

# CS 480/680

## Introduction to Machine Learning

### Lecture 3

### Maximum Likelihood Estimation and Entropy

Kathryn Simone  
17 September 2024

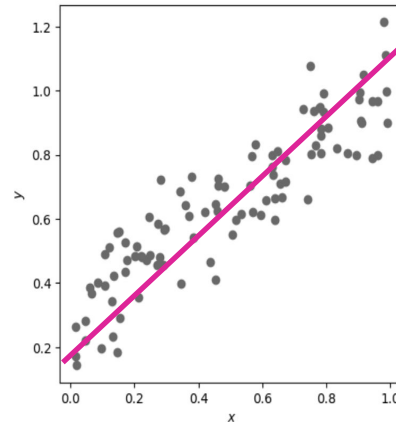
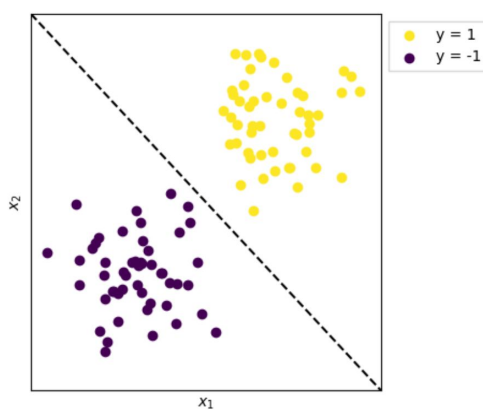


Good morning, everyone! Welcome to Lecture 3. Today, we will be diving deep into Maximum Likelihood Estimation (MLE) and key information-theoretic measures such as entropy and Kullback-Leibler (KL) divergence.

This week, we're stepping into the second phase of our course, where we broaden our understanding of the foundational statistical methods that underpin both supervised and unsupervised learning.

Our primary goal today is to connect the statistical framework of MLE with practical applications in machine learning, exploring how it allows us to estimate model parameters and evaluate their performance in various contexts.

# Classification and regression are supervised learning tasks

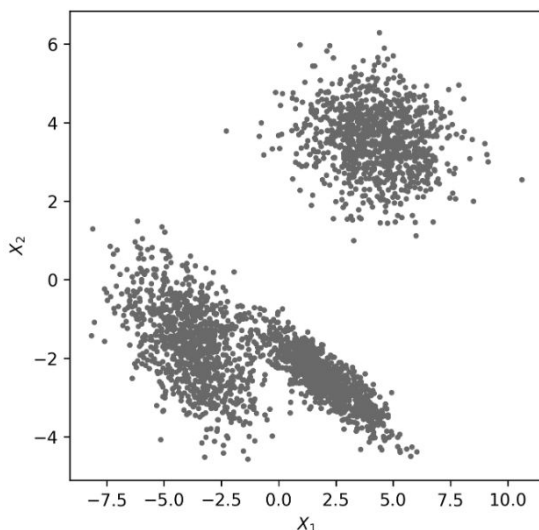


Last week, we focused primarily on supervised learning tasks, specifically classification and regression. In classification, think about separating data into two distinct clusters in a vector space using an algorithm like the perceptron, which finds a separating hyperplane that perfectly classifies the data points into two classes.

In regression, our goal shifted to fitting a line through data points in such a way that it minimizes the difference between predicted and actual outcomes, typically by minimizing the least squares loss.

Both classification and regression are examples of supervised learning, where the learning algorithm has access to true outcome values (labels) during training. This allows the model to learn the mapping between inputs and outputs effectively.

# Unsupervised learning concerns pattern identification



This week, however, we take a brief detour into unsupervised learning. Unsupervised learning involves scenarios where we don't have labeled data. There are no given outcomes to predict; instead, we aim to uncover hidden patterns, groupings, or structures within the data.

For example, imagine a scatter plot with two features,  $x_1$  and  $x_2$ . Visually, you might detect three clusters: one long and narrow with a strong linear relationship between the variables, and the other two more diffuse with weaker correlations. This clustering represents a type of unsupervised learning task where we identify inherent structures without specific guidance. A common technique in unsupervised learning is clustering, where we identify groups of data points that are similar to each other. Clustering methods, such as K-means, find these groups by minimizing the distance between points within each cluster. We'll cover clustering on Thursday, but this illustrates the core of unsupervised learning: discovering patterns without labeled outcomes.

So unsupervised learning is all about discovering patterns and structures in data, and today we'll be encountering Maximum Likelihood Estimation (MLE), which plays a big role here; it's a method we use to estimate the parameters of models that best explain what we observe.

	Lecture	Date	Topics	
	0	05/09/2024	Introduction + Administrative Remarks	
	1	10/09/2024	Halfspaces the Perceptron Algorithm	
	2	12/09/2024	Linear Regression and Convexity	
	3	17/09/2024	Maximum Likelihood Estimation	
	4	19/09/2024	k-means Clustering	
→	5	24/09/2024	k-NN Classification and Logistic Regression	
	6	26/09/2024	Hard-margin SVM	
	7	01/10/2024	Soft-margin SVM	
	8	03/10/2024	Kernel methods	
	9	08/10/2024	Decision Trees	
→	10	10/10/2024	Bagging and Boosting	
		15/10/2024	NO LECTURE - MIDTERM BREAK	
		17/10/2024	NO LECTURE- MIDTERM BREAK	
→	11	22/10/2024	Expectation Maximization Algorithm	
	12	24/10/2024	MLPs and Fully-Connected NNs	
		29/10/2024	NO LECTURE - MIDTERM EXAM	
	13	31/10/2024	Convolutional Neural Networks	
	14	05/11/2024	Recurrent Neural Networks	
	15	07/11/2024	Attention and Transformers	
	16	12/11/2024	Graph Neural Networks (Time permitting)	
→	17	14/11/2024	VAEs and GANs	
	18	19/11/2024	Flows	
	19	21/11/2024	Contrastive Learning (Time permitting)	
	20	26/11/2024	Robustness	
	21	28/11/2024	Privacy (Saber Malekmohammadi)	
	22	03/12/2024	Fairness	

Grasping MLE and its role in unsupervised learning helps you build a strong intuition for how probabilistic models work, which will also help when it comes time to deal with topics like logistic regression as well as bagging and boosting in a few weeks time. MLE in particular is the backbone of many unsupervised learning techniques, like Gaussian Mixture Models, the expectation maximization algorithm, Variational Autoencoders that we'll encounter later in the course.

My hope is that by getting comfortable with these foundational ideas now, we'll be in a much better position to understand and work with those advanced machine learning algorithms, to sort of make the leap from basic concepts to cutting-edge applications much smoother.

## Key Questions

- I. How can we represent and sample from a distribution?
- II. How do you estimate the parameters and evaluate the model?
- III. Could this also apply to supervised learning?
- IV. Summary + Housekeeping

So today we'll be focusing on answering 3 questions. First, we need to ask, **"How can we represent and sample from a distribution?"** because this forms the basis of modeling any data-generating process. From there, it's crucial to understand **"How do you estimate the parameters and evaluate the model?"**—this is where methods like Maximum Likelihood Estimation will come into play, as we'll see that it allows us to fit our models to real data. Finally, we'll contemplate whether the MLE framework **"Could also apply to supervised learning?"**.

# Lecture Objectives

At the end of the lecture, we should be able to:

- ★ Identify the probability density function and parameterization of widely used distributions and relate it to their use in constructing likelihood functions.
- ★ Construct the likelihood function for a dataset and maximize it to find the maximum likelihood estimates (MLE) of the parameters.
- ★ Define and apply information theoretic measures such as entropy and KL divergence to characterize and compare distributions.
- ★ Reformulate the linear regression objective using likelihood principles, and demonstrate that it can be viewed as a special case of MLE.

