YZM2011

Introduction to Machine Learning

Week 2: Probability Theory and Statistics Fundamentals

Instructor: Ekrem Çetinkaya

Date: 24.02.2026

Course Content

Fundamentals

- Introduction to probability theory
- Basic probability rules (Sum & Product)
- Conditional probability

Bayes' Theorem

- Bayes' theorem derivation
- Prior, Likelihood, Posterior
- Bayesian inference

Distributions

- Gaussian (Univariate & Multivariate)
- Bernoulli, Binomial, Beta
- Multinomial, Dirichlet

Statistics

- Expectations and Covariances
- MLE vs. Bayesian Inference
- Bias-Variance decomposition

Recommended Reading

This week's material covers the probabilistic foundations that underpin virtually every machine learning algorithm. The reference text is Bishop's *Pattern Recognition and Machine Learning* (PRML).

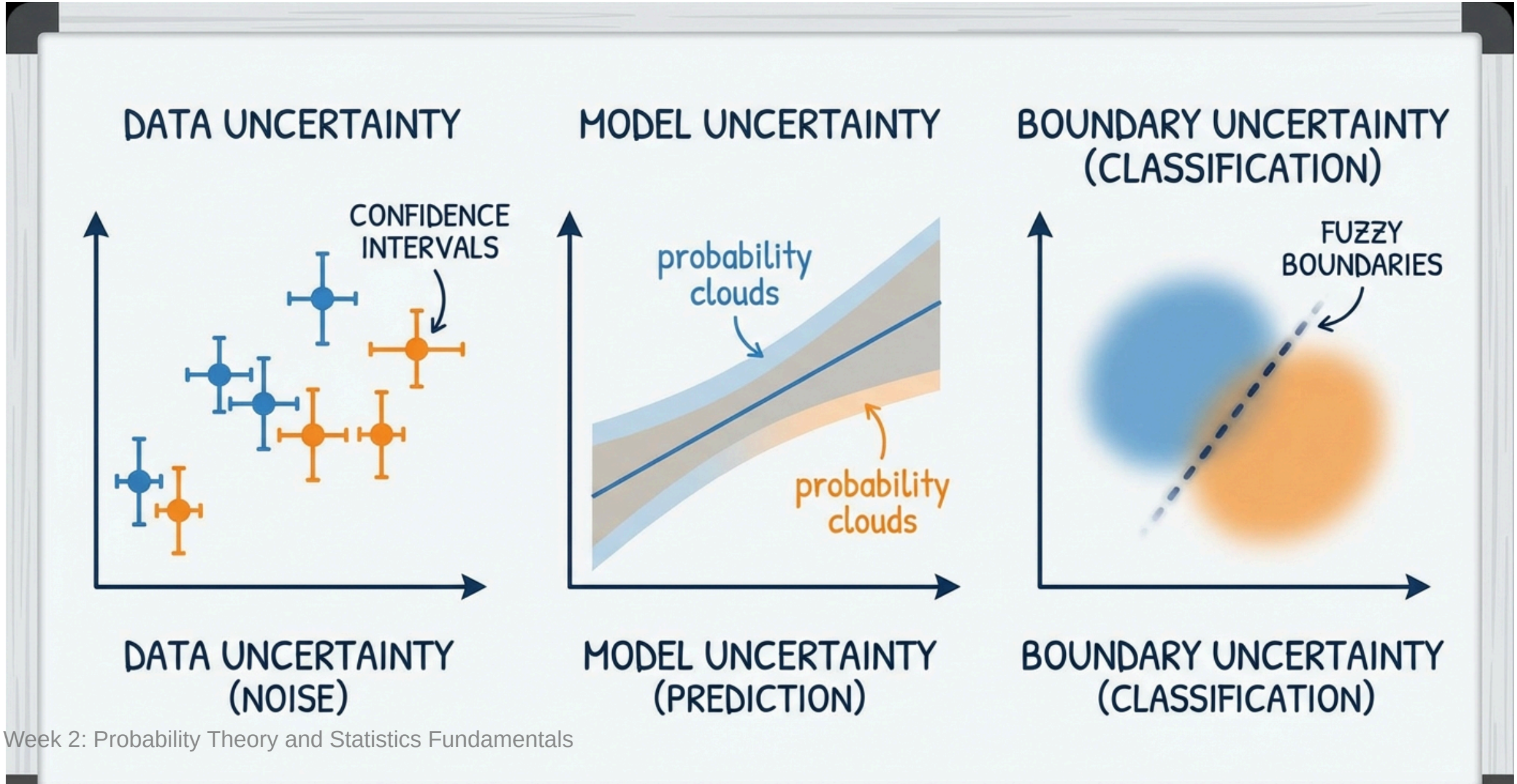
Primary Reading

- **Section 1.2: Probability Theory** (pp. 12-31) — Rules of probability, Bayes' theorem, Gaussian distribution
- **Chapter 2: Probability Distributions** (pp. 67-136) — Binary variables (Beta-Bernoulli), multinomial variables (Dirichlet-Multinomial), Gaussian distribution geometry and inference

Key Figures to Study

Figure	Topic
Figure 1.10	Grid diagram for sum/product rules
Figure 1.15	Bias of MLE variance
Figure 2.3	Sequential Bayesian updates
Figure 2.7	Geometry of multivariate Gaussian

Why Probability Theory?



Uncertainty in Machine Learning

Machine learning is fundamentally about making predictions under **uncertainty**. Every dataset is noisy, every model is an approximation, and every prediction carries some degree of doubt. Probability theory gives us the language and the tools to reason about this uncertainty rigorously.

- **Noisy data** — Measurements can always be erroneous or incomplete
- **Limited data** — We cannot see all possible cases; we work with a finite sample
- **Model uncertainty** — Which model is correct? There may be many plausible explanations for the same data
- **Prediction uncertainty** — How confident are we in a given prediction?

> *"Probability theory is nothing but common sense reduced to calculation."*
> — Pierre-Simon Laplace

Without probability, we cannot quantify how much we should trust a prediction, compare competing models, or update our beliefs as new data arrives.

Probability: Two Perspectives

Before we start calculating, it is worth pausing to ask: what does a probability actually *mean*? There are two fundamentally different answers, and both play important roles in machine learning.

Frequentist Approach

Probability is the **long-run frequency** of an event in repeated experiments.

- "If I roll a die 1000 times, approximately 167 times I'll get a 6"
- Requires repeatable experiments
- **Objective** interpretation — the probability is a property of the experiment, not of the observer

Bayesian Approach

Probability is a **degree of belief** about a proposition, given the available evidence.

- "I believe there's a 30% chance it will rain tomorrow"
- Valid for single, unrepeatable events
- **Subjective** but internally consistent interpretation — different people can assign different probabilities to the same event, and that is fine

> Both perspectives are used in machine learning. Frequentist methods dominate classical statistics (hypothesis tests, confidence intervals), while Bayesian methods are increasingly popular for their ability to incorporate prior knowledge and quantify uncertainty naturally.

Basic Concepts: Sample Space

To work with probability formally, we need a few definitions. The **sample space** Ω is the set of all possible outcomes of an experiment. An **event** is any subset of Ω — it is the thing we assign a probability to.

Experiment	Sample Space (Ω)	Size
Coin flip	{Heads, Tails}	2
Die roll	{1, 2, 3, 4, 5, 6}	6
Two dice roll	{(1,1), (1,2), ..., (6,6)}	36
Temperature measurement	$\mathbb{R}$ (continuous)	∞

Events

An event A is a **subset** of the sample space. For example:

- A = "Die shows even" = {2, 4, 6}
- A = "At least one 6" (when rolling two dice)

The probability of an event tells us how likely that subset is to occur.

Probability Axioms (Kolmogorov)

All of probability theory rests on just **three axioms**, formalized by Andrey Kolmogorov in 1933. Every rule we derive later — Bayes' theorem, the law of total probability, independence — follows from these.

For any event A:

1. Non-negativity

$$P(A) \geq 0$$

Probabilities are never negative.

2. Normalization

$$P(\Omega) = 1$$

Something must happen — the probability of the entire sample space is 1.

3. Countable Additivity

If $A_1, A_2, \ldots$ are mutually exclusive events:

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

The probability of "at least one" of several non-overlapping events is the sum of their individual probabilities.

The Two Fundamental Rules

Bishop builds all of probability from just two rules, derived from the grid diagram in Figure 1.10 of PRML. If you understand these two rules deeply, everything else follows.

1. Sum Rule (Marginalization)

$$p(X) = \sum_{Y} p(X, Y)$$

To find the probability of X alone, **sum out** (marginalize) the other variable Y from the joint distribution. This is how we go from a joint distribution to a marginal one.

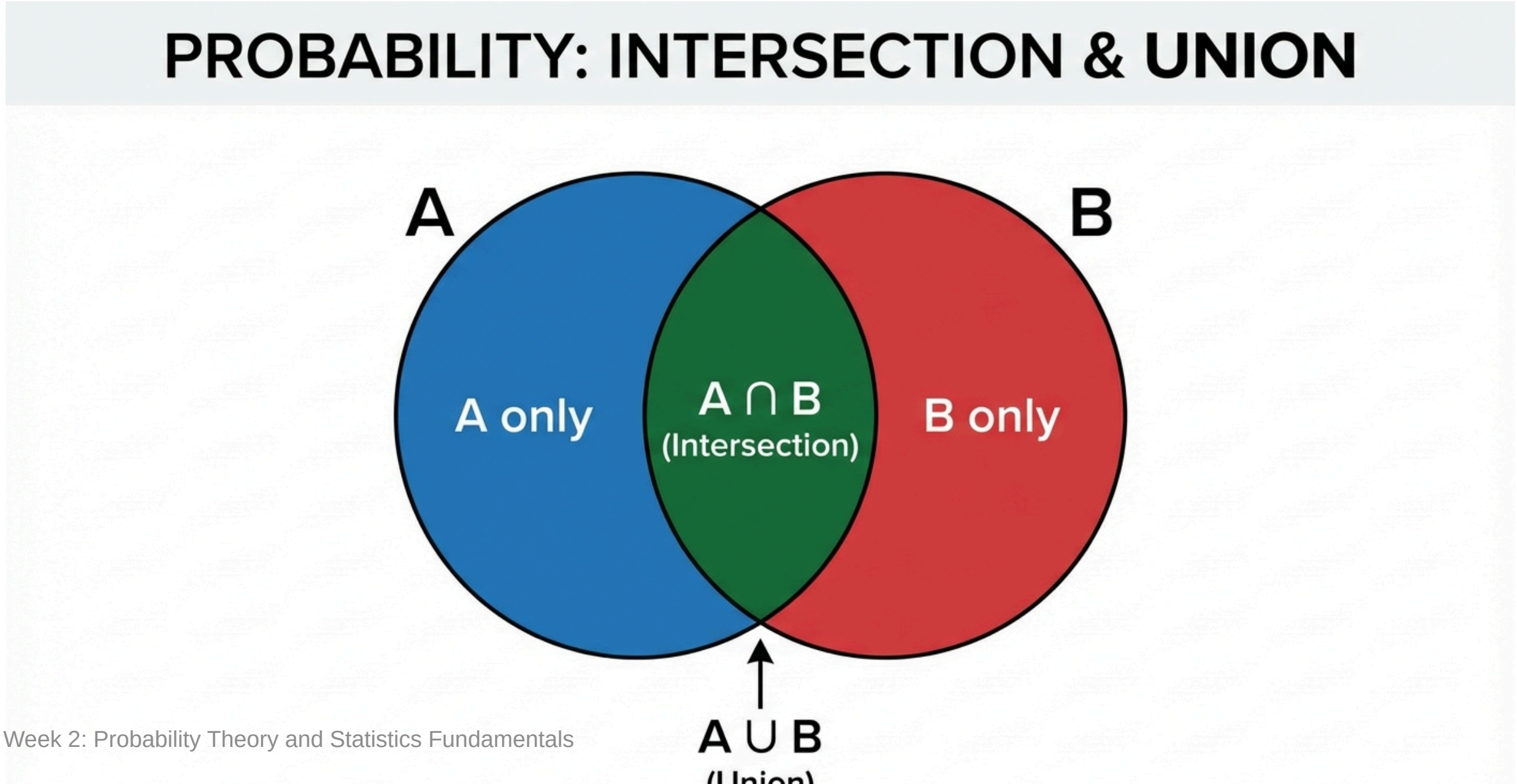
2. Product Rule

$$p(X, Y) = p(Y|X)\, p(X)$$

The joint probability of X and Y equals the conditional probability of Y given X, times the probability of X. This is the bridge between joint and conditional distributions.

> *"These two simple rules form the basis for all of the probabilistic machinery that we use throughout this book."* — Bishop, p. 15

Addition Rule: Venn Diagram



Addition Rule

When two events can overlap, simply adding their probabilities double-counts the intersection. The addition rule corrects for this:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

The Venn diagram on the right makes this clear — the overlap region belongs to both A and B, so it would be counted twice unless we subtract it once.

Example:

- $P(\text{Male}) = 0.5$
- $P(\text{Wears Glasses}) = 0.3$
- $P(\text{Male} \cap \text{Wears Glasses}) = 0.15$

$$P(\text{Male} \cup \text{Wears Glasses}) = 0.5 + 0.3 - 0.15 = 0.65$$

If the two events are **mutually exclusive** ($A \cap B = \emptyset$), the formula simplifies to $P(A \cup B) = P(A) + P(B)$ — no correction needed.

Independence

Two events A and B are **independent** if knowing that one occurred tells you nothing about the other. Formally:

$$P(A \cap B) = P(A) \cdot P(B)$$

This is equivalent to saying $P(A|B) = P(A)$ — conditioning on B does not change your belief about A.

Independent Examples

- Two consecutive die rolls — the die has no memory
- Heights of two unrelated people

Dependent Examples

- Same person's height and weight — taller people tend to weigh more
- Today's and tomorrow's weather — weather has temporal correlation

> ⚠️ **Independence ≠ Mutual Exclusivity** — these are two different concepts that students often confuse. We clarify the difference on the next slide.

Independence vs Mutual Exclusivity

These two concepts sound similar but are logically opposite in an important way.

Mutually Exclusive

$$A \cap B = \emptyset \quad \Rightarrow \quad P(A \cap B) = 0$$

Two events **cannot occur simultaneously**. If you know A happened, you know for certain B did not.

Example: Rolling a 3 and rolling a 5 on the same die throw.

Independent

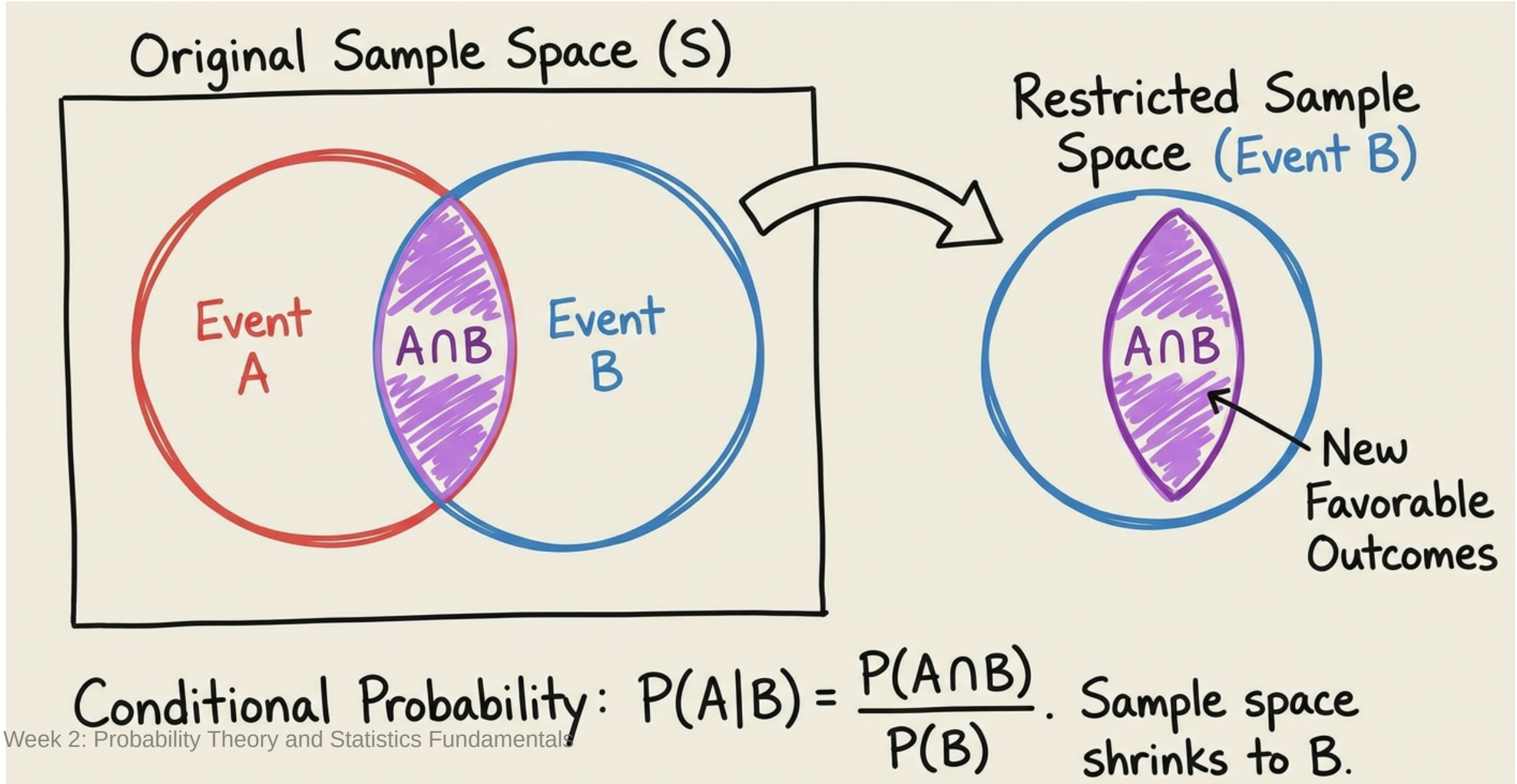
$$P(A \cap B) = P(A) \cdot P(B)$$

Knowing one event occurred gives you **no information** about the other. Both can happen at the same time.

