Lecture 1

- Big Picture of Statistical Learning
- Types of Learning Problems
- Statistical Decision Theory
- Nearest - Neighbor Methods
- Linear Regression Model

learning from data

next time

Statistical learning is a subfield of modern statistics that develops methods for understanding data, making predictions based on the data, and making data-informed decisions. Statistical learning plays a key role in many areas of science, engineering, finance, and medicine. As a result, people with SL skills are in high demand.

Nowadays the crucial role of data in scientific and engineering discoveries is obvious. But this has not been always the case!

- Plato (around 380 BC): astronomers should study stars ("decorations on a visible surface") by "reason and thought", without looking at the stars. The Republic, Book VII.
- Aristotle (384 - 322 BC): wrote a lot about biology and medicine, but never looked at the real data about human body. He believed that the heart, not the brain, was the seat of thought.

This could be easily disproved by observing the effects of trauma to the brain. (This was known by Hippocrates, who lived before Aristotle, and his school)

Of course, with time, people began to recognize the importance of data. One remarkable example of "learning from data":

- Johannes Kepler (1571 - 1630) discovered the laws of planetary motion (around 1610) from the extensive astronomical observations collected by Tycho Brahe (1546 - 1601).

The field of Statistics, solely dedicated to data analysis, has emerged from the work of

- John Graunt (1620 - 1674), who estimated the population of London by analyzing the statistics of deaths. He published his results in 1662.

Since then, the field of classical statistics has been developed by many prominent scientists and mathematicians such as Francis Galton, Karl Pearson, Ronald Fisher, William Gosset (aka Student), Egon Pearson (son of Karl Pearson), and Jerzy Neyman.

The availability of cheap computing power and the explosion of data truly revolutionized classical statistics:

- "Classical Statistics": problems come from surveys of human populations, agricultural and industrial experiments; data sets are small; solutions are analytical, done by hand.
- "Modern Statistics": problems come from everywhere; data sets are large; solutions are numerical & empirical, done on a computer.

Since computers are equally available to all research communities and data appear everywhere, modern methods for learning from data have been and continue to be developed by researchers from various fields:

Classical Field		Recent Subfield (focused on learning from data)
Statistics	$\longrightarrow$	Statistical Learning
Computer Science	$\longrightarrow$	Machine Learning
Engineering	$\longrightarrow$	Pattern Recognition
Biology / Medicine	$\longrightarrow$	Bioinformatics
Finance	$\longrightarrow$	Financial Machine Learning

All these subfield can be viewed as different facets of the same development.

<u>Remark</u>: For example, the wikipedia article on "Statistical Learning" redirects to "Machine Learning".

<u>Remark</u>: Each of those subfields has its own terminology, notation, and jargon. This complicates communication between researchers from different subfields.

<u>Main Goal of the Course</u>: to discuss the most fundamental ideas and methods for learning from data and explain them in a statistical framework.

The course can be viewed as:

- Natural continuation of my IDS/ACM/CS 157 (Statistical Inference).
- Statistical and theoretical analog of CMS/CS/CNS/EE/IDS 155 (ML & Data Mining)

<u>Types of Learning Problems</u>

There are three main types of learning:

- Statistical Learning
  - Supervised Learning (main focus)
    - Regression
    - Classification
  - Unsupervised Learning (will discuss)
    - Clustering
    - Generative Modeling
    - Dimensionality Reduction
  - Reinforcement Learning (not covered)

① <u>Supervised Learning</u>

This is the most studied and most often used in applications type of learning.

input $X \longrightarrow \boxed{\text{Complex System}} \longrightarrow Y$ output

The goal of supervised learning is to construct a statistical model for <u>predicting</u> output Y from input X based on "training data": $\{(X_1,Y_1)\ldots(X_N,Y_N)\} \leftarrow$ collection of examples.

- If output Y is <u>quantitative</u> (continuous) $\Longrightarrow$ the problem is called <u>regression</u>.

  <u>Example</u>: Predict the price of a stock Y in 1 months from now, based on the input X that describes the company performance measures and economic data.

- If output Y is <u>qualitative</u> (discrete, categorical) $\Longrightarrow$ the problem is called <u>classification</u>.

  <u>Example</u>: Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack ($Y=1$) or not ($Y=0$), based on the input $X=$ (demographic, diet, clinical measurements).

② Unsupervised Learning

Here the situation is more challenging : the training data consists only of inputs $\{X_1, \ldots, X_N\}$, there is no outputs $Y_1, \ldots, Y_N$ that could supervise our learning. The goal of unsupervised learning is less well defined and more ambiguous : to <u>understand</u> the structure of the data, find possible patterns among inputs.

