Fundamentals: Definition, Notation, and Principles

Qiang Sun

January 2026

Welcome

- Welcome to **STAD80: Analysis of Big Data (ABD)**.
- This is a **topics course**: content evolves from year to year.
- This year we focus on three pillars:
    1. **Fundamentals**: statistical principles and models, a little bit of predictive learning
    2. **Optimization**
    3. **Generative modeling and learning**

Roadmap for Today

1. GenAI: from prediction to generation; why big data & compute matter
2. Generative modeling as sampling (unconditional and conditional)
3. Fundamentals: distributions, models, estimators
4. Maximum likelihood estimation (MLE) and its role in generative modeling

Overview: GenAI, Big Data, Compute

From Prediction to Generation

- Classical ML: **prediction** (classification/regression)
- Modern GenAI: **generation** of new content conditioned on prompts
  - images, text, audio, video, molecular structures
- Distinction:
  - Predictive models estimate an unknown target from observed data.
  - Generative models produce realistic *samples* resembling draws from a complex data distribution.

Key Idea (GenAI)

The ability to *generate*, not just predict, is a defining feature of the current GenAI revolution.

Artistic Images

Realistic Videos

ChatGPT

Draft Texts

These systems are "creative": they generate new objects.

Why GenAI is Linked to Big Data and Computation

- Generative modeling approximates extremely complex distributions over high-dimensional objects.
- Doing so reliably typically demands:
  1. **Massive training datasets** (multimodal, curated, filtered)
  2. **Large-scale optimization** (stochastic methods, many iterations)
- Computational enablers:
  - GPUs/TPUs and distributed systems
  - scalable stochastic optimization algorithms
  - data pipelines (storage, preprocessing, streaming)

Generative Modeling as Sampling

Modalities: Representing Data Numerically

We begin by thinking about different data types (*modalities*) and how to represent them numerically.

1. **Image:** $H \times W$ pixels with 3 color channels

$$Z \in \mathbb{R}^{H \times W \times 3}$$

2. **Video:** a sequence of T frames

$$Z \in \mathbb{R}^{T \times H \times W \times 3}$$

3. **Molecular structure:** N atoms with 3D coordinates

$$Z = (Z^1, \ldots, Z^N) \in \mathbb{R}^{3 \times N}, \quad Z^i \in \mathbb{R}^3$$

High-Dimensional Viewpoint

After choosing a representation, we can *flatten* the object into a vector:

$$Z \in \mathbb{R}^d,$$

where d may be extremely large.

Key Idea (Representation)

Once data are represented as vectors, modeling and generation become questions about probability distributions on $\mathbb{R}^d$.

Examples of dimensionality

Images: $d = H \cdot W \cdot 3$; Videos: $d = T \cdot H \cdot W \cdot 3$.

Generation as Sampling: Intuition

Suppose we want to generate an image of a dog.

- There is no single "best" dog image.
- There are many acceptable images, varying in realism and diversity.

Statistical viewpoint

Replace the vague question "How good is this sample?" by

"How likely is it under the data distribution?"

The Data Distribution

We posit an (unknown) data distribution and denote it by p_{data}.

- Higher probability: objects that look like valid data
- Lower probability: implausible / out-of-distribution objects

Key Idea (Generation as Sampling)

Generating an object Z is modeled as sampling from the (unknown) data distribution:

$$Z \sim p_{\text{data}}.$$

Datasets as Finite Samples

We do not observe p_{data} directly; we observe a dataset.

Key Idea (Dataset)

A dataset consists of a finite collection of samples

$$Z_1, \ldots, Z_n \overset{\text{i.i.d.}}{\sim} p_{\text{data}}.$$

- Images: large collections of photos (public/curated datasets)
- Videos: curated repositories
- Molecules/proteins: experimental databases (e.g., PDB)

What is a Generative Model?

A *generative model* aims to approximate p_{data} well enough to produce realistic samples.

Two core tasks

1. **Learning:** fit a model distribution using $Z_{1:n}$.
2. **Sampling:** draw new synthetic samples that resemble the data.

Key Idea (Core objective)

Learn a distribution whose samples match the dataset in relevant ways.

Conditional Generation

In many applications, we want generation *conditioned* on some input y.

