Sparsity in Learning with the LASSO: Least Absolute Shrinkage and Selection Operator

Binh Nguyen – Telecom Paris

M2DS Alternants Research Seminar Course; 07/04/2022

Outline

Some Practical Info for this Course

- Each Thursday from now (April) to end of June
- First two sessions for each theme is lecture + having fun with practical Python (technically Jupyter) notebook for coding
- For interaction during classes: it's better for you to think as well, so prepare for some derivations/questions
- Four presentations as grading, final note is the average
- Advice: you should start working with the assigned paper as early as possible, because you might have some questions related to the paper and can check with me in the second session for each of the topics

Some Practical Info for this Course

Notice that:

- Group ordering will change for each topic
- 7 people - 3 groups (2-2-3) – expect that the group with 3 people will have to present more
- For now the plan for topic will be: Sparse regression with Lasso, Optimal Transport with Gromov-Wasserstein distance, and Flow-based Model (Normalizing Flows)

Overview

- Supervised learning with Linear Model
- The Lasso
- Solving Lasso with non-smooth optimization: proximal operators

Supervised Learning

- $y \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^p, \varepsilon \sim \mathcal{N}(0, \sigma^2), f : \mathbb{R}^p \to \mathbb{R}$
- General relationship

$$y = f(\mathbf{x}) + \varepsilon$$

- Goal: learn f from some realizations of $(\mathbf{x}, y)$

Linear Regression

$$y = f(\mathrm{x}) + \varepsilon$$

Goal: learn f from some realizations of (x, y)

- Assumption: f is linear:

$$f(\mathrm{x}) = \mathrm{x}^\top \boldsymbol{\beta}$$

- Question: what should we target now for learning f?

Linear Regression

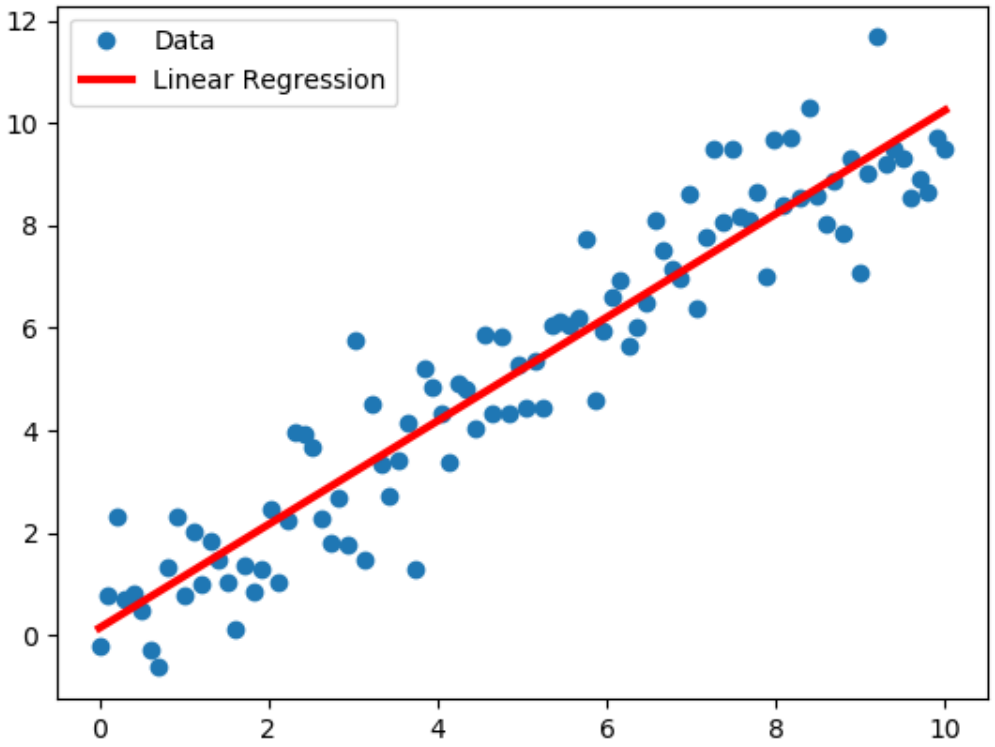
$$y = f(\mathrm{x}) + \varepsilon$$

Goal: learn f from some realizations of (x, y)

- Assumption: f is linear:
$$f(\mathrm{x}) = \mathrm{x}^\top \boldsymbol{\beta}$$

- Question: what should we target now for learning f?
- Answer: Estimate $\boldsymbol{\beta}$ from some realizations of (x, y)
- Question: how can we estimate $\boldsymbol{\beta}$ given data $(\mathrm{x}_i, y_i)_{i=1}^n$?

Linear Least Squares



$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{1}{2} (\mathrm{y} - \mathrm{x}^{\top} \boldsymbol{\beta})^2$$

Maximum (Log) Likelihood?

$$y = f(\mathbf{x}) + \varepsilon$$

- For one sample, define the likelihood:

$$\mathbb{P}(y = y_{obs} \mid \mathbf{x} = \mathbf{x}_{obs}) = \mathbb{P}(\varepsilon = y_{obs} - \mathbf{x}_{obs}^{\top}\boldsymbol{\beta})$$

Maximum (Log) Likelihood?

$$y = f(\mathbf{x}) + \varepsilon$$

- For one sample, define the likelihood:

$$\begin{aligned}\mathbb{P}(y = y_{obs} \mid \mathbf{x} = \mathbf{x}_{obs}) &= \mathbb{P}(\varepsilon = y_{obs} - \mathbf{x}_{obs}^\top \boldsymbol{\beta}) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_{obs} - \mathbf{x}_{obs}^\top \boldsymbol{\beta})^2}{\sigma^2}\right)\end{aligned}$$

Maximum (Log-)Likelihood?