More specific problems include :

- <u>Clustering</u> : to divide $\{X_1, \ldots, X_N\}$ into relatively distinct groups (clusters).

  <u>Example</u> : market segmentation : divide customers into groups based on their characteristics (income, zip code, shopping habits) $= X_i$

- <u>Dimensionality Reduction</u> : to represent the data $\{X_1 \ldots X_N\}$ in a smaller space, $\{X_i \in \mathbb{R}^D\} \rightsquigarrow \{\tilde{X}_i \in \mathbb{R}^d\}$, $d \ll D$, for visualization or as a pre-processing step for supervised learning.

- <u>Generative Modeling</u> : to model a generating mechanism that produced $\{X_1, \ldots X_N\}$, in order to generate artificial inputs that are similar to real data. (synthetic data in the input space)

  <u>Example</u> : synthetic earthquake generation (accelerograms similar to real ones $X_1 \ldots X_N$)

③ Reinforcement Learning

The goal of reinforcement learning is to infer optimal sequential <u>decisions</u> (actions) based on rewards or punishments received as a result of previous actions.

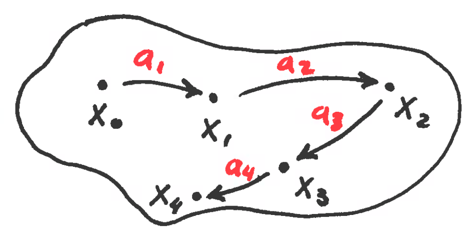
<u>Example</u> : training a robot to navigate a given environment in the presence of obstacles by penalizing decisions that result in collisions.

<u>Example</u> : training an algorithm to play chess (AlphaZero)

After 9 hours of training via self-play, AlphaZero defeated Stockfish 8, the strongest chess engine at that time (2017)

<u>Ref</u>: D. Silver at al (2018), Science, vol 362, pp 1140-44.

<u>Schematic Representation of RL</u>



Input space (aka Environment)

X_0 is the initial input (aka state)

1. Choose $a_1 \in \mathcal{A}_{X_0}$ ← set of available actions in state X_0
2. Execute a_1 :

   $X_0 \xrightarrow{a_1} X_1$ (new state)
3. Get reward $R_{a_1}(X_0, X_1) \in [-1, 1]$ (bad -1, good 1)

The following aspects make RL especially difficult :

a) Choosing $a_i \in \mathcal{A}_{X_{i-1}}$ is done <u>not</u> based on optimization $R_{a_i}(X_{i-1}, X_i) \to \max$ but based on previous examples. $\mathcal{A}_{X_{i-1}}$ is often very large (or ∞).

b) Executing a_i on X_{i-1} does not always bring us to deterministic state X_i. Environment may change during execution. So, executing a_i on X_{i-1} results into a probability distribution $\mathbb{P}(X_i = x \,/\, X_{i-1}, a_i)$ on the input space.

c) Action a_i may affect not only the immediate reward $R_{a_i}(X_{i-1}, X_i)$, but also rewards at all subsequent steps.

Let's start with a general discussion of supervised learning.

We are goinng to use the following notation and terminology:

- $X = (X_1, \dots, X_p)^T$ is a vector of inputs (aka predictors, features, independent variables)
  $p = \#$ inputs. We use X to refer to a generic input vector.

- $x_i = (x_{i1}, \dots, x_{ip})^T$ is the i^{th} observed value of X, $i = 1, \dots, N$.
  $N = \#$ observations. $x_i \in \mathbb{R}^p$ is a p-vector.

- $x^{(j)} = (x_{1j}, \dots, x_{Nj})^T$ is the vector of all observations of input X_j, $j = 1, \dots, p$.

- $$\mathbf{X} = \underbrace{\begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{N1} & \cdots & x_{Np} \end{bmatrix}}_{N \times p \text{ matrix}} = \underbrace{\begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}}_{\text{inputs}} = \Big[\underset{\substack{\uparrow \\ \text{observations} \\ \text{of } X_1}}{x^{(1)}} \cdots \underset{\substack{\uparrow \\ \text{observations} \\ \text{of } X_p}}{x^{(p)}}\Big]$$

  Remark: All vectors are assumed to be column vectors.

  x_{ij} = value of j^{th} input in the i^{th} observation.

- Y is a quatitative output corresponding to X (aka response, dependent variable) $Y \in \mathbb{R}$

- y_i is the i^{th} observed value of Y that correspond to x_i.

- G is a qualitative output corresponding to X, $G \in \mathcal{G}$ set of groups/classes.

- g_i is the i^{th} observation of G that correspond to x_i.