Example: Rolling a 3 on one die and rolling a 5 on a second die.

> **The critical insight:** If $P(A) > 0$ and $P(B) > 0$, then mutually exclusive events are always **dependent** — learning that one occurred immediately tells you the other did not. Conversely, independent events (with positive probability) are never mutually exclusive — both can occur together.

Conditional Probability: Visualization



Conditional Probability

Conditional probability answers the question: "Given that I already know B occurred, how likely is A?" It is defined as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0$$

The intuition is straightforward — once we learn B has occurred, we **restrict** our attention to the subset of outcomes where B is true, and then ask what fraction of that subset also contains A.

This is one of the most important concepts in probability, because in machine learning we are almost always conditioning on something — observed data, a particular class label, a specific feature value.

Conditional Probability: Medical Test Example

Let us work through a classic example that illustrates why conditional probability is so counterintuitive — and why we need Bayes' theorem.

Setup: A disease affects 1% of the population. A diagnostic test has the following characteristics:

- Disease prevalence: $P(D) = 0.01$ (1 in 100 people)
- Test sensitivity: $P(+|D) = 0.95$ (95% chance of positive result if the person is sick)
- False positive rate: $P(+|D') = 0.05$ (5% chance of positive result if the person is healthy)

Question: A patient tests positive. What is the probability that they actually have the disease?

$$P(D|+) = ?$$

Most people intuitively guess something high — 90% or more — because the test is 95% accurate. The actual answer, as we will see, is shockingly low. To find it, we need **Bayes' Theorem**.

Product Rule

The product rule follows directly from the definition of conditional probability. Rearranging $P(A|B) = P(A \cap B)/P(B)$ gives:

$$\boxed{P(A \cap B) = P(A|B) \cdot P(B)}$$

By symmetry (the joint probability does not care about ordering):

$$P(A \cap B) = P(B|A) \cdot P(A)$$

This symmetry is exactly what leads to Bayes' theorem — setting the two expressions equal and solving.

Chain Rule

The product rule generalizes to more than two events by conditioning repeatedly:

$$P(A \cap B \cap C) = P(A|B, C) \cdot P(B|C) \cdot P(C)$$

In general, for n events:

$$P(A_1, A_2, \ldots, A_n) = P(A_1) \prod_{i=2}^{n} P(A_i \mid A_1, \ldots, A_{i-1})$$

The chain rule is the foundation of many probabilistic models, including Bayesian networks and autoregressive language models.

Law of Total Probability

Sometimes we cannot compute $P(A)$ directly, but we *can* compute it conditioned on each of several mutually exclusive scenarios. The law of total probability lets us piece those conditional probabilities together.

Events $B_1, B_2, \ldots, B_n$ form a **partition** of Ω if they are mutually exclusive ($B_i \cap B_j = \emptyset$ for $i \neq j$) and exhaustive ($\bigcup B_i = \Omega$). Then:

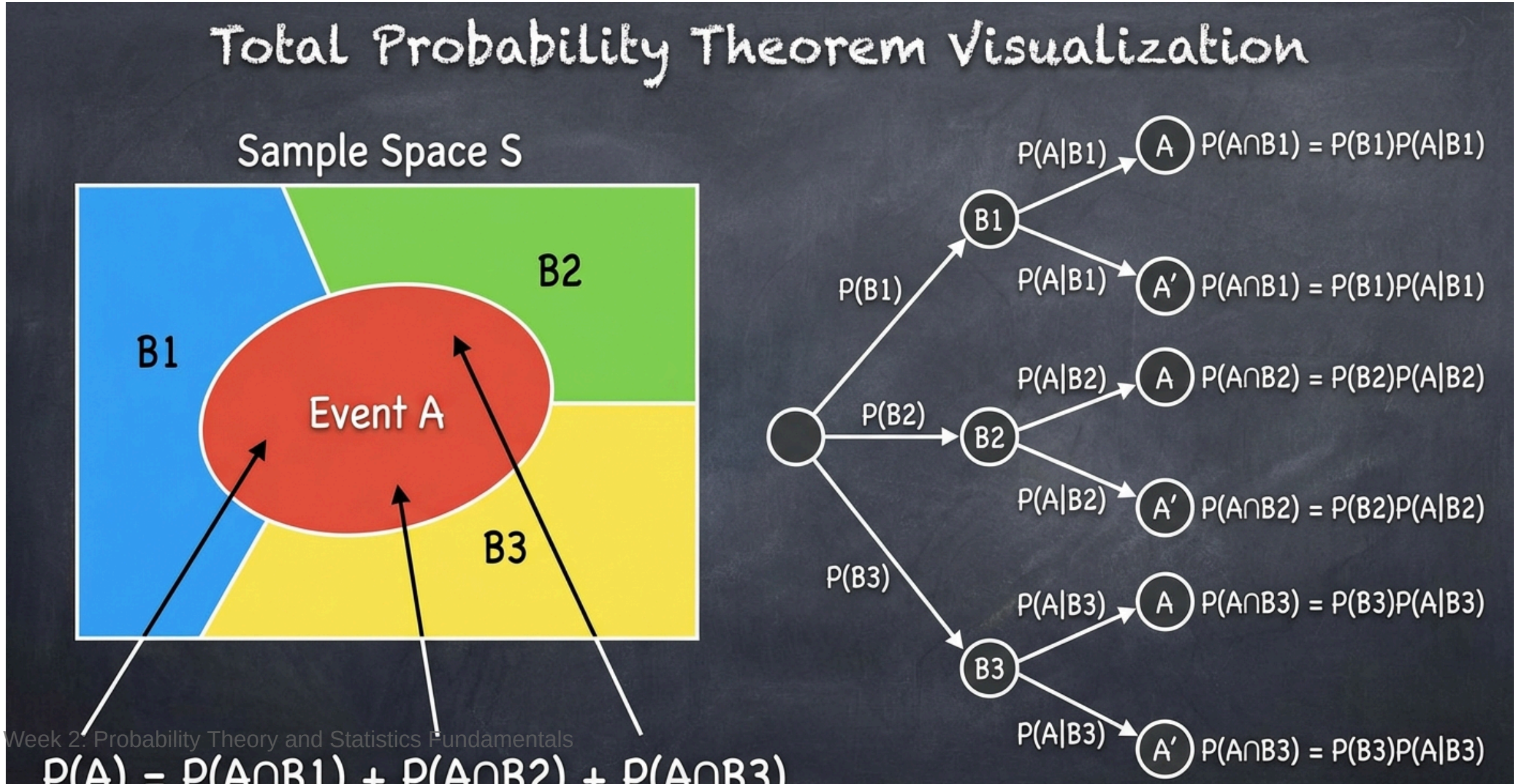
$$P(A) = \sum_{i=1}^{n} P(A|B_i) \cdot P(B_i)$$

In the most common **binary case** — which we will use heavily for Bayes' theorem — the partition is simply $\{B, B^c\}$:

$$P(A) = P(A|B) \cdot P(B) + P(A|B^c) \cdot P(B^c)$$

This rule is the key to computing the **evidence** (denominator) in Bayes' theorem.

Total Probability: Visualization



Total Probability: Factory Example

A factory has three machines that produce items at different rates and with different defect probabilities:

Machine	Share of Production	Defect Rate
M_1	50%	3%
M_2	30%	4%
M_3	20%	5%

Question: What is the overall probability that a randomly selected item is defective?

We cannot answer this directly — but we *can* answer it for each machine. By the law of total probability:

$$P(D) = P(D|M_1)P(M_1) + P(D|M_2)P(M_2) + P(D|M_3)P(M_3)$$

$$P(D) = 0.03 \times 0.5 + 0.04 \times 0.3 + 0.05 \times 0.2 = 0.037$$

Bayes' Theorem

Bayes' theorem is arguably the single most important result in probability for machine learning. It tells us how to **update our beliefs** in light of new evidence.

The derivation is simple. From the product rule, the joint probability can be written in two ways:

$$p(Y|X)\,p(X) = p(X|Y)\,p(Y)$$

Dividing both sides by $p(X)$ gives **Bayes' Theorem** (PRML Eq. 1.12):

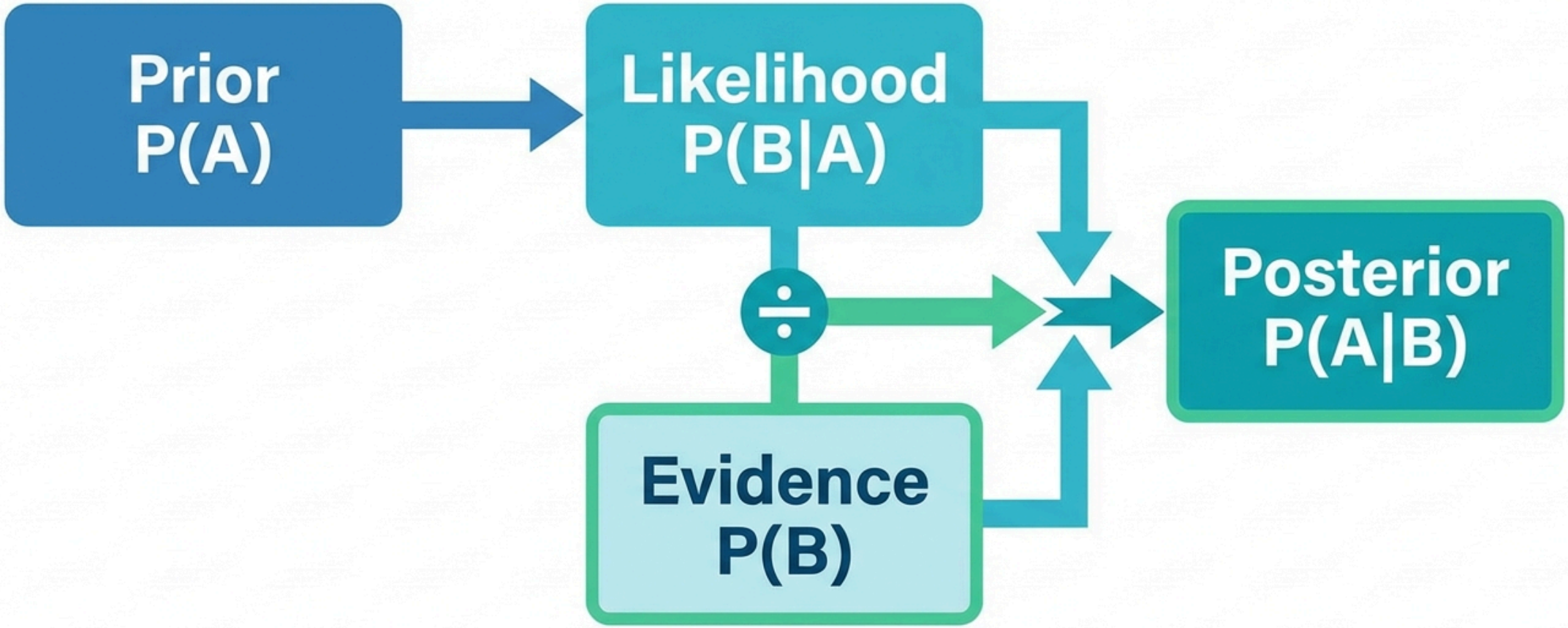
$$\boxed{p(Y|X) = \frac{p(X|Y)\,p(Y)}{p(X)}}$$

The denominator $p(X)$ can be computed using the sum rule (marginalization):

$$p(X) = \sum_{Y'} p(X|Y')\,p(Y')$$

This denominator is often called the **evidence** or **marginal likelihood**. It acts as a normalizing constant to ensure the posterior sums to 1.

Bayes' Theorem: Visual Overview



$$P(A|B) = P(B|A)P(A)/P(B)$$

Bayes' Theorem: Terminology

Each term in Bayes' theorem has a name and a specific role. Understanding these names is essential, because you will encounter them everywhere in machine learning literature.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Term	Name	Role
$P(A)$	**Prior**	What we believed about A *before* seeing data
$P(A \mid B)$	**Posterior**	What we believe about A *after* seeing data
$P(B \mid A)$	**Likelihood**	How probable the observed data is if A were true
$P(B)$	**Evidence**	Normalizing constant (same for all hypotheses)

Since the evidence $P(B)$ is the same regardless of which hypothesis we are evaluating, we often write:

> **Posterior ∝ Likelihood × Prior**

Bayes' Theorem: Medical Test Solution

Now we can answer the medical test question from earlier. We have the prior $P(D) = 0.01$, the likelihood $P(+|D) = 0.95$, and the false positive rate $P(+|D') = 0.05$.

Step 1: Compute the evidence $P(+)$ using total probability:

$$P(+) = P(+|D)\,P(D) + P(+|D')\,P(D') = 0.95 \times 0.01 + 0.05 \times 0.99 = 0.059$$

Step 2: Apply Bayes' theorem:

$$P(D|+) = \frac{P(+|D) \cdot P(D)}{P(+)} = \frac{0.95 \times 0.01}{0.059} = \frac{0.0095}{0.059} \approx \mathbf{0.161}$$

Despite a 95%-accurate test, a positive result means only a **16% chance** of actually having the disease.

The Base Rate Fallacy

Why is the result so low? The key insight is the **base rate** — the disease is very rare (1%), so the vast majority of people being tested are healthy. Even a small false-positive rate on a large healthy population generates many false alarms.

