

SD-TSIA : Ridge / Tikhonov

Joseph Salmon

<http://josephsalmon.eu>

Télécom ParisTech

Syllabus

Ridge Definitions

- SVD point of view

- Penalty point of view

- Bias analysis with the SVD

- Variance analysis with the SVD

Regularization parameter choice

- Regularization path

- Cross-Validation (CV)

Algorithms and computational aspects

Syllabus

Ridge Definitions

- SVD point of view

- Penalty point of view

- Bias analysis with the SVD

- Variance analysis with the SVD

Regularization parameter choice

- Regularization path

- Cross-Validation (CV)

Algorithms and computational aspects

Reminder

$$\mathbf{y} = X\boldsymbol{\theta}^* + \boldsymbol{\varepsilon}$$

- ▶ $\mathbf{y} \in \mathbb{R}^n$ observations
- ▶ $X \in \mathbb{R}^{n \times p}$ is the design matrix (column-wise: features)
- ▶ $\boldsymbol{\theta}^* \in \mathbb{R}^p$ is the **true** model parameter we aim at finding
- ▶ $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the noise

Rem: possibly a supplementary variable is added for the constants

Singular Value Decomposition (SVD)

Theorem Golub and Van Loan (2013)

For any matrix $X \in \mathbb{R}^{n \times p}$, there exists two orthogonal matrices $U = [\mathbf{u}_1, \dots, \mathbf{u}_n] \in \mathbb{R}^{n \times n}$ and $V = [\mathbf{v}_1, \dots, \mathbf{v}_p] \in \mathbb{R}^{p \times p}$, such that

$$U^\top X V = \text{diag}(s_1, \dots, s_{\text{rg}(X)}) = \Sigma \in \mathbb{R}^{n \times p}$$

with $s_1 \geq s_2 \geq \dots \geq s_{\text{rg}(X)} > 0$, with $\text{rg}(X) = \text{rang}(X)$.

$$X = U \Sigma V^\top \Leftrightarrow X = \sum_{i=1}^{\text{rg}(X)} s_i \mathbf{u}_i \mathbf{v}_i^\top$$

A least squares solution is then:

$$\hat{\boldsymbol{\theta}}^{\text{OLS}} = X^+ \mathbf{y} = \sum_{i=1}^{\text{rg}(X)} \frac{1}{s_i} \mathbf{v}_i \mathbf{u}_i^\top \mathbf{y}$$

Numerical instabilities

$$\hat{\boldsymbol{\theta}}^{\text{OLS}} = X^+ \mathbf{y} = \sum_{i=1}^{\text{rg}(X)} \frac{1}{s_i} \mathbf{v}_i \mathbf{u}_i^\top \mathbf{y}$$

If the smallest singular values s_i get close to zero then the numerical solution of the SVD is unstable!

Rem: this drawback is common not only for least squares, but also to other inverse problems (also referred to as “ill posed” in numerical analysis and signal processing)

Normal equations

A solution least squares solution $\boldsymbol{\theta}$ need to satisfy:

$$X^{\top}X\boldsymbol{\theta} = X^{\top}\mathbf{y} \Leftrightarrow V\Sigma^{\top}\Sigma V^{\top}\boldsymbol{\theta} = V\Sigma^{\top}U^{\top}\mathbf{y}$$

and if we look for $\boldsymbol{\theta}$ with the following form: $\boldsymbol{\theta} = V\boldsymbol{\beta}$, this is equivalent to

$$\Sigma^{\top}\Sigma\boldsymbol{\beta} = \Sigma^{\top}U^{\top}\mathbf{y}$$

$\Sigma^{\top}\Sigma$ diagonal with $r = \text{rang}(X)$ non zero elements that are the s_i^2

$$\Sigma^{\top}\Sigma = \left[\begin{array}{cc|c} s_1^2 & & 0 \\ & \ddots & \\ 0 & & s_r^2 \\ \hline & 0 & 0 \end{array} \right] \in \mathbb{R}^{p \times p}$$

Normal equations (continued)

Regularized alternative : solve normal equations where

$$\left[\begin{array}{ccc|c} s_1^2 & & 0 & 0 \\ & \ddots & & \\ 0 & & s_r^2 & 0 \\ \hline & 0 & & 0 \end{array} \right] \text{ replaced by } \left[\begin{array}{ccc|c} s_1^2 & & 0 & 0 \\ & \ddots & & \\ 0 & & s_r^2 & 0 \\ \hline & 0 & & 0 \end{array} \right] + \lambda \text{Id}_p$$

