What is Statistical “Learning”?

Course 229352: Statistical Learning for Data Science 2

What is Statistical Learning? Two Perspectives

In this course, we'll view "learning from data" through two main lenses:

Perspective 1: Learning Distributions

Finding the underlying probability that generates the data.

- How is data generated? $P(\mathbf{X})$
- How are outputs related to inputs? $P(Y|\mathbf{X})$

What is Statistical Learning? Two Perspectives

In this course, we'll view "learning from data" through two main lenses:

Perspective 1: Learning Distributions

Finding the underlying probability that generates the data.

- How is data generated? $P(\mathbf{X})$
- How are outputs related to inputs? $P(Y|\mathbf{X})$

Perspective 2: Learning Functions

Finding a function $f(\mathbf{X})$ of features $\mathbf{X}$.

- Prediction: $f(\mathbf{X}) = Y$
- Clustering: $f(\mathbf{X}) = \text{cluster}$
- Halfspaces: $f(\mathbf{X}) > 0$ or $f(\mathbf{X}) < 0$

What is Statistical Learning? Two Perspectives

In this course, we'll view "learning from data" through two main lenses:

Perspective 1: Learning Distributions

Finding the underlying probability that generates the data.

- How is data generated? $P(\mathbf{X})$
- How are outputs related to inputs? $P(Y|\mathbf{X})$

Perspective 2: Learning Functions

Finding a function $f(\mathbf{X})$ of features $\mathbf{X}$.

- Prediction: $f(\mathbf{X}) = Y$
- Clustering: $f(\mathbf{X}) = \text{cluster}$
- Halfspaces: $f(\mathbf{X}) > 0$ or $f(\mathbf{X}) < 0$

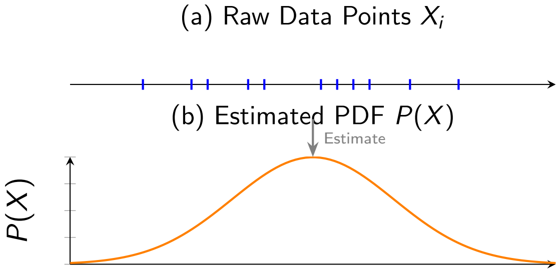
Our Overarching Goal

To use data to build models that help us understand the world, make predictions, or make informed decisions.

Perspective 1: Learning Probability Distributions

The Idea: Model the "data generating process."

- Learn $P(\mathbf{X})$.
    - What does 'typical' data look like?
    - Density estimation, quantiles, modes, etc.



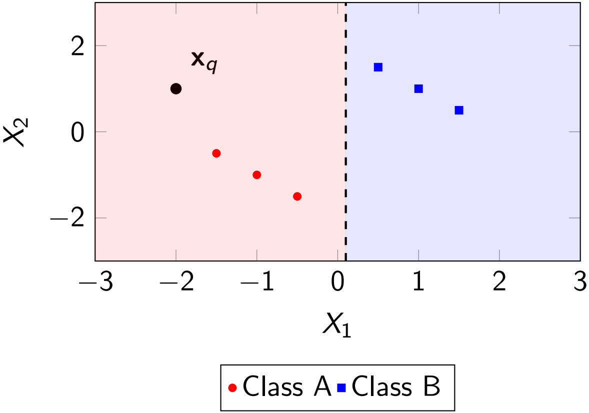
(a) Raw Data Points X_i

(b) Estimated PDF $P(X)$

Perspective 1: Learning Probability Distributions

The Idea: Model the "data generating process."

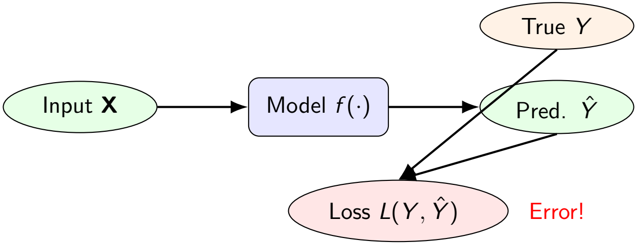
- Learn $P(\mathbf{X})$.
    - What does 'typical' data look like?
    - Density estimation, quantiles, modes, etc.
- Learn $P(Y|\mathbf{X})$.
    - Given inputs $\mathbf{X}$, what's the probability of output Y?
    - Probabilistic classification/regression.