So by the end of this lecture, I'm hoping you're in a strong position to

- **Represent and Sample from Distributions:** Demonstrate how to represent probability distributions and perform sampling from these distributions to gain a foundational understanding of different types of data-generating processes.
- **Estimate Parameters Using MLE:** Apply Maximum Likelihood Estimation (MLE) to estimate the parameters of various distributions, allowing you to fit models effectively to observed data.
- **Characterize and Compare Distributions:** Use information-theoretic measures like entropy and Kullback-Leibler (KL) divergence to evaluate and compare distributions, enabling you to quantify how well different models represent the data.
- **And last but not least, apply MLE to Regression Problems:** Reformulate the linear regression problem using the MLE framework, demonstrating your ability to view regression as a special case of MLE, which should in turn enhance your understanding of the connection between probability and supervised learning.

## Key Questions

- I. How can we represent and sample from a distribution?**
- II. How do you estimate the parameters and evaluate the model?
- III. Could this also apply to supervised learning?
- IV. Summary + Housekeeping

OK so turning to our first question about how we represent data through probability distributions. A key idea in statistical modeling is that any dataset can be thought of as being generated by some underlying probability distribution. By estimating this distribution, we can understand the data and make predictions about new observations.

# The PDF of a univariate Gaussian (normal) distribution

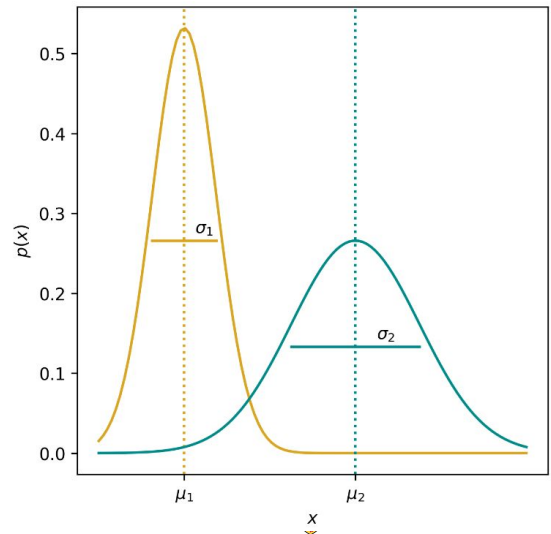
The probability density at a point  $x$  under a Gaussian distribution is given by:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

Where  $\mu$  and  $\sigma^2$  parameters referring to the mean (or center) and variance, respectively.

Probability density functions must satisfy

$$\int_{-\infty}^{\infty} p(x) = 1$$



For example, consider a simple Gaussian (normal) distribution, one of the most widely used distributions in statistics and machine learning. A Gaussian distribution is characterized by two parameters: the mean ( $\mu$ ), which indicates the center of the distribution, and the variance ( $\sigma^2$ ), which measures the spread or dispersion of the data around the mean

The probability density function (PDF) of a univariate Gaussian is given by this equation here.

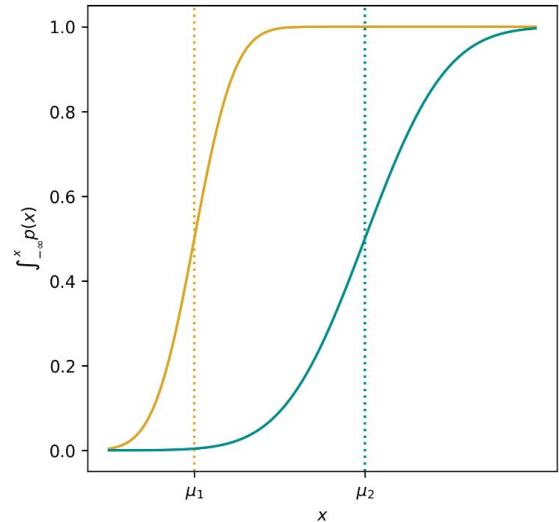
Here,  $\mu$  represents the average or expected value of the distribution, and  $\sigma^2$  measures how spread out the values are around the mean. The bell-shaped curve of the Gaussian distribution shows that values near the mean are more likely, while values further away are less likely. This shape is symmetric around the mean. One critical property of a PDF is that the total area under the curve must equal one, reflecting that the total probability of all possible outcomes is 1.



# The CDF of a univariate Gaussian (normal) distribution

Cumulative distribution function (CDF):

$$\Pr[X \leq x] = \int_{-\infty}^x p(x)$$



Closely related to the PDF is the Cumulative Distribution Function (CDF), which shows the probability that the random variable  $X$  is less than or equal to a particular value  $x$ . Mathematically, the CDF is the integral of the PDF from negative infinity to infinity.

The CDF starts near zero and gradually increases to one as you move from the left to the right of the distribution. For a Gaussian distribution, the CDF provides insights into the probability mass below a certain threshold.

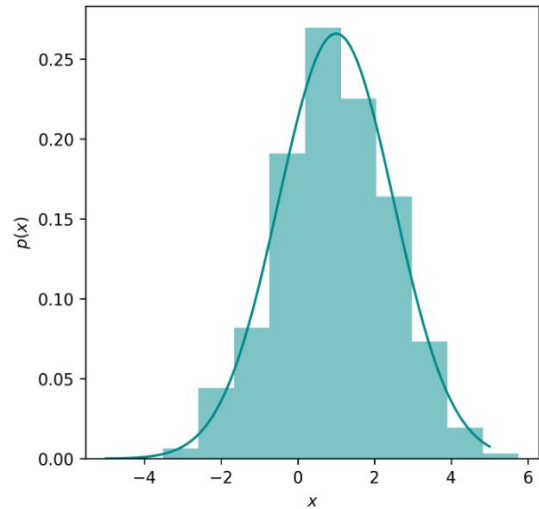
# Expectation and the first moment

We denote a continuous random variable  $X$  that follows a normal distribution with mean  $\mu$  and variance  $\sigma^2$  as

$$X \sim \mathcal{N}(\mu, \sigma^2).$$

The expectation of  $X$  is given by:

$$\begin{aligned} E[X] &= \int xp(x)dx = \mu \\ &\approx \frac{1}{n} \sum_{i=1}^n X_i \end{aligned}$$



**So in terms of notation,** we'll denote a continuous random variable denoted by the capital letter  $X$ .  $X$  follows a normal distribution with mean  $\mu$  and variance  $\sigma^2$ . The symbol  $\sim$  is used to denote that a variable “follows” or “is drawn from” the specified distribution, which in this case is a normal (Gaussian) distribution.

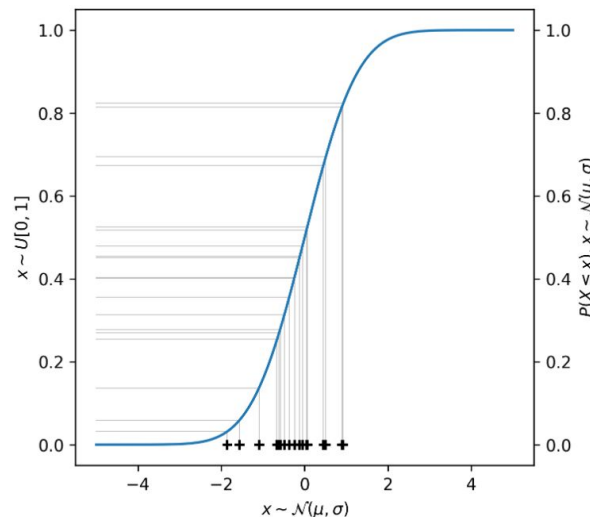
The expectation of a random variable, often referred to as the first moment, is essentially the average value that the variable takes on, weighted by its probability density. Mathematically, the expectation  $E[X]$  of  $X$  is defined by the integral over the probability density function. The integral sums over all possible values of  $X$ , effectively calculating a weighted average where each value of  $X$  is weighted by how likely it is to occur.

For practical purposes, especially when dealing with data, we often estimate the expectation using a sample mean. If we have a dataset consisting of  $n$  observations, the empirical mean provides an approximation of the expectation. This empirical mean is calculated by summing all observed values and dividing by the number of observations, capturing the average value in the dataset.

