Game Theory and ML

Start Recording!

Lecture 3: Machine Learning Background

Decision Problem

Goal: Given an input *x*, predict and output *y*

What we have: Observations: $(x_i, y_i)\ ,\ i = 1, \ldots, n$

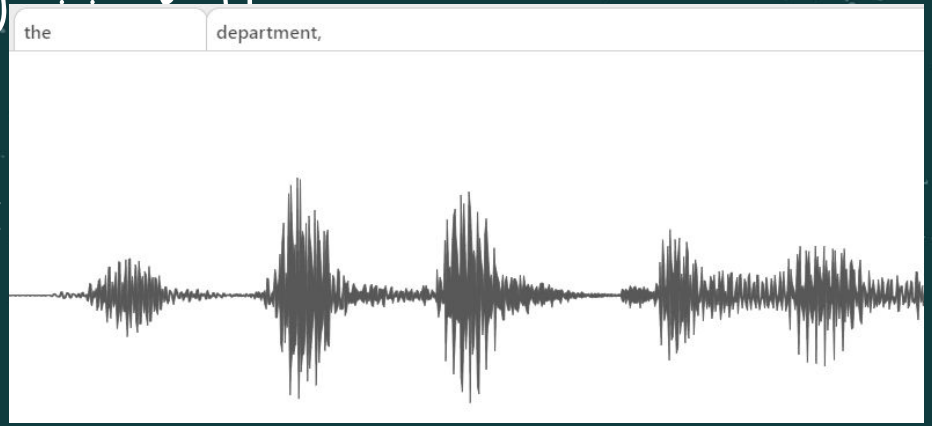
[Sagawa et al. 2020]

Source: Descript software

D

predict

vations:



Why it is "hard":

- Labels may be stochastic: $y = f(x) + \epsilon$
- The prediction may be "complex" (non-linear in high dimension)
- Only few pairs observed. (Bad labels)
- Criterion for performance not well defined. (worse in games)

Decision Theory

- Consider a fixed distribution: $(x, y) \sim p_{data}$
- A loss function. Examples:
  - Binary loss $\ell(y', y) = \mathbf{1}\{y' \neq y\}\,,\ \mathcal{Y} = \{0, 1\}$
  - Multi-class loss $\ell(y', y) = \mathbf{1}\{y' \neq y\}\,,\ \mathcal{Y} = \{1, \ldots, K\}$
  - Regression Loss $\ell(y', y) = (y - y')^2\,,\ \mathcal{Y} = \mathbb{R}$
  - Structured Prediction loss (See Simon's course IFT 6132)

$$\ell(y', y) = \frac{1}{p} \sum_{k=1}^{p} \mathbf{1}\{y'_k = y_k\}\,, \quad \mathcal{Y} = \{0, 1\}^p$$

Risks

Expected Risk:

Remarks:

1. Empirical risk is also called *Training Error*.
2. Expected risk is **not** the *test Error*. (But they are close)

Empirical Risk:

Depends on a Given dataset D_n

$$\hat{\mathcal{R}}(f, D_n) := \mathbb{E}_{(x_i, y_i) \sim \mathcal{D}_n} [\ell(f(x), y)] = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

Bayes Predictor

Question: what is the "best" predictor we can expect??

Answer: Bayes predictor.

Question: But wait.... What is the goal???

$$\text{find} \quad \min_{f \in \mathcal{F}} \mathcal{R}(f) := \mathbb{E}_{(x,y)\sim p_{data}} [\ell(f(x), y)]$$

Hypothesis class: the functions we have access to.
Examples: All measurable function, linear functions, a certain NN architecture

Expected risk: *Most of the time intractable (but it is our target)*

Goa

Remarks:

1. Optimal risk is zero iif the dependence between x and y is almost surely deterministic.
2. Minimizer may **not** be unique but the value at the minimum is.
3. Theoretical limit in terms of performance that depends on **the data**.
4. f* are called **Bayes predictors.**

Proposition 2.1 (Bayes predictor and Bayes risk) *The expected risk is minimized at a* Bayes predictor $f^* : \mathcal{X} \to \mathcal{Y}$ *satisfying for all* $x' \in \mathcal{X}$, $f^*(x') \in \arg\min_{z \in \mathcal{Y}} \mathbb{E}(\ell(y,z)|x = x') = \arg\min_{z \in \mathcal{Y}} r(z|x')$. *The Bayes risk* $\mathcal{R}^*$ *is the risk of all Bayes predictors and is equal to*

$$\mathcal{R}^* = \mathbb{E}_{x' \sim dp_x(x')} \inf_{z \in \mathcal{Y}} \mathbb{E}(\ell(y,z)|x = x').$$

Source: Francis Bach's book (see Refs at the end of the slides)

Point-wise expression for f*!!!