Perspective 2: Learning a Predictive Function $f(\mathbf{X})$

The Idea: Find a mapping from inputs to outputs.

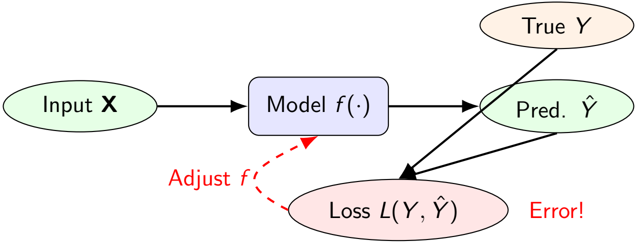
- Goal: Create a function $f(\mathbf{X})$ that produces predictions $\hat{Y}$.



Perspective 2: Learning a Predictive Function $f(\mathbf{X})$

The Idea: Find a mapping from inputs to outputs.

- Goal: Create a function $f(\mathbf{X})$ that produces predictions $\hat{Y}$.
- Criterion: Minimize a **loss function** $L(Y_{true}, \hat{Y})$ that penalizes errors.
  - Y_{true}: The actual observed value.
  - E.g., for classification: $L = \mathbf{1}_{Y_{true} \neq \hat{Y}}$ (0/1 loss)
  - E.g., for regression: $L = (Y_{true} - \hat{Y})^2$ (squared error)



Learning Probability Distributions:

Learning Probability Distributions: Parametric Estimation

Concept

- **Assumption-based:** We assume the data comes from a specific family of probability distributions (e.g., Gaussian/Normal, Bernoulli, Poisson).
- This family is characterized by a **fixed number of parameters**.
- **Learning** $\equiv$ **Estimating these parameters** from the data.

Learning Probability Distributions: Parametric Estimation

Concept

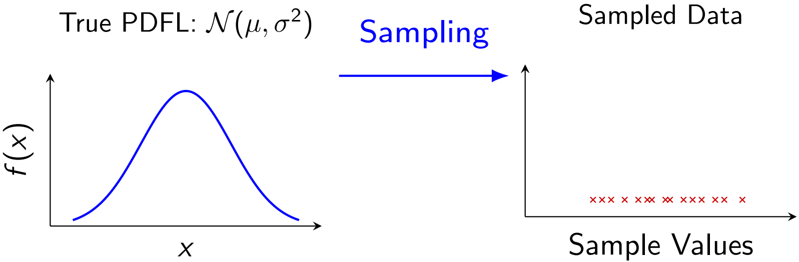
- **Assumption-based:** We assume the data comes from a specific family of probability distributions (e.g., Gaussian/Normal, Bernoulli, Poisson).
- This family is characterized by a **fixed number of parameters**.
- **Learning $\equiv$ Estimating these parameters** from the data.

Example

Imagine a coin. We assume it has a fixed 'probability of heads' (p). Learning p (e.g., by flipping it 100 times) is parametric estimation.

Example: Learning the Mean (Parametric)

- Imagine we have data $X_1, X_2, \dots, X_n$ and we assume it comes from a Normal distribution $\mathcal{N}(\mu, \sigma^2)$.
- Our goal is to "learn" the true mean μ and variance σ^2.