Key Idea (Conditional Generation)

Conditional generation is modeled as sampling

$$Z \sim p_{\text{data}}(\cdot \mid y),$$

where y is a conditioning variable (label, prompt, side information).

Practical goal

A **single** model that can condition on many possible values of y (e.g., many text prompts).

Unconditional vs Conditional Generation

Unconditional generation

We generate objects without any side information:

$$Z \sim p_{\text{data}}.$$

Conditional generation

We generate objects *given* some input $Y = y$ (prompt/label/context):

$$Z \sim p_{\text{data}}(\,\cdot \mid Y = y).$$

Key Idea (Two viewpoints)

Unconditional models learn the overall distribution of the dataset; conditional models learn how the distribution changes with y.

Examples of Conditioning Variables Y

- **Class-conditional images:** $Y \in \{\text{cat, dog, car, } \dots\}$.
- **Text-to-image:** $Y =$ a text prompt describing desired content/style.
- **Inpainting / editing:** $Y =$ partially observed image $+$ mask.
- **Molecules/proteins:** $Y =$ desired properties (binding affinity, solubility, constraints).

Interpretation

Conditioning tells the model *what kind of sample* to generate.

How Datasets Support Conditional Generation

For conditional generation, we typically observe paired data:

$$(Z_1, Y_1), \ldots, (Z_n, Y_n) \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(z, y).$$

Two equivalent targets

- Learn the **joint** distribution $p_{\text{data}}(z, y)$, then sample $Z \mid Y = y$.
- Learn the **conditional** distribution $p_{\text{data}}(z \mid y)$ directly.

Key Idea (Unconditional as a special case)

Unconditional generation corresponds to sampling from the marginal:

$$p_{\text{data}}(z) = \int p_{\text{data}}(z, y)\, dy.$$

Sampling: What Does It Mean Operationally?

Once a model is trained (e.g., p_θ), sampling means producing a new random draw.

Unconditional sampling

$$\text{Sample } Z \sim p_\theta.$$

Produces a diverse set of samples reflecting the overall training distribution.

Conditional sampling

$$\text{Fix } y, \text{ then sample } Z \sim p_\theta(\cdot \mid y).$$

Produces diverse samples *consistent with the same condition* y.

Key Idea (Diversity vs control)

Unconditional: diversity with less control. Conditional: diversity *given* control.

Fundamentals: Notation and Models

Random Samples and Notation

Convention

Capital letters denote random variables; lower-case letters denote observed values.

Random sample

$X_1, \ldots, X_n$ are a random sample if

$$X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} p(x).$$

We write $X_{1:n} = (X_1, \ldots, X_n)$ and $x_{1:n} = (x_1, \ldots, x_n)$.

CDF and PDF

CDF

$$F(x) = \mathbb{P}(X \leq x), \qquad x \in \mathbb{R}.$$

PDF (when it exists)

If F is differentiable, then

$$p(x) = \frac{d}{dx} F(x).$$

Remark

For discrete X, $p(x)$ typically denotes a probability mass function (pmf).

Statistical Models

Definition

A statistical model is a family of distributions indexed by Θ:

$$\mathcal{P} = \{p_\theta : \theta \in \Theta\}.$$

Parametric vs Nonparametric

- **Parametric:** Θ is finite-dimensional (e.g., $\Theta \subseteq \mathbb{R}^d$).
- **Nonparametric:** no finite-dimensional parameterization.

Gaussian family (parametric)

$$\mathcal{P} = \left\{ p_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \,:\, \mu \in \mathbb{R},\ \sigma^2 > 0 \right\}.$$

The parameter $\theta = (\mu, \sigma^2)$ is **2-dimensional** (finite-dimensional).

Examples: Parametric and Nonparametric Models

Gaussian family (parametric)

$$\mathcal{P} = \left\{ p_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right) \;:\; \mu \in \mathbb{R},\; \sigma^2 > 0 \right\}.$$

Why parametric? The parameter $\theta = (\mu, \sigma^2)$ is **2-dimensional** (finite-dimensional).

All PDFs (nonparametric)

$$\mathcal{P} = \{\text{all PDFs on } \mathbb{R}\}.$$

Why nonparametric? There is no finite-dimensional parameter θ that can index *all* PDFs.