Main Goal: Given the training data $\overbrace{(x_1, y_1), \dots, (x_N, y_N)}^{\text{regression}}$ $\Big($or $\overbrace{(x_1, g_1), \dots, (x_N, g_N)}^{\text{classification}}\Big)$ and the value of input X, make a good prediction $\hat{Y}$ (or $\hat{G}$) of the output Y (or G).

Statistical Decision Theory provides a framework for developing powerful prediction methods.

Let's start with the case of regression (quantitative output).

Let $X \in \mathbb{R}^p$ be a random input and $Y \in \mathbb{R}$ be the corresponding random output.

We want to find a function (prediction rule) $f : \mathbb{R}^p \to \mathbb{R}$ such that $\hat{Y} = f(X) \approx Y$ is an accurate prediction.

To measure the accuracy of prediction (the goodness of $f(x)$), we need a loss function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. If the expected loss is small, the prediction rule is good; otherwise it is bad:

$$\mathbb{E}\left[L(Y, \hat{Y})\right] = \mathbb{E}\left[L(Y, f(X))\right] = \begin{cases} \text{small} \Rightarrow f \text{ is good}, \\ \text{large} \Rightarrow f \text{ is bad}. \end{cases}$$

By far the most popular and often used loss function is the squared error loss : $\boxed{L(Y, \hat{Y}) = (Y - \hat{Y})^2}$ aka L_2 loss function.

The corresponding expected loss is then called the mean squared error (MSE) or expected prediction error (EPE) :

$$MSE(f) = \mathbb{E}\left[(Y-\hat{Y})^2\right] = \mathbb{E}\left[(Y - f(X))^2\right] \longleftarrow$$

this expected value is w.r.t. the joint distribution of X and Y.

So, our goal is to find the prediction rule f that minimizes $MSE(f)$.

We can find f by conditioning on X and using the law of total expectation :

$$MSE(f) = \underset{\text{wrt } X}{\mathbb{E}}\Big[\overbrace{\underset{\text{wrt } Y}{\mathbb{E}}\left[(Y - f(X))^2 \,|\, X\right]}\Big]$$

this is a random variable, whose value is $\underbrace{\mathbb{E}\left[(Y - f(x))^2 \,|\, X = x\right]}_{\text{conditional expectation}}$ when $X = x$.

$$= \int (y - f(x))^2 \underbrace{p_{Y|X}(y|x)}_{\text{conditional PDF of } Y \text{ given } X.} dy$$

The inner conditional expectation :

$$\mathbb{E}\left[(Y - f(x))^2 / X = x\right]$$

$$= \mathbb{E}\left[\left(Y - \mathbb{E}[Y/X=x] + \mathbb{E}[Y/X=x] - f(x)\right)^2 \,|\, X = x\right]$$

$$= \underbrace{\mathbb{E}\left[\left(Y - \mathbb{E}[Y/X=x]\right)^2 / X = x\right]}_{\mathbb{V}[Y|X=x]} + \mathbb{E}\Big[\underbrace{\left(\mathbb{E}[Y/X=x] - f(x)\right)}_{\text{constant}}^2 / X = x\Big]$$

$$+ 2\,\mathbb{E}\Big[\left(Y - \mathbb{E}[Y|X=x]\right) \cdot \overbrace{\left(\mathbb{E}[Y|X=x] - f(x)\right)}^{\text{constant}} \,\Big|\, X = x\Big]$$

$$= \mathbb{V}[Y|X=x] + \left(\mathbb{E}[Y|X=x] - f(x)\right)^2 + 2\left(\mathbb{E}[Y|X=x] - f(x)\right)\overbrace{\left(\mathbb{E}[Y/X=x] - \mathbb{E}[Y/X=x]\right)}^{0}$$

So : $\mathbb{E}\left[(Y - f(x))^2 | X = x\right] = \mathbb{V}[Y|X=x] + \left(\mathbb{E}[Y|X=x] - f(x)\right)^2$, and

$$MSE(f) = \mathbb{E}\left[\mathbb{V}[Y|X] + \left(\mathbb{E}[Y|X] - f(X)\right)^2\right]$$

Therefore, the prediction rule that minimizes $MSE(f)$ is

$$\boxed{f(X) = \mathbb{E}[Y/X]}$$

In other words, if we observe that input $X = x$, then the $\overbrace{\text{best prediction}}^{\text{in the MSE sense } (L_2 \text{ sense})}$ $\hat{Y}$ for the output is $\boxed{f(x) = \mathbb{E}[Y/X=x]}$ $\longleftarrow$ this conditional expectation is called the regression function.