MIDDLE EAST TECHNICAL UNIVERSITY

MMI714 Generative Models for Multimedia

The Mathematical Background

Welcome

This is MMI714

“Generative Models for Multimedia”

This week we will review the mathematical concepts required for this course.

This week, we will learn to speak the language of generative modelling!

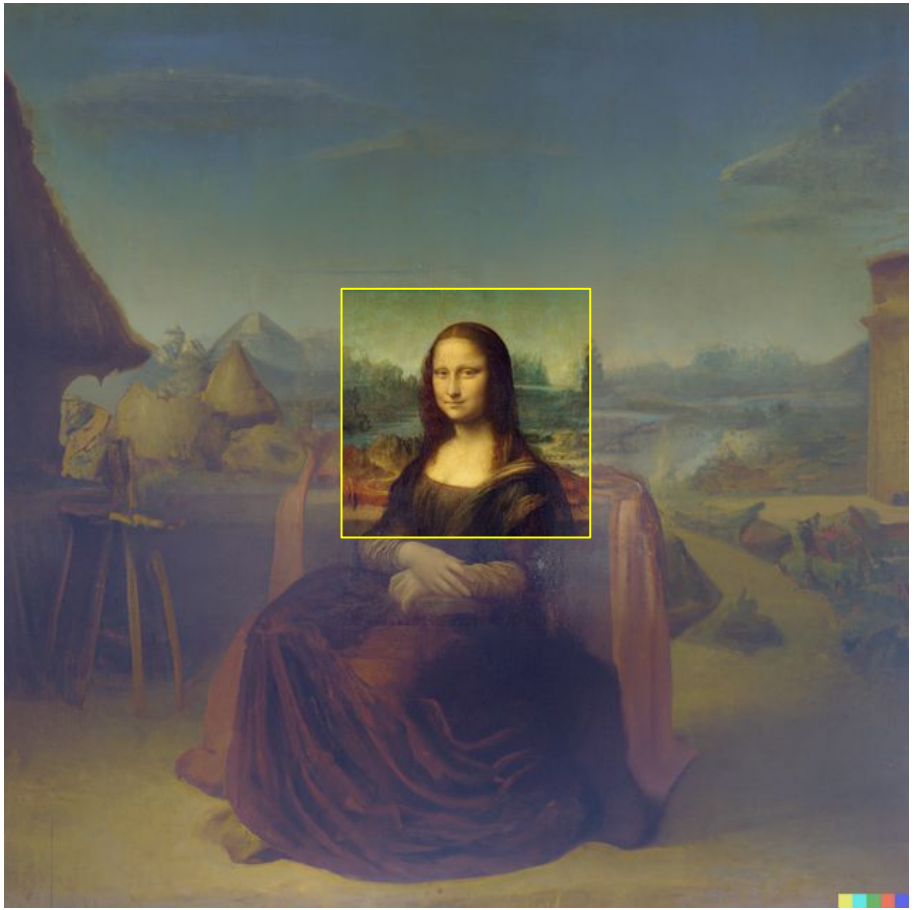


Image by openai.org

Welcome

This is MMI714

“Generative Models for Multimedia”

This review will cover key mathematical concepts needed for the course, including:

sampling, inference, evaluation, distributions, Bayes theorem, likelihood, Gaussian distributions, modality, complexity, expectation maximization, distribution distances, divergence, KL Divergence, Information Theory, Entropy

