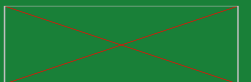


Do not edit
How to change the
design

**Join at slido.com
#3862491**



The Slido app must be installed on every computer you're presenting from



Lecture 4

Probability and Density Estimation

Review of Probability and Fitting Probability Models to Data

EECS 189/289, Fall 2026 @ UC Berkeley

Joseph E. Gonzalez and Narges Norouzi

Alexa, call my
mom!



Calling your
mom!



3862491

*Someone on YouTube just accidentally called their mom.

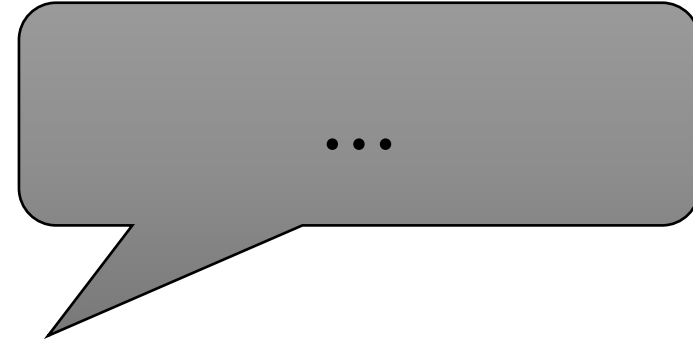


3862491

Siri, call my
mom!



Siri



*Unlikely that anyone accidentally calls their mom...

Try It Yourself



3862491

Take out your phone or laptop and say one of these out loud.

- “Hey Siri, what is the airspeed velocity of an unladen swallow?”
- “Alexa, set a timer for one second.”
- “OK Google, tell me a joke about Berkeley.”

Now say them again, but replace the first word: **“Hey Sardine...”, “Alaska...”, “OK Doodle...”**

Some devices woke up. Some did not. Some woke up for the wrong phrase.

- Every one of those devices was running a detector on **every second of sound in this room.**



3862491



Siri

How does a smart assistant
“know when to listen”?

Wake Words

A **wake word** is a verbal cue that triggers voice assistants to start actively listening.

Example: *“Siri, set an alarm for 6:00AM”*

Most voice assistants **continuously** run a wake word detector model on every sound they hear.



3862491



Wake Words

A **wake word** is a verbal cue that triggers voice assistants to start actively listening.

Example: *“Siri, set an alarm for 6:00AM”*

Most voice assistants **continuously** run a wake word detector model on every sound they hear.



Building a Wake-Word Detector



3862491

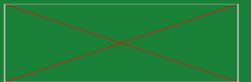


Is this a learning problem?

□ **Yes** – difficult to describe easy to demonstrate

What kind of learning problem?

**What kind of learning problem
is wake word detection.**



Building a Wake-Word Detector



3862491



Is this a learning problem?

- Yes – difficult to describe easy to demonstrate

What kind of learning problem is this?

- Supervised (we will collect labeled examples)

- Binary classification (is this sound a wake word? Yes/No)

What kind of model should we use?

- Many possible solutions, ... probably a small neural network.

How accurate
does the
detector need to
be?

Building a Wake-Word Detector



3862491



Streamed to the cloud
for processing.

For every segment of sound (e.g., 1 second):

- We observe a sound vector $\mathbf{X} \in \mathbb{R}^D$ (feature vector)
- Our model $f_w(\mathbf{X})$ (e.g., a Neural Network) predicts:

1: if a wake word was said

0: otherwise

Introduce **Random Variables** to
Model this Process

Building a Wake-Word Detector



3862491



For every segment of sound (e.g., 1 second) there are **three random variables**:



We ultimately observe X and Y but today we will focus on the relationship between Z and Y (the *joint probability*).



3862491

Today's Plan

Probability. Joint, marginal, conditional, independence, Bayes.

□ The detector fires. What is the chance a wake word was said?

Expectations. Turning probabilities into quantities with units.

□ What that accuracy costs per day.

Density estimation. Combining probability and optimization to fit models

□ **Maximum likelihood:** the objective behind most models in this course.

□ **Maximum a Posteriori:** accounting for priors.

A brief review of the

Basics of Probability



3862491

The Joint Probability Distribution

Let X and Y be **discrete random variables** that take values:

$$X \in \{x_1, \dots, x_L\} \quad \text{and} \quad Y \in \{y_1, \dots, y_M\}$$

The **joint probability function** $p(X = x_i, Y = y_j)$ is the probability of the event $X = x_i$ and $Y = y_j$.

The joint probability satisfies the **following two properties**:

$$\forall x_i, y_j: p(X = x_i, Y = y_j) \geq 0 \quad \text{(Non-negativity)}$$

$$\sum_{i=1}^L \sum_{j=1}^M p(X = x_i, Y = y_j) = 1 \quad \text{(Normalization)}$$

$p(X = x_i, Y = y_j)$	0	1	
	0.2	0.1	0.3
	0.25	0.45	0.7
	0.45	0.55	1



3862491

The Joint Probability Distribution

Let X and Y be **discrete random variables** that take values:

$$X \in \{x_1, \dots, x_L\} \quad \text{and} \quad Y \in \{y_1, \dots, y_M\}$$

The **joint probability function** $p(X = x_i, Y = y_j)$ is the probability of the event $X = x_i$ and $Y = y_j$.

The Sum Rule (Marginalization): defines the distribution over a subset of the random variables.

$p(X = x_i, Y = y_j)$	0	1	$p(X = x_i)$
	0.2	0.1	0.3
	0.25	0.45	0.7
$p(Y = y_j)$	0.45	0.55	1

$$p(X = x_i) = \sum_{j=1}^M p(X = x_i, Y = y_j)$$



3862491

Conditional Probability

We define the conditional probability as the chance of observing $Y = y_j$ given **we observed** $X = x_i$

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

Example:

What is the $p(Y = y_1 | X = x_2)$?

$$\frac{\begin{array}{|c|c|} \hline 0.25 & 0.45 \\ \hline \end{array}}{0.7} = \begin{array}{|c|c|} \hline 0.36 & 0.64 \\ \hline \end{array} \quad p(Y = y_1 | X = x_3)$$

$p(X = x_i, Y = y_j)$	0	1	$p(X = x_i)$
	0.2	0.1	0.3
	0.25	0.45	0.7
$p(Y = y_j)$	0.45	0.55	1



3862491

Product Rule: Chain Rule of Probability

Recall the definition of the conditional:

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

Multiply both sides by $p(X = x_i)$ you obtain the **product rule**:

$$p(X = x_i, Y = y_j) = p(Y = y_j | X = x_i) p(X = x_i)$$

The product rule is like a **chain rule** of probability, it allows us to **factorize** joint probability distributions into conditionals:

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

- *Notation:* we often drop x_i , y_j , and z_k for convenience.