It can be re-written by:

$$(\lambda \text{Id}_p + \Sigma^\top \Sigma) \beta = \Sigma^\top U^\top \mathbf{y}$$

i.e., we add a small $\lambda > 0$ to all the eigen values of $X^\top X$, λ being called the **regularization parameter**

$$\beta = (\lambda \text{Id}_p + \Sigma^\top \Sigma)^{-1} \Sigma^\top U^\top \mathbf{y}$$

and hence

$$\theta = V(\lambda \text{Id}_p + \Sigma^\top \Sigma)^{-1} \Sigma^\top U^\top \mathbf{y}$$

Ridge : explicit formulation

With the SVD, the following equation simplifies:

$$\boldsymbol{\theta} = V(\lambda \text{Id}_p + \Sigma^\top \Sigma)^{-1} \Sigma^\top U^\top \mathbf{y}$$

This gives a simple formulation for the *Ridge* estimator

$$\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} = (\lambda \text{Id}_p + X^\top X)^{-1} X^\top \mathbf{y}$$

Reminder: under the full rank hypothesis $\hat{\boldsymbol{\theta}}^{\text{OLS}} = (X^\top X)^{-1} X^\top \mathbf{y}$

Rem:

$$\lim_{\lambda \rightarrow 0^+} \hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} = \hat{\boldsymbol{\theta}}^{\text{OLS}}$$

$$\lim_{\lambda \rightarrow +\infty} \hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} = \mathbf{0} \in \mathbb{R}^p$$

Kernel trick

Kernel trick: According to whether $n > p$ or $n \leq p$, it is more suitable to consider the following equivalent formulation of the Ridge estimator:

$$X^{\top}(XX^{\top} + \lambda \text{Id}_n)^{-1}\mathbf{y} = (X^{\top}X + \lambda \text{Id}_p)^{-1}X^{\top}\mathbf{y}$$

- ▶ LHS: inverse/solve an $n \times n$ matrix
- ▶ RHS: inverse/solve an $p \times p$ matrix

Rem: this property is also useful for kernel method such as SVM
(cf. *machine learning* course)

Exo: Show the kernel trick using the SVD of X

Syllabus

Ridge Definitions

SVD point of view

Penalty point of view

Bias analysis with the SVD

Variance analysis with the SVD

Regularization parameter choice

Regularization path

Cross-Validation (CV)

Algorithms and computational aspects

Ridge / Tikhonov : penalized definition

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \left(\underbrace{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda \|\boldsymbol{\theta}\|_2^2}_{\text{regularization}} \right)$$

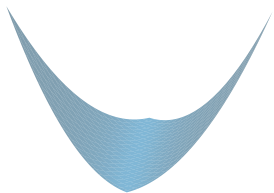
- ▶ Note that the *Ridge* estimator is **unique** for any fixed $\lambda > 0$
- ▶ We recover the limiting cases:

$$\lim_{\lambda \rightarrow 0} \hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = \hat{\boldsymbol{\theta}}^{\text{OLS}} \text{ (solution with smallest } \|\cdot\|_2 \text{ norm)}$$

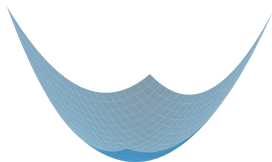
$$\lim_{\lambda \rightarrow +\infty} \hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = \mathbf{0} \in \mathbb{R}^p$$

- ▶ Link between the two formulation thanks to the first order conditions: for $f(\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2/2 + \lambda \|\boldsymbol{\theta}\|_2^2/2$
 $\nabla f(\boldsymbol{\theta}) = X^{\top}(X\boldsymbol{\theta} - \mathbf{y}) + \lambda\boldsymbol{\theta} = 0 \Leftrightarrow (X^{\top}X + \lambda \text{Id}_p)\boldsymbol{\theta} = X^{\top}\mathbf{y}$