- For n *i.i.d.* samples $(\mathtt{x}_i, y_i)_{i=1}^n$, the likelihood:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}) &\overset{\text{def.}}{=} \mathbb{P}(y_1, \ldots, y_n \mid \mathtt{x}_1, \ldots \mathtt{x}_n) \\ &= \mathbb{P}(y_1 \mid \mathtt{x}_1) \times \cdots \times \mathbb{P}(y_n \mid \mathtt{x}_n)\end{aligned}$$

Maximum Log-likelihood Estimatior:

$$\boldsymbol{\beta}_{MLE} = \operatorname*{argmax}_{\boldsymbol{\beta}} \log \mathcal{L}(\boldsymbol{\beta})$$

Maximum (Log-)Likelihood?

- For n *i.i.d.* samples $(\mathrm{x}_i, y_i)_{i=1}^n$, the likelihood:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}) &\overset{\text{def.}}{=} \mathbb{P}(y_1, \ldots, y_n \mid \mathrm{x}_1, \ldots \mathrm{x}_n) \\ &= \mathbb{P}(y_1 \mid \mathrm{x}_1) \times \cdots \times \mathbb{P}(y_n \mid \mathrm{x}_n)\end{aligned}$$

Maximum Log-likelihood Estimatior:

$$\boldsymbol{\beta}_{MLE} = \operatorname*{argmax}_{\boldsymbol{\beta}} \log \mathcal{L}(\boldsymbol{\beta}) = \operatorname*{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} \frac{1}{2}(\mathrm{y} - \mathrm{x}^\top \boldsymbol{\beta})^2$$

Matrix Notations

- $\mathrm{X} = [\mathrm{x}_1, \cdots, \mathrm{x}_n]^\top \in \mathbb{R}^{n \times p}, \mathrm{y} \in \mathbb{R}^n$
- $\sum_{i=1}^n \frac{1}{2}(\mathrm{y} - \mathrm{x}^\top \boldsymbol{\beta})^2 = \frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2$
- ℓ_2 norm of a vector: $\|\mathrm{v}\| = \sqrt{\sum_{i=1}^n v_n^2}$

Closed-form Solution in low-dimension

Objective:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2$$

- Question: low-dimension: $n > p$ – what is the closed-form?

Closed-form Solution in low-dimension

Objective:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2$$

- Question: low-dimension: $n > p$ – what is the closed-form?

$$\hat{\boldsymbol{\beta}} = (\mathrm{X}^\top \mathrm{X})^{-1} \mathrm{X}^\top \mathrm{y}$$

Maximum Likelihood Estimator/Linear Least Square

Objective:

$$\hat{\boldsymbol{\beta}}_{MLE} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2$$

- Pros:
  - Consistent: $\lim_{n\to\infty} \hat{\boldsymbol{\beta}}_{MLE} = \boldsymbol{\beta}^*$.
  - Clear statistical framework.
- Cons:
  - Can behave badly when f is misspecified (*i.e.* f is not linear).
  - Ill-posed when $n < p$: $\mathrm{X}^\top \mathrm{X}$ is not invertible.

So... what do we do when $n < p$?

Ockham's razor: only a few coefficients in $\boldsymbol{\beta}$ are important

- Or: induce some sparsity on $\hat{\boldsymbol{\beta}}$
- ℓ_0 (pseudo)-norm: number of non-zero coefficients in $\boldsymbol{\beta}$

$$\ell_0(\boldsymbol{\beta}) \overset{\text{def.}}{=} \sum_{i=1}^{n} \mathbb{1}_{\beta_i \neq 0} = |\text{supp}(\beta)|$$

Subset Selection/Matching Pursuit

$$\boldsymbol{\beta}_{subset} \stackrel{\text{def.}}{=} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq t$$

where $t > 0$ is an integer controlling the sparsity of the solution

Subset Selection/Matching Pursuit

$$\boldsymbol{\beta}_{subset} \stackrel{\text{def.}}{=} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq t$$

where $t > 0$ is an integer controlling the sparsity of the solution

Problems:

- Non-convexity
- Instability: adding a single sample may completely change $\boldsymbol{\beta}$ and hence its support
- NP-Hard to solve (*i.e.* very long time to solve)

Least Absolute Shrinkage and Selection Operator

- Idea: relax non-convexity from the ℓ_0 norm to a convex problem

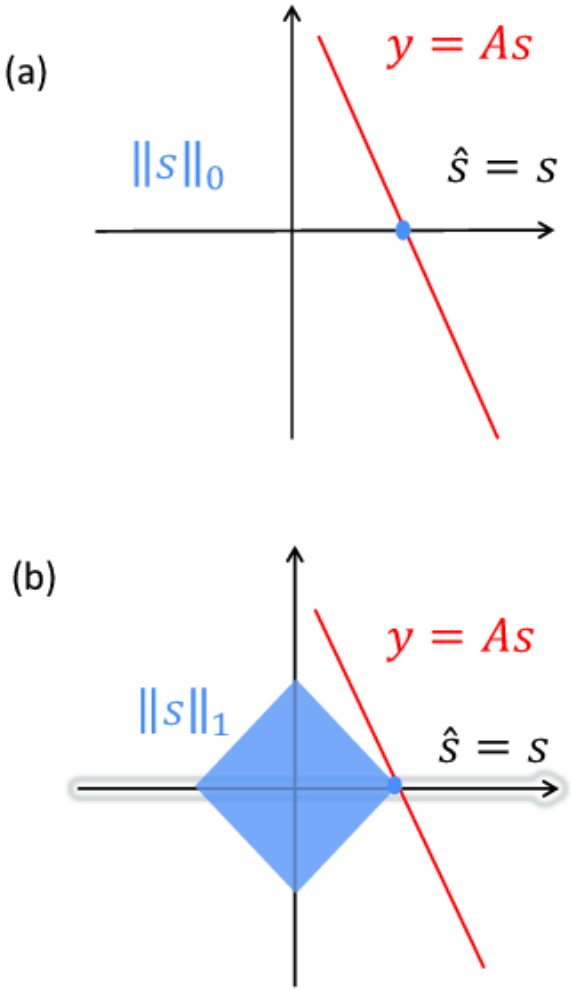
$$\|\boldsymbol{\beta}\|_1 \overset{\text{def.}}{=} \sum_{i=1}^{n} |\beta_i|$$

