Inmas Data Science Workshop

Instructor: Christian Kuemmerle
TAs: · Daniel Fuentes-Keuthan
· Patrick Martin

Dates & Times:

- Fr, March 19: 2-5 pm ET
- Sat, March 20: ▷ 9-12 am ET
  ▷ 2-5 pm ET
- Sun, March 21: ▷ 9-12 am ET

Goals:

▷ 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 statistical
- "reliance upon the test of experience as ultimate standard of validity"

Our focus: Prediction instead of Inference

"machine learning" "artificial intelligence (AI)"

Common Task Framework:

- Public "training" dataset: List of observations with labels
- Competitors with common task to infer label prediction rule from training data
- submit to $\longrightarrow$ Referees, report accuracy of prediction rule applied to (hidden) testing dataset.

Eg.:
- $1M Netflix (2006 - 2008)
- ImageNet

Other aspect:
- Available computer hardware
- Better software framework

Goal: **Learn from data.**

a) Ex.:
- Predict salary of professors based on discipline, employment length
- Detect spam e-mails based on large set of spam/non-spam e-mails.

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 all ~~64~~ 64×64 RBG pictures)
- $\mathcal{Y} \subset \mathbb{R}^q$: target set / set of labels
- Let $\mathcal{D}$ be a probability distribution $\mathcal{X} \times \mathcal{Y}$.

  Assume we are given a training set $S := \{x_i, y_i\}_{i=1}^{n} \overset{\text{i.i.d}}{\sim} \mathcal{D}$
- 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 function $\hat{h}_n \in \mathcal{F}$ of a hypothesis space $\mathcal{F} \subseteq \{h : \mathcal{X} \to \mathcal{Y}\}$ based on the information of training set S.

Many learning algorithms amount to: Empirical Risk Minimization

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

Example: <u>Linear Regression</u> $(q = 1)$

- $\ell(y, z) := (y - z)^2 \qquad + \frac{1}{2} \langle \gamma, M \gamma \rangle$
- $\mathcal{F} := \left\{ x \longmapsto \beta_0 + \langle x, \beta \rangle, \ \beta_0 \in \mathbb{R}, \ \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} \in \mathbb{R}^k \right\}$

Using ERM, we obtain

$$(\hat{\beta}_0, \hat{\beta}_0) = \underset{\beta_0, \beta}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (\beta_0 + \langle x_i, \beta \rangle - y_i)^2 \right\} \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$= \underset{\tilde{\beta} \in \mathbb{R}^{k+1}}{\operatorname{argmin}} \left\{ \frac{1}{n} \| X \tilde{\beta} - y \|_2^2 \right\} \qquad X = \begin{pmatrix} - x_1 - \\ \vdots \\ - x_n - \end{pmatrix}$$