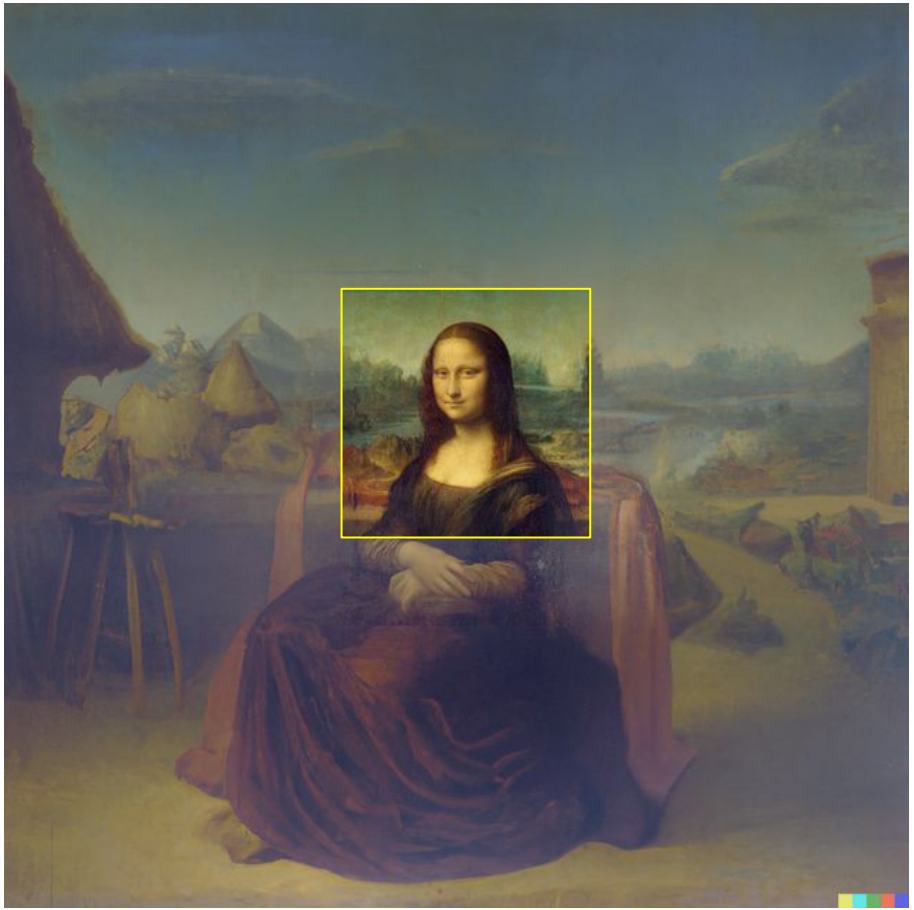
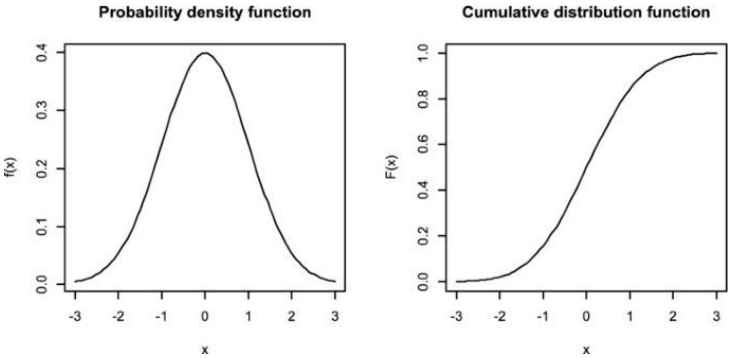


Image by openai.org

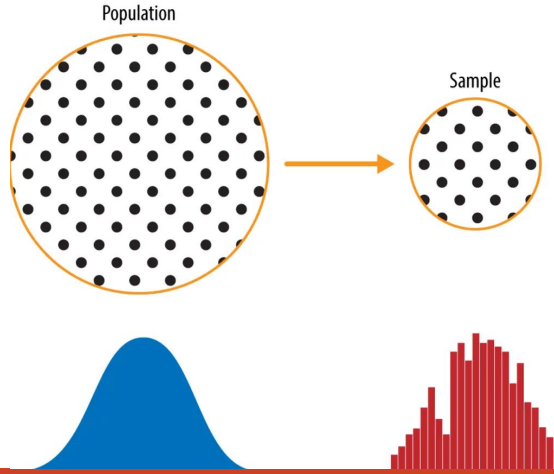
Cumulative Distribution vs Density

- Even though these words can be used interchangeably, in probability theory, they mean different things.
- A cumulative distribution describes the probability of a random variable taking on certain values, while a density function describes the probability of the random variable taking on values in a small interval around a particular point.
- For a continuous distribution, a density function, if it exists, is the derivative of the cumulative distribution.



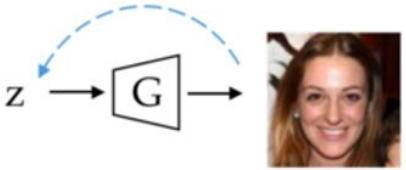
Sampling

- Sampling is a process of generating random variables from a given distribution
- In generative modelling, sampling is used to generate new samples from a learned distribution
- There are several methods for sampling from probability distributions, including analytical (i.e. GMMs*) and non-analytical methods (i.e. GANs*).