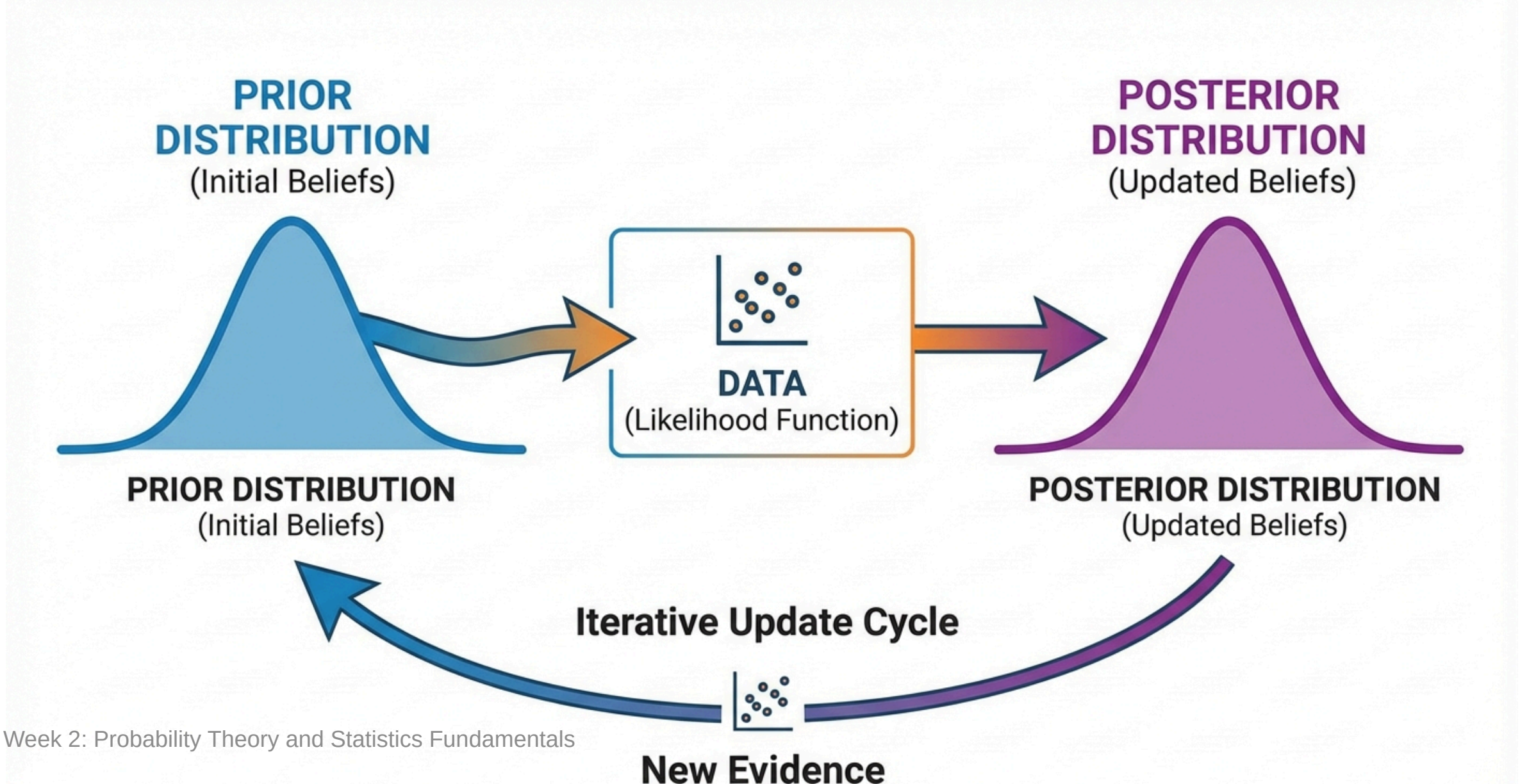
A frequency table makes this vivid. Imagine testing 10,000 people:

Group	People	Test Positive
Sick (1%)	100	95 (true positives)
Healthy (99%)	9,900	495 (false positives)
Total positive		**590**

Only 95 of the 590 positive results are actually sick — that is 95/590 ≈ **16%**.

> **Lesson:** When the base rate of a condition is low, false positives can vastly outnumber true positives. This is called the **base rate fallacy** and it has real consequences in medical screening, spam filtering, and fraud detection.

Bayesian Update Cycle



Bayes' Theorem: Intuitive Meaning

At its core, Bayes' theorem describes a **learning process** — how a rational agent should update its beliefs when it encounters new evidence.

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

The cycle works like this:

1. **Start** with a belief about the world (the *prior*)
2. **Observe** new data
3. **Update** your belief using Bayes' theorem (the *posterior*)
4. **Repeat** — the posterior from this round becomes the prior for the next

This is exactly what machine learning does: it starts with some initial model (prior), sees training data (evidence), and produces an updated model (posterior). In this sense, Bayes' theorem is the mathematical formula for **learning from data**.

Sequential Bayesian Update

One of the most elegant features of Bayesian reasoning is that it works **sequentially**. When we observe multiple data points, we do not need to process them all at once — we can update one observation at a time, and the final answer is the same.

$$P(A|B_1, B_2) \propto P(B_2|A, B_1) \cdot \underbrace{P(A|B_1)}_{\text{yesterday's posterior}}$$

The posterior after the first observation becomes the prior for the second.

Example: Is a coin biased? Suppose our prior is $P(\text{Biased}) = 0.1$. We flip the coin three times and get heads every time:

- After 1st heads: $P(\text{Biased}|H)$ increases
- After 2nd heads: $P(\text{Biased}|HH)$ increases further
- After 3rd heads: $P(\text{Biased}|HHH) \approx 0.47$

Each observation shifts our belief. With enough evidence, the prior becomes irrelevant and the data dominates — this is a fundamental property of Bayesian learning.

> Each observation updates our belief!

Practice: Conditional Probability

In a classroom:

- 60% of students are male, 40% female
- 30% of males wear glasses
- 20% of females wear glasses

Questions:

1. What is the probability that a randomly selected student wears glasses?
2. Given that a student wears glasses, what is the probability they are male?

Hint: Use the law of total probability for (1) and Bayes' theorem for (2).

Practice: Bayes' Theorem

Two factories produce the same product:

- Factory A: 70% market share, 2% defective
- Factory B: 30% market share, 5% defective

A product is found to be defective.

Question: What is the probability that this product came from Factory A?

Hint: Identify prior, likelihood, and compute the evidence using total probability.

Random Variables

So far we have talked about events (subsets of the sample space). A **random variable** is a function that assigns a numerical value to each outcome, which lets us do algebra with probability.

$$X : \Omega \to \mathbb{R}$$

Examples:

- X = "Sum of two dice" → $\{2, 3, \ldots, 12\}$
- X = "Person's height" → $\mathbb{R}^+$ (continuous)
- X = "Is email spam?" → $\{0, 1\}$ (binary)

Random variables come in two flavors, and the distinction matters because it determines how we define their probability distribution:

	Discrete	**Continuous**
Values	Countable (finite or infinite)	Uncountable (a range on $\mathbb{R}$)
Distribution	PMF: $P(X = x)$	PDF: $f(x)$, but $P(X = x) = 0$
Normalization	$\sum_x p(x) = 1$	$\int f(x)\, dx = 1$

Probability Mass Function (PMF)

For a **discrete** random variable, the PMF gives the probability of each possible value:

$$p(x) = P(X = x)$$

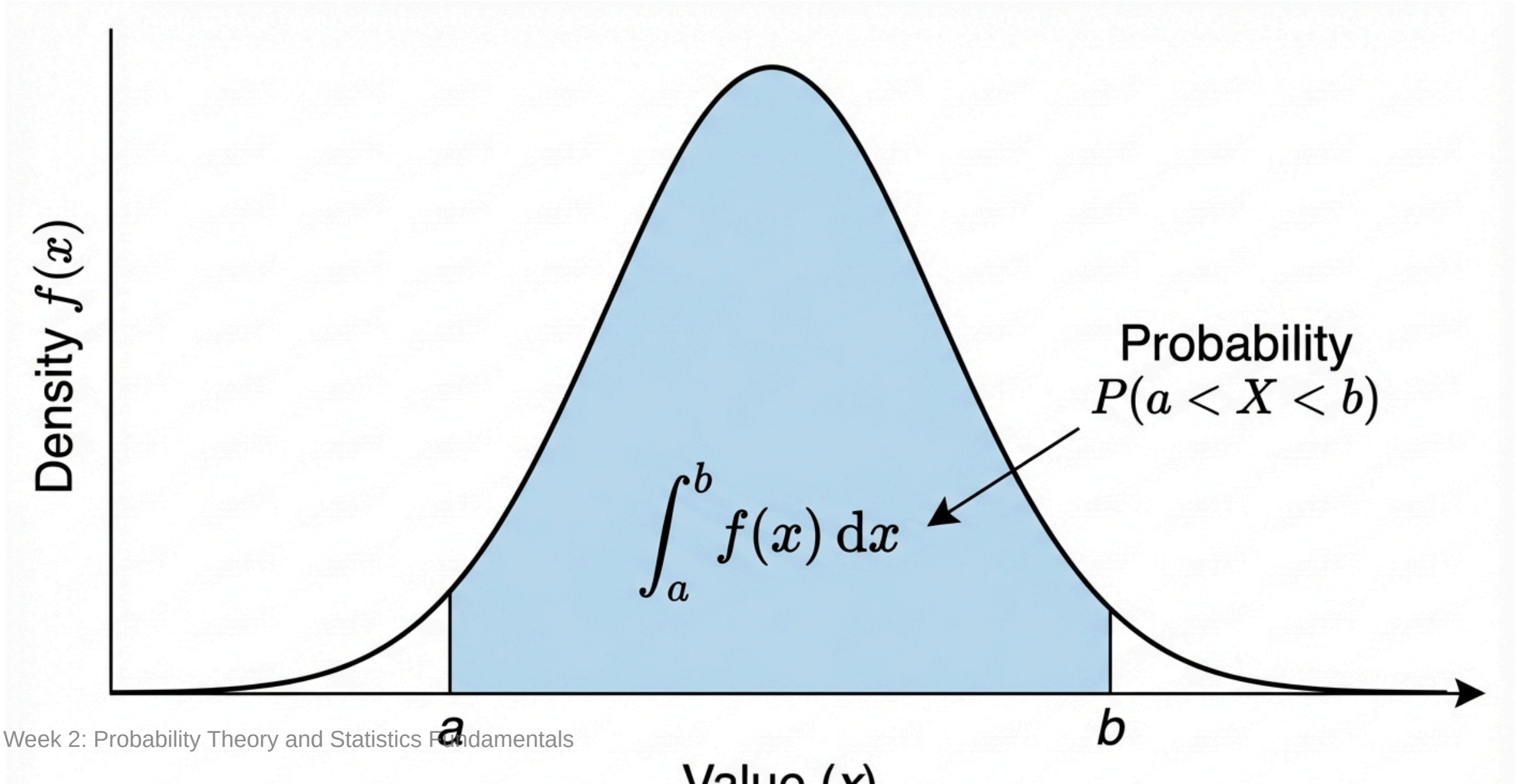
It must satisfy two properties: (1) $p(x) \geq 0$ for all x, and (2) $\sum_x p(x) = 1$ — probabilities are non-negative and must sum to one.

Example: Fair die

x	**1**	**2**	**3**	**4**	**5**	**6**
$p(x)$	1/6	1/6	1/6	1/6	1/6	1/6

Example: Sum of two dice — The value 7 has the highest probability: $p(7) = 6/36 = 1/6$, because there are six ways to roll a sum of 7 (1+6, 2+5, 3+4, 4+3, 5+2, 6+1) out of 36 equally likely outcomes.

Probability Density Function (PDF)



PDF: Definition and Properties

For a **continuous** random variable, we cannot assign probability to individual points — there are uncountably many of them. Instead, we define a density function $f(x)$ such that:

$$P(a \leq X \leq b) = \int_a^b f(x)\,dx$$

The probability comes from the **area under the curve**, not from the height at any single point. This means:

- $f(x) \geq 0$ for all x
- $\int_{-\infty}^{\infty} f(x)\,dx = 1$

Important: $P(X = x) = 0$

For any single value x, the probability is exactly zero. This is counterintuitive but follows from the integral of a single point being zero. It also means $P(X \leq a) = P(X < a)$ for continuous variables — the boundary point does not matter.

Cumulative Distribution Function (CDF)

The CDF answers the question "what is the probability that X is at most x?" It works for both discrete and continuous random variables, which makes it a universal tool.

$$F(x) = P(X \leq x)$$

Properties:

- Monotonically non-decreasing — more outcomes are included as x grows
- $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$
- For continuous variables: $f(x) = dF(x)/dx$ and $F(x) = \int_{-\infty}^{x} f(t)\, dt$

The CDF and PDF are two representations of the same information. The PDF tells you where probability is *concentrated* (the peaks); the CDF tells you how much probability has *accumulated* up to each point.

Joint Distribution

When we have **two or more** random variables, their joint distribution describes how they behave together — not just individually. This is essential in ML, where we almost always model relationships between variables.

Discrete

$$p(x, y) = P(X = x, Y = y)$$

$$\sum_{x} \sum_{y} p(x, y) = 1$$

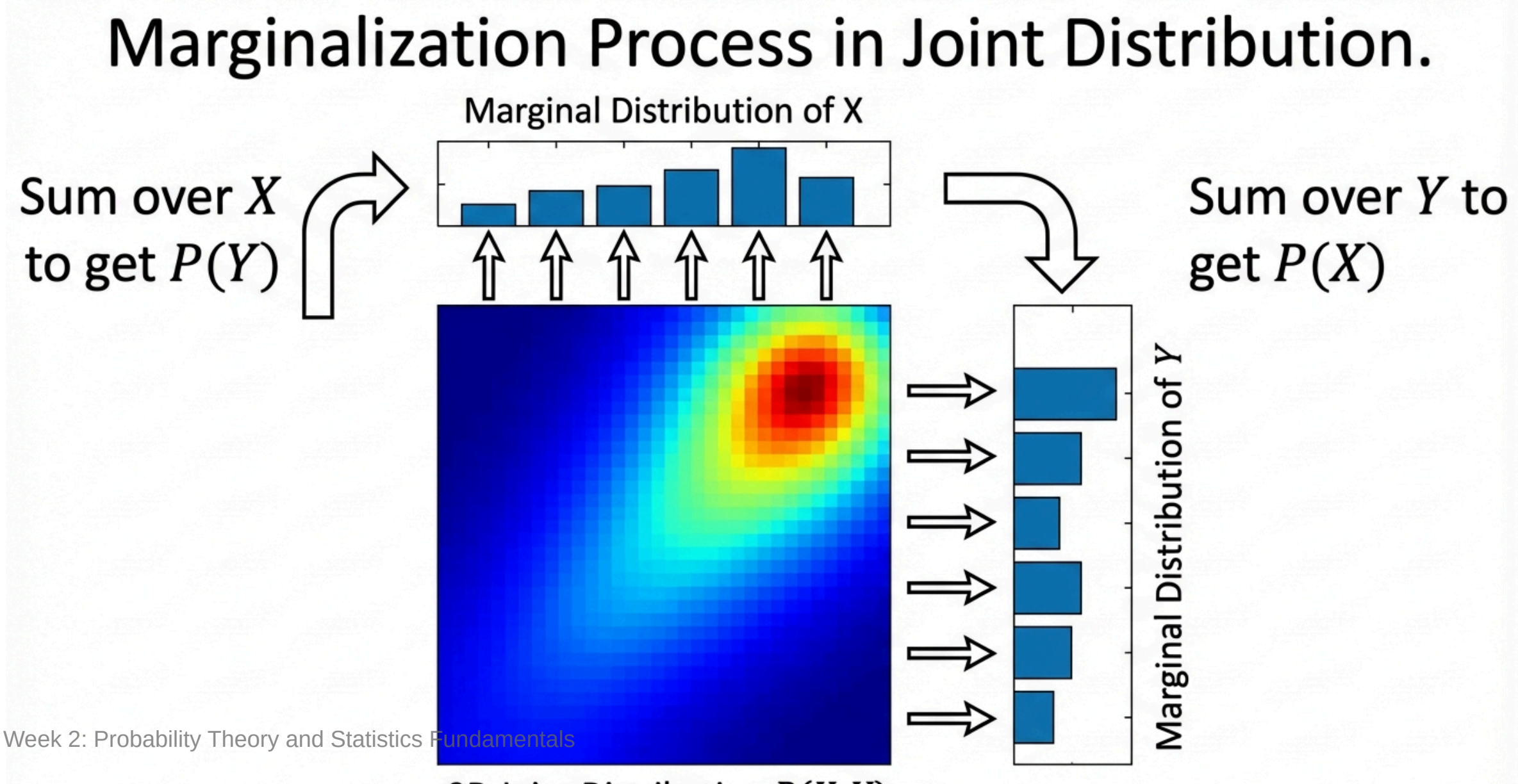
Continuous

$$P(a \leq X \leq b,\, c \leq Y \leq d) = \int_{a}^{b} \int_{c}^{d} f(x, y)\, dy\, dx$$

$$\int \int f(x, y)\, dx\, dy = 1$$

The joint distribution contains *all* the information about the relationship between X and Y. From it, we can derive marginals, conditionals, and test independence.