On the right side of the slide, we've got a histogram of data samples alongside the smooth curve of the probability density function (PDF) of the normal distribution. The histogram represents the observed frequency of data points within specific intervals, while the smooth curve shows the theoretical distribution of  $X$ . Conceptually, calculating the expectation can be thought of as finding the center of mass of this distribution, where each point's contribution is proportional to its probability density,

represented by the height of the PDF curve. So in this way expectation is a kind of “weighted average,” where values are weighted by their probability density. In the case of the normal distribution, the mean  $\mu$  represents this balance point where the distribution is centered.

# Inverse transform sampling from a parameterized distribution



So for a Gaussian distribution, we call the mean and the variance SUFFICIENT STATISTICS because they allow us to reproduce the characteristics of the data if they are known.

On a computer, we can do this with something called inverse transform sampling.

Suppose you have a random variable  $X$  that follows a normal distribution with mean  $\mu$  and standard deviation  $\sigma$ . The blue curve on the graph represents the cumulative distribution function (CDF) of this normal distribution. The CDF shows the probability that the random variable  $X$  takes on a value less than or equal to a certain point.

To generate samples, we generate a random number from a uniform distribution over the interval  $[0, 1]$ . This is shown on the y-axis on the left side of the plot. A uniform random variable is evenly distributed across this range, meaning every value between 0 and 1 is equally likely.

**We** use the uniform random number as an input to the inverse of the CDF, this means that we essentially “look up” the uniform value on the y-axis of the CDF curve. For example, if our uniform sample is 0.7, we trace horizontally across until we hit the CDF curve, and then we project down vertically to the x-axis to find the corresponding value of  $X$ .

The point where the line intersects the x-axis gives us a sample from the target distribution, in this case, the normal distribution. Repeating this process generates a series of samples that follow the shape of the target distribution.

The reason this works is that where the CDF curve is steeper, more uniform samples will map to closely spaced values on the x-axis, reproducing the shape of the density of the normal distribution. Essentially, the slope of the CDF determines the density of the generated samples: steeper slopes indicate regions where the normal distribution is denser (more likely to produce samples), such as near the mean. This technique is not limited to normal distributions—it can be applied to any distribution, making it a versatile tool in simulations, statistical modeling, and various computational applications.

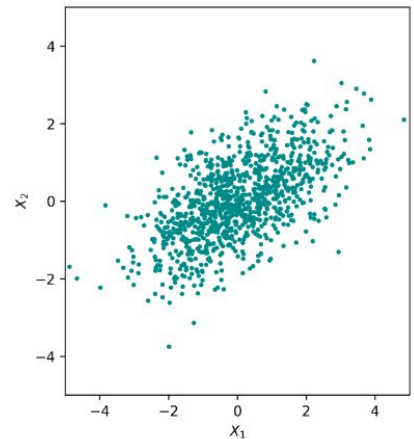
# Covariance: generalization for multidimensional data

A random vector  $\mathbf{X}, \mathbf{X} \in \mathbb{R}^d$  has covariance matrix

$$\begin{aligned}\Sigma &= \text{Cov}(\mathbf{X}) \\ &= E[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T] \\ &= \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \dots & \text{Cov}(X_1, X_d) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \dots & \text{Cov}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_d, X_1) & \text{Cov}(X_d, X_2) & \dots & \text{Var}(X_d) \end{bmatrix},\end{aligned}$$

which is symmetric and positive semidefinite.

$$\begin{aligned}\mathbf{X} &\sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \\ &\sim \frac{1}{\sqrt{2\pi^d \det(\Sigma)}} e^{-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{X} - \boldsymbol{\mu})}\end{aligned}$$



We can extend our understanding of the univariate Gaussian (normal) distribution to the multivariate case, where our random variable is a vector rather than a single value. We denote random vectors with bold capital letters, such as  $\mathbf{X}$ , representing a vector that exists in a multi-dimensional space of real numbers.

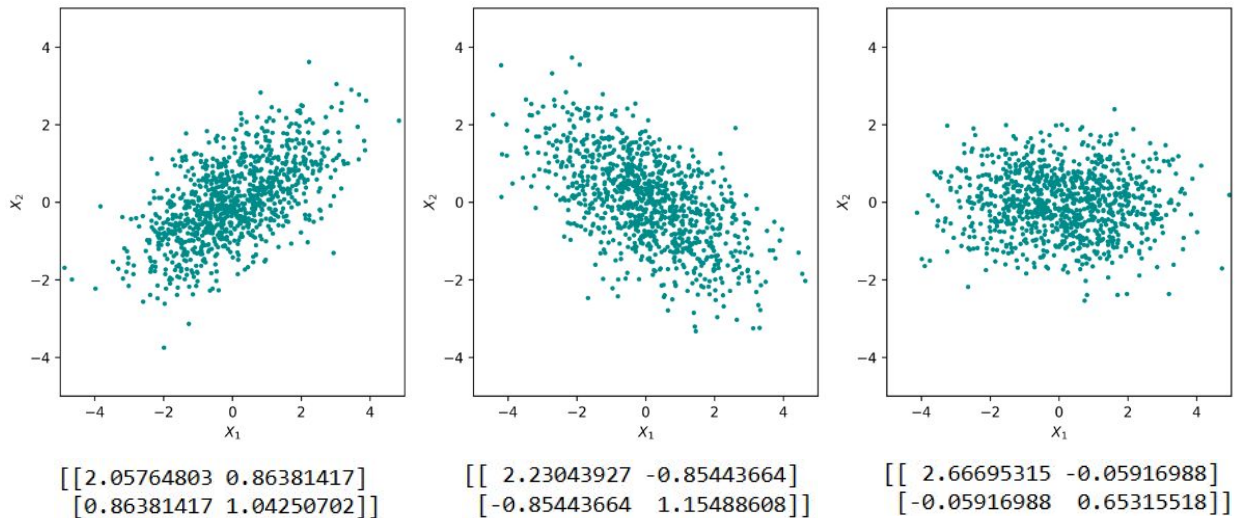
In this multivariate context, each component of the vector has its own mean and variance, and the relationships between these components are captured by the covariance matrix, denoted as  $\Sigma$ . This matrix is crucial because it defines the spread and orientation of the data in multidimensional space and captures how the variables interact with each other.

The diagonal elements of the covariance matrix represent the variances of individual components, while the off-diagonal elements represent covariances between pairs of variables, indicating how much two variables change together—positive values suggest they increase together, while negative values suggest one increases as the other decreases.

In a multivariate normal distribution, our random vector follows a distribution characterized by a mean vector and a covariance matrix, replacing the single variance seen in the univariate case. Understanding the covariance structure is vital because it captures dependencies and linear relationships between variables, which are often critical when modeling real-world phenomena. For example, in the scatter plot on the slide, you can see that a positive covariance results in an elliptical data shape tilted in the direction of the relationship, demonstrating that as one variable

increases, the other tends to do the same, showing the importance of visualizing and interpreting covariance in multivariate data.

## Covariance matrix examples



So here are a few examples.

In the first plot on the left, we observe a covariance matrix with positive diagonal values, showing variability in both directions, and a positive off-diagonal value, indicating a positive relationship between the two variables; as one variable increases, the other tends to increase as well, reflected in the upward slope of the scatter plot. The matrix is symmetric, which is a key property of covariance matrices.

In the second plot in the middle, the covariance matrix still shows positive variances but has negative off-diagonal values, indicating a negative covariance; as one variable increases, the other tends to decrease, which is evident in the downward slope of the scatter plot, highlighting an inverse relationship.

In the third plot on the right, the covariance matrix shows variances along the diagonal with minimal covariance between the variables, as the off-diagonal values are close to zero. This near-zero covariance is visually confirmed by the scatter plot, where the points are scattered without a clear trend, showing almost no correlation.

So the covariance matrix captures individual variability AS WELL AS the strength and direction of relationships between variables.