3862491

Bayes' Theorem

Named after [Thomas Bayes](#) who was an 18th century statistician, philosopher, and minister.

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} = \frac{p(B|A)p(A)}{\sum_{A'} p(B|A')p(A')}$$

- Derived from product rule: $p(A, B) = p(A|B)p(B) = p(B|A)p(A)$

Useful in Bayesian statistics:

- $p(A)$ is the **prior belief** about A
- $p(B|A)$ is the **likelihood** of an observation B given those beliefs
- $p(A|B)$ is our **posterior belief** about A given our observation B



3862491

Independent Random Variables

Two variables X and Y are independent if the joint probability factorizes:

$$X \perp Y \Rightarrow p(X, Y) = p(X) p(Y)$$

Using the definition of conditionals:

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

Independent and Identically Distributed (IID) data (X_i, Y_i) is:

$$p(X_1, Y_1, \dots, X_N, Y_N) = \prod_{n=1}^N p(X_n, Y_n)$$

- We will often assume IID data (though it is often not true in practice).



3862491

Summarizing the Four Rules

Sum rule (marginalization): collapse a variable by summing it out.

$$p(X = x_i) = \sum_{j=1}^M p(X = x_i, Y = y_j)$$

Product rule (chain rule): factor a joint into a chain of conditionals.

$$p(X = x_i, Y = y_j) = p(Y = y_j | X = x_i) p(X = x_i)$$

Bayes' theorem: reverse the direction of a conditional.

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} = \frac{p(B|A)p(A)}{\sum_{A'} p(B|A')p(A')}$$

Independence: the special case where conditioning changes nothing.

$$X \perp Y \implies p(X, Y) = p(X) p(Y)$$

What happens with rare events?

Analyzing the Wake Word Detector

Building a Wake-Word Detector



3862491

For every segment of sound there are **three random variables**:



Today we will focus on the relationship between Z and Y :

$$p(Z, Y) = p(Z) p(Y | Z)$$

Prior: $p(Z)$ the probability that a wake word is said

Likelihood: $p(Y | Z)$ the likelihood of detecting a wake word given whether a wake word was said.

Today we will focus on the relationship between Z and Y :

$$p(Z, Y) = p(Z) p(Y | Z)$$



3862491

Prior: $p(Z)$ the probability that a wake word is said

Likelihood: $p(Y | Z)$ the likelihood of detecting a wake word given whether a wake word was said.

Our **wake word detector** can be characterized by:

➤ **Detection rate (True Positive Rate / Recall):** $p(Y = 1 | Z = 1)$

➤ If a wake word is said, do we detect it?

➤ **False alarm rate (False Positive Rate):** $p(Y = 1 | Z = 0)$

➤ If a wake word is not said, do we say that it was?

Building a Wake-Word Detector



3862491

Today we will focus on the relationship between Z and Y :

$$p(Z, Y) = p(Z) p(Y | Z)$$

Prior: $p(Z)$ the probability that a wake word is said

Likelihood: $p(Y | Z)$ the likelihood of detecting a wake word given whether a wake word was said.

Our **wake word detector** can be characterized by:

➤ **Detection rate (True Positive Rate / Recall):** $p(Y = 1 | Z = 1)$

➤ If a wake word is said, do we detect it?

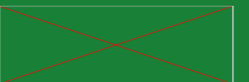
➤ **False alarm rate (False Positive Rate):** $p(Y = 1 | Z = 0)$

➤ If a wake word is not said, do we say that it was?

How could we improve the Wake Word Detector?



The Slido app must be installed on every computer you're presenting from



Building a Wake-Word Detector



3862491

Today we will focus on the relationship between Z and Y :

$$p(Z, Y) = p(Z) p(Y | Z)$$

➤ **Detection rate (True Positive Rate / Recall):** $p(Y = 1 | Z = 1)$

➤ **False alarm rate (False Positive Rate):** $p(Y = 1 | Z = 0)$

We really want to know the **Precision:** $p(Z = 1 | Y = 1)$

➤ Depends on the **detector** and **rarity of wake words** $p(Z = 1)$

Ultimately want high **precision** and **recall**.



3862491

Analyzing the Wake Word Detector

Let's assume wake words are rare and occur in about 0.01% of segments:

$$p(Z = 1) = 0.0001 \quad \text{and} \quad p(Z = 0) = 1 - p(Z = 1) = 0.9999$$

Suppose the wake word model **detects 99% of the wake words**:

$$p(Y = 1 | Z = 1) = 0.99$$

And **falsely detects (false positive)** a wake word just 0.1% of the time:

$$p(Y = 1 | Z = 0) = 0.001$$

If the **detector is positive**, what is the chance that **it was a wake word**?

$$p(Z = 1 | Y = 1) = \frac{p(Y = 1, Z = 1)}{p(Y = 1)} = \frac{p(Y = 1 | Z = 1) p(Z = 1)}{p(Y = 1)}$$

Bayes' Theorem



3862491

Analyzing the Wake Word Detector

We assumed rare wake words: $p(Z = 1) = 0.0001$

A **true positive rate (recall)** of 99%: $p(Y = 1 | Z = 1) = 0.99$

And a **false positive rate** of 0.1%: $p(Y = 1 | Z = 0) = 0.001$

If the **detector is positive**, what is the chance that **it was a wake word**?

$$p(Z = 1 | Y = 1) = \frac{p(Y = 1 | Z = 1) p(Z = 1)}{p(Y = 1 | Z = 1) p(Z = 1) + p(Y = 1 | Z = 0) p(Z = 0)}$$
$$p(Z = 1 | Y = 1) = \frac{0.99 * 0.0001}{0.99 * 0.0001 + 0.001 * 0.9999} = 0.09 = 9\%$$

Is that good (enough)?

How would we improve it?