Motivation



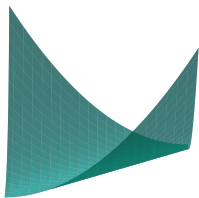
OLS



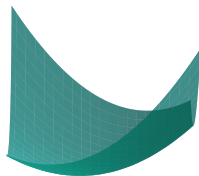
Ridge

Regularize: simplify the problem when ill-conditioned

Motivation



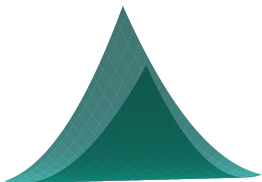
OLS



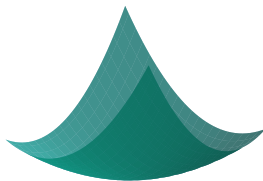
Ridge

Regularize: simplify the problem when ill-conditioned

Motivation



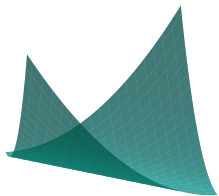
OLS



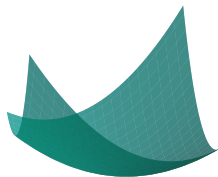
Ridge

Regularize: simplify the problem when ill-conditioned

Motivation



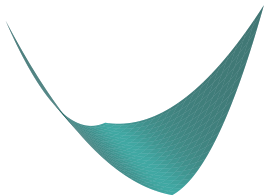
OLS



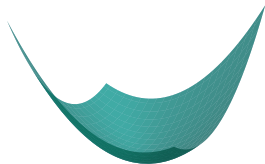
Ridge

Regularize: simplify the problem when ill-conditioned

Motivation



OLS



Ridge

Regularize: simplify the problem when ill-conditioned

Constraint interpretation

A “Lagrangian” formulation is as follows:

$$\arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2}_{\text{regularization}} \right)$$

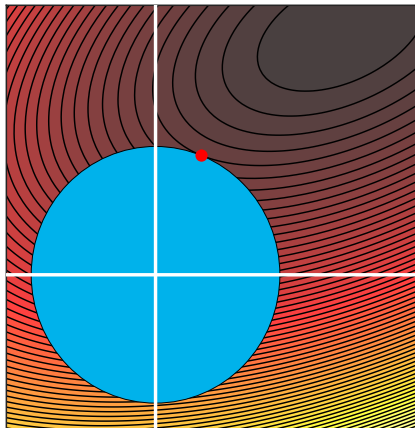
has for a certain $T > 0$ the same solution as:

$$\begin{cases} \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ \text{s.t. } \|\boldsymbol{\theta}\|_2^2 \leq T \end{cases}$$

Rem: the link $T \leftrightarrow \lambda$ is not explicit!

- ▶ If $T \rightarrow 0$ we recover the null vector: $\mathbf{0} \in \mathbb{R}^p$
- ▶ If $T \rightarrow \infty$ we recover $\hat{\boldsymbol{\theta}}^{\text{OLS}}$ (un-constrained)

Level lines and constraints set



Optimization under ℓ_2 constraints

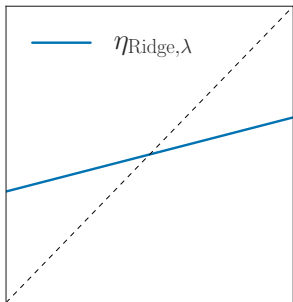
The orthogonal case

Consider the simple case: $X^\top X = \text{Id}_p$

$$\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} = (\lambda \text{Id}_p + X^\top X)^{-1} X^\top \mathbf{y}$$

$$\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} = (\lambda \text{Id}_p + \text{Id}_p)^{-1} X^\top \mathbf{y} = \frac{1}{\lambda + 1} X^\top \mathbf{y}$$

$$\hat{\mathbf{y}} = \frac{1}{\lambda + 1} \mathbf{y} = (\eta_{\text{rdg},\lambda}(\mathbf{y}_i))_{i=1,\dots,n}$$



Rem: the real function $\eta_{\text{rdg},\lambda}$ is a linear contraction (shrinkage)

Associated prediction

From the *Ridge* coefficient:

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = (\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} \mathbf{y}$$

the associated prediction is given by:

$$\hat{\mathbf{y}} = X \hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = X(\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} \mathbf{y}$$

Rem: the estimator $\hat{\mathbf{y}}$ is linear w.r.t. \mathbf{y}

Rem: the matrix

$$H_{\lambda} = X(\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} = \sum_{j=1}^{\text{rg}(X)} \frac{s_j^2}{s_j^2 + \lambda} \mathbf{u}_j \mathbf{u}_j^{\top}. \text{ is the}$$

equivalent of the **hat matrix**

If $\lambda \neq 0$, we do not have $H_{\lambda}^2 = H_{\lambda} = \sum_{j=1}^{\text{rg}(X)} \mathbf{u}_j \mathbf{u}_j^{\top}$ anymore, so H_{λ} is not a projection (in general).

