Inmas Machine Learning Workshop

Instructor: Christian Kuemmerle

TAs:
- Lanlan Ji
- Vittorio Loprinzo
- Salma Tarmoun
- Sichen Yang

Goals:

- ▷ Learn to use computational tools to learn from data
  - ↳ Intuition for their power & challenges
- ▷ Gain intuition of geometry of high-dimensional spaces
- ▷ Learn to use Python to appy these techniques

What is Data Science?

[Tukey '62]: "Data Analysis" as an empirical science:

- Procedures for gathering data, interpret data
- Uses mathematical statistics
- "reliance upon the test of experience as ultimate standard of validity"

Our focus: Prediction instead of Inference. ← focus of "statistics"

Last 15-20 years: Led to technological advances in pattern/image recognition, machine translation, targeted advertisement, (semi-) autonomous cars ...

"machine Learning", "artificial intelligence (AI)"

Common Task Framework: [Liberman '10, Donoho]

- Public "training" dataset: List of observations with labels
- Competitors with common task to infer label prediction rule from training data
- $\xrightarrow{\text{Submit to}}$ Referee, reports accuracy of prediction rule applied to (hidden) testing dataset

Ex: · $1M Netflix prize (2006-2008)
· ImageNet

Other aspects: · Available computer hardware
· Software | communication frameworks facilitating reproducibility

Goal: Learn from data.

a) **Ex.:** - Predict salary of professors from employment data
- Detect spam e-mails based on large set of spam/non-spam emails.

Supervised Learning

b) "Learning without teacher", find meaningful data representation / summary

Ex: - Find categories among pictures on phone
- Visualize complex genetics data to be interpreted by humans

Unsupervised Learning

The Framework of Statistical Learning

- $\mathcal{X} \subset \mathbb{R}^k$: domain set (e.g. space of images with fixed pixel number)
- $\mathcal{Y} \subset \mathbb{R}^q$: target set | set of labels (e.g. $\mathcal{Y} = \{0,1\}$ if spam/non-spam classification
- Let $\mathcal{D}$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$

  Assume we are given a training set $S := (x_i, y_i)_{i=1}^{n} \overset{i.i.d.}{\sim} \mathcal{D}$
- Goal: Find a predictor | classifier function $h : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected risk $L_{\mathcal{D}}(h) := \mathbb{E}_{\mathcal{D}}\left[\ell(Y, h(X))\right]$ where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{Z}$ is a given loss/error function

- Learning Algorithm:

  Specific algorithm that maps S to a specific member function $\hat{h}_n \in \mathcal{F}$ of a $\boxed{\text{hypothesis space } \mathcal{F}} \subset \{h : \mathcal{X} \to \mathcal{Y}\}$ based on the information of S.

Many learning algorithms: $\hat{h}_n = \boxed{\underset{h \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i))}$

<u>Empirical Risk Minimization</u>

<u>Ex:</u> · Linear Regression: ▷ $\ell(y, z) = (y - z)^2$

▷ $\mathcal{F} := \{ x \mapsto \underline{\underline{\beta_0}} + \langle x, \beta \rangle, \ \beta_0 \in \mathbb{R}, \ \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \in \mathbb{R}^n \}$

<u>Bias - Variance Tradeoff:</u>

$$\underbrace{L_{\mathcal{D}}(\hat{h}_n) - \min_h L_{\mathcal{D}}(h)}_{\text{generalization error}} = \underbrace{\left(L_{\mathcal{D}}(\hat{h}_n) - L_{\mathcal{D}}(h_{\mathcal{F}}) \right)}_{\text{estimation error}} + \underbrace{\left(L_{\mathcal{D}}(h_{\mathcal{F}}) - \min_h L_{\mathcal{D}}(h) \right)}_{\text{approximation error}}$$

$h_{\mathcal{F}} = \underset{h \in \mathcal{F}}{\operatorname{argmin}} L_{\mathcal{D}}(h)$

estimation error:

▷ induced by choice / size of S

▷ For fixed sample size $|S|$, Larger if $\mathcal{F}$ large

approximation error:

▷ smaller if $\mathcal{F}$ large

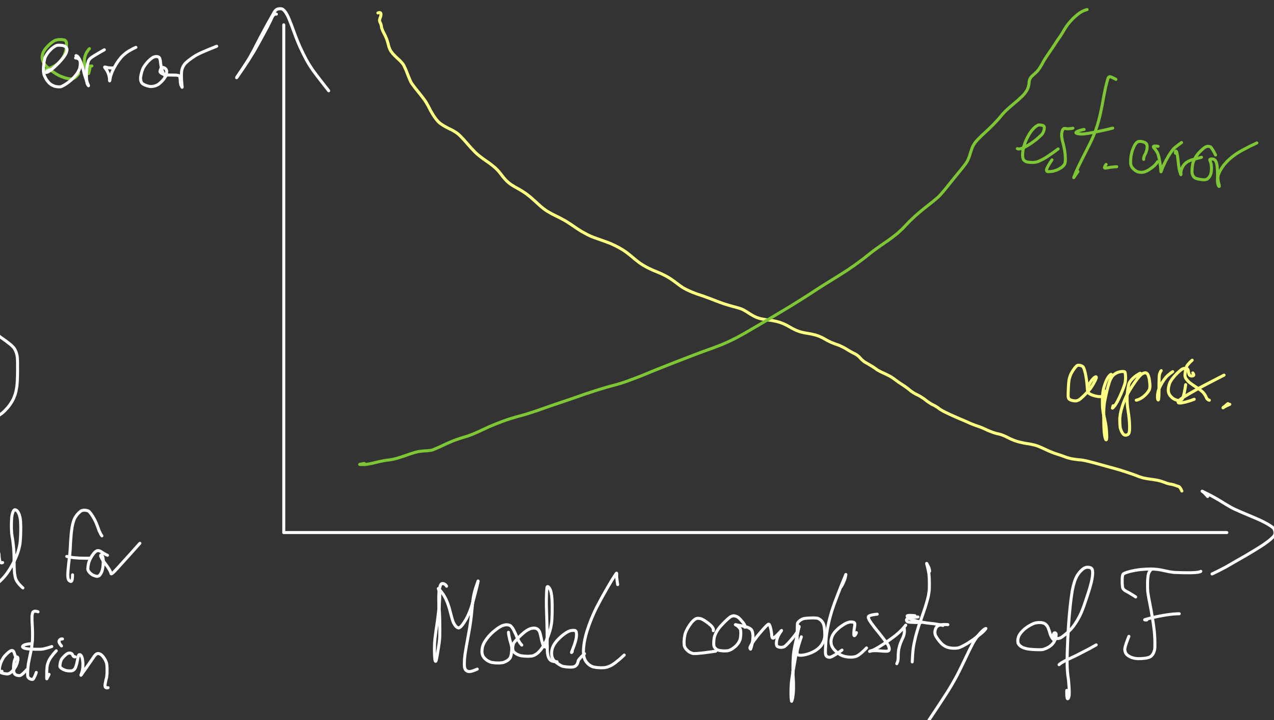
Cross Validation:

Goal: Reduce bias induced by training set S.

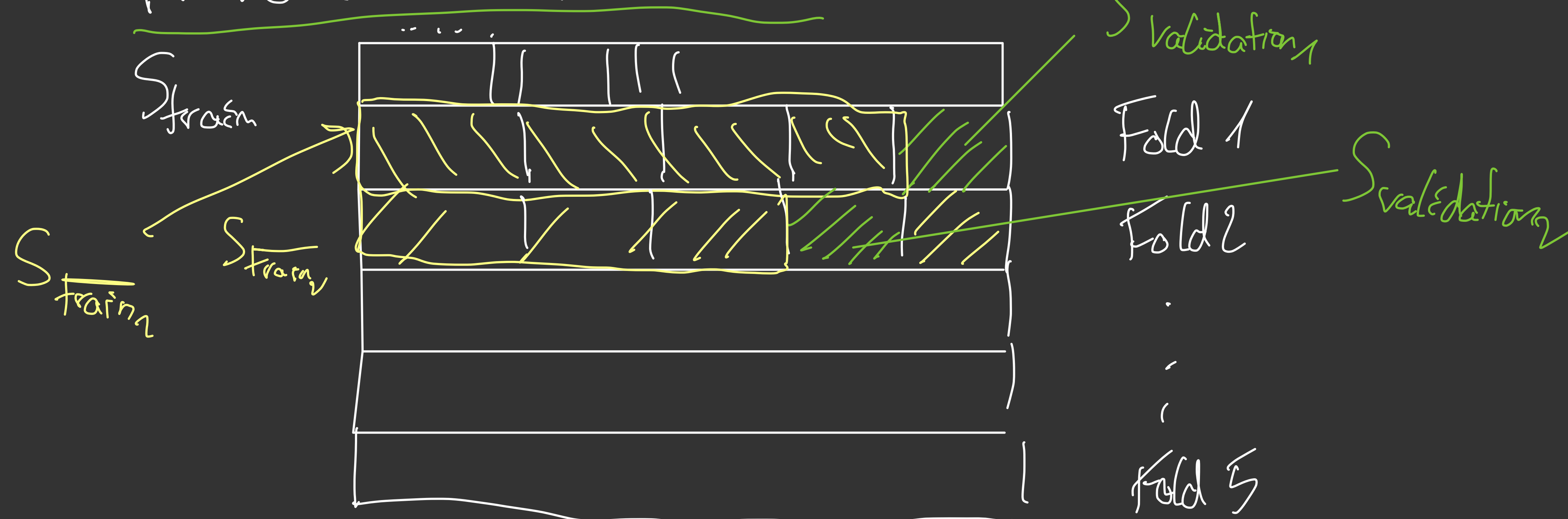
▷ Fair comparison only if: S_{train} , S_{test} $\sim \mathcal{D}$