3862491

Bayesian Updates: Wake Word Detector

•**Prior:** we assumed rare wake words $p(Z = 1) = 0.0001$

After observing the test $Y = 1$ we **updated our belief** about the presence of the wake word using Bayes Rule:

$$p(Z = 1 | Y = 1) = \frac{p(Y = 1, Z = 1)}{p(Y = 1)} = \frac{p(Y = 1 | Z = 1) p(Z = 1)}{p(Y = 1)} = 0.09$$

Posterior: the updated belief $p(Z = 1 | Y = 1)$ after the observation.

Bayesian Updating: the process of starting from a **prior belief** and then using **observations** to compute **posterior distribution** (updated belief) is central to **Bayesian machine learning**.



3862491

Revisiting the Math with Counts

Imagine 1,000,000 one-second segments.

- Wake words are said in 0.01% of them: **100 segments** contain a wake word, 999,900 do not.
- The detector catches 99% of the real ones: **99 true detections.**
- The detector fires on 0.1% of the silent ones: **999 false alarms.**

The detector fires $99 + 999 = 1,098$ times. **Only 99 of those are real.**

□ $99 / 1,098 \approx 9\%$

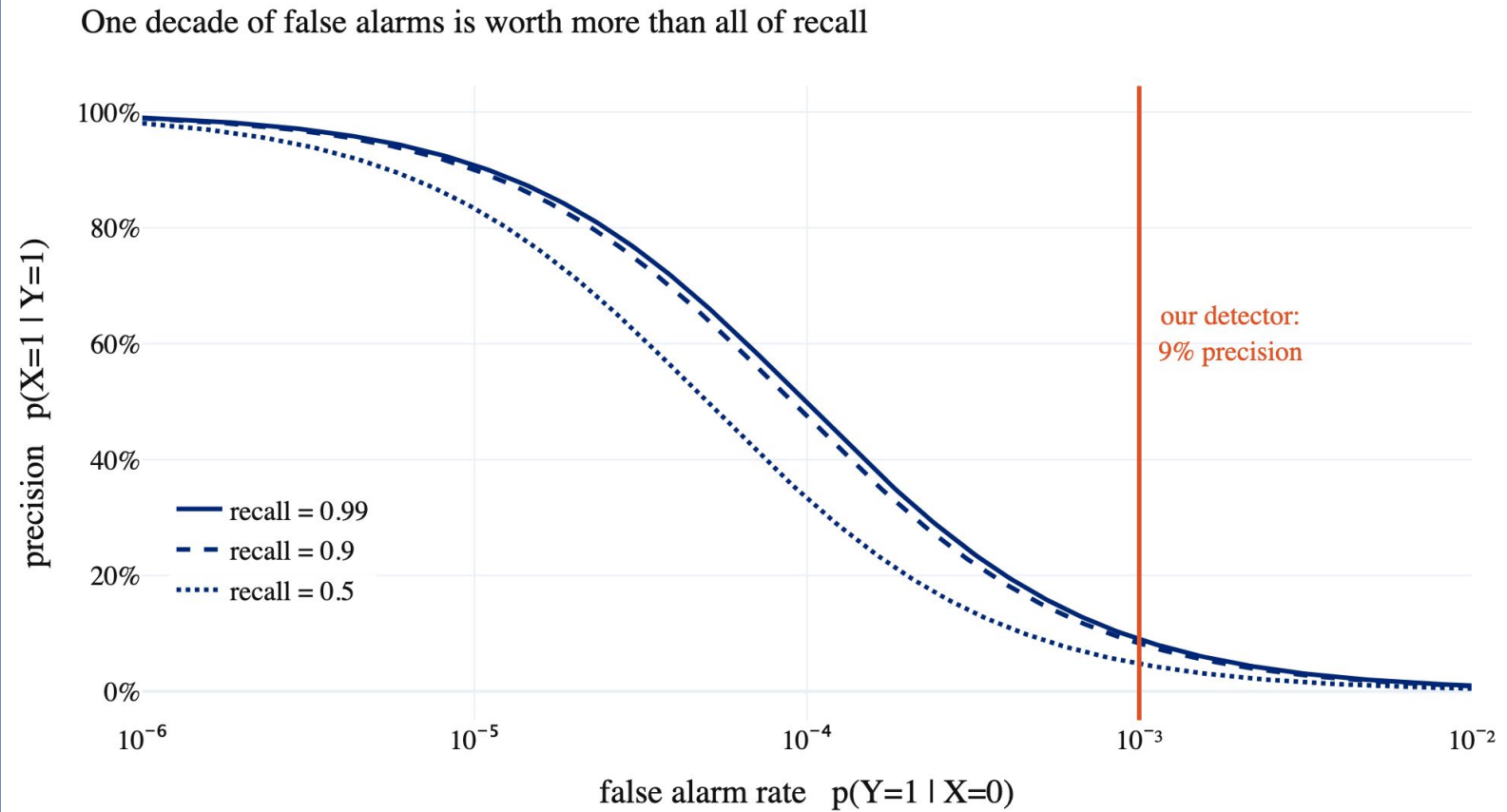
The base rate dominates. There are **10,000 times more silent segments** than wake-word segments, so a **small error rate on a huge population** becomes a real challenge.



3862491

Demo

Which knob
matters?





3862491

What Is 9% Precision Costing Us?

Every time the detector fires, the device **invokes the full speech model**. Invoking the full speech model can be expensive. Why?

- **Energy:** to **evaluate** the full model or **send the data** to the cloud
- **Money:** to evaluate the **model in a datacenter**.

Let C be the cost (e.g., \$ or Joules) of invoking the full speech model, what is the expected cost per day for false alarms?

- Keep in mind each segment of sound **is not independent**

To answer this, we will compute an **expectation**.



3862491

Expectations

The **expectation of a random variable** X is the weighted average with respect to a probability distribution $p(x)$:

$$\mathbb{E}[X] = \sum_x x p(x)$$

An expectation maps a random variable to a number, but a conditional expectation is a function of the variable being conditioned on.