Marginal Distribution: Visualization



Marginal Distribution

If we know the joint distribution of X and Y but only care about X, we **marginalize out** Y — summing (or integrating) over all its possible values:

$$p(x) = \sum_{y} p(x, y) \qquad \text{or} \qquad f(x) = \int f(x, y)\, dy$$

Intuitively, we are "ignoring" or "averaging over" the variable we do not care about.

Example:

	$Y = 0$	$Y = 1$	$p(x)$
$X = 0$	0.1	0.2	**0.3**
$X = 1$	0.3	0.4	**0.7**
$p(y)$	**0.4**	**0.6**	1.0

The row sums give $p(x)$, the column sums give $p(y)$. This is exactly the **sum rule** we saw earlier, applied to random variables.

Conditional Distribution

The conditional distribution tells us how one variable behaves when we fix or observe the other. It is defined by slicing the joint and renormalizing:

$$p(y|x) = \frac{p(x,y)}{p(x)} \qquad \text{or} \qquad f(y|x) = \frac{f(x,y)}{f(x)}$$

For each fixed value of x, $p(y|x)$ is a valid probability distribution over y — it sums (or integrates) to 1. And the product rule follows immediately: $p(x,y) = p(y|x) \cdot p(x)$.

In machine learning, conditional distributions appear everywhere. A classifier models $p(y|x)$ — the probability of a label y given features x. A generative model might model $p(x|y)$ — the probability of features given a class.

Independent Random Variables

Two random variables X and Y are **independent** if their joint distribution factorizes:

$$p(x, y) = p(x) \cdot p(y) \quad \text{for all } x, y$$

This is equivalent to $p(x|y) = p(x)$ — knowing Y tells you nothing about X.

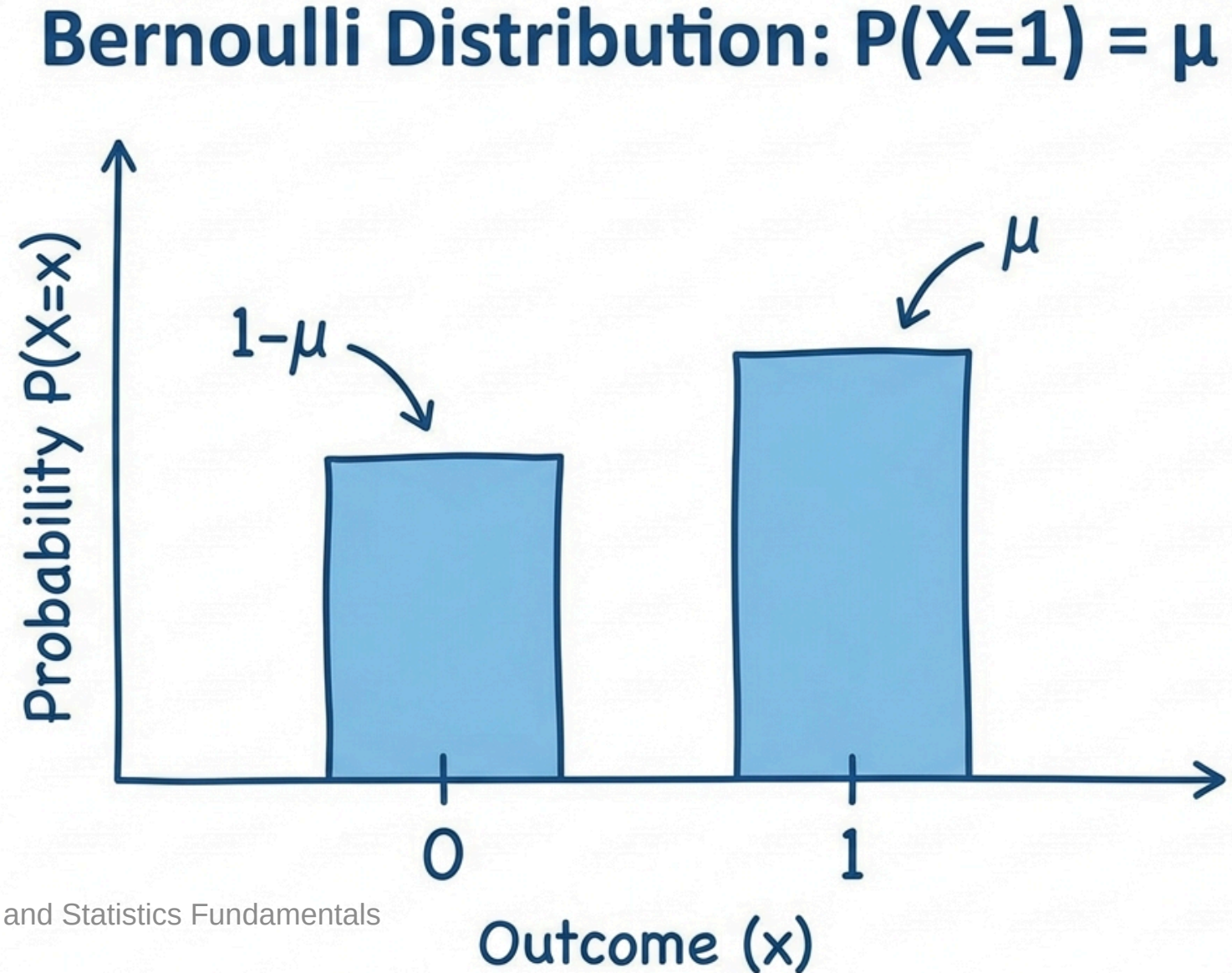
The i.i.d. Assumption

Perhaps the most important assumption in machine learning is that data points are **independent and identically distributed** (i.i.d.). This means every data point is drawn from the same distribution, and the draws do not influence each other:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i)$$

This assumption dramatically simplifies learning — the joint likelihood of the dataset becomes a product of individual likelihoods. Without it, we would need to model all possible dependencies between data points, which is usually intractable.

Bernoulli Distribution



Bernoulli Distribution: Definition

The simplest probability distribution — it models a single experiment with exactly two outcomes ($X \in \{0, 1\}$).

$$p(x|\mu) = \mu^x (1-\mu)^{1-x}$$

where $\mu \in [0, 1]$ is the probability of "success" ($X = 1$).

This compact notation captures both cases: when $x = 1$ we get μ, and when $x = 0$ we get $1 - \mu$.

Examples in ML:

- Coin flip (fair: $\mu = 0.5$, biased: $\mu \neq 0.5$)
- Spam / not spam classification
- Click / no click prediction

Despite its simplicity, the Bernoulli distribution is the building block for many important distributions: the Binomial (multiple Bernoulli trials), the Categorical (multiple outcomes), and the Beta (the conjugate prior for μ).

Bernoulli: Properties

The expected value and variance of the Bernoulli distribution have clean, intuitive forms:

$$E[X] = \mu \qquad\qquad \mathrm{Var}(X) = \mu(1-\mu)$$

The variance is a symmetric function of μ that reaches its **maximum at** $\mu = 0.5$ — this is the point of greatest uncertainty, where both outcomes are equally likely. At the extremes ($\mu = 0$ or $\mu = 1$), the outcome is certain and the variance is zero.

This parabolic shape of $\mu(1-\mu)$ appears repeatedly in ML — for example, in the entropy of a binary variable and in the gradient of logistic regression.

Binomial Distribution

If we repeat a Bernoulli experiment n times independently and count the number of successes, we get the **Binomial distribution**. It answers the question: "Out of n trials, how many times does the event occur?"

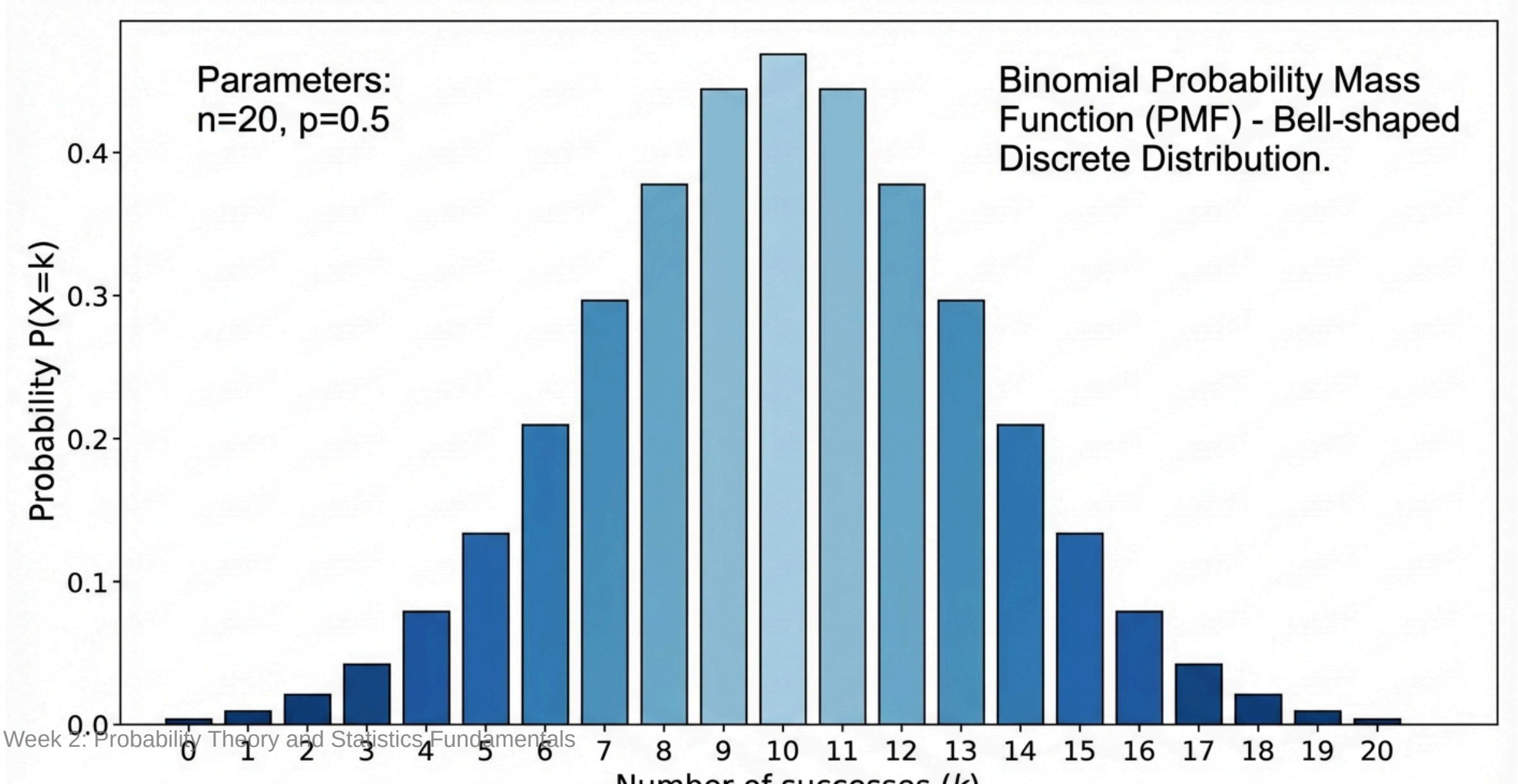
$$P(X = k) = \binom{n}{k} \mu^k (1 - \mu)^{n-k}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient — the number of ways to choose which k trials are successes.

Property	Value
Parameters	n (trials), μ (success probability)
Expected value	$E[X] = n\mu$
Variance	$\mathrm{Var}(X) = n\mu(1 - \mu)$

Note that Bernoulli is simply the special case where $n = 1$.

Binomial Distribution: Shape



Binomial: Quality Control Example

A production line has a 5% defect rate ($\mu = 0.05$). An inspector selects 20 products ($n = 20$).

Question 1: What is the probability of exactly 2 defective products?

$$P(X = 2) = \binom{20}{2}(0.05)^2(0.95)^{18} = 190 \times 0.0025 \times 0.397 \approx 0.189$$

Question 2: What is the probability of at most 1 defective product?

$$P(X \leq 1) = P(X = 0) + P(X = 1) \approx 0.736$$

So there is about a 74% chance that the inspector finds zero or one defects — even though 5% of all items are defective. This is because with only 20 samples, the expected number of defects is just $n\mu = 1$.

Multinomial Distribution

The Multinomial generalizes the Binomial to $K > 2$ categories. Each trial has K possible outcomes with probabilities $\mu_1, \ldots, \mu_K$, and we count how many times each outcome occurs in N trials.

$$P(x_1, \ldots, x_K | \boldsymbol{\mu}, N) = \frac{N!}{x_1! \cdots x_K!} \prod_{k=1}^{K} \mu_k^{x_k}$$

where $\sum_k x_k = N$ and $\sum_k \mu_k = 1$.

Property	Value
Expected count	$E[x_k] = N\mu_k$
Variance	$\mathrm{Var}(x_k) = N\mu_k(1 - \mu_k)$
Covariance	$\mathrm{Cov}(x_j, x_k) = -N\mu_j\mu_k$ for $j \neq k$

The negative covariance is intuitive: if more trials land in category j, fewer are left for category k. This distribution appears in text modeling (word counts), multi-class classification, and topic models.

Dirichlet Distribution (PRML 2.2)

Just as the Beta distribution is the conjugate prior for the Bernoulli/Binomial, the **Dirichlet** is the conjugate prior for the Multinomial. It is a distribution over probability vectors $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ that live on the simplex ($\sum_k \mu_k = 1$).

$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

where $\alpha_0 = \sum_k \alpha_k$. The hyperparameters α_k act as "pseudo-counts" — they encode our prior belief about how many times each category has been observed.

Key property (conjugacy): After observing counts $\mathbf{m} = (m_1, \ldots, m_K)$, the posterior is also Dirichlet:

$$p(\boldsymbol{\mu}|D, \boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha} + \mathbf{m})$$

The Dirichlet appears in Latent Dirichlet Allocation (LDA), Bayesian Naive Bayes, and many other generative models.

Categorical Distribution

The Categorical distribution is the $N = 1$ special case of the Multinomial — a single draw from K categories. It is the distribution behind every multi-class classifier's output.

We represent the outcome using **one-hot encoding**: $\mathbf{x} = (0, 0, 1, 0, \ldots, 0)^T$, where exactly one element is 1 and the rest are 0.

$$P(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$$

This notation is elegant — only the term where $x_k = 1$ contributes, giving $P(\mathbf{x}) = \mu_k$.

Where it appears in ML:

- **Softmax output** of a neural network classifier
- **Word distributions** in NLP (each word is a category)
- **State transitions** in Hidden Markov Models

Poisson Distribution

The Poisson distribution models the **count of events** occurring in a fixed interval of time or space, when those events happen independently at a constant average rate λ.

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, 2, \ldots$$

A remarkable property: $E[X] = \lambda$ and $\mathrm{Var}(X) = \lambda$ — the mean and variance are equal. When they are not equal in your data, the Poisson may not be a good fit.

Examples:

- Number of customers arriving per hour ($\lambda = 12$)
- Number of typos on a page ($\lambda = 2$)
- Website visits per minute ($\lambda = 50$)

The Poisson can be derived as the limit of the Binomial when $n \to \infty$ and $\mu \to 0$ with $n\mu = \lambda$ held constant — hence its association with "rare events."