True PDFL: $\mathcal{N}(\mu, \sigma^2)$

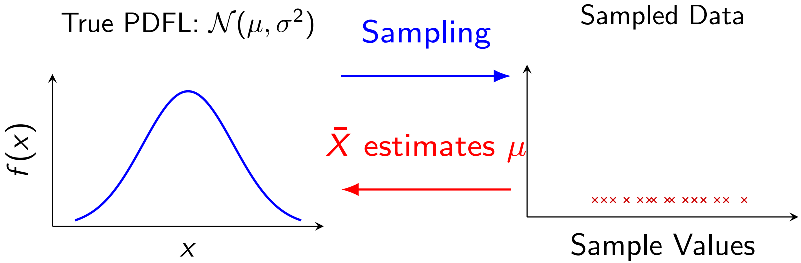
Sampling

Sampled Data

Example: Learning the Mean (Parametric)

- Imagine we have data $X_1, X_2, \ldots, X_n$ and we assume it comes from a Normal distribution $\mathcal{N}(\mu, \sigma^2)$.
- Our goal is to "learn" the true mean μ and variance σ^2.
- The simplest and most common estimate for the mean μ is the **sample mean**:

$$\hat{\mu} = \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$



True PDFL: $\mathcal{N}(\mu, \sigma^2)$

Sampling

Sampled Data

$\bar{X}$ estimates μ

Nonparametric Estimation

Concept

- **Data-driven:** Few/no strong assumptions about the underlying distribution.
- **Learning $\equiv$ Directly inferring distributional properties** (median, quantiles etc.) from data without a fixed parametric form.

Nonparametric Estimation

Concept

- **Data-driven:** Few/no strong assumptions about the underlying distribution.
- **Learning $\equiv$ Directly inferring distributional properties** (median, quantiles etc.) from data without a fixed parametric form.

Example

Instead of assuming data is Normal, we might estimate its Probability Density Function (PDF) or Cumulative Distribution Function (CDF) directly from observations.

Normal Distribution: PDF and CDF

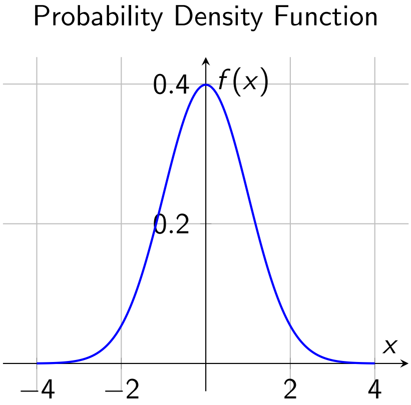


Figure: PDF of $\mathcal{N}(0,1)$

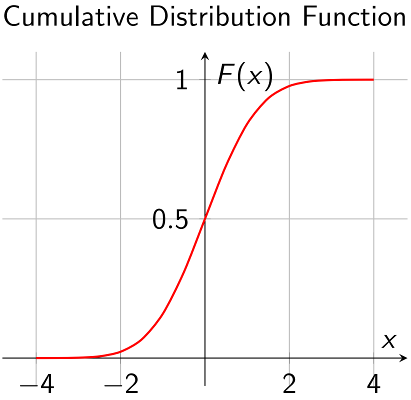


Figure: CDF of $\mathcal{N}(0,1)$

Example: Cumulative Distribution Function (CDF)

What is a CDF?

- $F(x) = P(X \leq x)$: Probability X is less than or equal to x.
- Non-decreasing, from 0 to 1.
- Completely characterizes the probability distribution.

Example

Standard normal: $F(0) = 0.5$ (50% chance value ≤ 0).

How can we estimate the CDF?

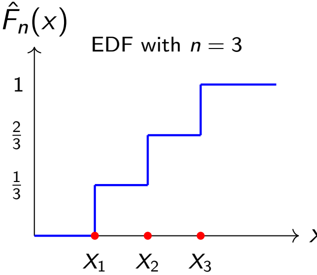
Estimating CDF: The Empirical Distribution Function (EDF)

How do we estimate $F(x)$ without assuming a distribution?

Use the **Empirical Distribution Function (EDF)**, $\hat{F}_n(x)$.

- Given i.i.d. observations $X_1, \dots, X_n$:

$$\hat{F}_n(x) = \frac{\text{Number of } X_i \leq x}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[X_i \leq x]$$



- $\mathbf{1}[X_i \leq x]$ is an indicator function (1 if $X_i \leq x$, 0 otherwise).
- The EDF is a **step function**, jumping at each data point.