$$\overset{\text{exercise}}{=} (X^T X)^{-1} (X^T y)$$

<u>Idea:</u> $L_S(\hat{h}_n) \approx L_{\mathcal{D}}(\hat{h}_n) \approx$ small

if S represents $\mathcal{D}$ well

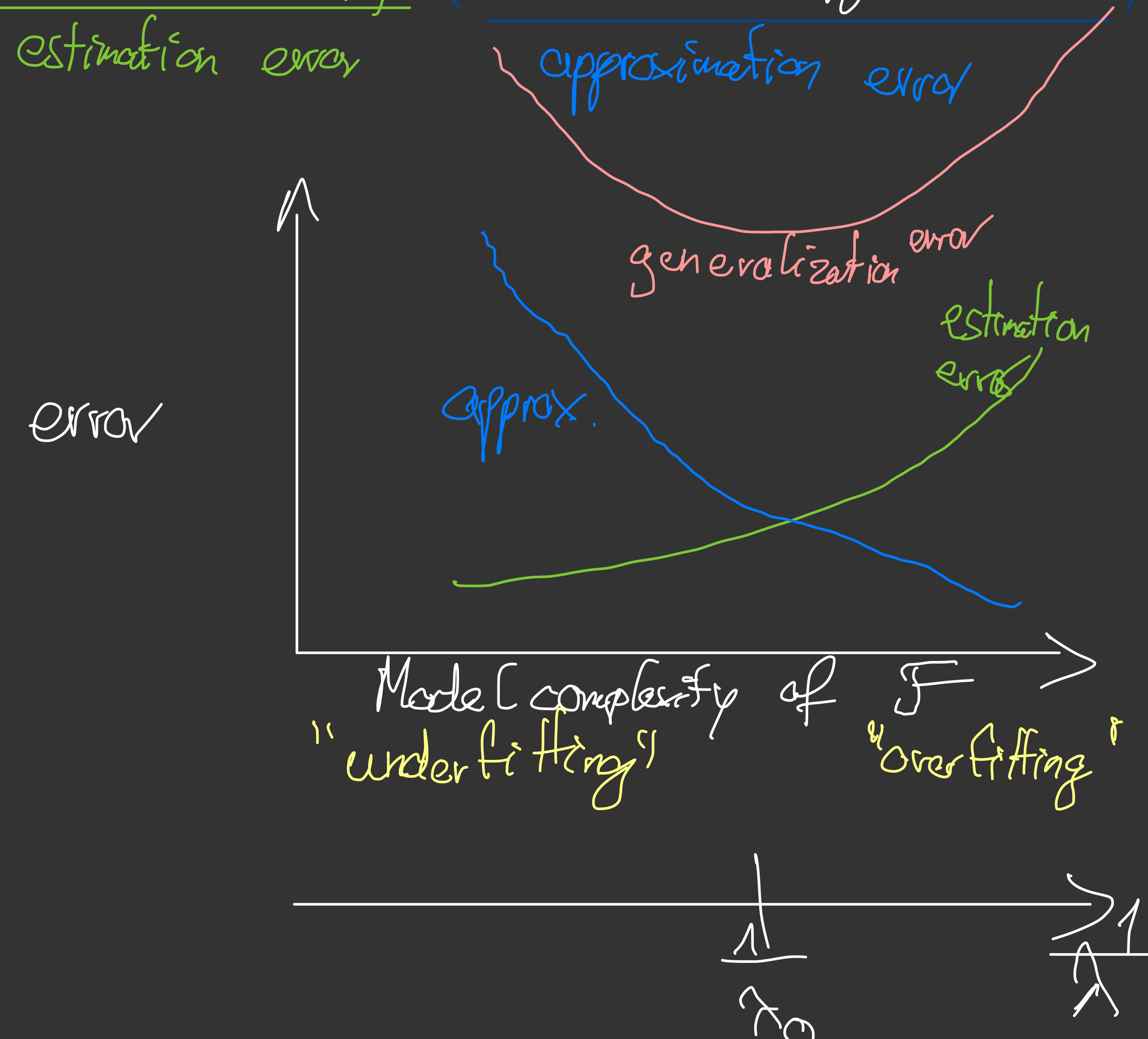
Bias-Complexity Trade-off

Q: How to choose $\mathcal{F}$?

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

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

minimizer of expected risk over hypothesis space $\mathcal{F}$



Preprocessing: If domain set s.t $\mathcal{X} \subset \mathbb{R}^k$, we can define a feature map $\phi : \mathcal{X} \longrightarrow \tilde{\mathcal{X}} \subset \mathbb{R}^l$ (with $k \ll l$), consider

$$\tilde{S} = (\phi(x_i), y_i)_{i=1}^n \quad \text{instead of} \quad S = (x_i, y_i)_{i=1}^n$$

Ex: ▷ Polynomial features: E.g. if $k = 1$, $l = 5$,

$$\Phi(x) = (1, x^1, x^2, x^3, x^4)$$

$\longrightarrow$ Often improves expressive power of a learning model).

▷ $\Phi(x) = \log(x)$

Controlling Model Complexity via Regularization

▷ Modify learning algorithm for some hypothesis space $\mathcal{F}$:

For $\lambda > 0$,

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

$\forall h \in \mathcal{F}$

$0 \leq R(h)$ is a term that "quantifies complexity" of $h \in \mathcal{F}$.

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

<u>Q:</u> How to choose λ?

1. Ridge Regression

λ: "Regularization parameter"

- Choose $\mathcal{F}$ space of linear functions
- Choose regularization term $R(h) := \|\beta(h)\|_2^2$

$$\Longrightarrow \hat{\beta}_n = \beta(\hat{h}_n) = \underset{\tilde{\beta} \in \mathbb{R}^{k+1}}{\operatorname{argmin}} \left\{ \frac{1}{n} \|X\tilde{\beta} - y\|_2^2 + \lambda \|\beta\|_2^2 \right\}$$

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

- If $\lambda = 0$: $\longrightarrow$ Linear regression
- If $\lambda \longrightarrow \infty$: coefficients $\hat{\beta}$ "shrinked to 0"
- λ in between: balancing fit of the linear model and size of coefficients.

- Strong connection to hypothesis set: $\mathcal{F}_t = \{ h: \mathbb{R}^{k+1} \to \mathbb{R} : h(x) = \langle \tilde{\beta}, x \rangle$ s.t. $\|\beta\|_2 \leq t \}$

2. Sparse Regression

- If features are designed to "explain" the target variable as a linear combination of few features (e.g., $s << K$)

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

where $\|\beta\|_0 = \sum_{i=1}^{n} \mathbb{1}_{\{|\beta_i| \neq 0\}}$ is nr. of non-zero coefficients of $\beta \in \mathbb{R}^K$.

- Problem: ERM on $\mathcal{F}_s^{sparse}$ in NP-hard

$\hookrightarrow$ computational challenges!

- Possible approach: Lasso Regression:

$$\hat{\beta}_n = \underset{\beta \in \mathbb{R}^{k+1}}{\operatorname{argmin}} \left\{ \frac{1}{n} \| A\beta - y \|_2^2 + \lambda \cdot \|\beta\|_1 \right\} \qquad (*)$$

▷ $(*)$ has no closed form solution, but is a convex optimization problem.

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

Generalization error = Optimization error + estimation error + approximation error

▷ Alternative algorithm: Orthogonal Matching Pursuit

Orthogonal matching pursuit (OMP)

Input: measurement matrix $\mathbf{A}$, measurement vector $\mathbf{y}$.
Initialization: $S^0 = \emptyset$, $\mathbf{x}^0 = \mathbf{0}$.
Iteration: repeat until a stopping criterion is met at $n = \bar{n}$:

$$S^{n+1} = S^n \cup \{j_{n+1}\}, \quad j_{n+1} := \underset{j \in [N]}{\operatorname{argmax}} \left\{ |(\mathbf{A}^*(\mathbf{y} - \mathbf{A}\mathbf{x}^n))_j| \right\}, \qquad (\mathrm{OMP}_1)$$

$$\mathbf{x}^{n+1} = \underset{\mathbf{z} \in \mathbb{C}^N}{\operatorname{argmin}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{z}\|_2, \operatorname{supp}(\mathbf{z}) \subset S^{n+1} \right\}. \qquad (\mathrm{OMP}_2)$$

Output: the $\bar{n}$-sparse vector $\mathbf{x}^\sharp = \mathbf{x}^{\bar{n}}$.

▷ "Greedy" algorithm.

- Sparse regression: Strongly related to problems in signal and image processing:
  - ▷ Speed-up of MRI measurements
  - ▷ Interpolation of geophysical data