Exercice: Prove that proposition.

Bayes Predictor

Exercices:

- Compute the Bayes predictor for the binary loss. (see slide 6)
- Compute the Bayes predictor for the regression loss. (see slide 6)
- Compute the Bayes predictor for the cross-entropy loss (y is binary):

$$\ell(y', y) = y \ln(y') + (1 - y) \ln(1 - y')$$

Learning with Empirical Risk Minimization

We

fi

Qu

ibution...

nd thus

We

$i), y_i)$

$i=1$

Answer: Learning Theory

Bias-Variance Trade-off

$$\underbrace{\mathcal{R}(f_\theta) - \mathcal{R}^*}_{\text{Prediction Error}} = \underbrace{\left(\mathcal{R}(f_\theta) - \inf_\theta \mathcal{R}(f_\theta)\right)}_{\text{Minimization term}} + \underbrace{\left(\inf_\theta \mathcal{R}(f_\theta) - \mathcal{R}^*\right)}_{\text{Bias}}$$

Prediction Error:
Goal: Find, the best $\boldsymbol{\theta}$ we can expect.
Problem: Challenging to estimate.

Minimization term:
How well we are minimizing the expected risk.
Problem: hard to minimize.

Bias:
Irreducible error
Problem:
We need to change ou class of function (larger) to get better.

Bias-Variance Trade-off

$$\underbrace{\mathcal{R}(f_\theta) - \mathcal{R}(f_{\theta^*})}_{\text{Prediction Error}} = \underbrace{\left(\mathcal{R}(f_\theta) - \hat{\mathcal{R}}(f_\theta)\right)}_{\text{First Deviation Term}} + \underbrace{\left(\hat{\mathcal{R}}(f_\theta) - \hat{\mathcal{R}}(f_{\theta^*})\right)}_{\text{Optimization Term}} + \underbrace{\left(\hat{\mathcal{R}}(f_{\theta^*}) - \mathcal{R}(f_{\theta^*})\right)}_{\text{Second Deviation Term}}$$

Prediction Error:
Goal: Find , the best we can expect.
Problem: Challenging to estimate.

First Deviation Term:
Problem: The empirical Risk and the expected risk can have drastically different values for some $\boldsymbol{\theta}$.

Optimization Term:
Relatively how well we can minimize the empirical risk.

Second Deviation Term

Bias-Variance Trade-off

$$\mathcal{R}(f_\theta) - \mathcal{R}(f_{\theta^*}) = \Big(\mathcal{R}(f_\theta) - \hat{\mathcal{R}}(f_\theta)\Big) + \Big(\hat{\mathcal{R}}(f_\theta) - \hat{\mathcal{R}}(f_{\theta^*})\Big) + \Big(\hat{\mathcal{R}}(f_{\theta^*}) - \mathcal{R}(f_{\theta^*})\Big)$$

$$\mathcal{R}(f_\theta) - \mathcal{R}^* \leq \underbrace{2 \sup_\theta \Big|\mathcal{R}(f_\theta) - \hat{\mathcal{R}}(f_\theta)\Big|}_{\text{Upper-bound on the "variance" terms.}} + \underbrace{Bias}_{\text{For this one we need to increase the size of the function space}}$$

Problem: If we have too much freedom with $\boldsymbol{\theta}$ then this supremum can be large.

First idea: More data (hard to get a lot)

Second idea: Being smart (regularization)

Third idea: Cross your fingers (Deep Learning)