Estimating CDF: The Empirical Distribution Function (EDF)

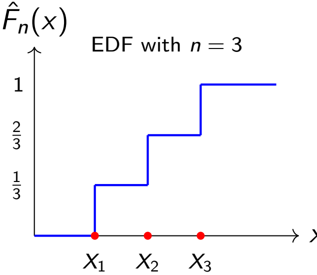
How do we estimate $F(x)$ without assuming a distribution?

Use the **Empirical Distribution Function (EDF)**, $\hat{F}_n(x)$.

- Given i.i.d. observations $X_1, \dots, X_n$:

$$\hat{F}_n(x) = \frac{\text{Number of } X_i \leq x}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[X_i \leq x]$$

- $\mathbf{1}[X_i \leq x]$ is an indicator function (1 if $X_i \leq x$, 0 otherwise).
- The EDF is a **step function**, jumping at each data point.



Does EDF accurately estimate the CDF?

As $n \to \infty$, $\hat{F}_n(x) \to F(x)$ (Glivenko-Cantelli, 1933).

Python Interactive: EDF Calculation

Link to Google Colab

Generative Models

Focus: The Data Itself (Perspective 1)

- Aim to learn **joint distribution** $P(\mathbf{X}, Y)$ or data distribution $P(\mathbf{X})$.
- Understand **how data is generated**.
- Can generate new, synthetic data.

Example

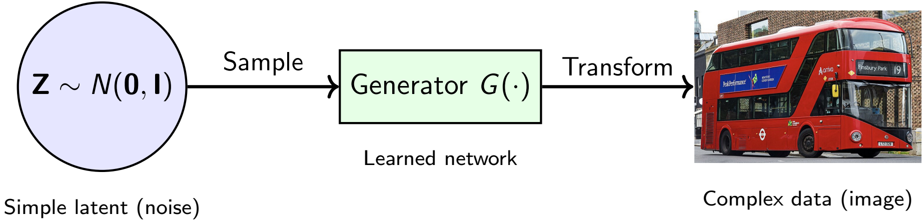
- Generating realistic images.
- Creating new music.
- Producing synthetic text.

Advanced Generative Models: Learning a Transformation

For complex data (e.g., images), directly modeling $P(\mathbf{X})$ is hard. Modern approach:

The Core Idea

- Assume simple distribution for **latent variable Z** (e.g., $N(\mathbf{0}, \mathbf{I})$).
- **Learn a complex function (generator)** $G : \mathbf{Z} \to \mathbf{X}$.
- Maps simple latent samples to realistic data samples.



$\mathbf{Z} \sim N(\mathbf{0}, \mathbf{I})$

Sample

Generator $G(\cdot)$

Transform

Learned network

Simple latent (noise)

Complex data (image)

Figure: Generative models by learning a transformation from random noises.

Summary: Diverse Learning Approaches

We've seen "learning" as:

- **Learning Distributions** $P(\mathbf{X}), P(Y|\mathbf{X})$
  - Parametric (assume model family)
  - Nonparametric (data-driven, e.g., EDF)
- **Learning Functions** $f(\mathbf{X}) \to \hat{Y}$
  - Minimize loss/error
  - Classification, Regression

- **Generative Models:** Learn $P(\mathbf{X})$ or $P(\mathbf{X}, Y)$ to understand data generation and create new samples.
- Advanced ones learn transformations from simple latent spaces.

Key Takeaway

"Learning" in data science is diverse, from parameter estimation to complex generative processes, each with its strengths and suitable applications. Our learning targets often guide our choice of methods.

What's Next?

- This was an introduction to "learning" concepts.
- In this course, we will explore many more advanced techniques:
  - **Classification algorithms** (SVMs, Trees, Boosting), linking them to $P(Y|\mathbf{X})$ and $Y = f(\mathbf{X})$.
  - **Clustering algorithm**, k-means and hierarchical clustering.
  - **Deep Learning Models**, CNNs, RNNs, Transformers.

Question?