Sampling

- In generative modeling, sampling refers to the process of generating new data points from a learned distribution.
- One popular approach to generative modeling is through the use of Generative Adversarial Networks (GANs)* .
- Once the GAN has been trained*, we can use it to sample new data points from the learned distribution.

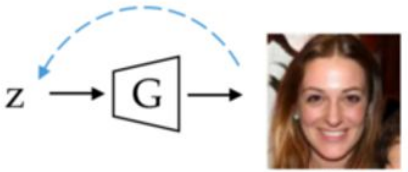


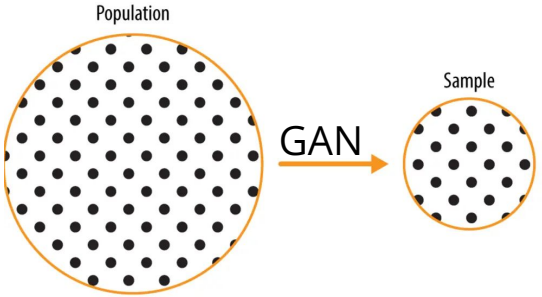
z is a random variable, so that GAN creates a different sample each time

*(which we will learn later)

Sampling

- In some cases, we want to draw samples from a probability distribution that we may not know analytically (like in GANs).
- Or in some other cases, we may know the functional form of the distribution and can use it to generate samples analytically (like in GMMs*)
- The random variable "***z***" is a mathematical construct that captures the randomness in the sampling process.

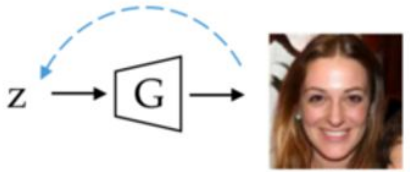


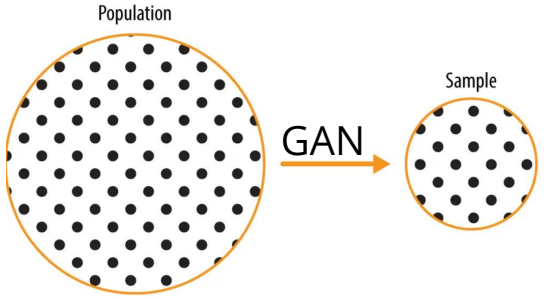


*(which we will learn later)

Sampling

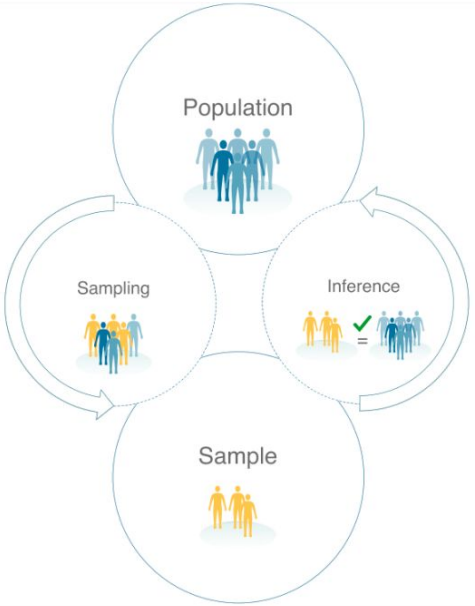
- The selection of the input random variable depends on the specific generative model used. In some cases, the input random variable may be uniformly distributed in a specific range, while in other cases, a more complex distribution may be used.
- In Gaussian Mixture Models (GMMs), for example, the input random variable is often chosen from a mixture of Gaussian distributions that approximate the target distribution.
- In Generative Adversarial Networks (GANs), the input random variable is typically chosen from a simple distribution, such as a uniform or normal distribution, and then