N remarks

Reminder: normalizing the p features the same way is necessary if you want the penalty to be similar for all features:

- ▶ center the observation and the features \Rightarrow no coefficient for the constants (hence no constraint on it)
- ▶ not centering features \Rightarrow do not put constraint on the constant feature (bias/intercept)

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \|\mathbf{y} - X\boldsymbol{\theta} - \theta_0 \mathbf{1}_n\|_2^2 + \lambda \sum_{j=1}^p \theta_j^2$$

Alternative (without normalization): change the penalty in

$$\arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^p \alpha_j \theta_j^2 \quad (\text{e.g., } \alpha_j = \|\mathbf{x}_j\|_2^2)$$

Rem: for cross validation one can use $\frac{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}{2n}$ rather than $\frac{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}{2}$ as the data fitting part

Normalization (continued)

Consider the estimator Ridge with variables $X' = [\mathbf{x}'_1, \dots, \mathbf{x}'_K]$, such that there exist a linear link $\sum_{k=1}^K \mu_k \mathbf{x}'_k = 1$ (e.g., qualitative variable):

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p, \boldsymbol{\theta}' \in \mathbb{R}^K} \|\mathbf{y} - X\boldsymbol{\theta} - X'\boldsymbol{\theta}' - \theta_0 \mathbf{1}_n\|_2^2 + \lambda \sum_{j=1}^p \theta_j^2 + \lambda \sum_{k=1}^K \theta_k'^2$$

is equivalent to solve :

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = \arg \min_{\substack{\boldsymbol{\theta} \in \mathbb{R}^p, \boldsymbol{\theta}' \in \mathbb{R}^K \\ \sum_{k=1}^K \theta_k' = 0}} \|\mathbf{y} - X\boldsymbol{\theta} - X'\boldsymbol{\theta}' - \theta_0 \mathbf{1}_n\|_2^2 + \lambda \sum_{j=1}^p \theta_j^2 + \lambda \sum_{k=1}^K \theta_k'^2$$

Exo: proof ; help: cf. [Par06, page 35]

Syllabus

Ridge Definitions

SVD point of view

Penalty point of view

Bias analysis with the SVD

Variance analysis with the SVD

Regularization parameter choice

Regularization path

Cross-Validation (CV)

Algorithms and computational aspects

General form of the bias

Under Additive White Gaussian Noise (AWGN) assumption

$\mathbf{y} = X\boldsymbol{\theta}^* + \boldsymbol{\varepsilon}$ with $\mathbb{E}(\boldsymbol{\varepsilon}) = 0$:

$$\begin{aligned}\mathbb{E}(\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}}) &= \mathbb{E}[(\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} \mathbf{y}] \\ &= \mathbb{E}[(\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} X \boldsymbol{\theta}^* + (\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} \boldsymbol{\varepsilon}] \\ &= (\lambda \text{Id}_p + X^{\top} X)^{-1} X^{\top} X \boldsymbol{\theta}^* \\ &= \sum_{i=1}^{\text{rg}(X)} \frac{s_i^2}{s_i^2 + \lambda} \mathbf{v}_i \mathbf{v}_i^{\top} \boldsymbol{\theta}^*\end{aligned}$$

Rem: one recovers $\mathbb{E}(\hat{\boldsymbol{\theta}}^{\text{OLS}}) = \sum_{i=1}^{\text{rg}(X)} \mathbf{v}_i \mathbf{v}_i^{\top} \boldsymbol{\theta}^*$ when $\lambda \rightarrow 0$

Rem: the bias is $\mathbb{E}(\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}}) - \boldsymbol{\theta}^* = -\lambda(X^{\top} X + \lambda \text{Id}_p)^{-1} \boldsymbol{\theta}^*$

Syllabus

Ridge Definitions

SVD point of view

Penalty point of view

Bias analysis with the SVD

Variance analysis with the SVD

Regularization parameter choice

Regularization path

Cross-Validation (CV)

Algorithms and computational aspects

Variance in the general case

Under the assumption $\mathbb{E}(\boldsymbol{\varepsilon}) = 0$, and with a homoscedastic model:
 $\mathbb{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top) = \sigma^2 \text{Id}_n$

Variance / Covariance

$$V_\lambda^{\text{rdg}} = \mathbb{E} \left((\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} - \mathbb{E}(\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}})) (\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} - \mathbb{E}(\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}}))^\top \right)$$

Explicit computation:

$$\begin{aligned} V_\lambda^{\text{rdg}} &= \mathbb{E}((\lambda \text{Id}_p + X^\top X)^{-1} X^\top \boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^\top X (\lambda \text{Id}_p + X^\top X)^{-1}) \\ &= (\lambda \text{Id}_p + X^\top X)^{-1} X^\top \mathbb{E}(\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^\top) X (\lambda \text{Id}_p + X^\top X)^{-1} \\ &= \sigma^2 (\lambda \text{Id}_p + X^\top X)^{-1} X^\top X (\lambda \text{Id}_p + X^\top X)^{-1} \\ &= \sum_{i=1}^{\text{rg}(X)} \frac{s_i^2 \sigma^2}{(s_i^2 + \lambda)^2} \mathbf{v}_i \mathbf{v}_i^\top \end{aligned}$$