Uniform Distribution

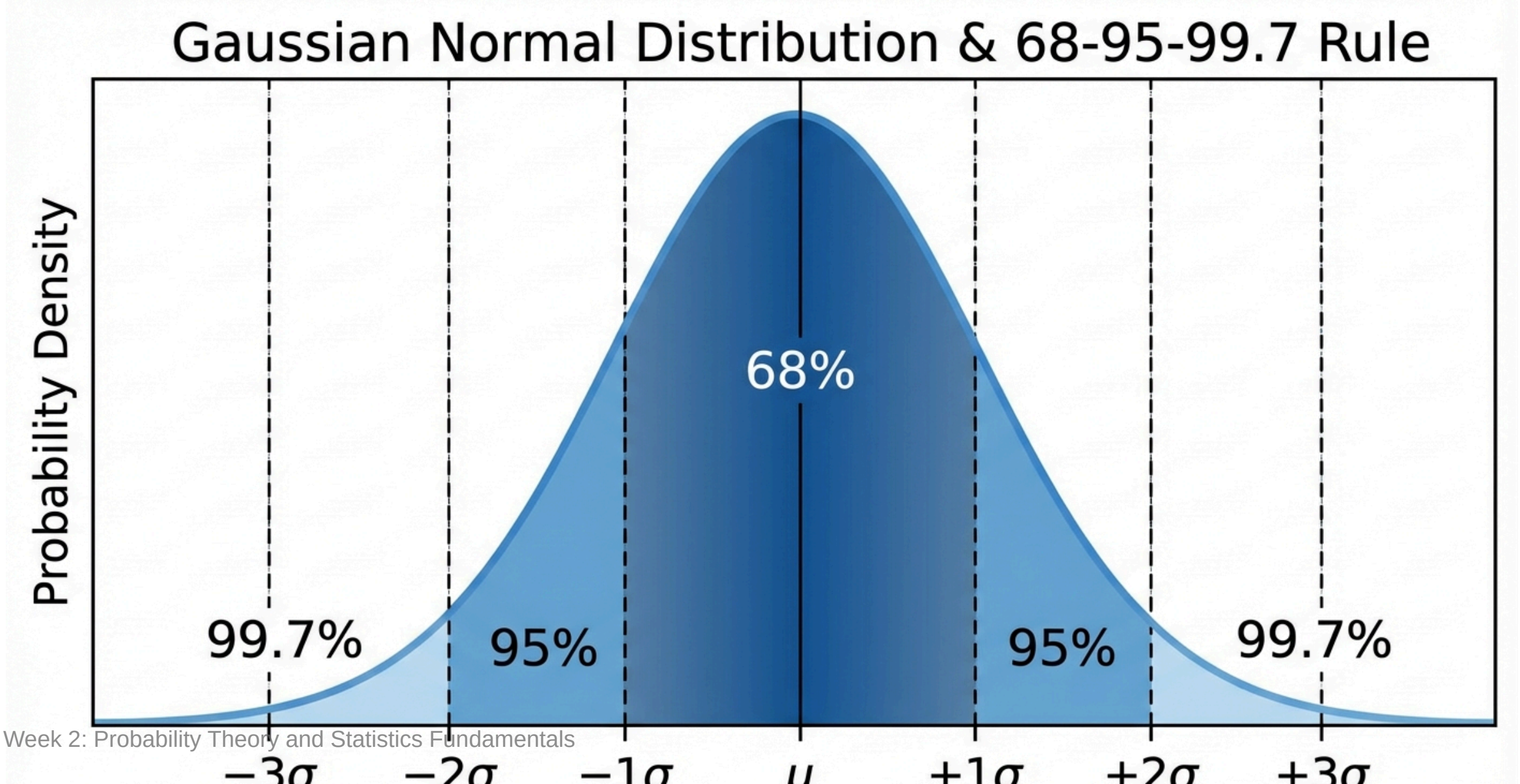
The Uniform distribution represents **complete ignorance** — every value in the interval $[a, b]$ is equally likely.

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Property	Value
Expected value	$E[X] = (a+b)/2$
Variance	$\text{Var}(X) = (b-a)^2/12$

In machine learning, the Uniform is often used as a **non-informative prior** — it says "I have no preference for any value" — and for random initialization of model parameters before training begins. It also plays a central role in Monte Carlo sampling methods.

Gaussian (Normal) Distribution



Gaussian: Definition

The Gaussian is the most important continuous distribution in machine learning. Its familiar bell-shaped curve appears throughout statistics, physics, and engineering.

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

It is fully characterized by just two parameters:

- μ — the **mean** (center of the bell)
- σ^2 — the **variance** (width of the bell)

The distribution is symmetric about μ and unimodal — there is exactly one peak.

Why is the Gaussian So Important?

Three reasons make the Gaussian uniquely central:

1. Central Limit Theorem (CLT)

The average of n independent random variables — regardless of their original distribution — converges to a Gaussian as n grows:

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{d} \mathcal{N}(\mu, \sigma^2/n)$$

This explains why so many real-world quantities (heights, measurement errors, stock returns) look approximately Gaussian.

2. Maximum Entropy

Among all distributions with a given mean and variance, the Gaussian has the **highest entropy** — it makes the fewest assumptions beyond those two statistics.

3. Closure Under Linear Operations

Gaussians are closed under summation, conditioning, and marginalization. This makes them analytically tractable: if you start with Gaussians, you stay with Gaussians.

Standard Normal and the 68-95-99.7 Rule

Any Gaussian can be converted to the **standard normal** $\mathcal{N}(0,1)$ via the Z-score transformation:

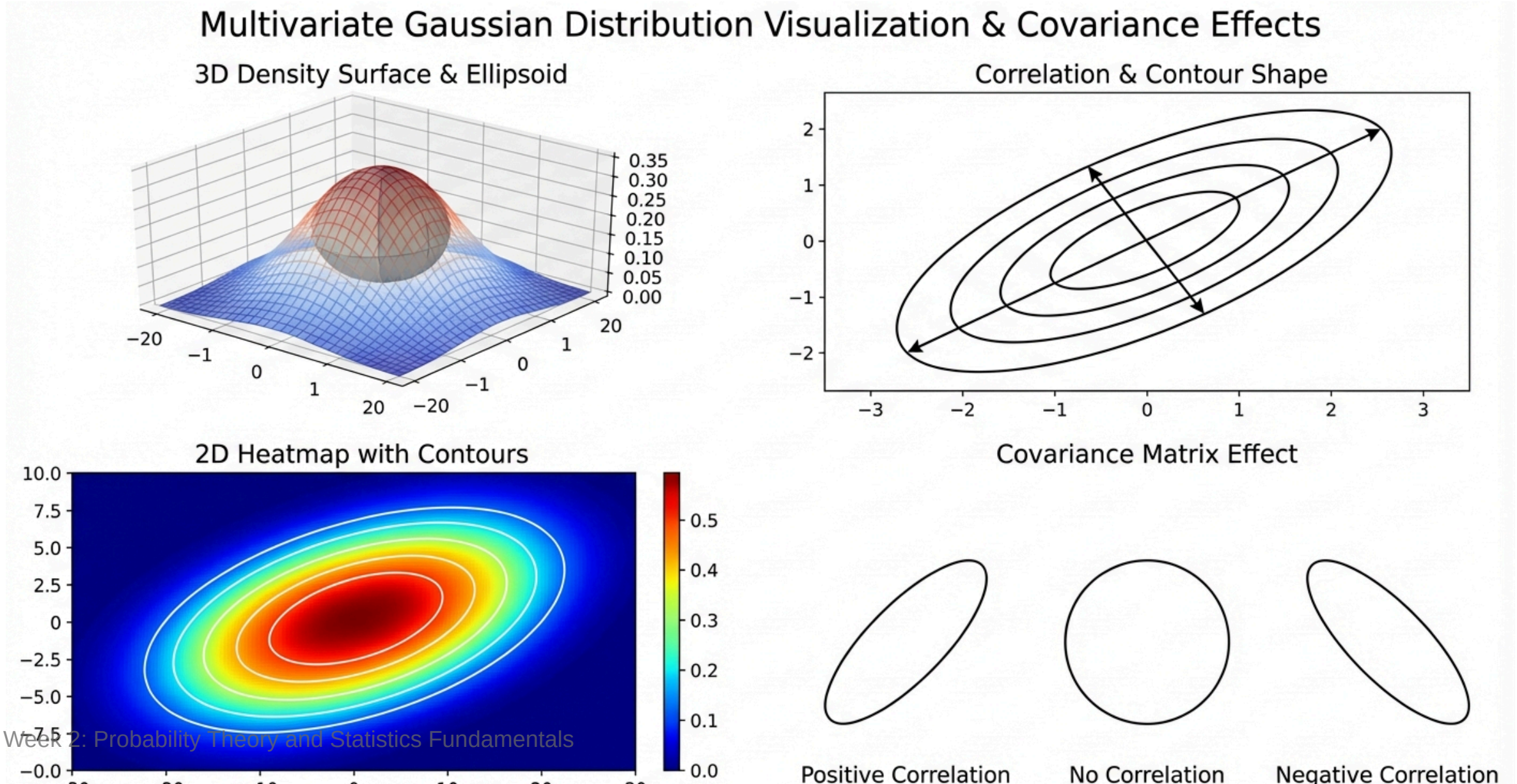
$$Z = \frac{X - \mu}{\sigma} \sim \mathcal{N}(0,1)$$

This gives us a universal reference. The **68-95-99.7 rule** tells us how probability concentrates around the mean:

Interval	Probability
$\mu \pm 1\sigma$	68.27%
$\mu \pm 2\sigma$	95.45%
$\mu \pm 3\sigma$	99.73%

Almost all the mass lies within three standard deviations of the mean. Observations beyond 3σ are rare enough to be considered potential **outliers** — a useful heuristic in data cleaning and anomaly detection.

Multivariate Gaussian: Contour Shapes



Multivariate Gaussian

When we move from a single variable to D dimensions, the Gaussian generalizes naturally. The bell curve becomes an ellipsoidal "cloud" in $\mathbb{R}^D$, and the scalar variance becomes a full covariance matrix that encodes both the spread and the orientation of that cloud.

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

- $\boldsymbol{\mu}$ — the $D \times 1$ **mean vector**, the center of the ellipsoid
- $\boldsymbol{\Sigma}$ — the $D \times D$ **covariance matrix**, which controls both the shape and orientation of the contours

The exponent $(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$ is the **Mahalanobis distance** — it measures how far $\mathbf{x}$ is from $\boldsymbol{\mu}$, scaled by the covariance structure. When $\boldsymbol{\Sigma}$ is diagonal, the contours are axis-aligned ellipses; when it has off-diagonal terms, the contours rotate to reflect correlations between dimensions.

Types of Covariance Matrices

The covariance matrix $\boldsymbol{\Sigma}$ controls the shape of the Gaussian's contour lines. Choosing the right structure is a modeling decision that balances expressiveness against the number of parameters you need to estimate.

Type	Form	Contour Shape	Parameters
Spherical	$\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$	Circle / sphere	1
Diagonal	$\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_D^2)$	Axis-aligned ellipse	D
Full	General positive-definite matrix	Rotated ellipse	$D(D+1)/2$

The spherical case assumes all dimensions have equal variance and no correlation — the simplest model but often too restrictive. The diagonal case allows different variances per dimension but still assumes independence. The full covariance captures arbitrary correlations but requires estimating $D(D+1)/2$ parameters, which can be problematic in high dimensions where data is scarce.

> **Trade-off in practice:** Many ML algorithms (e.g., Gaussian Mixture Models) let you choose the covariance type. Starting with diagonal and upgrading to full only when you have enough data is a common strategy.

Expected Value (Expectation)

The expected value is the **weighted average** of all possible values of a random variable, where the weights are the probabilities. Intuitively, it is the "center of mass" of the distribution — if you placed weights proportional to probability at each value, the expected value is the balance point.

Discrete

$$E[X] = \sum_{x} x \cdot p(x)$$

Continuous

$$E[X] = \int_{-\infty}^{\infty} x \cdot f(x)\, dx$$

An important subtlety: the expected value does not have to be a value the random variable can actually take. For the sum of two dice, $E[X] = 7$ — which happens to be the most likely outcome — but for a single die, $E[X] = 3.5$, which is not even a possible outcome. The expectation is a mathematical summary, not a prediction of any single trial.

Expected Value: Properties

The most powerful property of expectation is **linearity** — it holds unconditionally, regardless of whether the variables are independent or dependent. This makes expectations far easier to work with than variances.

$$E[aX + b] = aE[X] + b \qquad\qquad E[X + Y] = E[X] + E[Y]$$

The second equation is remarkable: the expected value of a sum is always the sum of the expected values, even for correlated variables. This property is used constantly in ML — for example, the expected loss of an ensemble is the average of the individual expected losses.

Expectation of a Function (LOTUS)

The **Law of the Unconscious Statistician** says that to compute $E[g(X)]$, you do not need the distribution of $g(X)$ itself — only the distribution of X:

$$E[g(X)] = \sum_x g(x) \cdot p(x) \qquad \text{or} \qquad E[g(X)] = \int g(x) \cdot f(x)\, dx$$

This is immensely useful because deriving the distribution of $g(X)$ can be difficult, but computing the weighted average directly is often straightforward.

Variance

While the expected value tells us where a distribution is centered, the **variance** tells us how spread out it is — how far values typically deviate from the mean. It is defined as the expected squared deviation from the mean:

$$\mathrm{Var}(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

The second form (computational formula) is usually easier to work with. Both yield the same result, but the first makes the interpretation transparent: we are averaging the squared distances from the mean.

Because variance is measured in squared units (e.g., meters²), we often use the **standard deviation** $\sigma = \sqrt{\mathrm{Var}(X)}$ instead, which returns us to the original units and is easier to interpret. In the context of the Gaussian distribution, σ directly controls the width of the bell curve.

Variance: Properties

Unlike expectation, variance is **not linear** — scaling by a constant a scales the variance by a^2, and adding a constant has no effect at all (shifting a distribution does not change its spread):

$$\mathrm{Var}(aX + b) = a^2 \,\mathrm{Var}(X)$$

For the variance of a sum, we must distinguish two cases. If X and Y are **independent**, their variances simply add:

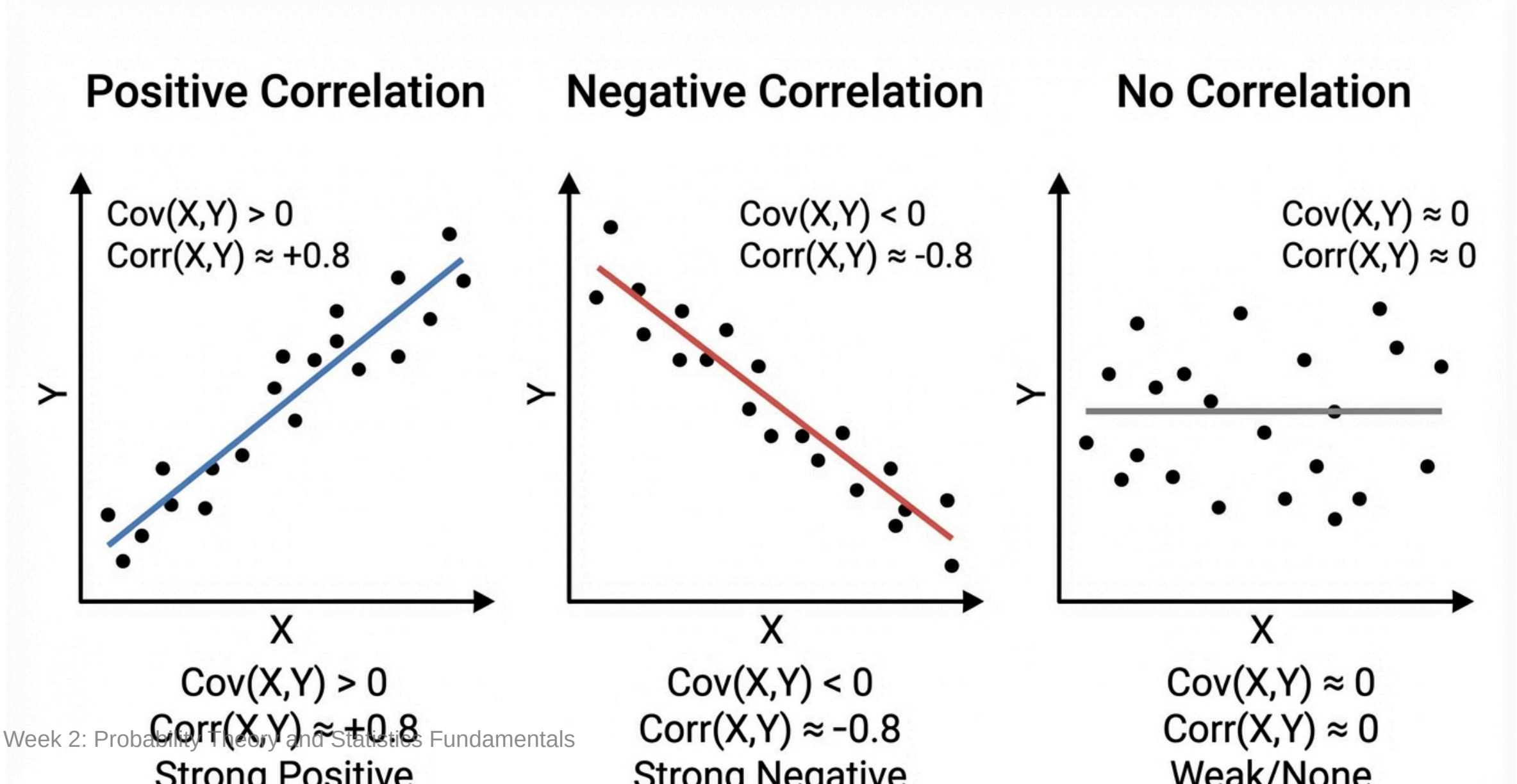
$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) \qquad (\text{independent only})$$

In the general case, the covariance term appears:

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X, Y)$$

This is why independence matters so much in statistics — when variables are correlated, the variance of their sum can be much larger (positive correlation) or smaller (negative correlation) than you would expect from summing their individual variances.