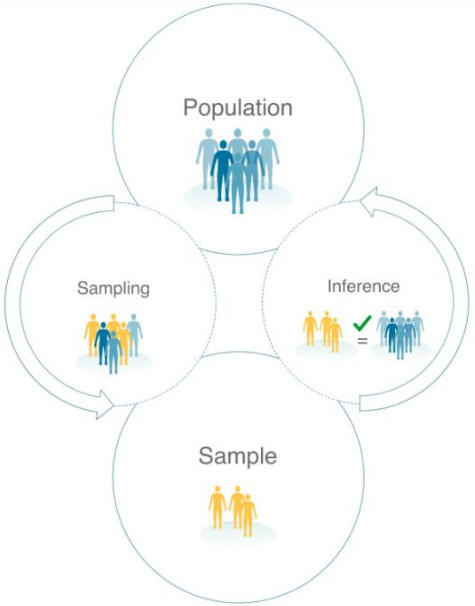
Inference

- So sampling is, when we draw random samples from the probability distribution defined by a model.
- However, in many real-world applications, we often want to do the opposite: given a new data point, we want to infer which model generated it.
- **This process is called inference, and it is the reverse process of sampling.**
- Inference involves using the observed data to update our beliefs about the parameters of the generative model.



Inference

- In discriminative deep learning models like AlexNet, inference is simply the forward run of the model, where we input a data point and obtain a prediction.
- However, in generative models like GANs, inference is (kind of like, but not necessarily) the reverse run of the model, where we use the observed data to update our belief about the generative process.
    - And also **an integral part of the training process.**



Evaluation

- Evaluation (similar to discriminative models) is a crucial aspect of generative modeling, allowing us to measure the quality and performance of the generated samples.
- However, evaluating generative models is challenging compared to discriminative models, where the task is often well-defined.
- In generative models, the goal is to generate realistic and diverse samples, but there is no single objective measure of sample quality.

Evaluation

- Moreover, the quality of the generated samples may depend on the specific application or domain, making it difficult to define a universal evaluation metric.
- The low quality of generated outputs may relate to different outcomes, such as low diversity, unnaturalness, overfitting,, etc...

This is a categorisation of approaches I came up with myself. I am open to ideas, discussions and improvement.

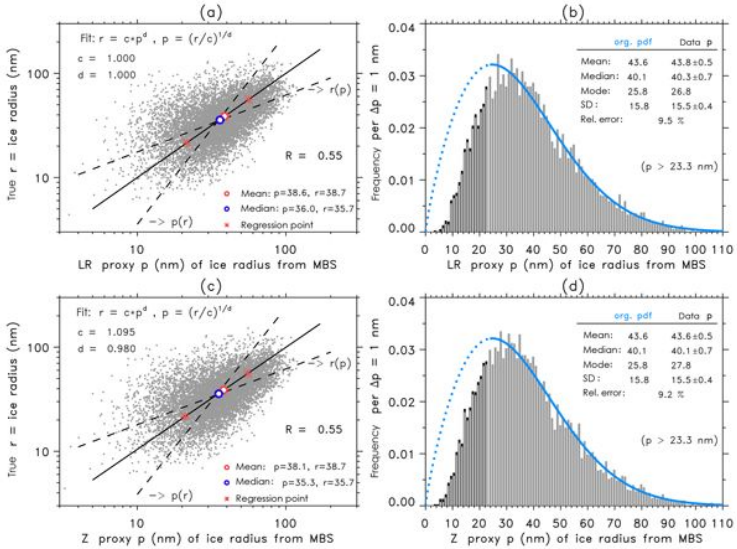
Evaluation Dimensions

- Before we cover various evaluation methods, at this point, we categorize evaluation approaches into three main groups (or say dimensions):
  - **Density**-based approach: These methods evaluate the quality of the generated samples by comparing their distribution to the target distribution. Examples include kernel density estimation (KDE), maximum mean discrepancy (MMD), and energy distance.
  - **Representative sample**-based approach: These methods evaluate the quality of the generated samples by comparing them to a set of "representative" samples from the target distribution. Examples include Precision and Recall, Coverage, etc.
  - **Feature similarity**-based approach: These methods use feature representations extracted from pre-trained models to evaluate the quality and diversity of generated samples. Examples include Frechet Inception Distance (FID), Inception Score (IS), and Learned Perceptual Image Patch Similarity (LPIPS).
- An evaluation method may include a combination of these approaches/dimensions. For example Fréchet Inception Distance* (FID) partially utilize all three approaches.