The **conditional expectation** of a random variable is a **weighted average** with respect to the **conditional distribution**:

$$\mathbb{E}[X \mid y] = \sum_x x p(x|y)$$



3862491

Functions of Random Variables

The function of a random variable is also a random variable:

$$Y = f(X)$$

The **expectation of a function** $f(X)$ is the weighted average with respect to a probability distribution $p(x)$:

$$\mathbb{E}[f] = \mathbb{E}[f(X)] = \sum_x f(x) p(x)$$

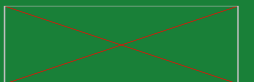
The expectation of a function of multiple variables is defined over the joint distribution:

$$\mathbb{E}[f(X, Y)] = \sum_{x, y} f(x, y) p(x, y)$$

**For any two correlated random
variables X and Y**



The Slido app must be installed on every computer you're presenting from





3862491

Linearity of Expectation

For any two *correlated* random variables X and Y :

$$\mathbb{E}[aX + bY + c] =$$

(a) $a \mathbb{E}[X] + b\mathbb{E}[Y] + c + \text{Cov}[X, Y]$

(b) $a^2 \mathbb{E}[X] + b^2 \mathbb{E}[Y] + c$

(c) $a \mathbb{E}[X] + b\mathbb{E}[Y] + c$

(d) $\mathbb{E}[aX + bY] + c$

(e) None of the above



3862491

Linearity of Expectation

For **any** two random variables X and Y :

$$\mathbb{E}[aX + bY + c] = a\mathbb{E}[X] + b\mathbb{E}[Y] + c$$

- Does not require any assumptions about $p(X, Y)$

$$\begin{aligned}\mathbb{E}[aX + bY + c] &= \sum_x \sum_y (a x + b y + c) p(x, y) \\&= \sum_x \sum_y (a x p(x, y) + b y p(x, y) + c p(x, y)) \\&= a \sum_x x \sum_y p(x, y) + b \sum_y y \sum_x p(x, y) + c \sum_x \sum_y p(x, y) \\&= a \mathbb{E}[X] + b \mathbb{E}[Y] + c\end{aligned}$$



3862491

Variance

The **variance** of a random variable defines its **spread**:

$$\text{var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$$

Using linearity of expectation, we can rewrite the variance as:

$$\text{var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

What is $\text{var}[aX + b]$?

$$\begin{aligned}\text{var}[aX + b] &= \underbrace{\mathbb{E}[(aX + b)^2]}_{\mathbb{E}[a^2X^2 + 2abX + b^2]} - \underbrace{\mathbb{E}[aX + b]^2}_{(a\mathbb{E}[X] + b)^2} \\ &= a^2\mathbb{E}[X^2] + \cancel{2ab\mathbb{E}[X]} + \cancel{b^2} &= a^2\mathbb{E}[X]^2 + \cancel{2ab\mathbb{E}[X]} + \cancel{b^2} \\ \text{var}[aX + b] &= a^2(\mathbb{E}[X^2] - \mathbb{E}[X]^2) = a^2 \text{var}[X]\end{aligned}$$



3862491

Covariance

The covariance of two random variables defines the degree to which they vary together:

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

The covariance of two **random vectors** $\mathbf{x}$ and $\mathbf{y}$ is the matrix:

$$\text{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T])] = \mathbb{E}[\mathbf{x}\mathbf{y}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T]$$

We will return to covariance in greater detail in future lectures.



3862491

What Is 9% Precision Costing Us?

Let C be the cost (e.g., \$ or Joules) of invoking the full speech model, what is the expected cost per day for false alarms?

➤ Note: each segment of sound **is not independent**

If each segment of sound is **1 second long**, then there about $24*60*60=86,400$ segments in a day.

➤ Let I_t be the indicator variable that the segment t is a **false alarm**

$$\mathbb{E}[\mathbf{Total\ Cost}] = \mathbb{E}\left[\sum_{t=1}^{86,400} C I_t\right] = C \sum_{t=1}^{86,400} \mathbb{E}[I_t] = C \sum_{t=1}^{86,400} p(Z = 0, Y = 1)$$

$$p(Z = 0, Y = 1) = p(Y = 1 \mid Z = 0) p(Z = 0) = 0.001 * 0.9999$$

$$\mathbb{E}[\mathbf{Total\ Cost}] = 86.4 * C$$

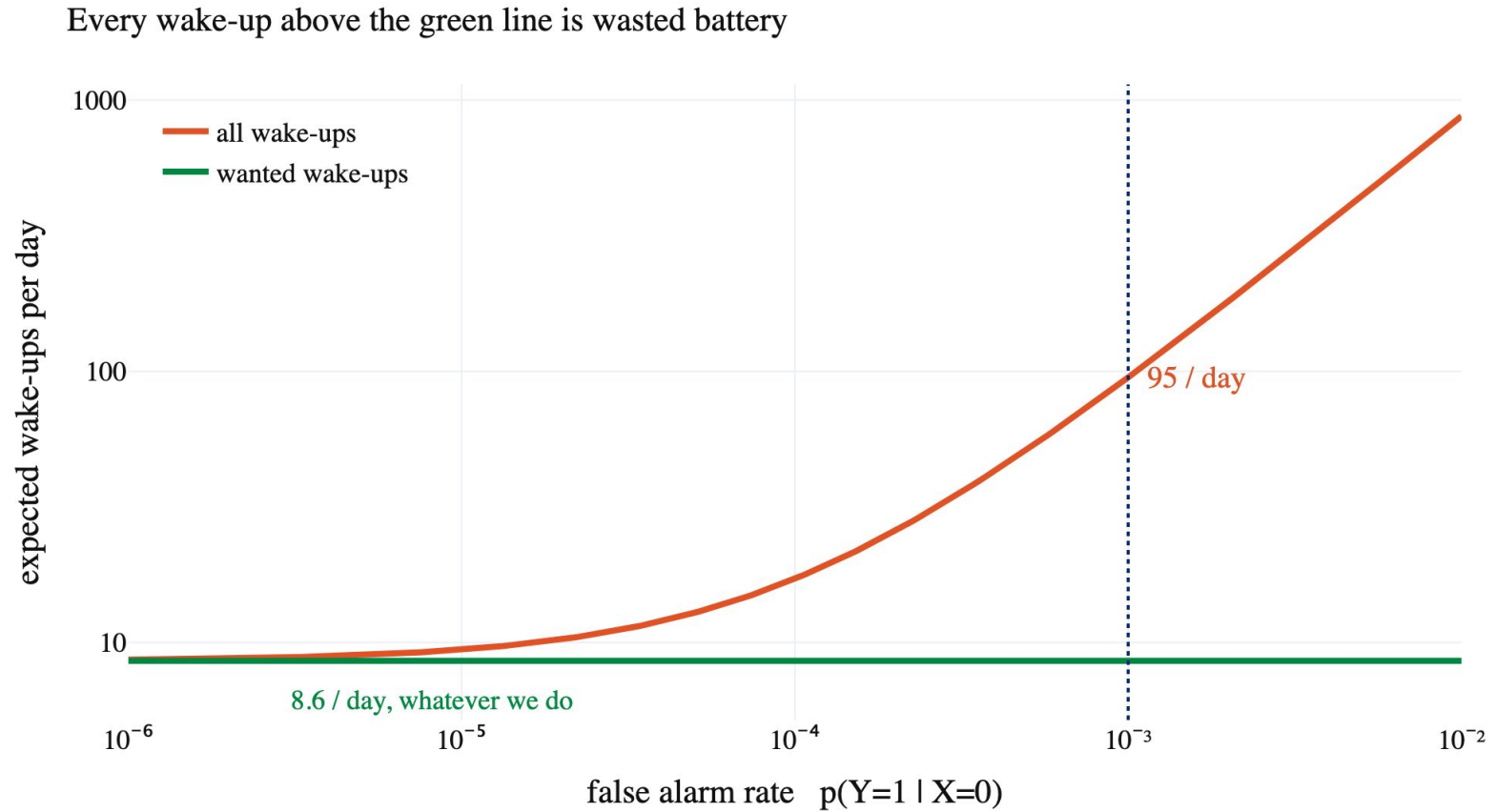
In lecture, I had incorrectly said $\mathbb{E}[I_t] = p(Z = 0 \mid Y = 1)$. This is **incorrect** since we want an indicator for all events not just when the detector triggered.