- Which makes

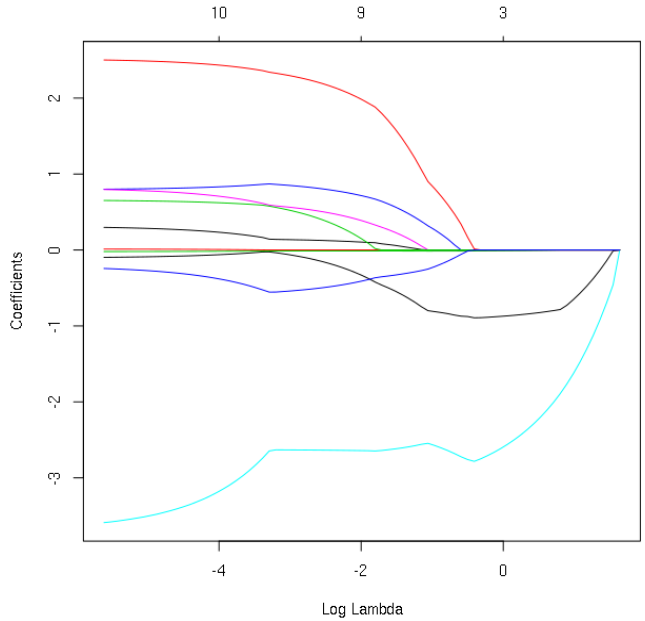
$$\boldsymbol{\beta}_{lasso} \overset{\text{def.}}{=} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2} \|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2$$

$$\text{subject to} \quad \|\boldsymbol{\beta}\|_1 \leq t$$

where $t > 0$ controls the sparsity of the solution



Lagrangian Formulation



$$\boldsymbol{\beta}_{lasso} \overset{\text{def.}}{=} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

where $\lambda > 0$ controls the sparsity of the solution

- Promote sparsity: there is a threshold λ_{max} such that $\lambda > \lambda_{max}$ implies $\beta_{lasso} = 0$
- Question: how to find this λ_{max}?

Finding λ_{max}

Reminder: the sub-gradient ∂f: a vector $g \in \mathbb{R}^p$ is a subgradient of $f : \mathbb{R}^p \to \mathbb{R}$ at x if

Finding λ_{max}

Reminder: the sub-gradient ∂f: a vector $g \in \mathbb{R}^p$ is a subgradient of $f : \mathbb{R}^p \to \mathbb{R}$ at x if

$$f(y) \geq f(x) + g^\top (y - x)$$

- Idea: $\beta = 0$ is the solution to the lasso iff:

$$0 \in \partial_{\beta=0} \left(\frac{1}{2} \|y - X\beta\|^2 + \lambda \|\beta\|_1 \right)$$

Finding λ_{max}

Reminder: the sub-gradient ∂f: a vector $g \in \mathbb{R}^p$ is a subgradient of $f : \mathbb{R}^p \to \mathbb{R}$ at x if

$$f(\mathrm{y}) \geq f(\mathrm{x}) + \mathrm{g}^\top(\mathrm{y} - \mathrm{x})$$

- Idea: $\boldsymbol{\beta} = 0$ is the solution to the lasso iff:

$$0 \in \partial_{\boldsymbol{\beta}=0} \left(\frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \right)$$

- We know that

$$\partial \left(\frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 \right) = \mathrm{X}^\top(\mathrm{y} - \mathrm{X}\boldsymbol{\beta})$$

$$\partial_{\boldsymbol{\beta}=0} \left(\lambda\|\boldsymbol{\beta}\|_1 \right) = [-\lambda, \lambda]^p$$

- So: for every $i \in [p]$: $(\mathrm{X}^\top \mathrm{y})_i \in [-\lambda, \lambda]$

Finding λ_{max}

Reminder: the sub-gradient ∂f: a vector $g \in \mathbb{R}^p$ is a subgradient of $f : \mathbb{R}^p \to \mathbb{R}$ at x if

$$f(\mathrm{y}) \geq f(\mathrm{x}) + \mathrm{g}^\top(\mathrm{y} - \mathrm{x})$$

- Idea: $\boldsymbol{\beta} = 0$ is the solution to the lasso iff:

$$0 \in \partial_{\boldsymbol{\beta}=0} \left(\frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \right)$$

- We know that

$$\partial \left(\frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 \right) = \mathrm{X}^\top(\mathrm{y} - \mathrm{X}\boldsymbol{\beta})$$

$$\partial_{\boldsymbol{\beta}=0} \left(\lambda\|\boldsymbol{\beta}\|_1 \right) = [-\lambda, \lambda]^p$$

- So: for every $i \in [p]$: $(\mathrm{X}^\top \mathrm{y})_i \in [-\lambda, \lambda]$

$$\lambda_{max} = \|\mathrm{X}^\top \mathrm{y}\|_\infty$$

Why sparsity?