Rem: one recovers $V^{\text{OLS}} = \sum_{i=1}^{\text{rg}(X)} \frac{\sigma^2}{s_i^2} \mathbf{v}_i \mathbf{v}_i^\top$ when $\lambda \rightarrow 0$

Rem: one find a null variance when $\lambda \rightarrow \infty$

Prediction risk

Homoscedastic assumption: $\mathbb{E}(\epsilon\epsilon^\top) = \sigma^2 \text{Id}_n$

Quadratic prediction risk $\mathbb{E}\|X\boldsymbol{\theta}^\star - X\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}}\|^2$

Under the Homoscedastic assumption:

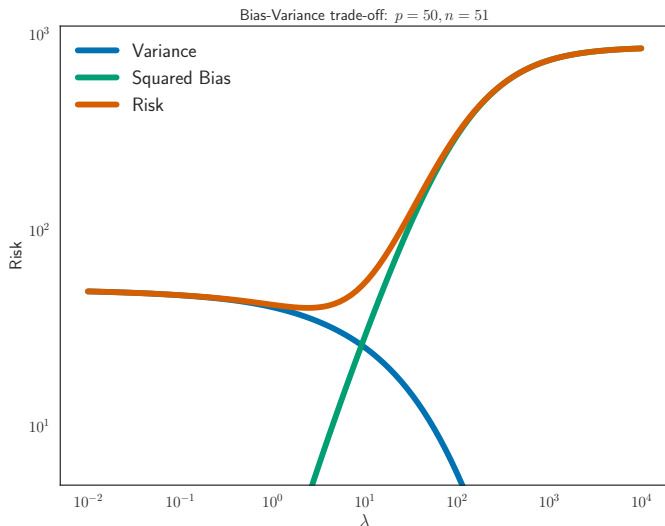
$$R_{\text{pred}}(\boldsymbol{\theta}^\star, \hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}}) = \mathbb{E} \left[(\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} - \boldsymbol{\theta}^\star)^\top (X^\top X) (\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} - \boldsymbol{\theta}^\star) \right]$$

Explicit computation (begins as for OLS):

$$\begin{aligned} R_{\text{pred}}(\boldsymbol{\theta}^\star, \hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}}) &= \mathbb{E} \left[(\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} - \boldsymbol{\theta}^\star)^\top (X^\top X) (\hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}} - \boldsymbol{\theta}^\star) \right] \\ &= \mathbb{E} \left[(X(X^\top X + \lambda \text{Id}_p)^{-1} X^\top \boldsymbol{\epsilon})^\top (X(X^\top X + \lambda \text{Id}_p)^{-1} X^\top \boldsymbol{\epsilon}) \right] \\ &\quad + \lambda^2 \boldsymbol{\theta}^{\star\top} (X^\top X + \lambda \text{Id}_p)^{-2} \boldsymbol{\theta}^\star \\ &= \sum_{i=1}^{\text{rg}(X)} \frac{s_i^4 \sigma^2}{(s_i^2 + \lambda)^2} + \lambda^2 \boldsymbol{\theta}^{\star\top} (X^\top X + \lambda \text{Id}_p)^{-2} \boldsymbol{\theta}^\star \end{aligned}$$

Rem: $\lim_{\lambda \rightarrow 0} R_{\text{pred}}(\boldsymbol{\theta}^\star, \hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}}) = \text{rg}(X) \sigma^2$, $\lim_{\lambda \rightarrow \infty} R_{\text{pred}}(\boldsymbol{\theta}^\star, \hat{\boldsymbol{\theta}}_\lambda^{\text{rdg}}) = \|X\boldsymbol{\theta}^\star\|_2^2$

Bias / Variance: simulated example



$$X \in \mathbb{R}^{51 \times 50}, \boldsymbol{\theta}^* = (2, 2, 2, 2, 2, 0, \dots, 0)^\top$$