S_{train}: used in learning method

S_{test}: used for evaluation



▷ K-Fold Cross validation



Controlling Model Complexity via Regularization

"Regularized loss minimization" (RLM)

$$\hat{h}_n := \boxed{\operatorname*{argmin}_{h \in \mathcal{F}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i)) + \lambda \cdot R(h) \right\}$$

$R(h)$ a term that quantifies "complexity" of $h \in \mathcal{F}$

$\lambda > 0$

Ridge Regression: Linear Regression with $R(h) = \|\beta\|_2^2$

- Choose $\mathcal{F}$ space of linear functions

$$R(h) := \|\beta(h)\|_2^2$$

$$\Rightarrow \hat{\beta}_n := \beta(\hat{h}_n) = \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n} \|X\beta - y\|_2^2 + \lambda \|\beta\|_2^2$$

$$= (X^T X + \lambda I)^{-1} X^T (y)$$

▷ If $\lambda = 0$: $\longrightarrow$ Linear regression

▷ If $\lambda \to \infty$: coefficients $\hat{\beta}$ "shrinked to 0".

▷ λ in between: balancing fit of linear model and size of coefficents.

- Strong connection to hypothesis set $\mathcal{F}_t := \{h : \mathbb{R} \to \mathbb{R} : h(x) = \langle \beta, x \rangle \text{ s.t. } \underbrace{\|\beta\|_2 \leq t}_{\text{bound on coefficents}}\}$

Cross Validation: Idea: Reduce bias induced by training set S.

▷ Fair comparisons only if: S_{train}, $S_{test} \sim \mathcal{D}$

S_{train}: used in learning method

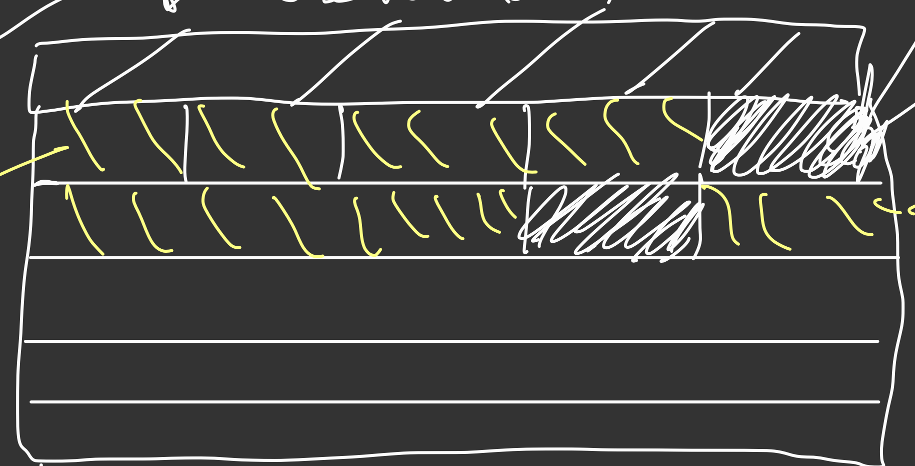
S_{test}: used for evaluation only

▷ K-Fold Crossvalidation

S_{train}

S_{train_1}

$S_{validation_1}$



Fold 1

Fold 2

⋮

Fold 5

$\Rightarrow$ Used for "Model Selection"

Controlling Model Complexity via Regularization

▷ Modify learning algorithm for some hypothesis space $\mathcal{F}$: For some $\lambda > 0$,

$$\hat{h}_n := \underset{h \in \mathcal{F}}{\operatorname{argmin}} \{ \underbrace{L_S(h)}_{\text{loss term}} + \underbrace{\lambda R(h)}_{\text{regularization term}} \}$$

, $R(h)$ a term that quantifies "complexity" of $h \in \mathcal{F}$.