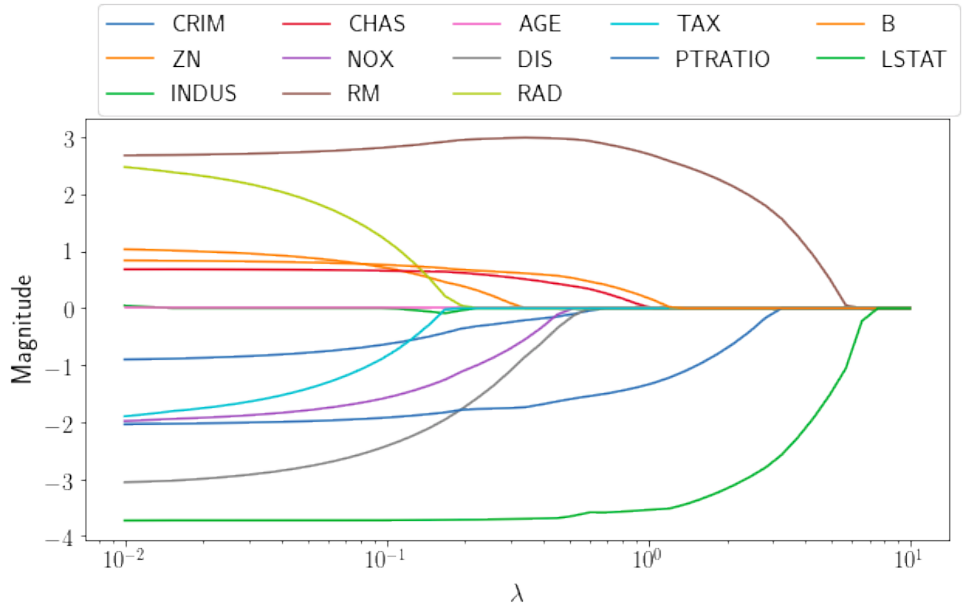


Figure: Lasso Path on Boston dataset

- Perform *model selection* and *estimation* at the same time: which variables in x are important
- Sparsity = faster computation and solvers

Why sparsity?

- Perform *model selection* and *estimation* at the same time: which variables in x are important
- Sparsity = faster computation and solvers
  - Example: $p = 1000, n = 1000$, computing $\mathrm{X}\boldsymbol{\beta}$ takes approx. $n \times p = 10^6$ operations
  - But: if we know only 10 of the coefficients in $\boldsymbol{\beta}$ is non-zero, it takes only 10^4 operations – or 100 times faster

The Simplest Lasso

- $n = 1, p = 1, x = 1$
- The problem reduces to

$$\min_{\beta} \frac{1}{2}(y - \beta)^2 + \lambda|\beta|$$

- Proximity operator:

$$\text{prox}_{|\cdot|}(y, \lambda) \overset{\text{def.}}{=} \underset{\beta}{\text{argmin}} \frac{1}{2}(y - \beta)^2 + \lambda|\beta|$$

- Solution: soft-thresholding

$$\text{st}(y, \lambda) \overset{\text{def.}}{=} \text{prox}_{|\cdot|}(y, \lambda) = \begin{cases} 0, & \text{if } |y| \leq \lambda \\ y - \lambda & \text{if } y > \lambda \\ y + \lambda & \text{if } y < -\lambda \end{cases}$$

Soft-Thresholding

Nice property: separability, if $x, \beta \in \mathbb{R}^p$

$$\text{prox}_{\|\cdot\|_1}(x, \lambda) = \underset{\beta}{\text{argmin}} \frac{1}{2}(x - \beta)^2 + \lambda|\beta| = (\text{st}(x_1, \lambda), \dots \text{st}(x_p, \lambda))$$

But How About p-dimension problem?

ISTA: Iterative Soft-Thresholding Algorithm

- Recall: $\boldsymbol{\beta}_{lasso} = \text{argmin}_{\beta} \frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2 + \lambda|\beta|$
- Gradient of the smooth term:

$$\nabla_{\boldsymbol{\beta}} \left(\frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2 \right) = \mathrm{X}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y})$$

- ISTA:

$$\boldsymbol{\beta}^{t+1} = \text{prox}_{\|\cdot\|_1}(\boldsymbol{\beta}^t - \eta \mathrm{X}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y}), \lambda\eta)$$

- Question: Where does this come from?

But How About p-dimension problem?

ISTA: Iterative Soft-Thresholding Algorithm

- Recall: $\boldsymbol{\beta}_{lasso} = \text{argmin}_{\beta} \frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2 + \lambda|\beta|$
- Gradient of the smooth term:

$$\nabla_{\boldsymbol{\beta}} \left(\frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2 \right) = \mathrm{X}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y})$$

- ISTA:

$$\boldsymbol{\beta}^{t+1} = \text{prox}_{\|\cdot\|_1}(\boldsymbol{\beta}^t - \eta \mathrm{X}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y}), \lambda\eta)$$

- Question: Where does this come from?

A Reminder: Convex Optimization 101

- Recall: Gradient descent, minimize $f(x)$ for a smooth f:

$$\text{Iterate } \mathbf{x}^{t+1} = \mathbf{x}^t - \eta \nabla f(\mathbf{x}^t)$$

- Its *quadratic surrogate*:

$$\mathbf{x}^{t+1} = \operatorname*{argmin}_{\mathbf{x}} f(\mathbf{x}^t) + \nabla f(\mathbf{x}^t)^\top (\mathbf{x} - \mathbf{x}^t) + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{x}^t\|_2^2$$

Back to ISTA...