Syllabus

Ridge Definitions

- SVD point of view

- Penalty point of view

- Bias analysis with the SVD

- Variance analysis with the SVD

Regularization parameter choice

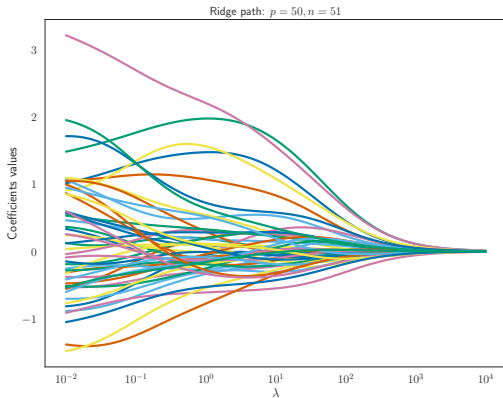
- Regularization path

- Cross-Validation (CV)

Algorithms and computational aspects

Choosing λ

```
n_features = 50; n_samples = 50
X = np.random.randn(n_samples, n_features)
theta_true = np.zeros([n_features, ])
theta_true[0:5] = 2.
y_true = np.dot(X, theta_true)
y = y_true + 1. * np.random.rand(n_samples,)
```



Syllabus

Ridge Definitions

- SVD point of view

- Penalty point of view

- Bias analysis with the SVD

- Variance analysis with the SVD

Regularization parameter choice

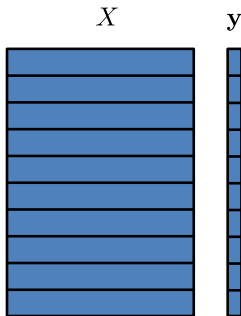
- Regularization path

- Cross-Validation (CV)

Algorithms and computational aspects

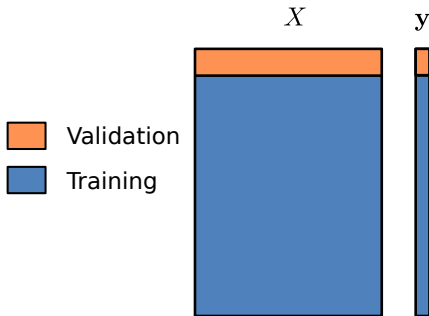
K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):



K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):

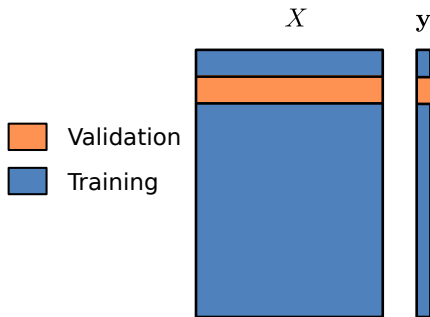


$k = 1$

1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

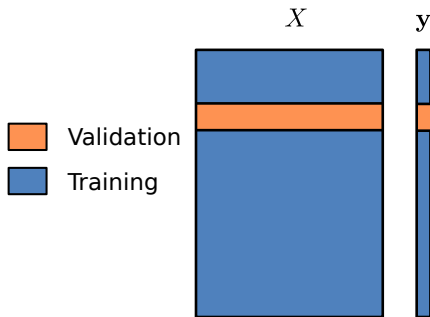
- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):



1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

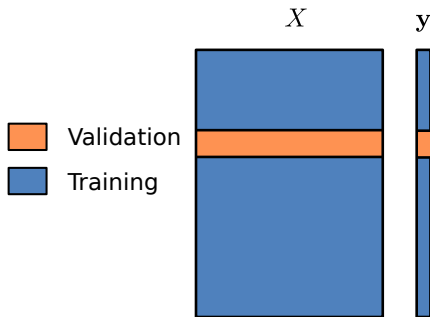
- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):



- $k = 3$
1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
 2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):

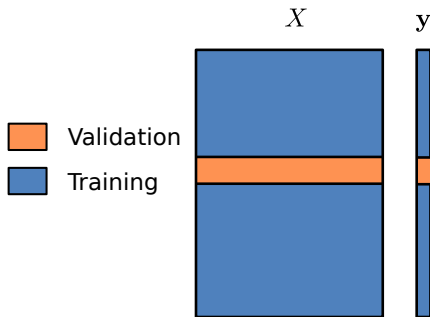


$$k = 4$$

1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):

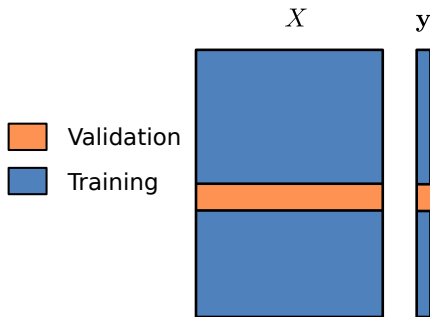


$k = 5$

1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

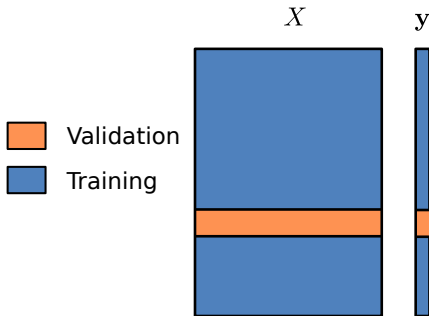
- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):