## Key Questions

- I. How can we represent and sample from a distribution?
- II. How do you estimate the parameters and evaluate the model?**
- III. Could this also apply to supervised learning?
- IV. Summary + Housekeeping

OK so now that we have a probability density function identified - which is the hypothesis class to use the language we've been working with in this course so far, let's figure out how we can estimate the parameters.

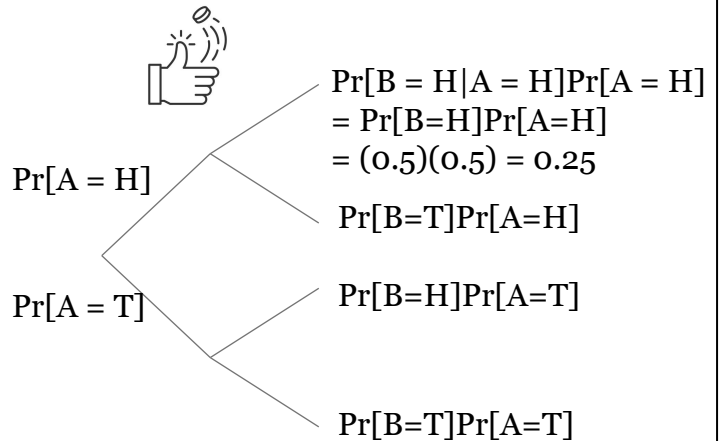
## The joint distribution describes behavior of combined densities

Suppose we have two independent univariate random variables,  $X_1$  and  $X_2$ . Their joint probability distribution,

$$p(X_1, X_2) = p(X_2 | X_1)p(X_1),$$

describes the probability of the variables occurring together. If variables  $X_1$  and  $X_2$  are *independent*, this simplifies to:

$$p(X_1, X_2) = p(X_2)p(X_1).$$



In many cases, we deal with multiple random variables simultaneously, which requires understanding their joint distribution. The joint probability distribution describes the probability of observing a combination of values for these variables. For example, the joint distribution of two variables  $X_1$  and  $X_2$  can be decomposed into the conditional probability of  $X_2$  given  $X_1$  and the MARGINAL probability of  $X_1$ . If  $X_1$  and  $X_2$  are independent, then the joint probability simplifies to a simple product.

An example of joint probability involving independent events is flipping a coin. Each flip is an independent event, meaning that the outcome of one flip does not influence the outcome of the next. For a single coin flip, the probability of getting heads (H) is 50%, or 0.50. Similarly, the probability of getting tails (T) is also 50%. The independence of these events means that each flip has the same probability regardless of previous outcomes. Even if you flipped five tails in a row, the probability of heads on the next flip remains 50%.

So if we were to consider two events: Event A: as the first coin flip results in heads (H), and event B: as the second coin flip also results in heads (H). Then we could find the joint probability of both events occurring by simply multiplying the probabilities of each individual event:  $P(A) \times P(B) = 0.5 \times 0.5 = 0.25$ . This multiplication works because the events are independent, meaning the joint probability is simply the product of the probabilities of each event.

## Example: estimating the parameters of a distribution

Suppose we have a set of  $n$  observations  $\{x_1, x_2, \dots, x_n\}$ ,  $x_i \in \mathbb{R}$ , and we assume that they are realizations of a univariate Gaussian (normal) distribution with some mean  $\mu$  and variance  $\sigma^2$ :

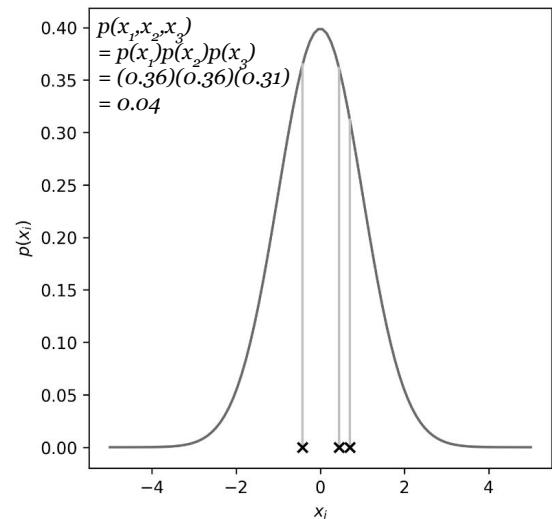
$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$

The probability density for each observation  $x_i$  is

$$p(x_i | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i - \mu)^2},$$

with joint density

$$\begin{aligned} p(\mathbf{x} | \mu, \sigma^2) &= \prod_{i=1}^n p(x_i | \mu, \sigma^2) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i - \mu)^2}. \end{aligned}$$



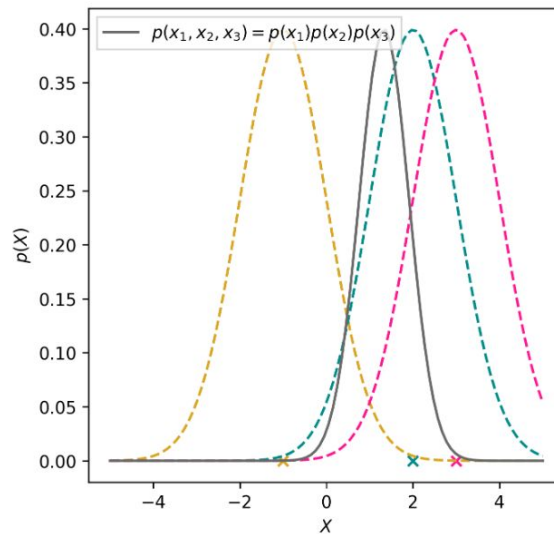
The concept of joint probability can be extended to continuous random variables.

Imagine we have a set of  $n$  observations, represented as  $\{x_1, x_2, \dots, x_n\}$  drawn from the real number line, where each observation is a scalar value. We assume these observations are independent realizations from a univariate Gaussian distribution with some unknown mean and variance. For each observation, the probability density is given by the standard Gaussian density function, which indicates how likely it is to observe a specific value given the mean and variance.

To find the joint density of all observations, we multiply the individual densities together. This is the meaning of this intimidatingly large  $\pi$  symbol which denotes a product over the elements. It's similar to the summation operation but we do the product. This multiplication combines the likelihood of each observation given the distribution's parameters into an overall joint density.

For example, if we have three observations—let's say 0.36, 0.36, and 0.31 as approximate densities—the joint density of these observations would be their product, approximately 0.04. It's important to remember that this value represents a density, not a probability, indicating the concentration of probability mass around these values.

# Visual interpretation of joint probability distribution



For a visual interpretation of how joint probability distributions work, particularly in the context of independent variables, we can think about it as dropping a gaussian at every observation.

On the graph, we have three individual density functions, each represented by a different dashed line. The **yellow dashed line** represents the density function of the first observation, the **pink dashed line** represents the density function of a second observation, and the **blue dashed line** represents the density function of a third observation.

The **solid gray line** represents the joint density of the three observations combined, which is calculated as the product of the individual densities. Because these observations are independent, multiplying the individual densities accurately reflects the likelihood of all three occurring together.

Notice how the solid gray line is narrower compared to each of the individual density curves. This is because, when you multiply densities, the estimate of the mean gets more and more certain.