- The quadratic surrogate + ℓ_1 regularization term:

$$\boldsymbol{\beta}^{t+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, f(\boldsymbol{\beta}^t) + \nabla f(\boldsymbol{\beta}^t)^\top (\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta}\|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- Take $f(\boldsymbol{\beta}) = \frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2$

Back to ISTA...

- The quadratic surrogate + ℓ_1 regularization term:

$$\begin{aligned}\boldsymbol{\beta}^{t+1} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, f(\boldsymbol{\beta}^t) + \nabla f(\boldsymbol{\beta}^t)^\top(\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta}\|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \nabla f(\boldsymbol{\beta}^t)^\top(\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta}\|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\end{aligned}$$

- Take $f(\boldsymbol{\beta}) = \frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2$

Back to ISTA...

- The quadratic surrogate + ℓ_1 regularization term:

$$\begin{aligned}\boldsymbol{\beta}^{t+1} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, f(\boldsymbol{\beta}^t) + \nabla f(\boldsymbol{\beta}^t)^\top(\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta}\|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \nabla f(\boldsymbol{\beta}^t)^\top(\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta}\|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \frac{1}{2\eta}\|\boldsymbol{\beta} - (\boldsymbol{\beta}^t - \eta\nabla f(\boldsymbol{\beta}^t))\|_2^2 + \lambda\eta\|\boldsymbol{\beta}\|_1\end{aligned}$$

- Take $f(\boldsymbol{\beta}) = \frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2$

Back to ISTA...

- The quadratic surrogate + ℓ_1 regularization term:

$$\begin{aligned}
\boldsymbol{\beta}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{\beta}} f(\boldsymbol{\beta}^t) + \nabla f(\boldsymbol{\beta}^t)^\top (\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta} \|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \\
&= \operatorname*{argmin}_{\boldsymbol{\beta}} \nabla f(\boldsymbol{\beta}^t)^\top (\boldsymbol{\beta} - \boldsymbol{\beta}^t) + \frac{1}{2\eta} \|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \\
&= \operatorname*{argmin}_{\boldsymbol{\beta}} \frac{1}{2\eta} \|\boldsymbol{\beta} - (\boldsymbol{\beta}^t - \eta \nabla f(\boldsymbol{\beta}^t))\|_2^2 + \lambda\eta \|\boldsymbol{\beta}\|_1 \\
&\overset{\text{def.}}{=} \operatorname{prox}_{\|\cdot\|_1}(\boldsymbol{\beta}^t - \mathrm{X}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y}), \lambda\eta)
\end{aligned}$$

- Take $f(\boldsymbol{\beta}) = \frac{1}{2}(y - \mathrm{X}\boldsymbol{\beta})^2$

Back to ISTA...

$$\boldsymbol{\beta}^{t+1} = \text{prox}_{\|\cdot\|_1}(\boldsymbol{\beta}^t - \eta \mathrm{X}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y}), \lambda\eta)$$

- Due to the `prox` operator many coefficients are 0
- One iteration takes $\mathcal{O}(\min(n, p) \times p)$

  $\longrightarrow$ problematic for large p

Coordinate Descent

Idea: freeze others, update only one coefficient β_j at each iteration, *i.e.*

$$\boldsymbol{\beta}_j^{t+1} = \text{prox}_{\|\cdot\|_1}(\boldsymbol{\beta}_j^t - \eta \mathrm{X}_{*,j}^\top(\mathrm{X}\boldsymbol{\beta} - \mathrm{y}), \lambda\eta)$$

$$\text{and } \beta_k^{t+1} = \beta_k^t \text{ for all } k \neq j$$

- Iteration cost $\mathcal{O}(n)$ if we know the *residual* $\mathrm{r} = \mathrm{X}\boldsymbol{\beta}^t - \mathrm{y}$
- Update residual: Once we update the coordinate j:

$$\mathrm{r} \leftarrow \mathrm{r} + (\beta_j^{t+1} - \beta_j^t)\mathrm{X}_{*,j}$$

A Bit More Advanced: FISTA - Fast(er) ISTA

Initialize $\boldsymbol{\beta}^1 = \mathrm{z}^1 = 0, \boldsymbol{\gamma}^1 = 1$, then for each update

- $\boldsymbol{\beta}_j^{t+1} = \mathrm{prox}_{\|\cdot\|_1}(\mathrm{z}^t - \eta \mathrm{X}^\top(\mathrm{X}\mathrm{z}^t - \mathrm{y}), \lambda\eta)$
- $\boldsymbol{\gamma}^{t+1} = \dfrac{1 + \sqrt{1 + 4(\gamma^t)^2}}{2}$
- $z^{t+1} = \boldsymbol{\beta}^{t+1} + \dfrac{\gamma^t - 1}{\gamma^{t+1}}(\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t)$

Some form of acceleration, which make convergence rate of fista $\mathcal{O}(1/T^2)$ instead of $\mathcal{O}(1/T)$ of ISTA