3862491

Demo

What 9% precision
costs per day



From counting to

Modeling Distributions



3862491

Modeling Distributions

Many machine learning models attempt to model the (joint) **probability distribution** of the **data**.

□ Instead of modeling “Does this sound contain a wake word” they model “The probability that this sound is a wake word”.

There are **many ways** to model a (joint) probability distribution.

□ Tabular representations

□ Classic probability models

□ Nonparametric models (histograms, kernel density estimation)



3862491

Tabular Representations

Already saw a **tabular representation** of a **joint distribution**:

	0.2	0.1
	0.1	0.1
	0.15	0.35



$$p_w(X = x_i, Y = y_j) =$$

Here p_w is a **parametric probability model** and $w \in \mathbb{R}^{3,2}$ satisfies:

$$\forall x_i, y_j: w_{i,j} \geq 0 \text{ (Non-negativity)}$$

$$\sum_{i=1}^L \sum_{j=1}^M w_{i,j} = 1 \text{ (Normalization)}$$

For **discrete distributions**, the tabular representation is **complete**: can express any discrete joint distribution

➤ Often not efficient (can be exponential in variables)

Bernoulli Distribution



3862491

The distribution of binary random variable $X \in \{0,1\}$ defined by:

$$X \sim \text{Bern}(\mu) \implies p(x|\mu) = \mu^x(1 - \mu)^{1-x}$$

where the parameter $0 \leq \mu \leq 1$.

$$\mathbb{E}[X] = 0(1 - \mu) + 1\mu = \mu$$

$$\text{var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \mu - \mu^2 = \mu(1 - \mu)$$

$$\mathbb{E}[X^2] = 0^2(1 - \mu) + 1^2\mu = \mu$$

Note that for $\mu \notin \{0,1\}$, X never takes the expected value!

The Bernoulli distribution is used extensively in classification.



3862491

Continuous Random Variables

For a **continuous random variable** X we use a **probability density** $p(x)$. We can use the density to compute the probability that X is between a and b :

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

A density $p(x)$ has the standard properties:

- **Normalization** $\int p(x) dx = 1$
- **Non-negativity** $p(x) \geq 0$.

A density **is not a probability**:

- $p(x)$ can be larger than 1,
- $P(X = x) = 0$ for any x .

Previous rules transfer with sums replaced by integrals:

- **Sum Rule:** $p(x) = \int p(x, y) dy$
- **Product Rule:** $p(x, y) = p(y | x) p(x)$
- **Expectation of a Function:** $\mathbb{E}[f(x)] = \int f(x) p(x) dx$



3862491

Normal (Gaussian) Distribution

If a single real-valued variable X is **normally distributed**, then:

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

Where the parameters μ and σ^2 are called the mean and variance.

With some calculus you can show $\mathbb{E}[X] = \mu$ and $\text{var}[X] = \sigma^2$.

Fitting distributions to data with

Density Estimation and MLE



3862491

Density Estimation

Density estimation is the process of trying to **infer the probability distribution** $p(\mathbf{X})$ given observations (data) $\mathcal{D} = \{x_1, \dots, x_N\}$.

There are *infinitely many* density functions from which our data could have been sampled.

- Could be any function that assigns **non-zero probability to the data**
- The choice of function is a **modeling decision (inductive bias!)**

Density estimation is a fundamental technique in machine learning and is part of most modern methods.



3862491

Empirical Probability Distributions

Consider N trials $\{(X_1, Y_1), \dots, (X_N, Y_N)\}$ in which we sample (X, Y) .

Let n_{ij} be the number (count) of trials where $(X = x_i, Y = y_j)$, we can construct a discrete **empirical probability distribution**:

$$\hat{p}(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Frequentist View: In the limit as $N \rightarrow \infty$, the empirical distribution converges (with probability 1, by the law of large numbers) to the true distribution:

$$p(X = x_i, Y = y_j) = \lim_{N \rightarrow \infty} \hat{p}(X = x_i, Y = y_j)$$



3862491

The Empirical Distribution Is a Model

The empirical distribution is a **model** of the probability of events:

$$\hat{p}(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

- It satisfies **non-negativity** and **normalization**, we can marginalize it, condition on it, and apply Bayes' theorem to it.

Is it a good model of the probability of events?

- What are some issues with this approach when N is **small**?
- What happens when there are **many variables**?
- What happens when the variables take on **many values** or are **continuous**?



3862491

Estimating the Parameters of a Distribution

Given a collection of data $\mathcal{D} = \{x_1, \dots, x_N\}$, there are several ways to estimate the “best” parameters for a given distribution.

- **Maximum Likelihood Estimation (MLE):** choose the parameters that maximize the **likelihood** of the data. (Frequentist)

$$w_{\text{MLE}}^* = \arg \max_w p(\mathcal{D} | w)$$

- **Maximum A Posteriori (MAP) Estimation:** choose the parameters that maximize the posterior given the data and the **prior**. (Bayesian)

$$w_{\text{MAP}}^* = \arg \max_w p(\mathcal{D} | w) p(w)$$

Both methods rely on computing the **likelihood of the data**.