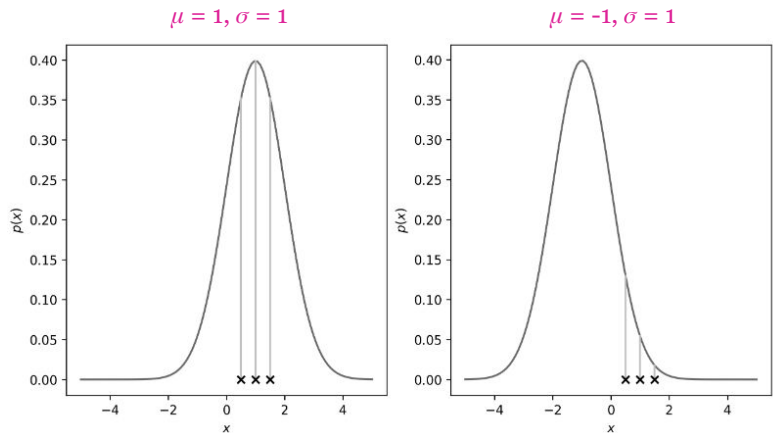
# Likelihood considers the probability of parameters, given data

Joint density:

$$p(\mathbf{x} \mid \mu, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} (x_i - \mu)^2}$$

Likelihood:

$$\mathcal{L}(\mu, \sigma^2 \mid \mathbf{x}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} (x_i - \mu)^2}$$



A closely related concept to joint density is LIKELIHOOD. While these concepts are mathematically similar, their interpretations and roles are different, particularly when we think about how we use data to inform our understanding of a model. So this equation here expresses the joint density, which represents the probability density of observing a set of data points given specific parameters of the distribution, such as the mean and variance. This density is considered as a function of the data points - in pink - given the fixed parameters of the distribution.

The likelihood, denoted by a scripted  $\mathcal{L}$ , is very similar to the joint density but with a critical difference in interpretation.

Instead of considering the joint density as a function of the observed data given the parameters, the likelihood function treats the parameters of the distribution (mean and variance) as the variables of interest. The likelihood function uses the same product formulation as the joint density, but is viewed as a function of the parameters, not the data.

So in essence, the JOINT DENSITY answers: "How likely is it to observe this data given specific parameters of the model?", whereas the **likelihood** flips this around and asks: "How likely are these parameter values given the observed data?" This subtle shift transforms the problem from one of prediction to inference.

So to consider a practical example, suppose you have three data points: 0.75, 1, and 1.75.

And you are debating between two possible guesses for the mean of the distribution:

one guess is that the mean  $\mu = 1$ , and the other guess is that the mean  $\mu = -1$ . And someone told you that the variance was one.

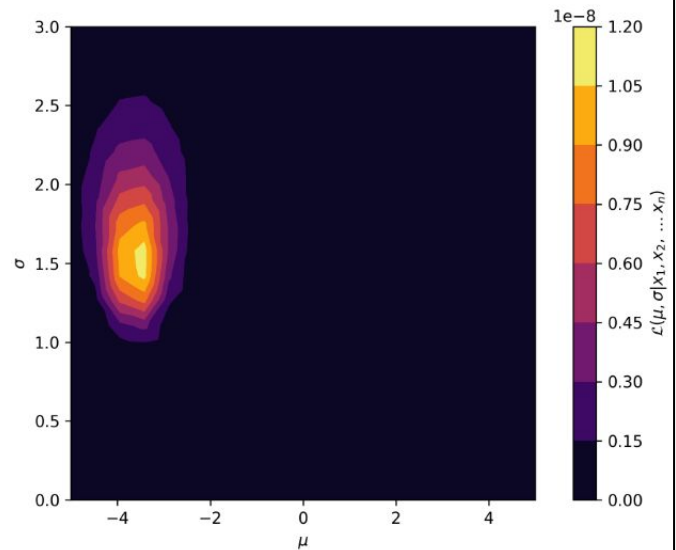
Using the likelihood function, we calculate the joint density of the data under each assumption:

- If you assume a mean of 1, the data fits the model well, and the likelihood function returns a relatively high value, indicating that this model is plausible given the observed data.
- On the other hand, if you assume a mean of -1, the observed data points do not fit well, resulting in a low likelihood, indicating that this set of parameters is less plausible given the data.

The concept of likelihood is central to statistical inference because it provides a framework for evaluating how well different parameter sets explain the data. It turns data into evidence, guiding us toward the most reasonable parameter estimates given what we observe. For continuous random variables, likelihood does not describe the PROBABILITY of parameters, because we're dealing with densities, but it provides a relative measure of how well each set of parameters explains the data

# Finding the Maximum Likelihood Estimate (MLE)

$$\mathcal{L}(\mu, \sigma^2 | \mathbf{x}) = \prod_{i=1}^n \frac{1}{\sigma^2} e^{-\frac{1}{2\sigma^2} (x_i - \mu)^2}$$



Alright, so now, we're ready to get at the central concept of today's lecture, which is Maximum Likelihood Estimation or (MLE), which will allow us to move beyond comparing between discrete guesses to find the best parameters given the data. As the name suggests, it does this by maximizing the likelihood function.

So let's work through a specific example, and apply MLE to estimate the parameters of a Gaussian distribution. Suppose we have a dataset consisting of  $n$  observations and we assume that these data points are drawn from a Gaussian distribution with unknown mean  $\mu$  and variance  $\sigma^2$ .

ANYONE have any idea how we could proceed?

In practice, we often work with the log-likelihood function,  $\log L(\theta)$ , instead of the likelihood itself. Taking the logarithm turns the product into a sum, which simplifies differentiation and maximization. Also, as Log is a monotonic function, it doesn't change location of the optimum. In other words, maximizing the log-likelihood is equivalent to maximizing the likelihood because the logarithm is a monotonic function, meaning it preserves the ordering of values.

To find the maximum likelihood estimates (MLEs) of  $\mu$  and  $\sigma^2$ , we first take partial derivatives of the log-likelihood with respect to each parameter. For the mean, we get this expression here, and for the variance, we get this down here. I won't prove the partial derivatives with respect to either of the parameters, but that could be a good exercise to practice your calculus!

We then set them to zero to find out where the likelihood is at its maximum. When we do that, the best estimate of the mean is how we would calculate it empirically. Excluding the trivial possibility of a zero variance distribution, we are left with this factor in the brackets here, showing that the MLE estimate of the variance recovers that of the standard definition of the variance. and the best estimate of variance reflects the average squared deviation from the mean.

So for example, if the true parameters of a gaussian were  $\mu = -3$  and  $\sigma$  of 1.5, the likelihood over the parameter space could be depicted by this heatmap, which has a clear, single peak. TEST -- is this function convex? No, it is concave.

So yeah, to summarize before we move on, MLE finds the parameters that make the observed data most probable under the assumed model.



# Entropy characterizes the uncertainty in a distribution

$$\begin{aligned}h(X) &= E[-\log p(X)] \\&= - \sum_{i=1}^n \Pr[x_i] \log \Pr[x_i] \text{ (Discrete Random Variable)} \\&= - \int_{\mathcal{X}} p(x) \log p(x) dx \text{ (Continuous Random Variable)}\end{aligned}$$

Alright so we've used MLE to essentially find the most probable explanation for our data, at least within the constraints of our chosen probability density function, which was a gaussian in this case. But once we have our model, a natural question arises: how does it compare to the true generating process?

And so one way to answer that question meaningfully for probability distributions involves a concept called ENTROPY. I'll first define and give some examples of entropy, and then show how we use it to answer the question.

Entropy is a fundamental concept in information theory and statistics, which quantifies the UNCERTAINTY or RANDOMNESS inherent in a distribution -- It tells us unpredictable or the outcomes are, how SPREAD OUT we would expect them to be. For example, in a perfectly predictable system, entropy would be low, whereas a highly uncertain system would have high entropy.

## Mathematical Definition of Entropy:

The entropy of a random variable  $X$  is denoted as  $h(X)$ . It is mathematically defined as the expected value of the negative logarithm of the probability density or mass function:  $h(X) = E[-\log p(X)]$

## Discrete Random Variables:

For a discrete random variable  $X$ , entropy is computed using a summation, over all

possible outcomes of  $X$ .  $\Pr[x_i]$  is the probability of the discrete outcome  $x_i$ , so each possible outcome contributes to the overall uncertainty of the distribution.