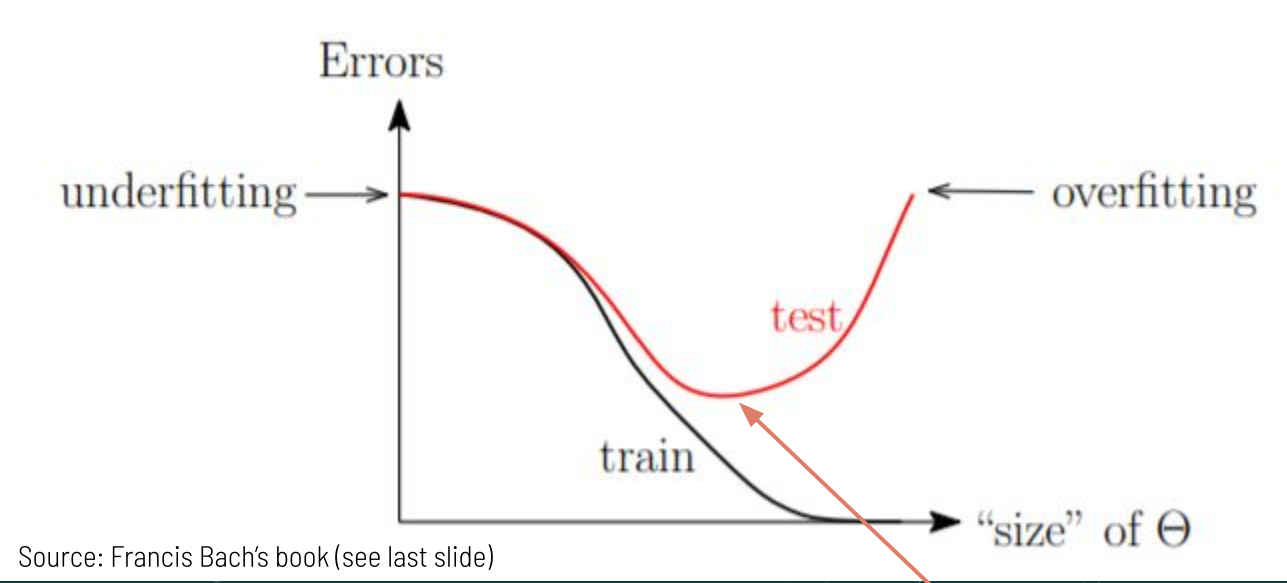
Yann Lecun's Cake Theory on Data

- "Pure" Reinforcement Learning (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- Supervised Learning (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- Unsupervised/Predictive Learning (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

Plenary talk at NeurIPS 2016

Bias-Variance in One Picture



Source: Francis Bach's book (see last slide)

Very important: Here the number of sampe n is **fixed**.

From previous slide:
Being smart or lucky means being here

Example: Linear Regression

- Also Known as least-squares.
- (Square) Regression loss (see slide 6).

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (f_\theta(x_i) - y_i)^2$$

- Linear function with **arbitrary fixed features.**

$$f_\theta(x) = \langle \theta, \varphi(x) \rangle$$

Example: Linear Regression

Exercice: Show that when the design matrix X is full rank the optimal solution for the square loss regression (slide 16) is:

$$\theta^* = (X^\top X)^{-1} X^\top y \quad \text{where} \quad X = \begin{pmatrix} \varphi(x_1)^\top \\ \cdots \\ \varphi(x_n)^\top \end{pmatrix}$$

Exercice: Show that when the design matrix X is not full rank we just need to replace inverse by pseudo-inverse in the formula above.

Example: Linear Regression

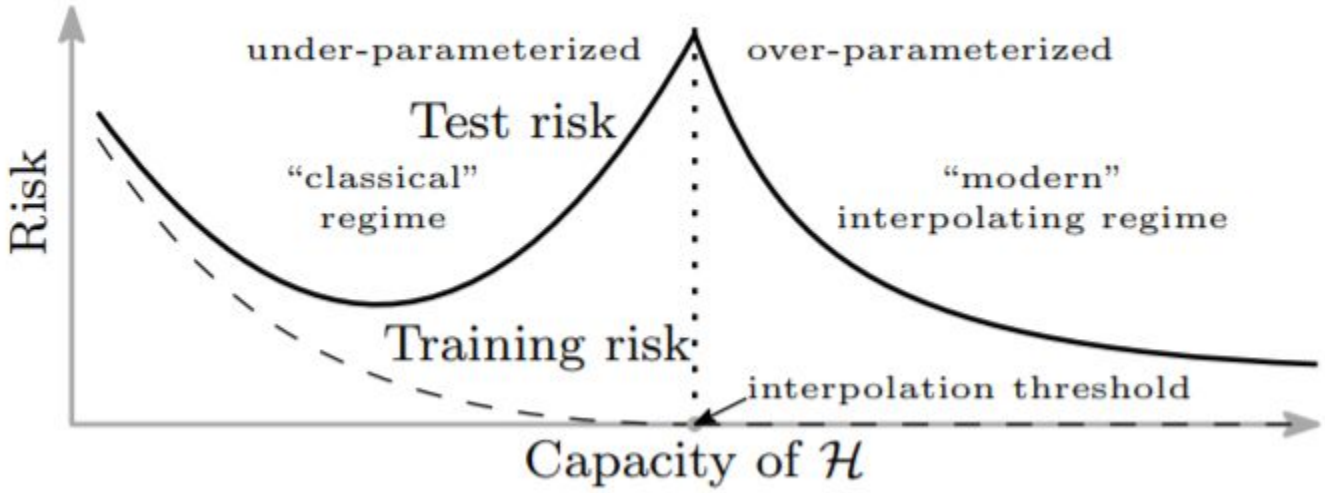
Important Exercice: (You should at least try)