Covariance and Correlation



Covariance

Covariance measures the **linear co-movement** of two random variables — whether they tend to increase together, decrease together, or move independently.

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]$$

The sign of the covariance tells you the direction of the relationship:

Sign	Meaning	Example
$\text{Cov} > 0$	X and Y increase together	Height and weight
$\text{Cov} < 0$	One increases, the other decreases	Price and demand
$\text{Cov} = 0$	No **linear** relationship	(but could still be dependent!)

Two useful identities: $\text{Cov}(X, X) = \text{Var}(X)$ (covariance with yourself is variance), and covariance is symmetric: $\text{Cov}(X, Y) = \text{Cov}(Y, X)$.

Correlation

The problem with covariance is that its magnitude depends on the units of X and Y — making it hard to compare across different pairs of variables. **Correlation** solves this by normalizing:

$$\rho(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \, \sigma_Y}$$

This normalization guarantees $-1 \leq \rho \leq 1$, where $|\rho| = 1$ means a perfect linear relationship and $\rho = 0$ means no linear relationship. Correlation is dimensionless, so $\rho = 0.8$ between height and weight is directly comparable to $\rho = 0.8$ between temperature and ice cream sales.

> **Critical warning:** $\rho = 0$ does **not** imply independence. Independence always implies zero correlation, but the converse is false — variables can be strongly dependent yet have zero correlation, as the next slide demonstrates.

Correlation ≠ Independence

This is one of the most common mistakes in statistics, so let us see a concrete counterexample.

Let $X \sim \mathcal{N}(0, 1)$ and define $Y = X^2$. The variable Y is completely determined by X — there is no randomness left once you know X. They are maximally dependent. Yet:

$$\text{Cov}(X, Y) = E[X \cdot X^2] - E[X] \cdot E[X^2] = E[X^3] - 0 = 0$$

The covariance is zero because X^3 is an odd function of a symmetric distribution, so its expectation vanishes. If you scatter-plot X vs Y, you see a clear parabola — an obvious relationship that correlation completely misses.

> **The lesson:** Correlation captures only **linear** relationships. For nonlinear dependencies, you need other tools — mutual information, distance correlation, or simply plotting the data.

Covariance Matrix

When we have D random variables, all pairwise covariances are collected into a single $D \times D$ matrix — the **covariance matrix**. This is the multivariate generalization of variance.

$$\boldsymbol{\Sigma} = E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T] = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

The diagonal contains the variances (how much each variable varies on its own), and the off-diagonal entries contain the covariances (how pairs of variables co-vary). The matrix is always **symmetric** ($\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T$) and **positive semi-definite** ($\mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} \geq 0$ for all $\mathbf{v}$).

The covariance matrix appears everywhere in ML: it parameterizes the multivariate Gaussian, it defines the geometry of PCA, and its inverse (the precision matrix) encodes conditional independence structure in Gaussian graphical models.

Practice: Variance and Covariance

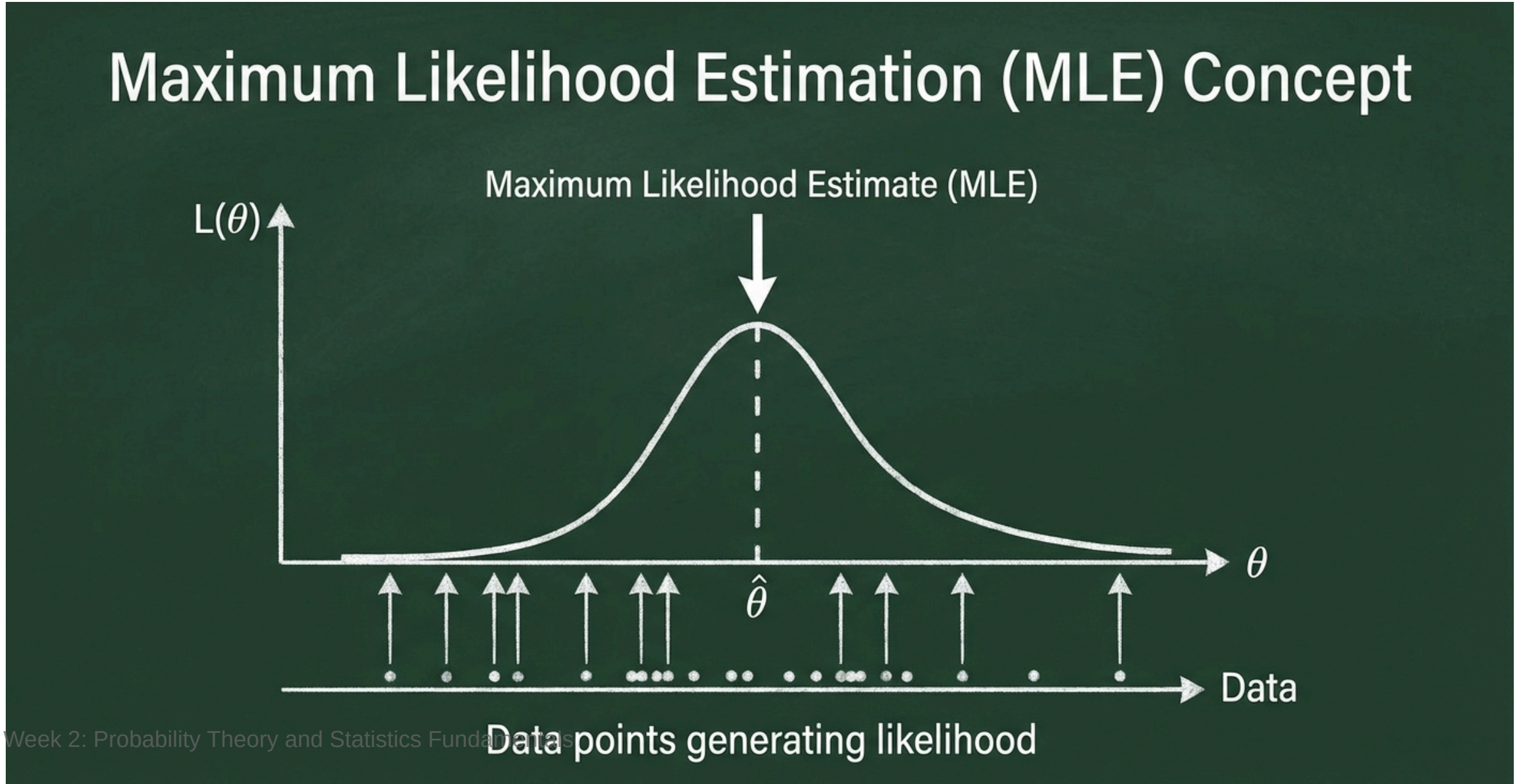
X and Y are random variables with: $E[X] = 3$, $\text{Var}(X) = 4$, $E[Y] = 5$, $\text{Var}(Y) = 9$, $\text{Cov}(X, Y) = 2$.

Questions:

1. $E[2X + 3Y] = ?$
2. $\text{Var}(2X + 3Y) = ?$
3. $\rho(X, Y) = ?$

Hint: Use linearity of expectation for (1), the general variance formula $\text{Var}(aX + bY) = a^2\text{Var}(X) + b^2\text{Var}(Y) + 2ab\,\text{Cov}(X, Y)$ *for (2), and* $\rho = \text{Cov}/(\sigma_X \sigma_Y)$ *for (3).*

Maximum Likelihood Estimation



MLE: Formulation

We now turn to the central question of statistics: given observed data, how do we estimate the parameters of our model? The most widely used approach is **Maximum Likelihood Estimation** — find the parameter values that make the observed data as probable as possible.

Given i.i.d. data $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ and a model $p(x|\theta)$, the **likelihood function** treats the data as fixed and views probability as a function of the parameter:

$$L(\theta) = p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

The MLE is the value of θ that maximizes this product:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} L(\theta)$$

> The key conceptual shift: in probability, θ is fixed and we reason about x; in likelihood, x is fixed (observed) and we reason about θ.

Log-Likelihood

In practice, we almost never maximize the likelihood directly — instead, we maximize its logarithm. Since $\log$ is a strictly increasing function, the maximizer does not change, but the math becomes much cleaner.

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$$

Three reasons make the log-likelihood preferable:

1. **Products become sums** — derivatives of sums are far easier to compute than derivatives of products
2. **Numerical stability** — multiplying N small probabilities quickly underflows to zero; summing log-probabilities avoids this entirely
3. **Convexity** — for many common models (exponential family), the log-likelihood is concave, which guarantees a unique global maximum

The MLE is equivalently:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \ell(\theta)$$

In deep learning, we minimize the **negative** log-likelihood as a loss function — this is exactly the same optimization, just flipped.

MLE Example: Bernoulli

Let us work through the simplest non-trivial example. We have N independent coin flips $x_1, \ldots, x_N \in \{0, 1\}$ and want to estimate the probability of heads μ under the Bernoulli model $p(x|\mu) = \mu^x (1-\mu)^{1-x}$.

Log-likelihood:

$$\ell(\mu) = \sum_{i=1}^{N} \left[x_i \log \mu + (1 - x_i) \log(1 - \mu) \right]$$

Setting the derivative to zero:

$$\frac{d\ell}{d\mu} = \frac{\sum x_i}{\mu} - \frac{N - \sum x_i}{1 - \mu} = 0$$

Result:

$$\hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^{N} x_i = \bar{x}$$

The MLE is simply the **sample proportion** — the fraction of ones in the data. This is intuitive: if you see 7 heads in 10 flips, the maximum likelihood estimate is $\hat{\mu} = 0.7$.

MLE: The Small-Sample Problem

The Bernoulli MLE reveals a fundamental weakness. Suppose you flip a coin 3 times and get 3 heads. The MLE says $\hat{\mu} = 3/3 = 1.0$ — it is absolutely certain that the coin always lands heads. After just three observations, MLE has committed to the most extreme possible estimate.

This is clearly unreasonable. No sensible person would conclude from 3 heads that tails is impossible. The problem is that MLE has no mechanism to express uncertainty or incorporate prior knowledge — it simply picks the parameter that maximizes the likelihood of what it has seen, no matter how little data there is.

> **This is the core motivation for Bayesian methods:** by introducing a prior distribution over μ, we can temper these extreme estimates. The prior "pulls back" the estimate toward more reasonable values, especially when data is scarce. As we collect more data, the prior's influence fades and the Bayesian estimate converges to the MLE.

MLE Example: Gaussian

Now let us apply MLE to the Gaussian — the workhorse distribution of ML. Given N observations $x_1, \ldots, x_N$ from $\mathcal{N}(x|\mu, \sigma^2)$, the log-likelihood is:

$$\ell(\mu, \sigma^2) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2$$

Taking derivatives with respect to μ and σ^2 and setting them to zero yields the familiar formulas:

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i = \bar{x} \qquad\qquad \hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

The MLE for the mean is just the sample average, and the MLE for the variance is the average squared deviation from that mean. These are the most natural estimates — but the variance estimate has a subtle problem, as we will see next.

MLE: Gaussian Variance Bias

The MLE variance estimate has a systematic flaw — it **underestimates** the true variance on average (PRML Section 1.2.4, Figure 1.15):

$$E[\hat{\sigma}^2_{MLE}] = \frac{N-1}{N}\sigma^2 < \sigma^2$$

The intuition is that we are measuring deviations from the sample mean $\bar{x}$ rather than the true mean μ. Since $\bar{x}$ is computed from the same data, it is always closer to the data points than μ would be — this "uses up" one degree of freedom and systematically shrinks our variance estimate.

The **unbiased** correction divides by $N-1$ instead of N:

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

This is known as **Bessel's correction**. For large N, the difference between N and $N-1$ is negligible, but for small samples it matters significantly. This bias is a concrete example of the broader problem with MLE — it is a purely data-driven estimate with no mechanism to account for estimation uncertainty.

Practice: MLE — Bernoulli

7 out of 10 emails are classified as spam. Estimate the probability of spam (μ) using the Bernoulli model.

Questions:

1. What is the MLE estimate?
2. Write the log-likelihood function $\ell(\mu)$.
3. Would the MLE change if there were 100 emails with 70 spam? Why or why not?

Practice: MLE — Gaussian

5 measurements: $\{2, 4, 6, 8, 10\}$

Questions:

1. MLE estimate for μ?
2. MLE estimate for σ^2?
3. Unbiased estimate for σ^2 (using Bessel's correction)?

Properties of MLE

Despite the bias issue, MLE has excellent theoretical properties that make it the default choice when data is plentiful.

Advantages

- **Consistency** — converges to the true value as $N \to \infty$
- **Asymptotic efficiency** — achieves the lowest possible variance among all unbiased estimators (Cramér-Rao bound)
- **Asymptotic normality** — the sampling distribution of $\hat{\theta}_{MLE}$ approaches a Gaussian, enabling confidence intervals
- **Invariance** — the MLE of $g(\theta)$ is $g(\hat{\theta}_{MLE})$

Disadvantages

- **Overfitting** — overconfident with small samples (as we saw with Bernoulli)
- **No prior information** — cannot incorporate domain knowledge
- **Point estimate only** — gives no measure of uncertainty about θ

> The asymptotic properties are powerful but require "enough" data. In the small-data regime — common in medicine, robotics, and scientific experiments — Bayesian methods often outperform MLE precisely because they can leverage prior knowledge.

Bayesian Approach

The Bayesian approach represents a fundamentally different philosophy of inference. Instead of treating the parameter θ as a fixed unknown number and producing a single point estimate, the Bayesian treats θ as a **random variable** and computes an entire **probability distribution** over it.

	Frequentist (MLE)	**Bayesian**
Parameter	Fixed, unknown constant	Random variable with a distribution
Data	Random (imagined repeated experiments)	Fixed (the data we actually observed)
Output	Point estimate $\hat{\theta}$	Full posterior distribution $p(\theta
Prior knowledge	Not incorporated	Encoded via the prior $p(\theta)$

The Bayesian update rule is simply Bayes' theorem applied to parameters:

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

This formula says: start with what you believed before seeing data (prior), multiply by how well each parameter value explains the observed data (likelihood), and normalize to get your updated belief (posterior).