**Continuous Random Variables:**

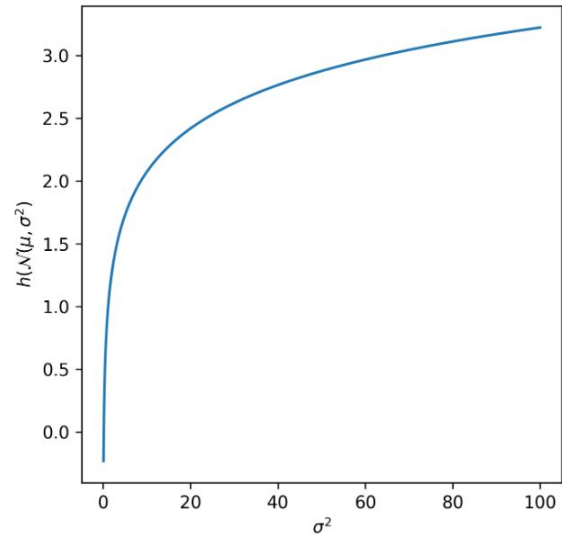
For a continuous random variable, the entropy is defined using an integral:

$h(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx$  over the domain of  $X$ , accounting for the continuous nature of the variable.

Here,  $p(x)$  represents the probability density function (PDF) of the random variable  $X$ .

## Example: Entropy of a Gaussian (normal) distribution

$$\begin{aligned}h(X) &= E[-\log p(X)] \\&= - \int_{\mathcal{X}} p(x) \log p(x) dx \\h(X \sim \mathcal{N}(\mu, \sigma^2)) &= \frac{1}{2} \log[2\pi e \sigma^2]\end{aligned}$$



The meaning of entropy is probably easiest to grasp with a few examples.

For example, if the continuous random variable  $X$  follows a normal distribution with mean and variance  $\sigma^2$ , the entropy has this closed-form expression

This shows that the entropy of a Gaussian distribution depends only on the variance and not on the mean. Since the mean represents the center of the distribution and does not affect the spread, it does not influence the level of uncertainty. In terms of the effect of sigma on entropy, we can see that entropy increases logarithmically with the variance. Graphically, this means that entropy starts near zero when the variance is close to zero and then increases monotonically as sigma grows. The curve shows that a small change in variance has a large effect on entropy initially, but that this tapers off as variance grows.

## Example: Entropy of a Bernoulli random variable

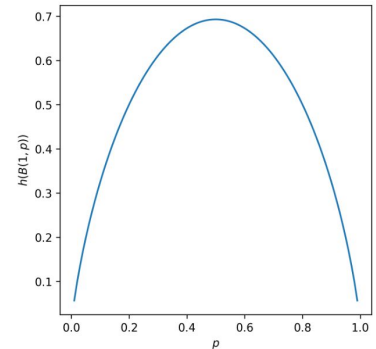
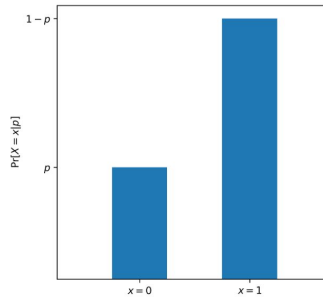
Bernoulli random variable:

$$\Pr[X = x] = \begin{cases} p & \text{if } x = 1, \\ 1 - p & \text{if } x = 0, \end{cases}$$

where  $0 \leq p \leq 1$ .

Entropy:

$$\begin{aligned} H(X) &= - \sum_{i=1}^n \Pr[x_i] \log \Pr[x_i] \\ &= - [P(X=1) \log P(X=1) + P(X=0) \log P(X=0)] \\ &= - [p \log p + (1-p) \log(1-p)] \end{aligned}$$



As another example, we could consider a discrete random variable: the Bernoulli random variable.

A Bernoulli random variable models a simple binary outcome, such as a coin flip, where there are only two possible outcomes: if  $x = 1$  with probability  $p$ , then the probability of  $x = 0$  is  $1-p$ . And of course, as where  $0 \leq p \leq 1$ . So this bar chart shows the probabilities for the two outcomes:  $X=0$  and  $X=1$ . The height of each bar corresponds to the probability  $p$  or  $1-p$ . As you adjust  $p$ , the heights of these bars will change, but they would always sum to one.

Since the Bernoulli variable only has two possible outcomes, the formula expands to this simple expression. Substituting the probabilities for each outcome, we get:  $H(X) = - [p \log p + (1-p) \log(1-p)]$ .

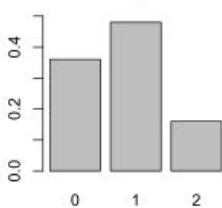
The curve on the bottom right here shows the entropy of the Bernoulli random variable as a function of  $p$ . We can see that the entropy of the curve has a peak at  $p=0.5$ . This peak represents the maximum uncertainty—when each outcome (0 or 1) is equally likely, the system is most unpredictable. As  $p$  approaches 0 or 1, the entropy decreases toward zero because one outcome becomes almost certain, and there is little uncertainty left; Entropy decreases as one outcome becomes more predictable. So in contrast to the Gaussian random variable, the entropy of the Bernoulli for its parametrization does not produce a monotonic function.

## Kullback-Leibler divergence measures dissimilarity between a reference and model distribution (aka relative entropy)

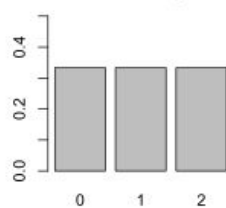
$$D_{\text{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \text{ (Discrete Random Variable)}$$

$$D_{\text{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \text{ (Continuous Random Variable)}$$

**Distribution P**  
Binomial with  $p = 0.4$ ,  $N = 2$



**Distribution Q**  
Uniform with  $p = 1/3$



	P	Q	Plog(P/Q)
X = 0	0.36	$\approx 0.33$	$\approx 1.08$
X = 1	0.48	$\approx 0.33$	$\approx 1.44$
X = 2	0.16	$\approx 0.33$	$\approx -0.11$
			$D_{\text{KL}} \approx 0.085$

Source: Wikipedia

So entropy measures the uncertainty within a single distribution, but what we really want to evaluate the MLE result is a way to describe how well one distribution matches another -- specifically, how well our model fits the true generating distribution?

So for this, we can calculate the RELATIVE ENTROPY, which is called the Kullback-Leibler or KL divergence. This is a measure of how one probability distribution DIVERGES from a second, REFERENCE probability distribution.

Specifically, if we denote the true or reference distribution P which represents the real world or data-generating process, and a model distribution Q, which could be the result of MLE, for a discrete random variable, KL divergence is defined as the product of the probability mass for an outcome under the reference, times the log of the RATIO of the probability mass under the reference to the model, over all possible values of x. It's effectively the SURPRISE we would expect to be by an outcome, if we used Q as a model instead of the actual distribution is P. KL divergence is defined similarly for continuous random variables, computing the integral over the domain of the probability density functions.

In the seminal paper -- and also available on Wikipedia -- Kullback gave this simple example for a discrete probability distribution. T

The true distribution is  $P$  which has unequal masses for each of the 3 outcomes, and we want to compare a naive model  $Q$  which assumes that all outcomes are equally likely.

So to do that, we compute  $P \log P$  over  $Q$ , for each outcome, and then sum the results.

If get a question about this.

- KL div is not a true distance because it is not symmetric
- Something else you can do is kolmogorov-smirnov distance - compares CDF. rough for large dimensions
- KL divergence of a gaussian will not go into

## Key Questions

- I. How can we represent and sample from a distribution?
- II. How do you estimate the parameters and evaluate the model?
- III. Could this also apply to supervised learning?**
- IV. Summary + Housekeeping

Now that we've explored Maximum Likelihood Estimation, it's important to highlight that this framework is not just confined to unsupervised learning or purely probabilistic contexts- it's a very general method of parameter estimation. When we train these models, we are essentially finding the parameters that maximize the likelihood of the observed labeled data.