Code in colab: (in the description of the video)

- Copy the colab! (you cannot edit it)
- Try to compete the code in `def solve_least_squares`
- Try to complete the code in: `def solve_least_squares_with_GD`
- Have a way to visualize the results.

Double Descent Phenomenon

Figure from [Belkin et al. 2018]



- Complex Picture.
- Still and active research area. Basically we do not fully understand why.
- Related to why Deep Learning works (and shouldn't according to standard LT).
- See [Neal et al. 2018] and [Belkin et al. 2018]

Back to Classification

We want to minimize:

$$\hat{\mathcal{R}}(f, D_n) := \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}_n}[\ell(f(x), y)] = \frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i), y_i)$$

Problem:

- With 0-1 Loss it is NP-Hard!!!! [Feldman et al. 2009]

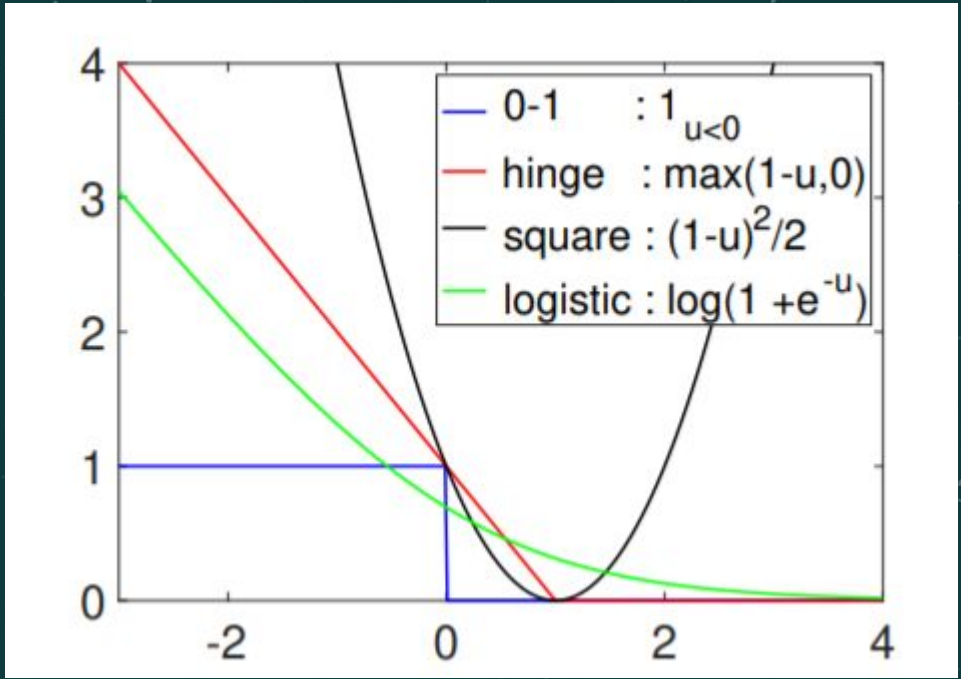
Solution:

- Solve a simpler problem.
- What about differentiable losses

Convex Surrogates

Here we have labels in {-1,1}:

- 0-1 loss
- Quadratic loss
- Hinge loss
- Logistic Loss



Logistic Regression

Warning: Often two conventions

- Labels in {-1,1}
- Labels in {0,1}

If labels in {0,1} we get:

$$\ell(y, f(x)) = y \log(\sigma(f(x))) + (1 - y) \log(1 - \sigma(f(x)))$$

(more frequent in ML) In the future we will keep this one.

$$\prod_{i=1} \mathbb{P}(y_i | f(x_i))$$

Logistic Regression

Exercice: Compute the optimal Bayes Classifier for

$$\ell(y, f(x)) = y \log(\sigma(f(x))) + (1 - y) \log(1 - \sigma(f(x)))$$

References:

The Book by Francis Bach (currently a draft)
https://www.di.ens.fr/~fbach/ltfp_book.pdf

Pattern Recognition and Machine Learning

Authors: **Bishop**, Christopher