Some concerns (that may be true for any complex / blackbox model):

- GAM are complex models which need lots of data to be trained so ω_t can not go to fast to 0
- GAM are over-parametrised linear models
  - → Trained to be good on all the data points (for each ω_t is high enough)
  - → Is a re-training of all the parameters necessary (interpretability, robustness)?
- Costly in terms of computing time and memory

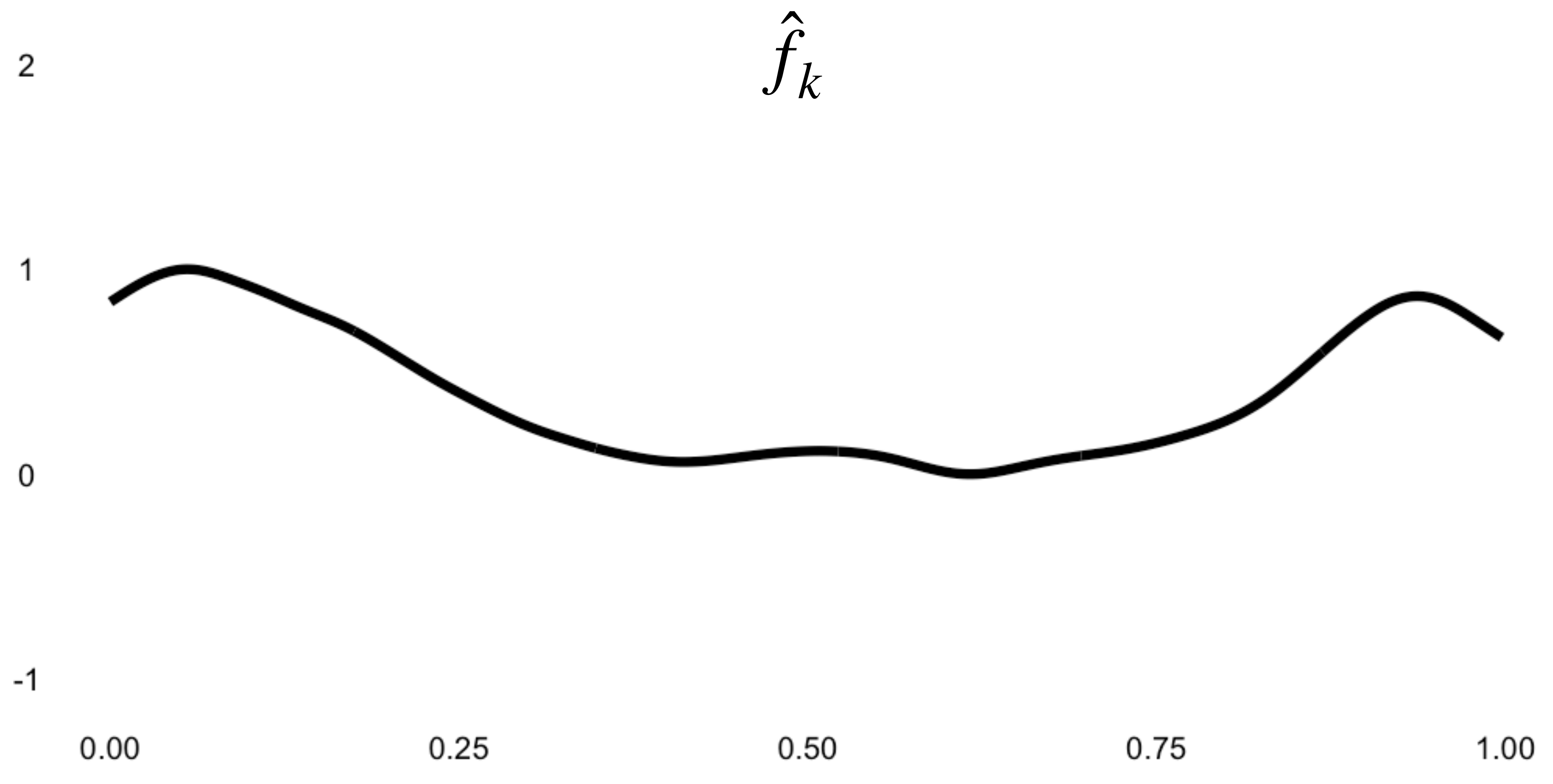
  Remark: in the mgcv R-package, `bam()` function updates an existing GAM with new data
- Need of model which reacts rapidly and locally