# Interpreting the linear regression problem with MLE

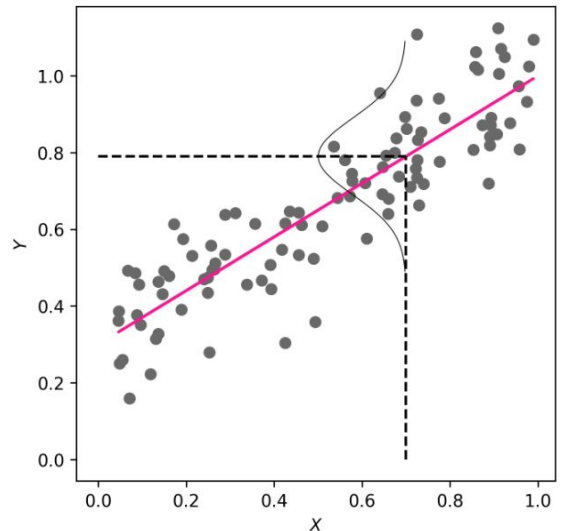
Consider a random variable  $Y$  that follows a normal distribution with mean  $\mu = w^T X$ , where  $X$  is another random variable, and variance  $\sigma^2$ :

$$Y \sim \mathcal{N}(w^T X, \sigma^2)$$
$$y_i = w^T x_i + \mathcal{N}(0, \sigma^2)$$

c.f.  $X \sim \mathcal{N}(\mu, \sigma^2)$

$$\mathcal{L}(\mathbf{y} \mid \mathbf{x}, w, \sigma^2) = \prod_{i=1}^n p(y_i \mid x_i, w, \sigma^2)$$

$$\mathcal{L}(\mathbf{z} \mid \mathbf{A}, w, \sigma^2) = \prod_{i=1}^n p(z_i \mid a_i, w, \sigma^2)$$



To see how this works, we can revisit the linear regression problem...

So how can we interpret linear regression through the lens of Maximum Likelihood Estimation (MLE)? Instead of just seeing the outcome  $Y$  as a set of fixed values, we can think of  $Y$  as a random variable that follows a normal distribution. But now the mean of this distribution isn't just a constant—it's a function of another random variable,  $X$ , and a parameter vector,  $w$ . Specifically, we model the mean as the inner product of  $w$  and  $X$ , which aligns perfectly with our linear regression setup. So, we're essentially saying that each observed value of  $Y$  is generated from this normal distribution centered around our predicted value, plus some normally distributed noise that accounts for the variability around the line.

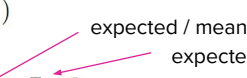
To move forward with this framework, we can take the next step by constructing a likelihood function that captures the probability of observing our data given the parameters  $w$  and the variance  $\sigma^2$ . To align with the notation we had last class - we'll use  $\mathbf{z}$  to denote the set of outcomes  $Y$ , and  $\mathbf{A}$  the  $n \times d$  matrix of feature vectors.

Our goal with MLE is to maximize this likelihood, which is the same as finding the set of parameters that make our observed data most probable under this model. This process isn't just about fitting a line—it's about understanding the data through the statistical structure of our assumptions.



## Maximizing the likelihood with respect to the parameters $w$

$$\begin{aligned}
 \mathcal{L}(z \mid A, w, \sigma^2) &= \prod_{i=1}^n p(z_i \mid a_i, w, \sigma^2) \\
 &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(z_i - w^T a_i)^2}
 \end{aligned}$$



$$\begin{aligned}
 \log \mathcal{L}(z \mid A, w, \sigma^2) &= -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} (z - Aw)^T (z - Aw) \\
 \operatorname{argmax}_w \log \mathcal{L} &= \operatorname{argmin}_w -\log \mathcal{L} \\
 -\log \mathcal{L} &= \frac{n}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (z - Aw)^T (z - Aw) \\
 \frac{\partial -\log \mathcal{L}}{\partial w} &= \frac{1}{2\sigma^2} (z - Aw)^T (z - Aw) \\
 &= \frac{1}{2\sigma^2} (w^T A^T A w - 2A^T w^T z + z^T z) \\
 \frac{\partial -\log \mathcal{L}}{\partial w} &= 0 \\
 \implies w &= (A^T A)^{-1} A^T z
 \end{aligned}$$

To maximize the likelihood with respect to the parameter vector  $w$ , our goal is to find the set of parameters that make the observed data most probable under our assumed model.

This top equation here specifies the likelihood function for the gaussian distribution.

Ooops, I see a mistake. Does anyone see what I did wrong here?

We first take the logarithmic transform as we did before, which simplifies our calculations since working with logarithms turns the product of probabilities into a sum, making the math more manageable. Our objective is to maximize this expression, which is equivalent to minimizing the negative log of the likelihood function. So we're just flipping the sign.

Starting from the negative log-likelihood expression, we take its partial derivative with respect to  $w$ . The first term drops off because it does not depend on  $w$ , leaving us with a term involving the squared difference between the observed values and the predicted values, scaled by the variance. We then set this derivative to zero to find the critical points, which leads us to the equation that recovers the parameter vector  $w$ .

Remarkably, this expression is identical to the least squares solution for linear regression, showing that under the assumption of normally distributed errors, maximizing the likelihood gives us the same result as minimizing the least squares loss. This connection highlights how MLE and least squares are fundamentally linked. It provides a deeper statistical rationale for the regression solution we encountered

last week.

## What about the variance?

$$\begin{aligned}\frac{\partial -\log \mathcal{L}}{\partial \sigma} &= \frac{\partial}{\partial \sigma} \left[ \frac{n}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (\mathbf{z} - A\mathbf{w})^T (\mathbf{z} - A\mathbf{w}) \right] \\ &= -\frac{n}{\sigma} + \frac{1}{\sigma^3} (\mathbf{z} - A\mathbf{w})^T (\mathbf{z} - A\mathbf{w})\end{aligned}$$

$$\frac{\partial -\log \mathcal{L}}{\partial \sigma} = 0$$

$$\Rightarrow \frac{n}{\sigma} = \frac{1}{\sigma^3} (\mathbf{z} - A\mathbf{w})^T (\mathbf{z} - A\mathbf{w})$$

$$\begin{aligned}\Rightarrow \sigma^2 &= \frac{1}{n} (\mathbf{z} - A\mathbf{w})^T (\mathbf{z} - A\mathbf{w}) \\ &= \frac{1}{n} \sum_{i=1}^n (a_i \mathbf{w} - z_i)^2\end{aligned}$$

If  $y = \log[f(x)]$ , then

$$\frac{dy}{dx} = \frac{1}{f(x)} f'(x)$$

$$\Rightarrow \frac{d}{d\sigma} \left( -\frac{n}{2} \log(2\pi\sigma^2) \right) = -\frac{n}{2} \frac{1}{2\pi\sigma^2} 4\pi\sigma = -\frac{n}{\sigma}$$

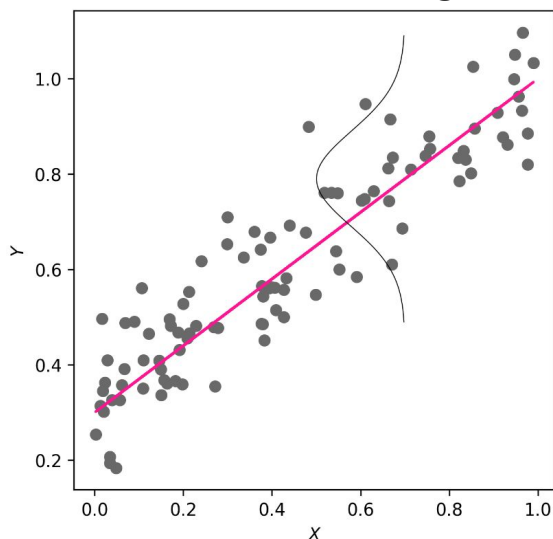
After having found the parameter vector  $\mathbf{w}$ , we can turn our attention to the variance by taking the partial derivative of the negative log-likelihood with respect to  $\sigma$ .

The first term of the negative log-likelihood, involving the logarithm of the variance term, simplifies due to the rules of differentiation of logarithmic functions that I've shown on the right here, and contributes a negative term proportional to the number of observations divided by  $\sigma$ .

The second term, involving the squared differences between observed and predicted values, contributes another component dependent on the variance and the residuals. By setting this derivative equal to zero, we isolate the variance term, leading us to an expression where the estimated variance is equal to the average of the squared residuals, or the mean squared error, between the observed values and the predicted values from our linear model.