Parameter Selection for the Lasso

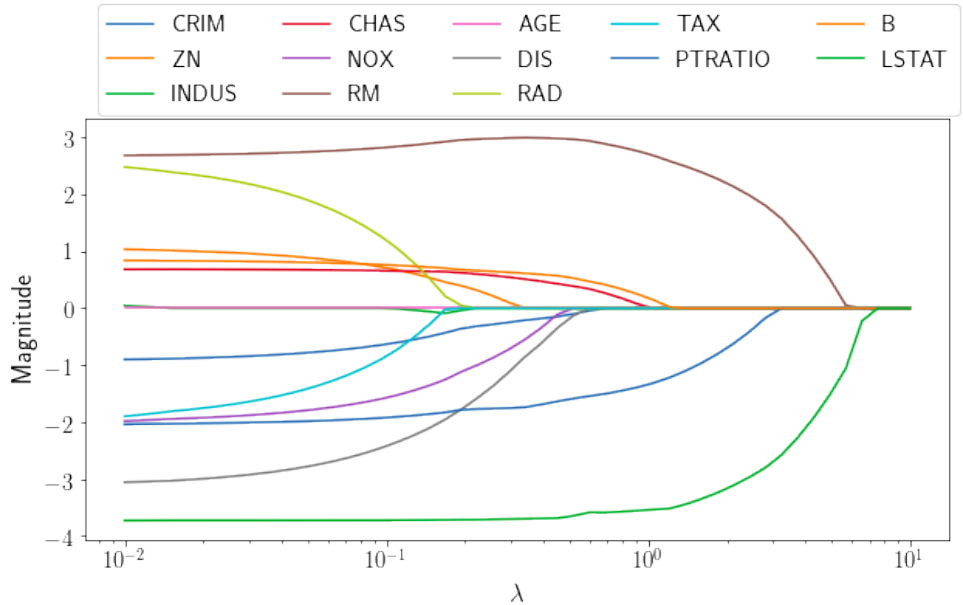


Figure: Lasso Path on Boston dataset

$$\boldsymbol{\beta}_{lasso} \overset{\text{def.}}{=} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{y} - \mathrm{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- We have talked about the role of λ as to control the sparsity level – but which sparsity level is optimal?

Selection of sparsity parameter λ

Caveat: to simplify things, definitions are taken from sklearn page – for a specific theoretical formula for the Lasso, check Zou et al. (2007)

- Equivalent to model selection: we can do that via a criterion
- Akaike's Information Criterion (AIC):

$$\mathrm{AIC}(\mathcal{L}(\hat{\boldsymbol{\beta}})) \overset{\text{def.}}{=} -2\log(\mathcal{L}(\hat{\boldsymbol{\beta}})) + 2d$$

where d is the degrees of freedom (number of parameters of models)

Zou, Hui, Trevor Hastie, and Robert Tibshirani. "On the degrees of freedom of the lasso." The Annals of Statistics 35.5 (2007): 2173-2192.

Selection of sparsity parameter λ

Caveat: to simplify things, definitions are taken from sklearn page – for a specific theoretical formula for the Lasso, check Zou et al. (2007)

- Equivalent to model selection: we can do that via a criterion
- Akaike's Information Criterion (AIC):

$$\text{AIC}(\mathcal{L}(\hat{\boldsymbol{\beta}})) \stackrel{\text{def.}}{=} -2\log(\mathcal{L}(\hat{\boldsymbol{\beta}})) + 2d$$

where d is the degrees of freedom (number of parameters of models)

- Bayesian Information Criterion (BIC):

$$\text{BIC}(\mathcal{L}(\hat{\boldsymbol{\beta}})) \stackrel{\text{def.}}{=} -2\log(\mathcal{L}(\hat{\boldsymbol{\beta}})) + \log(n)2d$$

- Question: What can we say about these two criteria as a function of the likelihood?

Zou, Hui, Trevor Hastie, and Robert Tibshirani. "On the degrees of freedom of the lasso." The Annals of Statistics 35.5 (2007): 2173-2192.

Selection of sparsity parameter λ

- Question: What's the formula for AIC and BIC in our settings?
  ($i.e.$ linear regression with with least squares loss and Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$?)

Selection of sparsity parameter λ

- Question: What's the formula for AIC and BIC in our settings? (*i.e.* linear regression with with least squares loss and Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$?)

$$\mathrm{AIC}(\hat{\boldsymbol{\beta}}_\lambda) = n \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \|\mathrm{y} - \mathrm{X}\hat{\boldsymbol{\beta}}_\lambda\|^2 + 2d$$

$$\mathrm{BIC}(\hat{\boldsymbol{\beta}}_\lambda) = n \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \|\mathrm{y} - \mathrm{X}\hat{\boldsymbol{\beta}}_\lambda\|^2 + \log(n)\, d$$

Selection of sparsity parameter λ

- Question: What's the formula for AIC and BIC in our settings? (*i.e.* linear regression with with least squares loss and Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$?)

$$\mathrm{AIC}(\hat{\boldsymbol{\beta}}_\lambda) = n \log(2\pi\sigma^2) + \frac{1}{\sigma^2}\|\mathrm{y} - \mathrm{X}\hat{\boldsymbol{\beta}}_\lambda\|^2 + 2d$$

$$\mathrm{BIC}(\hat{\boldsymbol{\beta}}_\lambda) = n \log(2\pi\sigma^2) + \frac{1}{\sigma^2}\|\mathrm{y} - \mathrm{X}\hat{\boldsymbol{\beta}}_\lambda\|^2 + \log(n)\, d$$

- The first term on the RHS is a constant, so:

$$\lambda_{AIC} \overset{\text{def.}}{=} \operatorname*{argmin}_\lambda \frac{1}{\sigma^2}\|\mathrm{y} - \mathrm{X}\hat{\boldsymbol{\beta}}_\lambda\|^2 + 2d$$

$$\lambda_{BIC} \overset{\text{def.}}{=} \operatorname*{argmin}_\lambda \frac{1}{\sigma^2}\|\mathrm{y} - \mathrm{X}\hat{\boldsymbol{\beta}}_\lambda\|^2 + \log(n)\, d$$

- Note: we also have to estimate σ^2 from the residual

Selection of sparsity parameter λ – with AIC/BIC

Pros:

- Relatively fast as the regularization path only computed once
- Guarantee to be optimal theoretically

Selection of sparsity parameter λ – with AIC/BIC

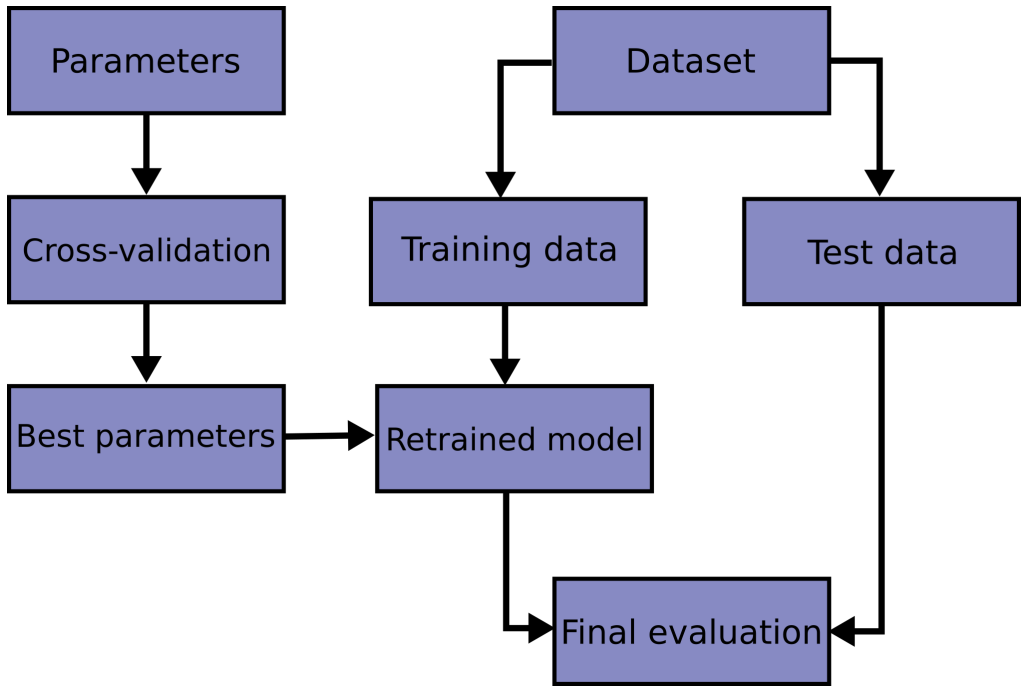
Pros:

- Relatively fast as the regularization path only computed once
- Guarantee to be optimal theoretically

Cons:

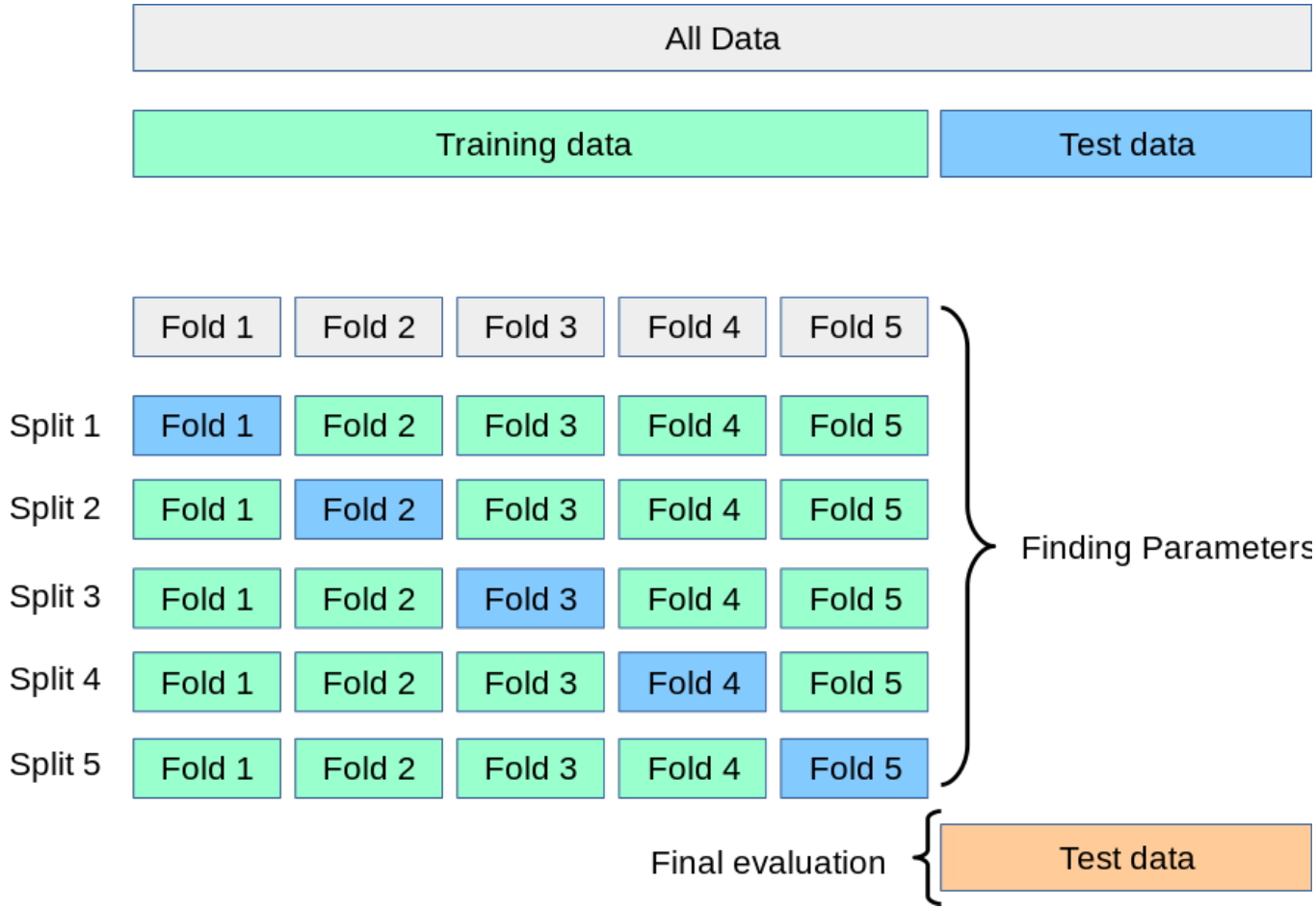
- Theoretical guarantees only with large n (asymptotic)
- Estimation of the term d (degrees of freedom) is tricky when $n \ll p$