- We cover both today: MLE first, then MAP.



3862491

The Likelihood Function

The likelihood function describes the **probability of the observed data** $\mathcal{D} = \{x_1, \dots, x_N\}$ under a particular distribution.

If we assume each observation $X_i \sim p(x|w)$ is an **Independent and Identically Distributed (IID)** sample from some parametric distribution $p(x|w)$ parameterized by w then the likelihood is:

$$p(\mathcal{D}|w) = \prod_{n=1}^N p(x_n | w)$$

Independent

Identically



3862491

The Log Likelihood Function

In practice, we will typically work with the **log likelihood function**:

$$p(\mathcal{D}|w) = \prod_{n=1}^N p(x_n | w) \implies \ln p(\mathcal{D}|w) = \sum_{n=1}^N \ln p(x_n | w)$$

We take the **log** of the likelihood function for several reasons:

- It's **easier to manipulate summations** than products (e.g., calculus).
- The log probabilities and summation are **more numerically stable**.

Does this change our **optimization problem**?

- No – log is **monotonically increasing function** of its arguments:

$$\arg \max_w p(\mathcal{D}|w) = \arg \max_w \ln p(\mathcal{D}|w)$$



3862491

Maximum Likelihood Estimation

Maximum likelihood estimation has two steps:

1. **Modeling:** Define the **log likelihood function** of the **data**: $\ln p(\mathcal{D}|w)$
2. **Optimization:** Compute the parameters that **maximize** the log-likelihood:

$$w_{\text{ML}} = \arg \max_w \ln p(\mathcal{D}|w)$$

Some machine learning people prefer to **minimize** a loss (same thing):

$$w_{\text{ML}} = \arg \min_w -\ln p(\mathcal{D}|w)$$

- Maximum likelihood estimation (MLE) → **minimize the log loss**.

Maximum likelihood estimation is a core technique in ML and is the basic objective for everything from the **logistic regression** to **ChatGPT**



3862491

The MLE for ChatGPT

• We model the likelihood of all the documents on the internet:

$$\begin{aligned} & \ln p(\text{Internet} \mid w) \\ &= \sum_{\text{Doc} \in \text{Internet}} \sum_{\text{Word} \in \text{Doc}} \ln p(\text{Word} \mid \text{Words before it in Doc}, w) \end{aligned}$$

Pre-Training: billions of dollars spent annually computing an MLE:

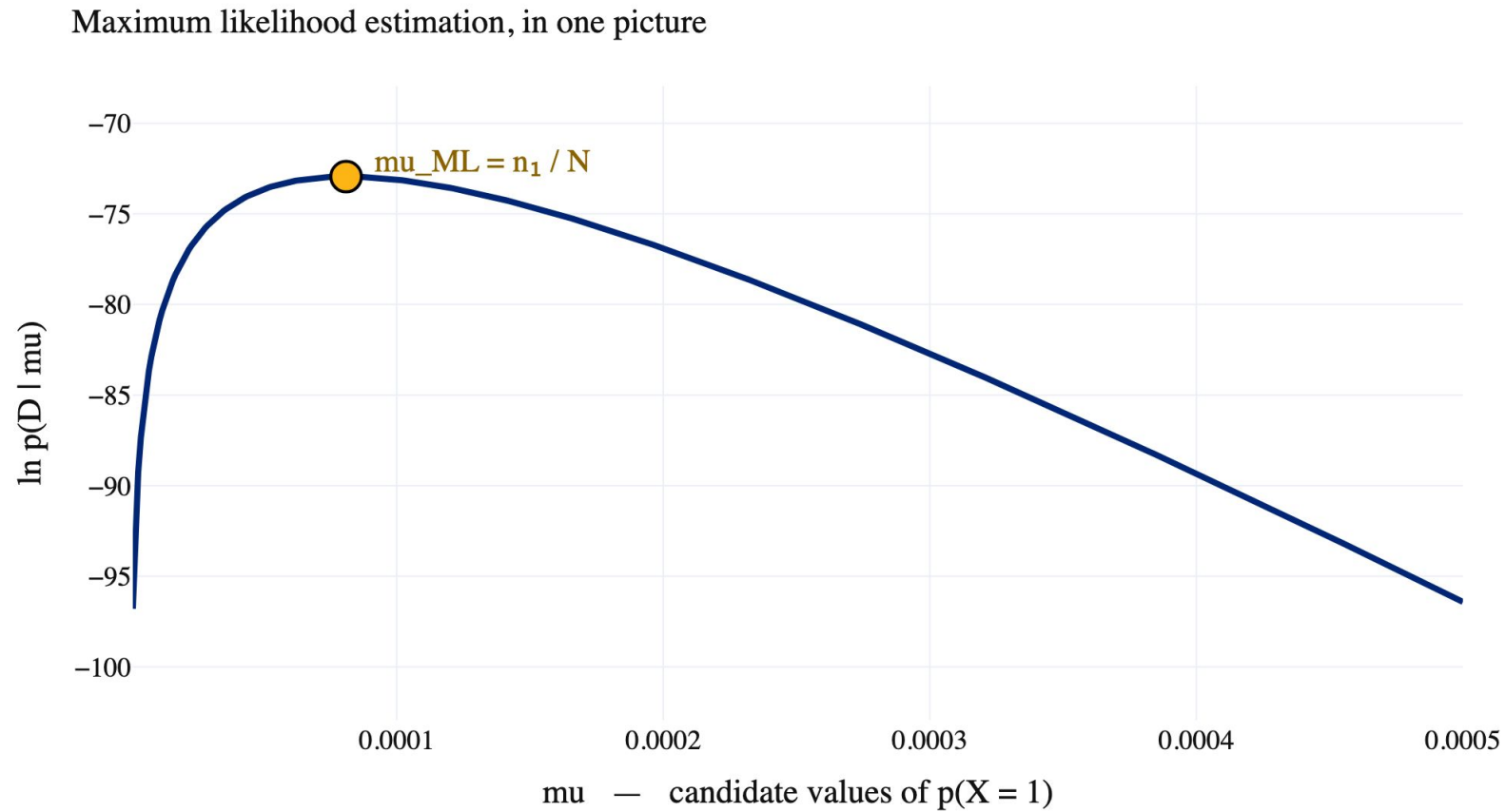
$$\arg \max_w \ln p(\text{Internet} \mid w)$$



3862491

Demo

Maximum likelihood,
in one picture





3862491

The MLE for IID Bernoulli Samples (Part 1)

Example: Bernoulli Distribution

We observe N samples $\mathcal{D} = \{x_1, \dots, x_N\}$ from a Bernoulli Distribution.