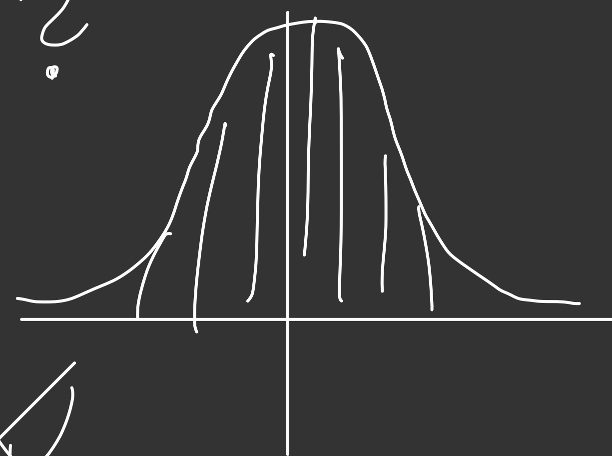
"regularized loss minimization" (RLM),

▷ Compared to ERM, term $\lambda R(h)$ penalizes too complex instances of $\mathcal{F}$.

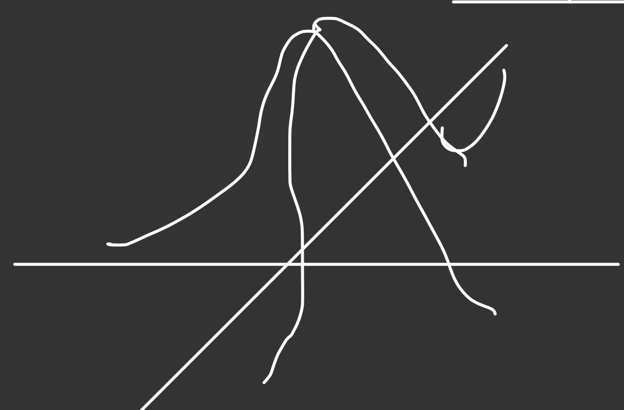
High Dimensional Geometry

Q: How do $x_1, x_2 \in \mathbb{R}^d$, $d \gg 1$ relate to each other when "generic"?

- 1-dim Gaussian:



$d = 2$



- d-dim Gaussian:

$\|X\| \sim \sqrt{d} \pm O(1)$

$\Rightarrow$ All very far from origin!

- If $x_1, x_2 \in \mathbb{R}^d$ indep. d-dim Gaussian

$\Rightarrow \|x_1 - x_2\|_2 \sim$ close to const. & large.

Preprocessing: If domain set s.t. $X \subset \mathbb{R}^k$, we can define a feature map $\phi : X \to \tilde{X} \subset \mathbb{R}^\ell$ (often with $k \ll \ell$) such that $S = (\phi(x_i), y_i)_{i=1}^n$ is used as training set.

Ex: Polynomial features. E.g., if $k = 1$, $\ell = 5$:

$$\phi(x) = (x, x^2, x^3, x^4, x^5).$$

$\longrightarrow$ Often improves expressive power of a learning model!

2. Sparse Regression

- If features are designed to "explain" the target variable as a linear combination of $\boxed{\text{few}}$ features (e.g., $k \ll n$), we can use

$$\mathcal{F}_k^{\text{sparse}} := \{ h : \mathbb{R} \longrightarrow \mathbb{R} : h(x) = \langle \beta, x \rangle \quad \text{s.t.} \quad \|\beta\|_0 \leq k \},$$

where $\|\beta\|_0 = \sum_{i=1}^{n} \mathbb{1}_{\{|\beta_i| \neq 0\}}$ is the number of non-zero coefficients of β.

- Problem: ERM on $\mathcal{F}_k^{\text{sparse}}$ is NP-hard

  $\hookrightarrow$ computational challenges!

- Possible approach: Lasso Regression:

$$\hat{\beta}_n = \underset{\beta \in \mathbb{R}^2}{\arg\min} \; \|A\beta - y\|_2^2 + \lambda \|\beta\|_1 \qquad (*)$$

- Unlike linear/ridge regression, $(*)$ has no closed form solution, but convex optimization problem $\longleftarrow$ well-established theory/methods exist.

- With respect to original class $\mathcal{F}_k^{\text{sparse}}$:

Generalization error = Optimization error + estimation error + approximation error