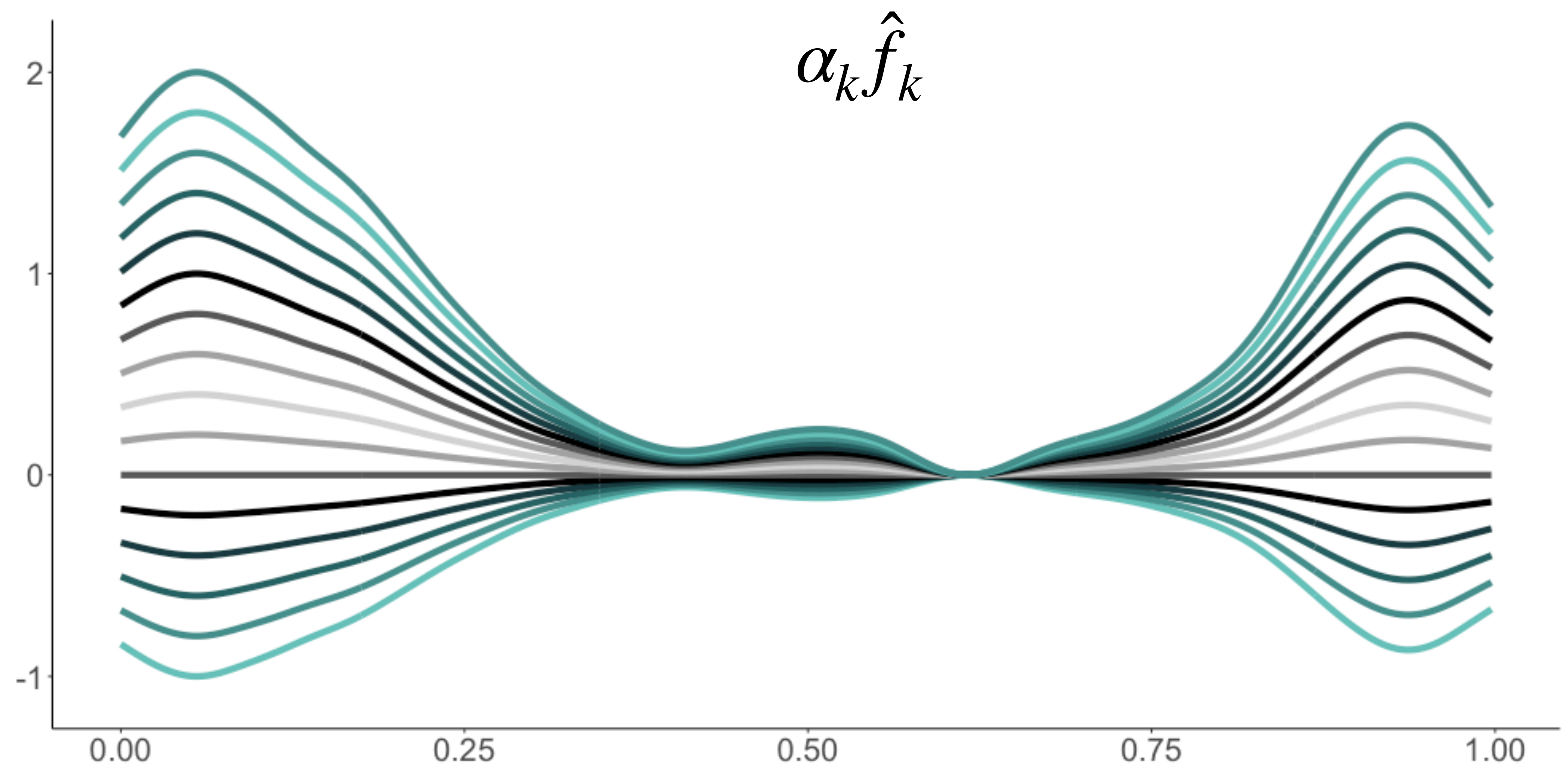
Online Generalised Additive Models

Idea:

Keep the estimated functions $\hat{f}_k$

But introduce some coefficients $\alpha_{t,k}$ that will be re-estimated at each time step t to allow the effect to evolve: $\hat{f}_{t,k} = \alpha_{t,k}\hat{f}_k$