- $k = 6$
1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
 2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

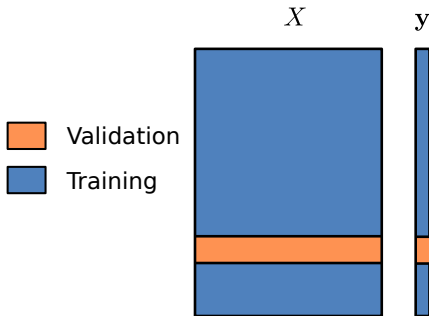
- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):



1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

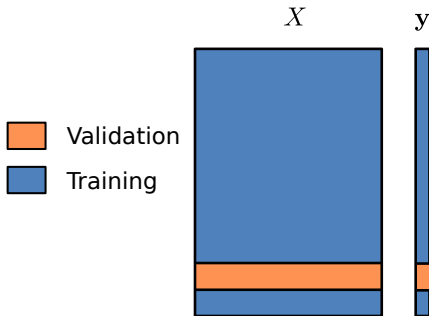
- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):



- $k = 8$
1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
 2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):

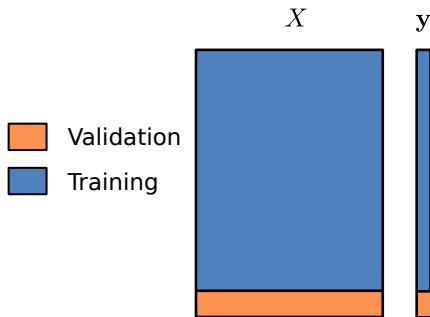


$k = 9$

1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, \mathbf{y}) into K blocks (sample-wise):

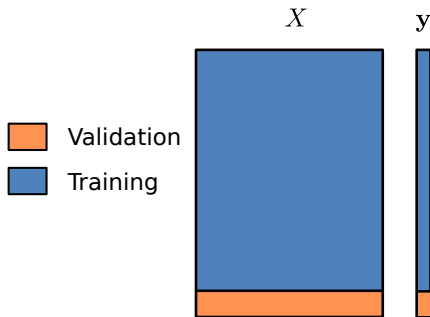


$k = 10$

1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, y) into K blocks (sample-wise):



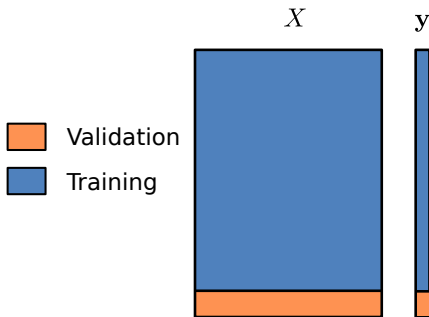
$k = 10$

1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

Parameter choice: compute $\widehat{\text{Error}}_1, \dots, \widehat{\text{Error}}_r$, average the errors and choose $\hat{i}^{\text{CV}} \in \llbracket 1, r \rrbracket$ achieving the smallest one

K -fold Cross-Validation ($K = 10$)

- ▶ Choose a grid of r λ 's to test: $\lambda_1, \dots, \lambda_r$
- ▶ Divide (X, y) into K blocks (sample-wise):



1. Compute with the training part the estimators for $\lambda_1, \dots, \lambda_r$:
 $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Evaluate the (prediction) error $\text{Error}_1^k, \dots, \text{Error}_r^k$ over the validation part,

Parameter choice: compute $\widehat{\text{Error}}_1, \dots, \widehat{\text{Error}}_r$, average the errors and choose $\hat{i}^{\text{CV}} \in \llbracket 1, r \rrbracket$ achieving the smallest one

Re-calibration: compute $\hat{\theta}^{\lambda_{\hat{i}^{\text{CV}}}}$ this time over the whole sample

CV in practice

Extreme cases of CV

- ▶ $K = 1$ impossible, needs $K = 2$
- ▶ $K = n$, “leave-one-out” strategy (cf. **Jackknife**): as many blocks as observations
Rem: $K = n$ (often) computationally efficient but instable

Practical advice:

- ▶ “randomise the sample” : having samples in random order avoid artifacts block (each fold needs to be representative of the whole sample!)
- ▶ standard choices: $K = 5, 10$