This result tells us that the MLE estimate of variance captures the average deviation of the observed data around our fitted regression line, reinforcing that variance estimation is tightly linked to how well our model fits the data.

## Under assumption of normally distributed errors, least-squares regression can be viewed as maximizing likelihood



So what does this mean? Well, under the assumption of normally distributed errors, least-squares regression is essentially maximizing the likelihood of the observed data. When we perform linear regression, we usually think about finding the line that minimizes the sum of squared errors between our observed data points and the predicted values on the line. However, this can also be viewed from a probabilistic perspective: we assume that the errors—or residuals—are normally distributed around the regression line with constant variance.

By modeling this, we set up a framework where the probability of observing our data is maximized when the predicted values from the regression line align closely with the actual observed values, accounting for the normal error distribution. In other words, the regression line that minimizes the sum of squared errors is also the one that maximizes the likelihood of the data under the normal error assumption.

## Key Questions

- I. How can we represent and sample from a distribution?
- II. How do you estimate the parameters and evaluate the model?
- III. Could this also apply to supervised learning?
- IV. Summary + Housekeeping**

We'll encounter maximum likelihood estimation as a very general framework for parameter estimation

# Lecture Objectives

At the end of the lecture, we should be able to:

- ★ Identify the probability density function and parameterization of widely used distributions and relate it to their use in constructing likelihood functions.
- ★ Construct the likelihood function for a dataset and maximize it to find the maximum likelihood estimates (MLE) of the parameters.
- ★ Define and apply information theoretic measures such as entropy and KL divergence to characterize and compare distributions.
- ★ Reformulate the linear regression objective using likelihood principles, and demonstrate that it can be viewed as a special case of MLE.

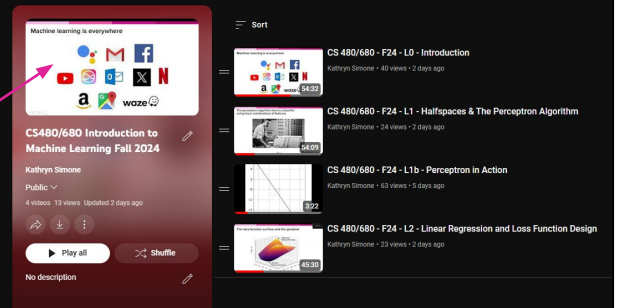
# Last lecture's slides have been updated

## Errata

- On the slide titled, “Equivalent notation of loss to leverage the gradient” a reference was made to constructing a loss *matrix*. This has been corrected to “We can write the total loss,  $L$  as ...”.
- A previous version of the slide deck had a slide titled “If a function is convex, its second derivative is positive.” This statement was incorrect because a convex function requires the second derivative to be non-negative ( $\geq 0$ , not strictly positive,  $> 0$ ). Additionally, a function may be convex but not necessarily everywhere twice differentiable. The corrected statement reads: “A twice-differentiable function of more than one variable is convex *if and only if* its Hessian is everywhere positive semidefinite,” emphasizing that this condition must hold for all points in the function’s domain. This clarification highlights that having a positive semidefinite Hessian matrix everywhere in the domain is a sufficient and necessary condition for convexity in the context of twice-differentiable functions. However, convexity as a broader property does not inherently require the function to be twice differentiable or the Hessian to be defined everywhere.

## Lecture videos linked from course homepage, playlist also on YouTube

LECTURE	TITLE	MATERIALS	SUPPLEMENTARY READINGS
0	Logistics & Introduction	<a href="#">Slides</a> <a href="#">Video Lecture</a>	N/A
1	Halfspaces & The Perceptron Algorithm	<a href="#">Slides</a> <a href="#">Video Lecture</a> <a href="#">Perceptron</a> <a href="#">Video</a>	UML Section 9.1 ESL Section 4.5 <a href="#">Yaoliang Yu's Lecture Notes</a> <a href="#">Varun Kanade's Lecture Notes</a>
2	Linear Regression & Loss Function Design	<a href="#">Slides</a> <a href="#">Video Lecture</a>	





	Lecture	Date	Topics	
	0	05/09/2024	Introduction + Administrative Remarks	
	1	10/09/2024	Halfspace the Perceptron Algorithm	
	2	12/09/2024	Linear Regression and Convexity	
	3	17/09/2024	Maximum Likelihood Estimation	
	4	19/09/2024	k-means Clustering	
	5	24/09/2024	k-NN Classification and Logistic Regression	
	6	26/09/2024	Hard-margin SVM	
	7	01/10/2024	Soft-margin SVM	
	8	03/10/2024	Kernel methods	
	9	08/10/2024	Decision Trees	
	10	10/10/2024	Bagging and Boosting	
		15/10/2024	NO LECTURE - MIDTERM BREAK	
		17/10/2024	NO LECTURE- MIDTERM BREAK	
	11	22/10/2024	Expectation Maximization Algorithm	
	12	24/10/2024	MLPs and Fully-Connected NNs	
		29/10/2024	NO LECTURE - MIDTERM EXAM	
	13	31/10/2024	Convolutional Neural Networks	
	14	05/11/2024	Recurrent Neural Networks	
	15	07/11/2024	Attention and Transformers	
	16	12/11/2024	Graph Neural Networks (Time permitting)	
	17	14/11/2024	VAEs and GANs	
	18	19/11/2024	Flows	
	19	21/11/2024	Contrastive Learning (Time permitting)	
	20	26/11/2024	Robustness	
	21	28/11/2024	Privacy (Saber Malekmohammadi)	
	22	03/12/2024	Fairness	

	Lecture	Date	Topics	
<b>Non-parametric methods:</b> <ul style="list-style-type: none"> <li>- Kernel Density Est.</li> <li>- K-means</li> <li>- Clustering</li> <li>- K-NN Classification</li> </ul>	0	05/09/2024	Introduction + Administrative Remarks	
	1	10/09/2024	Halfspaces the Perceptron Algorithm	
	2	12/09/2024	Linear Regression and Convexity	
	3	17/09/2024	Maximum Likelihood Estimation	
	4	19/09/2024	Non-parametric Methods	
	5	24/09/2024	Logistic Regression	
	6	26/09/2024	Hard-margin SVM	
	7	01/10/2024	Soft-margin SVM	
	8	03/10/2024	Kernel methods	
	9	08/10/2024	Decision Trees	
	10	10/10/2024	Bagging and Boosting	
		15/10/2024	NO LECTURE - MIDTERM BREAK	
		17/10/2024	NO LECTURE - MIDTERM BREAK	
	11	22/10/2024	Expectation Maximization Algorithm	
	12	24/10/2024	MLPs and Fully-Connected NNs	
		29/10/2024	NO LECTURE - MIDTERM EXAM	
	13	31/10/2024	Convolutional Neural Networks	
	14	05/11/2024	Recurrent Neural Networks	
	15	07/11/2024	Attention and Transformers	
	16	12/11/2024	Graph Neural Networks (Time permitting)	
	17	14/11/2024	VAEs and GANs	
	18	19/11/2024	Flows	
	19	21/11/2024	Contrastive Learning (Time permitting)	
	20	26/11/2024	Robustness	
	21	28/11/2024	Privacy (Saber Malekmohammadi)	
	22	03/12/2024	Fairness	

# On the horizon

Questions?  
Ask Saber! :)

**Table 2: Grading Scheme**

Assessment	Assessment Date	Weighting (CS480)	Weighting (CS680)
→ Assignment 1	September 27	7.5%	7.5%
Assignment 2	October 14	7.5%	7.5%
Assignment 3	November 8	7.5%	7.5%
Assignment 4	November 22	7.5%	7.5%
Exams			
Midterm	October 29	30%	15%
Final	TBD	40%	30%
Project (CS 680 only)			
Thursday! → Pitch	September 19	N/A	2%
Proposal	October 8	N/A	8%
Report	December 3	N/A	15%
Total		100%	100%