Adaptive GAM with online linear regression

Underlying assumption: $Y_t = \sum_k \alpha_{k,t} \hat{f}_k (X_{t,k_1}, X_{t,k_2}, \ldots) + \text{noise} = \hat{f}(X_t)^{\mathrm{T}} \alpha_t + \varepsilon_t$

with $\alpha = \begin{bmatrix} \vdots \\ \alpha_k \\ \vdots \end{bmatrix}$ and $\hat{f}(X) = \begin{bmatrix} \vdots \\ \hat{f}(X) \\ \vdots \end{bmatrix}$

These coefficients can be estimated using online linear regression:

$$\hat{\alpha}_{t+1} \in \arg\min_{\alpha_k} \sum_{s=1}^{t} \omega_s \Big(Y_s - \sum_k \alpha_k \hat{f}_k (X_{s,k_1}, X_{s,k_2}, \ldots) \Big)^2$$

Adaptive GAM with Kalman filter

Underlying assumption:

$$Y_t = \hat{f}(X_t)^{\mathrm{T}} \alpha_t + \varepsilon_t \text{ where } \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

$$\alpha_t = \alpha_{t-1} + \eta_t \text{ where } \eta_t \sim \mathcal{N}\left(\mathbf{0}, \Sigma\right)$$

Kalman filter algorithm:

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} + \frac{P_{t-1\hat{f}(X_{t-1})}}{\hat{f}(X_{t-1})^{\mathrm{T}} P_{t-1} \hat{f}(X_{t-1}) + \sigma^2} \left(Y_{t-1} - \alpha_{t-1}^{\mathrm{T}} \hat{f}(X_{t-1}) \right)$$

$$P_t = P_{t-1} - \frac{P_{t-1} \hat{f}(X_{t-1}) \hat{f}(X_{t-1})^{\mathrm{T}} P_{t-1}}{\hat{f}(X_{t-1})^{\mathrm{T}} P_{t-1} \hat{f}(X_{t-1}) + \sigma^2} + \Sigma$$

Generalisation of these two approaches

Functions f_k could be

- Trees of a random forest
- Outputs of the last layer of a neural network
- …

Quantile regression

Motivation

Whereas the least squares method provides an estimate of the expectation (conditional on the explanatory variables) of the random variables Y, quantile regression seeks to approximate the median or other quantiles

It is useful for predicting thresholds

When several regressions are performed, it is possible to get a good idea of the general distribution of Y

Quantile regression is less sensitive to outliers (L_1-loss)