Alternatives: random partition validation/test, time series variants, etc. http://scikit-learn.org/stable/modules/cross_validation.html

Rem: in prediction the best predictors can be averaged/combined instead of recomputing an estimators over the whole set

CV variants sklearn

Crucial points: the structures train/test artificially created should represent faithfully the underlying learning problem

Classical alternatives:

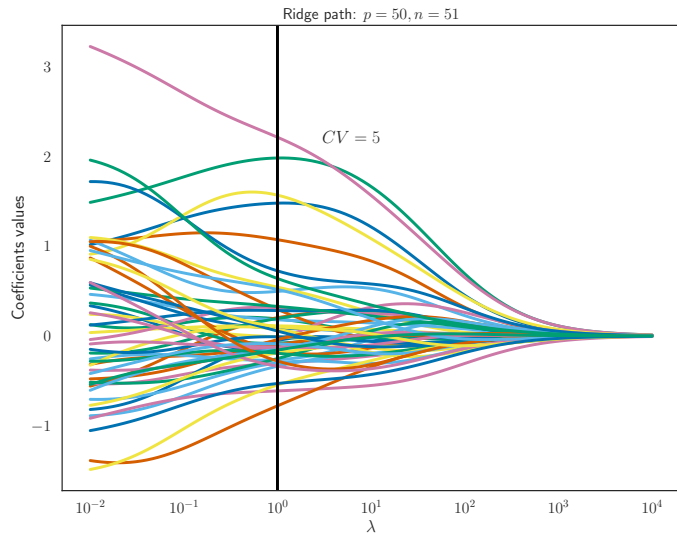
- ▶ random partitioning in train/test sets
(cf. `train_test_split`)
- ▶ Time series variant: `TimeSeriesSplit` (never predict the past with future information)
- ▶ For classification tasks with unbalanced classes
`StratifiedKFold`

Rem: averaging estimators (with weights reflecting their performance) is also relevant for prediction

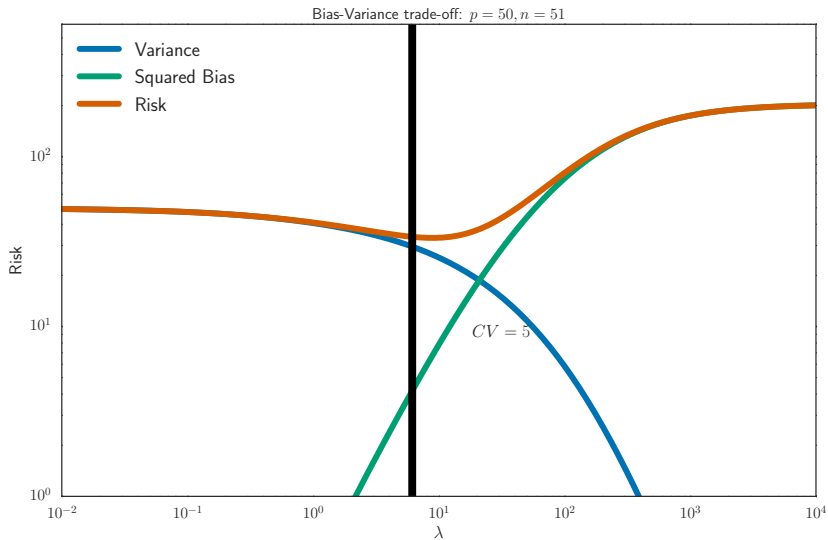
More details:

http://scikit-learn.org/stable/modules/cross_validation.html

Choosing λ : example with $CV = 5$ (I)



Choosing λ : example with $CV = 5$ (II)



Algorithms to compute the *Ridge* estimator

- ▶ 'svd': most stable method, useful for computing many λ 's cause the SVD price is paid only once
- ▶ 'cholesky' : matrix decomposition leading to a close form solution `scipy.linalg.solve`
- ▶ 'sparse_cg': conjugate gradient descent, useful also for sparse cases and high dimension (set `tol/max_iter` to a small value)
- ▶ stochastic gradient descent approaches : if n is huge

cf. the code of `Ridge`, `ridge_path`, `RidgeCV` in the module `linear_model` of `sklearn`

Rem: it is rare to compute the *Ridge* estimator only for one single λ

Rem: crucial issue of computing SVD for huge matrices...

References I

- ▶ G. H. Golub and C. F. van Loan.
Matrix computations.
Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.
- ▶ M. Y. Park.
Generalized linear models with regularization.
PhD thesis, Stanford University, 2006.