1. Modeling: Let's assume IID Bernoulli trials:

$$\ln p(\mathcal{D}|\mu) = \sum_{n=1}^N \ln(\mu^{x_n}(1-\mu)^{(1-x_n)}) = \sum_{n=1}^N x_n \ln(\mu) + (1-x_n) \ln(1-\mu)$$

Since each x_n is either **zero** or **one** we can rewrite this expression as:

$$\ln p(\mathcal{D}|\mu) = n_1 \ln(\mu) + n_0 \ln(1-\mu)$$

Where n_0 and n_1 are the number of observed zeros and ones, respectively.

The counts n_0 and n_1 are called **sufficient statistics** because they are all we need from the data to estimate the parameter.



3862491

The MLE for IID Bernoulli Samples (Part 2)

2. Optimization: Maximize the log likelihood function:

$$\arg \max_{\mu} n_1 \ln(\mu) + n_0 \ln(1 - \mu)$$

Here we use the **stationary-points approach** to optimization:

1. Take the derivative $\frac{\partial}{\partial \mu} \ln p(\mathcal{D}|\mu)$

$$\frac{\partial}{\partial \mu} (n_1 \ln(\mu) + n_0 \ln(1 - \mu)) = \frac{n_1}{\mu} - \frac{n_0}{1 - \mu}$$

2. Solve for the stationary point $\frac{\partial}{\partial \mu} \ln p(\mathcal{D}|\mu) = 0$

$$\frac{n_1}{\mu} - \frac{n_0}{1 - \mu} = 0 \Rightarrow n_1(1 - \mu) = n_0\mu \Rightarrow \mu_{ML} = \frac{n_1}{n_0 + n_1}$$

3. Verify that stationary points are maximum: $\frac{\partial^2}{\partial \mu^2} \ln p(\mathcal{D}|\mu) < 0$

Left as an
exercise
(for your AI).



3862491

The MLE for IID Bernoulli Samples (Part 3)

3. Verify that critical points are maxima $\frac{\partial^2}{\partial \mu^2} \ln p(\mathcal{D}|\mu) < 0$

$$\frac{\partial^2}{\partial \mu^2} \ln p(\mathcal{D}|\mu) = -\frac{n_1}{\mu^2} - \frac{n_0}{(1-\mu)^2}$$

Because $\mu \in (0,1)$ and n_0, n_1, μ^2 , and $(1-\mu)^2$ are all non-negative:

$$\frac{\partial^2}{\partial \mu^2} \ln p(\mathcal{D}|\mu) = -\frac{n_1}{\mu^2} - \frac{n_0}{(1-\mu)^2} \leq 0$$

Stopped Here



3862491

The Bernoulli MLE Is Just Counting

The maximum likelihood estimate (MLE) is:

$$\mu_{ML} = \frac{n_1}{n_0 + n_1} = \frac{n_1}{N}$$

This is **the fraction of ones in the data**, which is the **empirical distribution** defined earlier in lecture.

➤ μ_{ML} is the parameter value that makes the **observed data most probable** under a Bernoulli model.

The same argument applied to the **tabular model** gives:

$$\hat{p}(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Therefore, not only are **counts** the intuitive estimate but they are **also the maximum likelihood estimate**.



3862491

Your Turn: Let's flip a coin

• flip a coin $N = 10$ times and observe 7 heads. What is $\mu_{ML} = p(\text{heads})$

$$\mu_{ML} = \frac{n_1}{N} = \frac{7}{10}$$

I flip a different coin 10 times and observe 0 heads. What is $\mu_{ML} = p(\text{heads})$?

$$\mu_{ML} = \frac{n_1}{N} = \frac{0}{10}$$

Your model assigns **zero probability** to heads $p(\text{heads}) = 0$. Are you happy with that estimate?

- It is the correct maximum likelihood estimate.
- It is also almost certainly wrong! (Assuming my coin has two sides...)

Rare events produce confident but uninformative estimates. The remedy is to introduce a prior, which is **MAP estimation**.



3862491

The Issue with Maximum Likelihood and Rare Events

The Bernoulli MLE $\mu_{ML} = \frac{n_1}{N}$ is the observed frequency.

➤ When the event is rare, the **observed frequency is often zero**.

With $N = 10,000$ segments and a true rate of 0.0001:

$$p(n_1 = 0) = (1 - 0.0001)^{10,000} \approx 0.37$$

we observe $n_1 = 0$ about a ***third of the time***.

➤ A third of the time we would say wake words are never said.

We know something the data does not contain; wake words can occur.

➤ We can incorporate this using a **Maximum a Posteriori (MAP) estimate**.



3862491

The Parameter as a Random Variable

Maximum likelihood estimation treats μ as an **unknown constant**.

The Bayesian view treats μ as a **random variable** with a distribution. Applying Bayes' theorem to the parameter:

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

- **Prior** $p(\mu)$: what we believe about μ before seeing data.
- **Likelihood** $p(\mathcal{D} | \mu)$: is the quantity we maximized for the MLE.
- **Posterior** $p(\mu | \mathcal{D})$: the updated belief after seeing the data.

Maximum a Posteriori (MAP) estimate maximizes the posterior:

$$\mu_{\text{MAP}} = \arg \max_{\mu} \ln p(\mathcal{D} | \mu) + \ln p(\mu)$$

- The denominator $p(\mathcal{D})$ does not depend on μ , so it does not affect the maximization.



3862491

Modeling the Prior

The **choice** and **parameterization** of model for the prior $p(\mu)$ is a **design decision**.

- Capture our prior beliefs about the μ .
- Model is often chosen for convenience (conjugate to the likelihood).
- The prior parameters are set using *prior knowledge* or *another prior*.

What would be a good prior $p(\mu)$ for the Bernoulli coin flip example?

- The parameter $0 \leq \mu \leq 1$ is the probability of landing heads
- We expect the coin (and the flipper) to be approximately fair.