Formulation

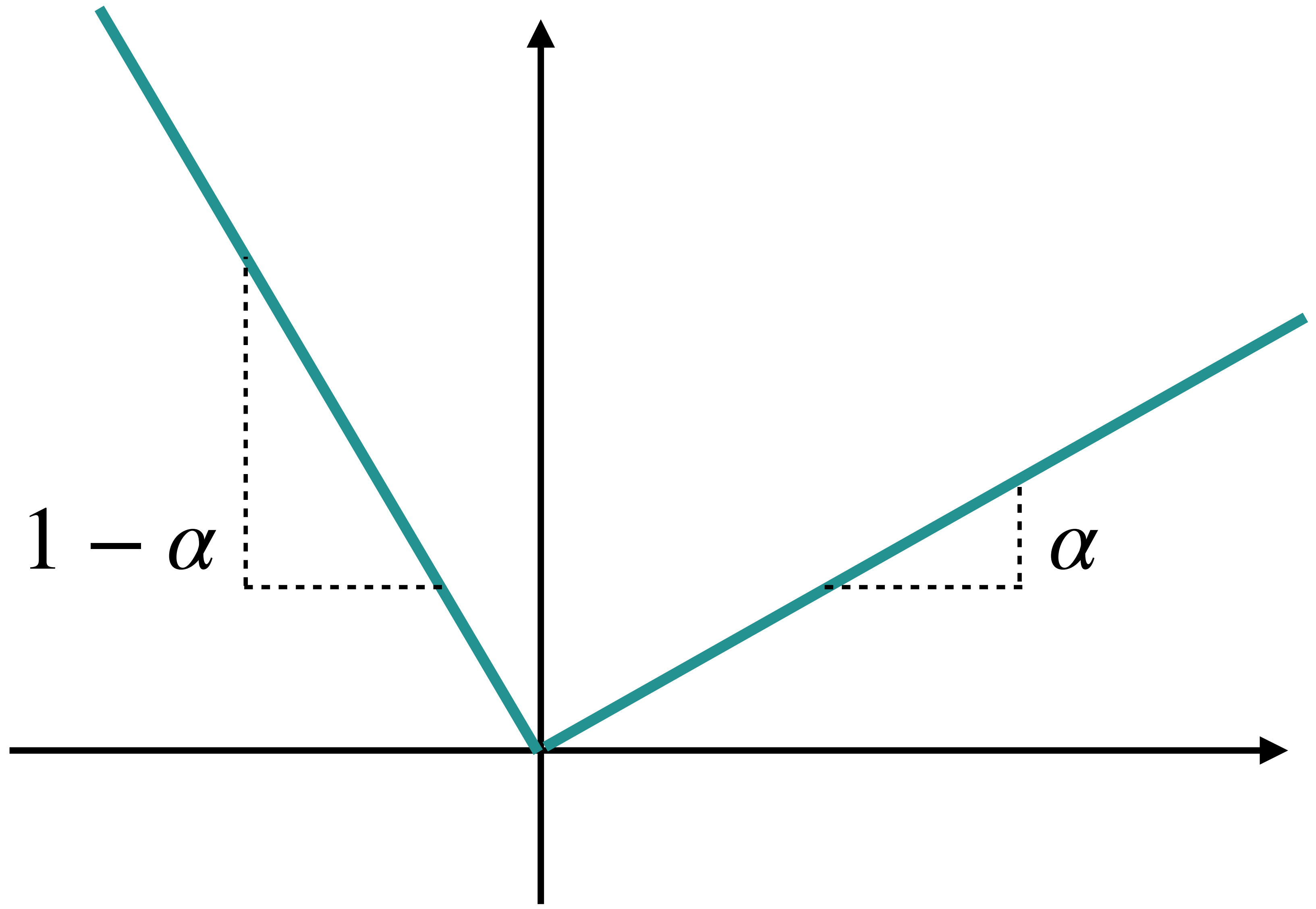
With f_Y the density and F_Y the cumulative distribution function of the random variable Y, by definition, the quantile q_α satisfies

$$F_Y(q_\alpha) = \int_{-\infty}^{q_\alpha} f_Y(y)\mathrm{d}y = \mathbb{P}\left(Y \leq q_\alpha\right) = \alpha$$

With ℓ_α the pinball loss function

$$\ell_\alpha(y-q) = \alpha|y-q|^+ + (1-\alpha)|y-q|^-$$

where $|x|^+ = \max(x,0)$ and $|x|^- = \max(-x,0)$



The quantile q_α minimise the function

$$q \mapsto \mathbb{E}_Y\left[\ell_\alpha(Y-q)\right]$$

Proof

We solve the convexe minimisation problem $q^\star \in \arg\min_q \mathbb{E}\left[\ell_\alpha(Y-q)\right]$ by differentiation

$$
\begin{aligned}
0 \;=\; \mathbb{E}\left[\frac{\partial \ell_\alpha(Y-q)}{\partial q}\right] &= \int_{-\infty}^{+\infty} \frac{\partial \ell_\alpha(y-q)}{\partial q} f(y)\mathrm{d}y \\
&= -(1-\alpha)\int_{-\infty}^{q} f(y)\mathrm{d}y + \alpha \int_{q}^{+\infty} f(y)\mathrm{d}y \\
&= (\alpha-1)F(q) + \alpha(1-F(q)) = \alpha - F(q)
\end{aligned}
$$

Thus, the solution $q^\star$ satisfies $F(q^\star) = \alpha$

Estimation

Let $(Y_i, X_{i1}, \dots X_{ip})_{i=1,\dots,n}$ be n observations independent and identically distributed of $p+1$ reals random variables $Y, X_1, \dots, X_p$, an estimator of the quantile α can be found by solving

$$\hat{\beta}^{\alpha} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell_{\alpha}\left(Y_i - X_i \beta\right)$$

It is possible to use a gradient descent method since the function to be is almost universally derivable

The Iteratively Reweighted Least Squares algorithm (IRLS) can also be used

That’s all folks!