Bayes' Theorem for Parameters

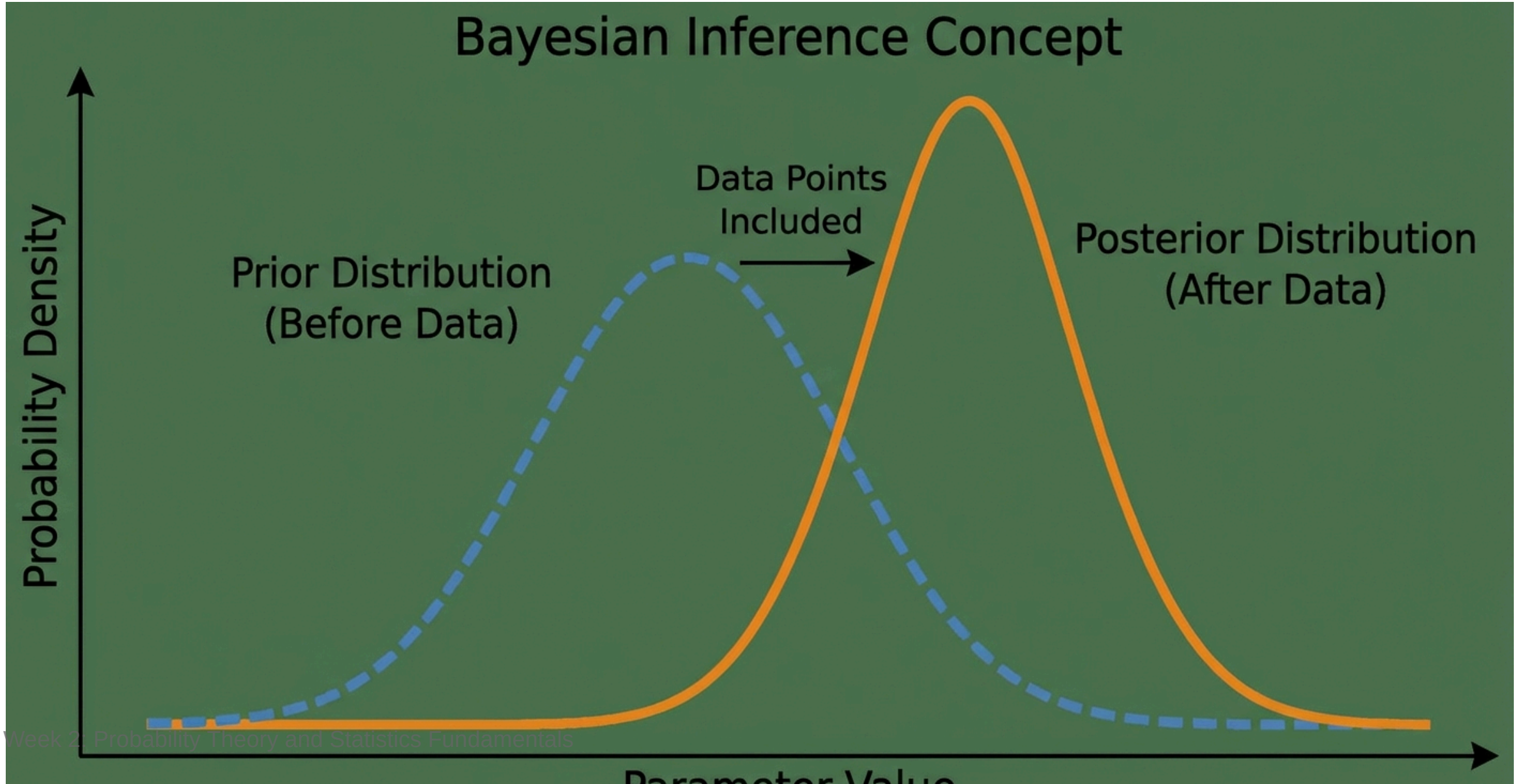
Applying Bayes' theorem with θ as the unknown and $\mathcal{D}$ as the observed evidence:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$$

Term	Name	Interpretation
$p(\theta)$	**Prior**	Our belief about θ before seeing any data
$p(\mathcal{D}	\theta)$	**Likelihood**
$p(\theta	\mathcal{D})$	**Posterior**
$p(\mathcal{D})$	**Evidence**	Normalization constant: $p(\mathcal{D}) = \int p(\mathcal{D}

The evidence $p(\mathcal{D})$ is the same for all values of θ, so for the purpose of finding the most probable θ we can ignore it. However, it plays a crucial role in **model comparison** — comparing which model (not which parameter) best explains the data.

Prior to Posterior



Prior Distribution

Choosing the prior is both the most powerful and most controversial aspect of Bayesian inference. The prior encodes what we know (or assume) about θ before seeing any data.

Informative priors incorporate genuine domain knowledge — for example, a physicist might know that a physical constant is positive, or a clinician might have data from previous studies. These priors help most when data is scarce.

Non-informative (vague) priors try to "let the data speak" by being as broad as possible — a Uniform or a very wide Gaussian. The goal is to add minimal bias, though truly non-informative priors can be surprisingly tricky to define rigorously (see Jeffreys priors).

Conjugate priors are chosen for mathematical convenience: when paired with a specific likelihood, the posterior stays in the same distributional family. This makes the posterior available in closed form, avoiding expensive numerical integration.

Conjugate Priors

A prior is **conjugate** to a likelihood if their product (the unnormalized posterior) belongs to the same distributional family as the prior. This is a purely mathematical convenience — it lets us write the posterior in closed form without numerical integration.

Likelihood	Conjugate Prior	Posterior
Bernoulli / Binomial	Beta	Beta
Multinomial	Dirichlet	Dirichlet
Gaussian (μ unknown)	Gaussian	Gaussian
Gaussian (σ^2 unknown)	Inverse-Gamma	Inverse-Gamma
Poisson	Gamma	Gamma

The pattern is always the same: observe data, add the sufficient statistics to the prior's hyperparameters, and read off the posterior. No integrals, no sampling, no approximation. This is why conjugate priors dominated Bayesian statistics before modern computational methods (MCMC, variational inference) made arbitrary priors feasible.

> **In practice:** conjugate priors are the starting point for many Bayesian models. When they are too restrictive, we switch to MCMC or variational methods — but we still often initialize with conjugate families.

Beta Distribution (PRML 2.1)

The Beta distribution is a distribution over probabilities — it lives on the interval $[0, 1]$ and is the conjugate prior for the Bernoulli and Binomial likelihoods.

$$\text{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \qquad \text{(PRML Eq. 2.13)}$$

The hyperparameters a and b control the shape. Intuitively, they act as "pseudo-counts": $a-1$ prior successes and $b-1$ prior failures. The expected value is $E[\mu] = a/(a+b)$.

Conjugacy in action: when we multiply a Beta prior by a Bernoulli likelihood with m successes and $l = N - m$ failures, the posterior is also Beta:

$$p(\mu|m,l,a,b) \propto \mu^{m+a-1}(1-\mu)^{l+b-1} = \text{Beta}(\mu \,|\, a+m,\ b+l) \qquad \text{(PRML Eq. 2.17)}$$

The update rule is remarkably simple: just add the observed counts to the prior hyperparameters.

Beta Distribution: Shapes

The Beta family is remarkably flexible — by varying a and b, it can express a wide range of prior beliefs about a probability parameter.

(a, b)	**Shape**	**Interpretation**
$a = b = 1$	Uniform (flat)	Complete ignorance — all values of μ equally likely
$a = b = 2$	Symmetric bell	Mild preference for $\mu \approx 0.5$
$a > b$	Skewed right	Prior belief that μ is large (more successes expected)
$a < b$	Skewed left	Prior belief that μ is small (more failures expected)
$a = b \gg 1$	Narrow peak at 0.5	Strong conviction that $\mu \approx 0.5$

The "effective sample size" of the prior is $a + b - 2$. A prior of $\text{Beta}(10, 10)$ carries the weight of 18 observations — so it takes at least that much real data before the posterior is dominated by the likelihood rather than the prior. This gives you direct control over how quickly you want the data to override your prior beliefs.

Bayesian Inference: Bernoulli Example

Let us trace the full Bayesian computation step by step. We start with a Beta prior $\mu \sim \text{Beta}(a, b)$ and observe m heads in N coin flips.

Step 1 — Write down the unnormalized posterior:

$$p(\mu|m, N) \propto \underbrace{\mu^m(1-\mu)^{N-m}}_{\text{Binomial likelihood}} \cdot \underbrace{\mu^{a-1}(1-\mu)^{b-1}}_{\text{Beta prior}}$$

Step 2 — Combine the exponents:

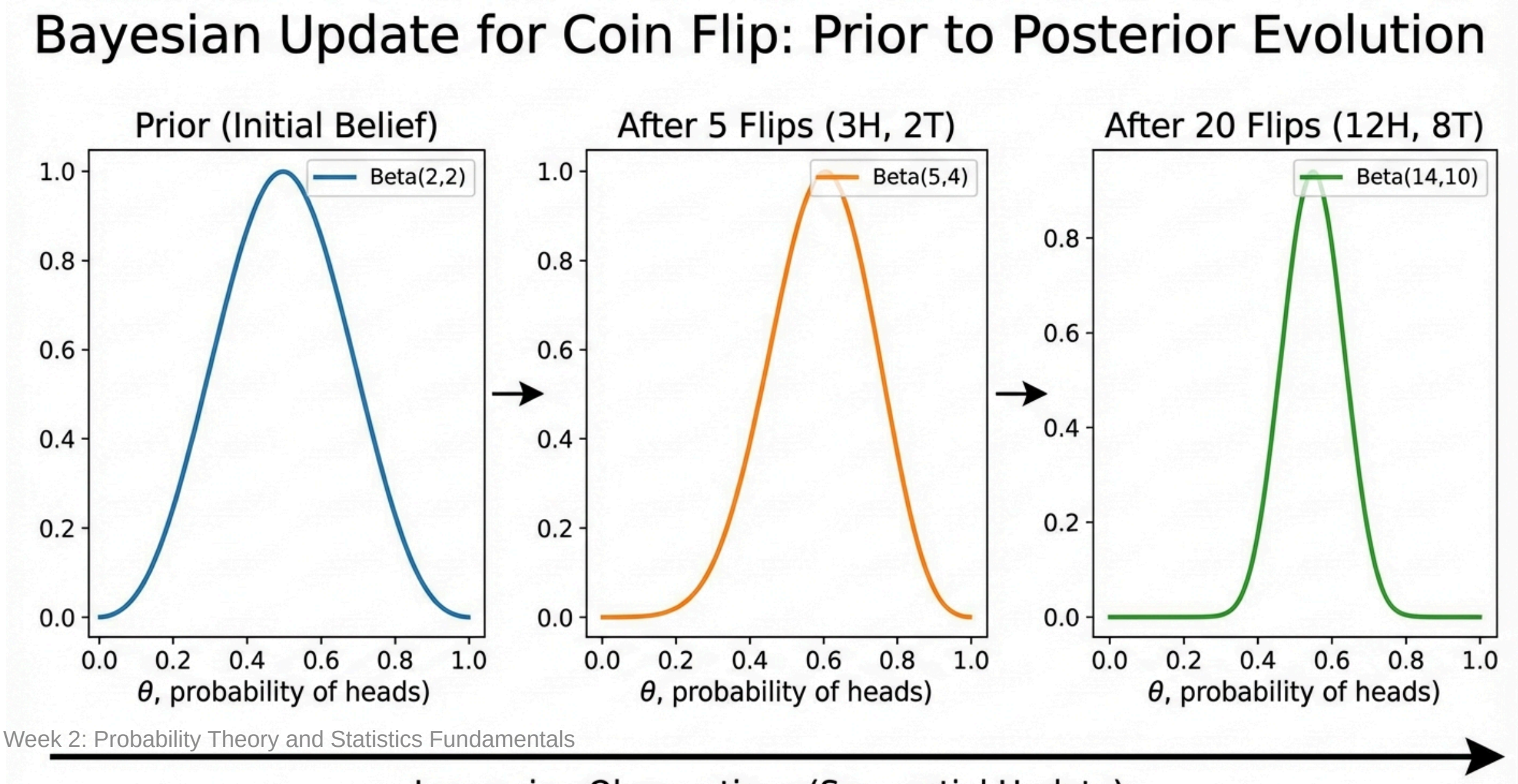
$$\propto \mu^{m+a-1}(1-\mu)^{N-m+b-1}$$

Step 3 — Recognize the kernel of a Beta distribution:

$$p(\mu|m, N) = \text{Beta}(\mu \,|\, a+m,\ b+N-m)$$

The posterior mean is $(a+m)/(a+b+N)$ — a weighted average of the prior mean $a/(a+b)$ and the MLE m/N, with the weights determined by the effective sample sizes of prior and data respectively.

Bayesian Update: Bernoulli



Bayesian Update: Numerical Example

Prior → Posterior Evolution

Start: Beta(2, 2) — uncertain

10 flips, 7 heads:

- Posterior: Beta(9, 5)
- E[μ] = 9/14 ≈ 0.64

100 flips, 70 heads:

- Posterior: Beta(72, 32)
- E[μ] ≈ 0.69

Observation:

- More data → Narrower posterior
- Prior's effect diminishes
- Approaches MLE

Maximum A Posteriori (MAP)

Computing the full posterior is ideal but often expensive. A simpler alternative is **MAP estimation** — find the single most probable parameter value under the posterior:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \; p(\theta|\mathcal{D}) = \arg\max_{\theta} \left[\log p(\mathcal{D}|\theta) + \log p(\theta)\right]$$

The evidence $p(\mathcal{D})$ drops out because it does not depend on θ. Comparing with MLE:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \log p(\mathcal{D}|\theta) \qquad \hat{\theta}_{MAP} = \arg\max_{\theta} \left[\log p(\mathcal{D}|\theta) + \log p(\theta)\right]$$

The only difference is the additional $\log p(\theta)$ term — the prior acts as a **penalty** that discourages extreme parameter values. As $N \to \infty$, the likelihood dominates and MAP converges to MLE.

MAP: Bernoulli Example

With a $\text{Beta}(a, b)$ prior and m successes in N trials, the MAP estimate has a clean closed form — it is the mode of the Beta posterior:

$$\hat{\mu}_{MAP} = \frac{m + a - 1}{N + a + b - 2}$$

Let us revisit the problematic small-sample case (3 flips, 3 heads) and compare:

Estimate	Formula	3 flips, 3 heads	100 flips, 70 heads
MLE	m/N	**1.0**	0.70
MAP ($a = b = 2$)	$(m+1)/(N+2)$	**0.8**	0.70

With small data, the prior pulls the MAP away from the extreme MLE estimate. With large data, both converge to the same value — the data overwhelms the prior.

> MAP is effectively MLE with a **regularization penalty** from the prior — a connection we will make precise when we discuss L1/L2 regularization.

Practice: MAP and Prior Effect

Same spam problem (10 emails, 7 spam). Prior: $\text{Beta}(2, 8)$ — we believe spam is rare.

Questions:

1. What is the MAP estimate? (Use the formula $\hat{\mu}_{MAP} = (m + a - 1)/(N + a + b - 2)$)
2. How does the MAP change if we use $\text{Beta}(5, 5)$ as prior instead?
3. What does the MAP converge to as $N \to \infty$?

MAP vs Full Bayesian

MAP is convenient, but it is still a **point estimate** — it discards all information about the shape of the posterior. Two posteriors can have the same mode but very different spreads, and MAP cannot distinguish between them.

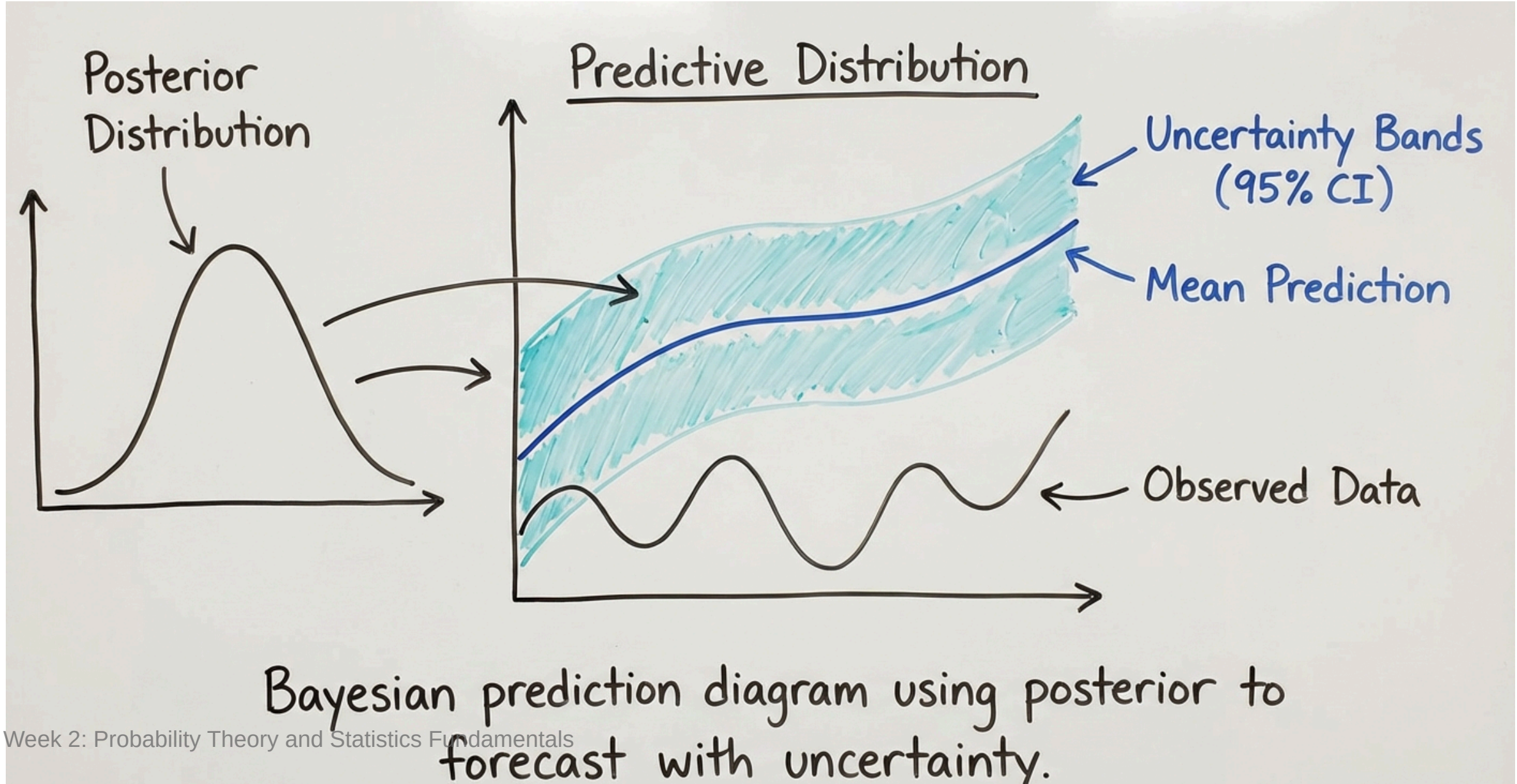
The full Bayesian approach uses the **entire posterior distribution** for both prediction and decision-making:

$$p(x_{new}|\mathcal{D}) = \int p(x_{new}|\theta)\, p(\theta|\mathcal{D})\, d\theta$$

This is the **posterior predictive distribution** — instead of plugging in a single $\hat{\theta}$, we average over all possible parameter values, weighted by how plausible each one is given the data. The result naturally accounts for parameter uncertainty: when we are unsure about θ, our predictions are appropriately more spread out.