Key Idea (Course plan)

We start with parametric models (cleaner theory), then return to nonparametric models later.

Estimators and MLE

Estimators: Key Definitions

Point estimation

Given $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} p_\theta(x)$, point estimation produces a single value intended to approximate θ.

Estimator

$$\hat{\theta}_n = g(X_1, \ldots, X_n),$$

where $g : \mathbb{R}^n \to \Theta$ is measurable.

Consistency

$\hat{\theta}_n$ is consistent if $\hat{\theta}_n \xrightarrow{P} \theta$ as $n \to \infty$.

Bias and Unbiasedness

Bias

$$\mathrm{Bias}(\hat{\theta}_n) = \mathbb{E}[\hat{\theta}_n] - \theta.$$

Unbiased estimator

$\hat{\theta}_n$ is unbiased if $\mathrm{Bias}(\hat{\theta}_n) = 0$.

Key Idea (Important distinction)

Unbiasedness and consistency are different properties, and neither implies the other.

Example: Unbiased vs Consistent

Normal mean example

If $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} N(\mu, 1)$:

- $\hat{\mu}^{(1)} = X_1$ is unbiased but **not** consistent.
- $\hat{\mu}_n^{(2)} = \frac{1}{n}\sum_{i=1}^{n} X_i$ is unbiased and consistent.
- $\hat{\mu}_n^{(3)} = \frac{1}{n+1}\sum_{i=1}^{n} X_i$ is biased but consistent.

Likelihood and Log-Likelihood

Likelihood (single observation)

$$L(\theta; x) = p_\theta(x).$$

Joint likelihood (i.i.d. data)

$$L_n(\theta; x_{1:n}) = \prod_{i=1}^{n} p_\theta(x_i).$$

Log-likelihood

$$\ell_n(\theta; x_{1:n}) = \sum_{i=1}^{n} \log p_\theta(x_i).$$

Maximum Likelihood Estimator (MLE)

Definition

$$\hat{\theta}_n \in \arg\max_{\theta \in \Theta} L_n(\theta; x_{1:n}) \quad \Longleftrightarrow \quad \hat{\theta}_n \in \arg\max_{\theta \in \Theta} \ell_n(\theta; x_{1:n}).$$

Key Idea (Computation)

Working with ℓ_n is numerically more stable and turns products into sums.

MLE Example: Gaussian

Gaussian distribution

If $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} N(\mu, \sigma^2)$, then

$$\hat{\mu}_n = \bar{X}_n, \qquad \hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2.$$

Note

The MLE for σ^2 uses $1/n$ (the unbiased sample variance uses $1/(n-1)$).

Asymptotics of the MLE (Informal)

Asymptotic normality

$$\sqrt{n}\,(\hat{\theta}_n - \theta) \xrightarrow{D} N\big(0,\, I(\theta)^{-1}\big).$$

Scalar Fisher information

$$I(\theta) = \mathbb{E}_\theta\left[\left(\frac{\partial}{\partial\theta}\log p_\theta(X)\right)^2\right] = -\mathbb{E}_\theta\left[\frac{\partial^2}{\partial\theta^2}\log p_\theta(X)\right].$$

Cramér–Rao type bound (informal)

For unbiased $\tilde{\theta}_n$,

$$\mathrm{Var}(\tilde{\theta}_n) \geq \frac{1}{n\,I(\theta)}.$$

Connecting MLE to Generative Modeling

- In generative modeling, the target is the (unknown) p_{data}.
- In a **parametric** generative model, posit $\{p_\theta : \theta \in \Theta\}$ and estimate θ via MLE:

$$\hat{\theta} \in \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p_\theta(z_i).$$

- After learning $\hat{\theta}$, we can **sample**:

$$Z \sim p_{\hat{\theta}}.$$

Key Idea (Big picture)

Modeling + Optimization + Compute $\Rightarrow$ scalable learning and sampling.

Wrap-Up

Wrap-Up

- GenAI: generating new content $\approx$ sampling from a learned distribution.
- Big data + compute make modern GenAI feasible at scale.
- Foundations today:
  - representations and modalities; p_{data}; datasets as i.i.d. samples
  - models (parametric/nonparametric) + examples
  - estimators: bias, consistency; MLE and Gaussian example
- Next: predictive modeling