3862491

The Beta Distribution

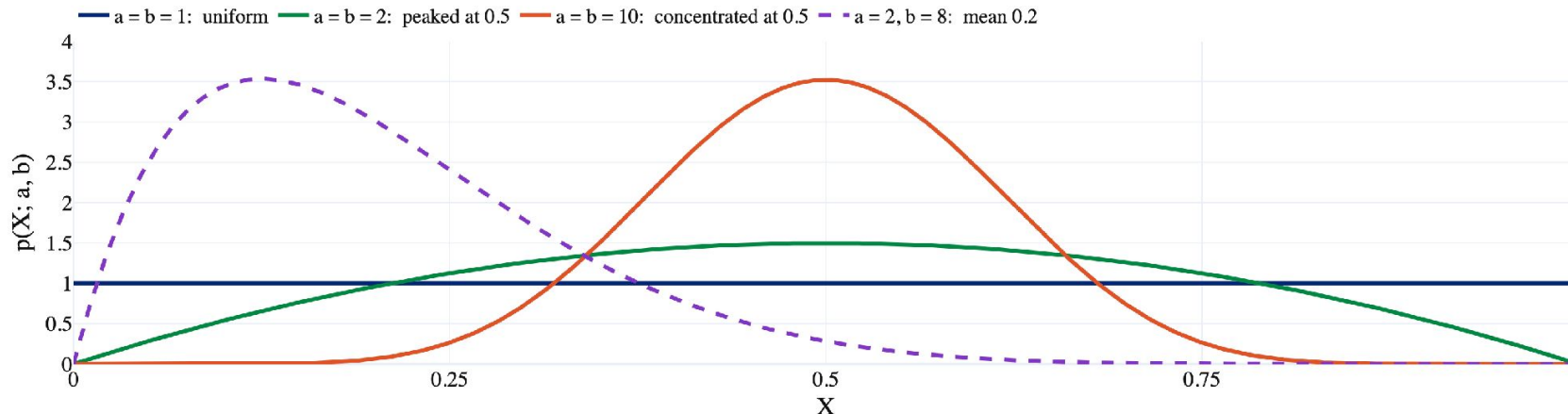
The **Beta distribution** for the **continuous random** variable $0 \leq X \leq 1$:

$$p(X; a, b) = \frac{1}{B(a, b)} X^{a-1} (1 - X)^{b-1}$$

with **parameters by α and β** with the following shape:

- $a = b = 1$: every value of X is equally plausible.
- $a = b = 2$: values of X close to 0.5 are more likely
- Large a and b concentrate the distribution near $\frac{a}{a+b}$

The beta function $B(a, b)$ is the normalizer (computed numerically) and outside the scope of this class.





3862491

Deriving the Posterior for Bernoulli + Beta

Recall we are trying to compute:

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

Substituting the Bernoulli and the Beta distributions (without normalization):

$$p(\mu | \mathcal{D}) \propto p(\mathcal{D} | \mu) p(\mu) = \mu^{n_1} (1 - \mu)^{n_0} \mu^{(a-1)} (1 - \mu)^{(b-1)}$$

Collecting terms, we can see that the **posterior is also a Beta distribution**:

$$p(\mu | \mathcal{D}) \propto \mu^{(n_1+a-1)} (1 - \mu)^{(n_0+b-1)}$$

with parameters $a' = (n_1 + a)$ and $b' = (n_0 + b)$.

When a prior paired with a likelihood has the same form as the posterior it is called a **conjugate prior**.

➤ The Beta distribution is the conjugate prior for the Bernoulli likelihood.



3862491

Computing the MAP

Maximum a Posteriori (MAP) estimate maximizes the posterior:

$$\mu_{\text{MAP}} = \arg \max_{\mu} \ln p(\mathcal{D} | \mu) + \ln p(\mu)$$

When deriving the posterior we already showed (collecting terms):

$$\ln p(\mathcal{D} | \mu) + \ln p(\mu) = (n_1 + a - 1) \ln \mu + (n_0 + b - 1) \ln(1 - \mu)$$

When computing the MLE we derived the **stationary point** of:

$$\frac{n_1}{n_1 + n_0} = \arg \max_{\mu} n_1 \ln \mu + n_0 \ln(1 - \mu)$$

Substituting $n'_1 = (n_1 + a - 1)$ and $n'_0 = (n_0 + b - 1)$.

$$\mu_{\text{MAP}} = \frac{n_1 + a - 1}{n_1 + n_0 + a + b - 2} = \frac{n_1 + a - 1}{N + a + b - 2}$$



3862491

The Prior as Pseudo-Counts

The prior acts like a form of pseudo-counts (assumed binary observations) by contributing $a - 1$ imaginary successes $b - 1$ imaginary failures.

$$\mu_{\text{MAP}} = \frac{n_1 + a - 1}{n_1 + n_0 + a + b - 2} = \frac{n_1 + a - 1}{N + a + b - 2}$$

With $a = b = 2$ this becomes $\mu_{\text{MAP}} = \frac{n_1 + 1}{N + 2}$ one imaginary observation of each outcome.

➤ Regardless of the observations $0 < \mu_{\text{MAP}} = \frac{n_1 + 1}{N + 2} < 1$ the probability is never 0 or 1.

This is also called **Laplace Smoothing**

➤ Add **pseudo counts to empirical estimates** to act as a prior when there is **limited data** or **events are rare**.



3862491

How μ_{MAP} and μ_{ML} scale with N

What happens as N grows and how do μ_{MAP} and μ_{ML} compare?

$$\mu_{\text{ML}} = \frac{n_1}{n_0 + n_1} = \frac{n_1}{N}, \quad \mu_{\text{MAP}} = \frac{n_1 + a - 1}{n_1 + n_0 + a + b - 2} = \frac{n_1 + a - 1}{N + a + b - 2}$$

As N grows, the constants $a - 1$ and $a + b - 2$ become negligible and the **two estimates converge**.

- **Small N :** the prior dominates, and the estimate is pulled toward the prior mean.
- **Large N :** the data dominates, and MAP and MLE agree.



3862491

What We Did Today

Reviewed basic probability, Bayes Theorem, Expectations, and Variance

Maximum likelihood determine model parameters by maximizing the likelihood of the data under the model.

Maximum a posteriori determine model parameters that maximize the posterior distribution.

Next Lecture: we will combine ideas learned today to develop Gaussian Mixture Models.

Lecture 4

Probability and Density Estimation

Credit: Joseph E. Gonzalez and Narges Norouzi

Reference Book Chapters:

- **Probability:** Chapter 2.[1-2]
- **Density Estimation and MLE:** Chapter 2.3