The challenge is computational: the integral is often analytically intractable. This is where modern approximate methods — **MCMC** (Markov Chain Monte Carlo) and **Variational Inference** — come in, which we will study later in the course.

Predictive Distribution



Posterior Predictive: Derivation

The posterior predictive distribution answers the practical question: "Given what I have learned from the data, what will the next observation look like?"

$$p(x_{new}|\mathcal{D}) = \int p(x_{new}|\theta)\, p(\theta|\mathcal{D})\, d\theta$$

For the Bernoulli-Beta model, this integral has a beautiful closed form. With a $\text{Beta}(a, b)$ prior and m successes in N trials:

$$p(x_{new} = 1|\mathcal{D}) = E[\mu|\mathcal{D}] = \frac{a + m}{a + b + N}$$

This is the **posterior mean**, not the posterior mode (MAP). With a $\text{Beta}(1, 1)$ prior (uniform), this becomes $(1 + m)/(2 + N)$ — which is exactly **Laplace smoothing**, one of the oldest and most widely used techniques in NLP and text classification.

MLE vs MAP vs Bayesian: The Complete Picture

These three approaches form a spectrum from simplest to most complete. Choosing among them is a practical trade-off between computational cost and the quality of uncertainty estimates.

Property	MLE	MAP	Full Bayesian
Output	Point estimate	Point estimate	Full distribution
Uses prior	❌	✅	✅
Quantifies uncertainty	❌	❌	✅
Computational cost	Low	Low	High
Overfitting risk	High	Medium	Low
Small-data performance	Poor	Good	Best

In practice, most production ML systems use MLE (or equivalently, minimizing a loss function) because it scales to billions of parameters. MAP appears whenever you add regularization. Full Bayesian methods are reserved for settings where uncertainty quantification is critical — medical diagnosis, autonomous driving, active learning — and the model is small enough for the computation to be feasible.

Regularization and the Bayesian Connection

One of the most beautiful connections in ML: the regularization terms that practitioners add to prevent overfitting are not arbitrary — they correspond to specific **prior distributions** in the Bayesian framework.

$$\hat{\theta}_{MAP} = \arg\min_{\theta} \underbrace{[-\log p(\mathcal{D}|\theta)]}_{\text{Loss (data fit)}} + \underbrace{[-\log p(\theta)]}_{\text{Regularization (prior)}}$$

Prior Distribution	$-\log p(\theta)$ **becomes**	**Regularization Name**
Gaussian: $p(\theta) \propto e^{-\lambda\|\theta\|^2}$	$\lambda\|\theta\|^2$	**L2 (Ridge)**
Laplace: $p(\theta) \propto e^{-\lambda\|\theta\|_1}$	$\lambda\|\theta\|_1$	**L1 (Lasso)**

So when a deep learning practitioner adds "weight decay" ($\lambda\|\theta\|^2$) to the loss function, they are implicitly assuming a Gaussian prior over the weights. When using L1 regularization (which encourages sparsity), they are implicitly assuming a Laplace prior. The regularization strength λ corresponds to the precision (inverse variance) of the prior.

> This connection gives principled guidance: the choice of regularizer should reflect your genuine beliefs about the parameter structure.

Information Theory: Entropy

Information theory, developed by Claude Shannon in 1948, provides a mathematical framework for quantifying uncertainty and information. The central concept is **entropy** — a measure of how uncertain or "surprising" a random variable is on average.

$$H(X) = -\sum_{x} p(x) \log p(x) = E[-\log p(X)]$$

The entropy is maximized when all outcomes are equally likely (maximum uncertainty) and minimized when the outcome is deterministic (zero uncertainty).

Distribution	Entropy	Interpretation
Fair coin ($p = 0.5$)	1 bit	Maximum uncertainty for binary
Biased coin ($p = 0.9$)	0.47 bits	Mostly predictable
Certain ($p = 1$)	0 bits	No uncertainty at all

In ML, entropy appears in **decision trees** (information gain), **variational inference** (ELBO), and as a regularizer encouraging exploration in reinforcement learning.

KL Divergence

The **Kullback-Leibler divergence** measures how different one probability distribution is from another. Despite often being called a "distance," it is not symmetric — $KL(p||q) \neq KL(q||p)$ — so it is technically not a distance metric.

$$KL(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} = E_p \left[\log \frac{p(X)}{q(X)} \right]$$

Key properties:

- **Non-negative:** $KL(p||q) \geq 0$ (Gibbs' inequality)
- **Zero iff equal:** $KL(p||q) = 0 \Leftrightarrow p = q$
- **Asymmetric:** the two "directions" penalize different kinds of mistakes

The deep connection to ML: minimizing the KL divergence from data to model is equivalent to maximizing the likelihood:

$$\arg\min_\theta \; KL(p_{data}||p_\theta) = \arg\max_\theta \; E_{p_{data}}[\log p_\theta(x)]$$

This means **MLE is secretly minimizing KL divergence** — fitting the model distribution as close as possible to the empirical data distribution.

Cross-Entropy

Cross-entropy combines entropy and KL divergence into the quantity that ML practitioners use most directly as a **loss function**:

$$H(p, q) = -\sum_{x} p(x) \log q(x) = H(p) + KL(p \| q)$$

Since $H(p)$ is a constant (the entropy of the true distribution does not depend on model parameters), minimizing cross-entropy is equivalent to minimizing KL divergence.

In classification, the cross-entropy loss takes the familiar form:

$$L = -\sum_{c=1}^{C} y_c \log \hat{y}_c$$

where $\mathbf{y}$ is the one-hot true label and $\hat{\mathbf{y}}$ is the softmax output of the model. This is the default loss for virtually every neural network classifier — from logistic regression to transformers. Its gradient has the elegant form $\hat{y}_c - y_c$, which makes optimization efficient.

> **The information-theoretic view:** training a classifier with cross-entropy loss is equivalent to finding the model distribution q that is closest (in KL divergence) to the true data distribution p.

Practice: Entropy

Three binary probability distributions:

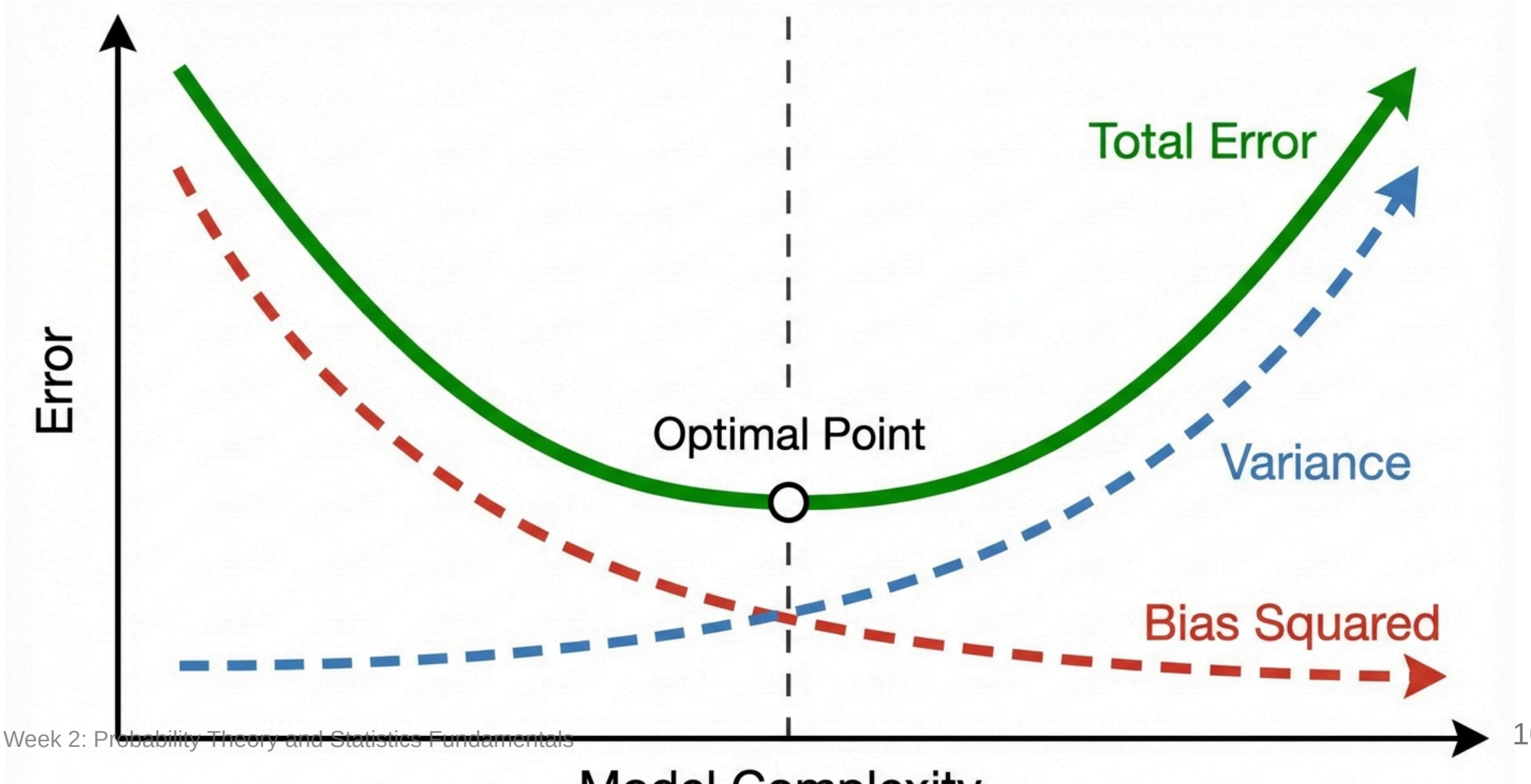
- A: $(1/2,\ 1/2)$
- B: $(1/4,\ 3/4)$
- C: $(1/8,\ 7/8)$

Questions:

1. Which distribution has the highest entropy? Calculate H for each.
2. Why does maximum entropy correspond to maximum uncertainty?

Hint: $H = -\sum p(x) \log_2 p(x)$

Bias-Variance Tradeoff



Bias-Variance Decomposition

One of the most fundamental concepts in all of machine learning: the expected prediction error can be decomposed into three components:

$$E[(y - \hat{f}(x))^2] = \underbrace{\text{Bias}^2}_{\text{systematic error}} + \underbrace{\text{Variance}}_{\text{sensitivity to data}} + \underbrace{\sigma^2}_{\text{irreducible noise}}$$

Bias measures how far the model's average prediction is from the truth — it reflects the limitations of the model family itself. A linear model fitting a curved relationship has high bias.

Variance measures how much the model's predictions fluctuate when trained on different datasets. A high-degree polynomial that fits every training set perfectly but gives wildly different predictions each time has high variance.

Noise (σ^2) is the irreducible error inherent in the data — no model can eliminate it.

Bias-Variance: The Fundamental Tension

As model complexity increases, bias decreases (the model can fit more complex patterns) but variance increases (the model becomes more sensitive to the specific training set). The total error is U-shaped, with a sweet spot in the middle.

Underfitting (High Bias)

- Model too simple for the data
- Cannot capture the underlying pattern
- **Training error:** high
- **Test error:** high
- Example: linear model on nonlinear data

Overfitting (High Variance)

- Model too complex for the data
- Memorizes noise in the training set
- **Training error:** low
- **Test error:** high
- Example: degree-20 polynomial on 10 points

> **The goal of ML:** find the model complexity that minimizes total error. Regularization, cross-validation, and Bayesian methods are all tools for navigating this trade-off.

Bias-Variance and the Bayesian Advantage

The Bayesian approach offers an elegant way to navigate the bias-variance tradeoff. Instead of committing to a single model (and accepting its specific bias-variance profile), Bayesian inference **averages over all models**, weighted by their posterior probability:

$$p(y|x, \mathcal{D}) = \int p(y|x, \theta)\, p(\theta|\mathcal{D})\, d\theta$$

This "model averaging" reduces variance — because averaging smooths out the idiosyncrasies of any single model — without necessarily increasing bias. It is one of the strongest theoretical arguments for the Bayesian framework.

In practice, exact Bayesian model averaging is often intractable, but approximations like **dropout** in neural networks (which can be interpreted as approximate Bayesian inference) and **ensemble methods** (which average over multiple trained models) achieve similar variance-reducing effects.

> Bayesian model averaging provides natural protection against overfitting — this is why Bayesian methods often outperform MLE on small datasets.

Summary: Probability and Statistics for ML

This week covered the probabilistic foundations that underpin every machine learning algorithm. Here is the conceptual map of what we learned (PRML Chapters 1-2):

Probability Foundations

- **Sum & Product Rules** — the two rules from which everything else derives (Eqs. 1.10-1.11)
- **Bayes' Theorem** — Posterior ∝ Likelihood × Prior (Eq. 1.12)
- **Key distributions** — Bernoulli, Binomial, Multinomial, Beta, Dirichlet, Gaussian

Statistical Inference

- **MLE** — maximizes $p(\mathcal{D}|\theta)$; efficient but prone to overfitting
- **MAP** — maximizes $p(\theta|\mathcal{D})$; equivalent to regularized MLE
- **Full Bayesian** — computes $p(\theta|\mathcal{D})$; captures uncertainty but computationally expensive
- **Information Theory** — entropy, KL divergence, cross-entropy loss

> The single most important takeaway: probability theory is not just a prerequisite — it **is** the language of machine learning. Every loss function, every regularizer, and every prediction has a probabilistic interpretation.

Critical Connections in ML

Where Will These Concepts Be Used?

Concept	ML Application
Bayes' Theorem	Naive Bayes, Bayesian Networks
Gaussian	Linear Regression, GMM, GP
MLE	Logistic Regression, Neural Networks
MAP	Regularization (L2, L1)
Bayesian	Uncertainty estimation, BNN
Entropy	Decision Trees, Information Gain
Cross-Entropy	Classification Loss

> Modern ML cannot be understood without these fundamentals!

Next Week: Linear Regression

Next week we put this probabilistic machinery to work on our first concrete ML algorithm — **linear regression**. We will derive the least squares solution, interpret it probabilistically (as MLE under Gaussian noise), add regularization (Ridge and Lasso — now you know these are Bayesian!), and analyze the bias-variance tradeoff in detail.

Reading: Bishop PRML Chapter 3.1

Thank You!

Questions & Contact

Instructor: Ekrem Çetinkaya

Office Hours	Tuesday 14:00 - 16:00
Location	Room F-B21
Booking	https://cal.com/ekremcetinkaya
Course Repo	https://github.com/ekremcet/ytu-private-lectures