In Practice: K-Folds Cross-Validation



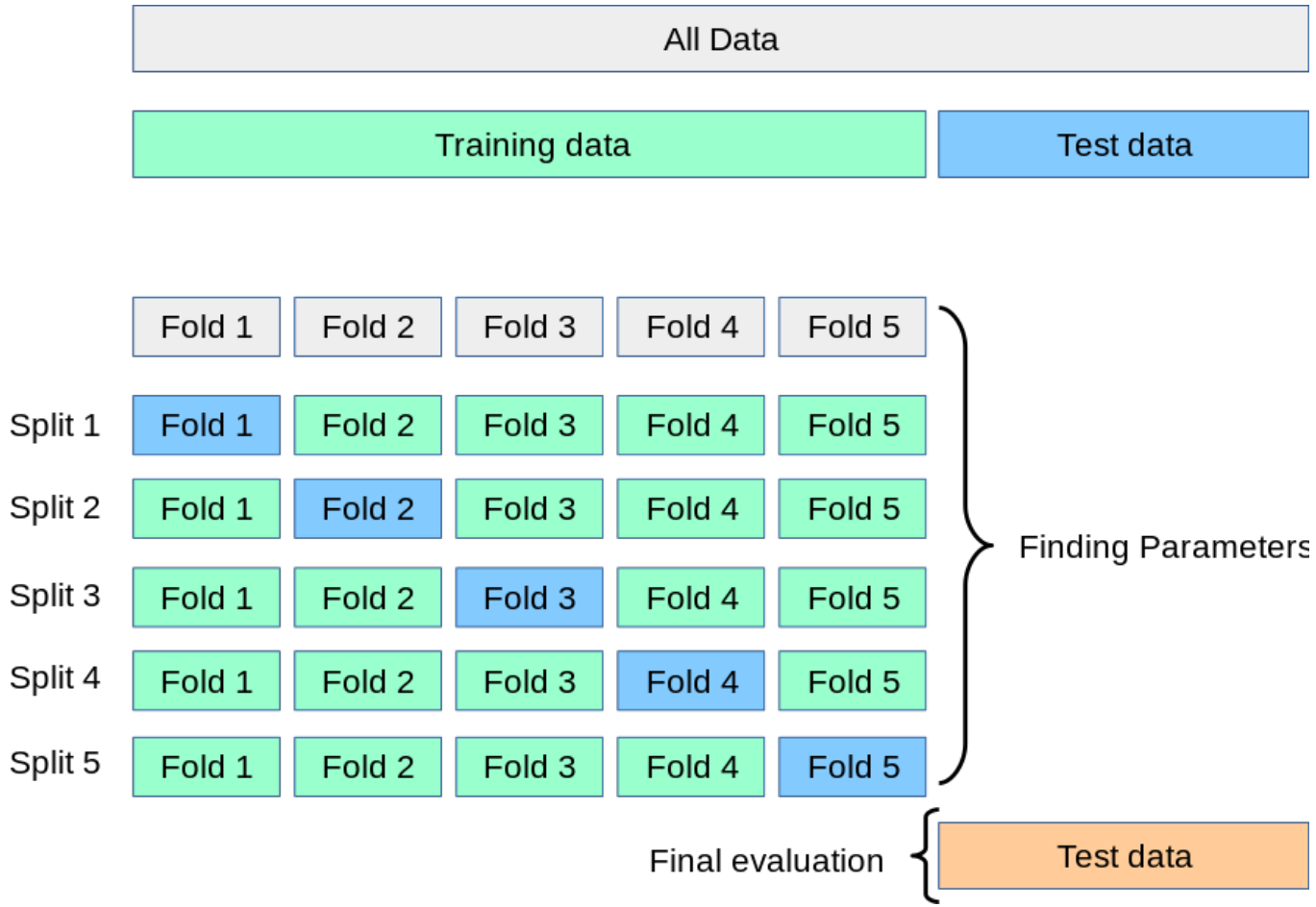
Cross-validation: based on `sklearn` page:
`scikit-learn.org/stable/modules/cross_validation.html`

In Practice: K-Folds Cross-Validation



- Performing model fitting on each (K-1) folds of the training data, evaluate on the remaining fold
- Which performance measure? (accuracy, MSE – mean squared errors, etc.)

In Practice: K-Folds Cross-Validation



- Easy with sklearn: `sklearn.model_selection` and even better: `sklearn.linear_model.LassoCV`

Next session

- Brief intro on a bit more advanced: hyper-parameter selection for Lasso with bi-level optimization
- Practical session with Jupyter notebook: playing with `sklearn.linear_model.Lasso`, etc. , and some optimization scheme