*(which we will learn later)

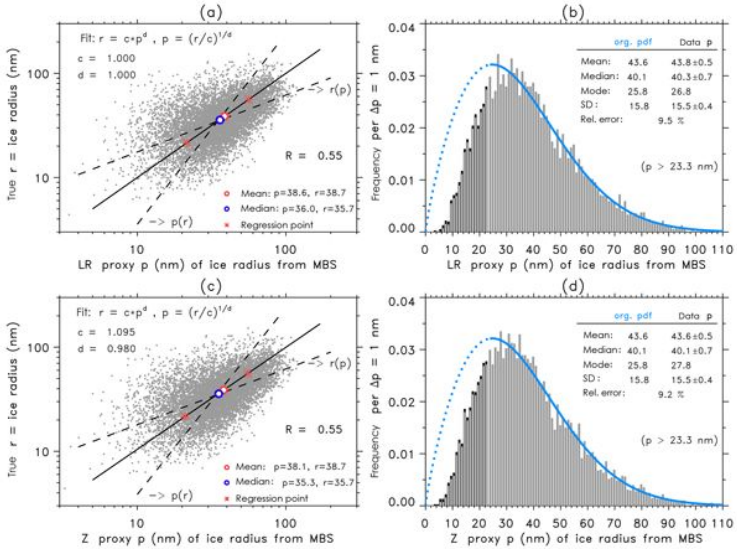
Distributions

- Distribution is a fundamental concept in statistics and probability theory.
- In the context of generative modeling, a distribution is a mathematical function that describes the probability of occurrence of each possible outcome in a given set of outcomes.

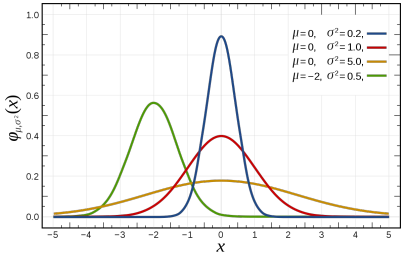


Distributions

- In generative modeling, the aim is to learn the probability distribution of a set of data, so that new data points can be generated from this learned distribution.
- The distribution can be either explicitly defined, as in the case of parametric models such as Gaussian Mixture Models (GMMs); or implicitly defined, as in the case of GANs. (remember previous week, but it was called density ?!)



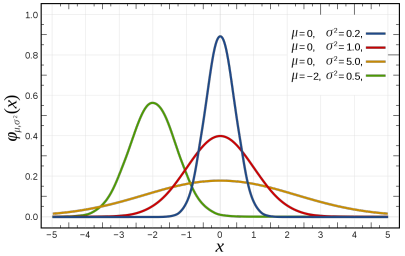
Gaussian Distribution



- The Gaussian distribution, also known as the normal distribution, is a probability distribution that describes how a continuous variable is likely to be distributed.
- It is characterized by two parameters: the mean (μ) and the standard deviation (σ).
- The Gaussian function has a bell-shaped curve, with the peak at the mean.
- The Gaussian distribution is widely used in statistics, machine learning, and other fields because of its mathematical properties and applicability to real-world phenomena.

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

Gaussian Distribution (real world phenomena? Like what?)

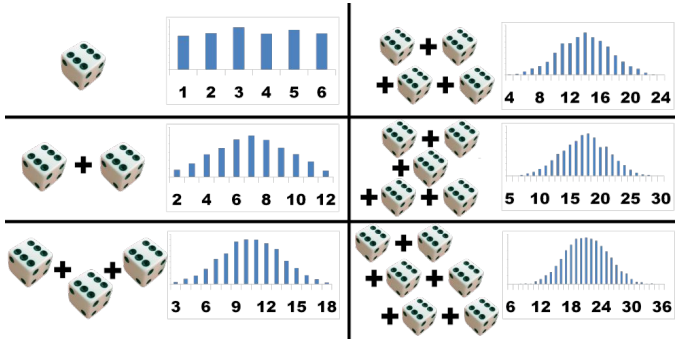


- The central limit theorem states that the sum of independent and identically distributed random variables approaches a Gaussian distribution as the number of variables increases.
- In practice, many real-world phenomena can be modeled as a sum of multiple small contributions, which leads to a Gaussian distribution according to the central limit theorem.

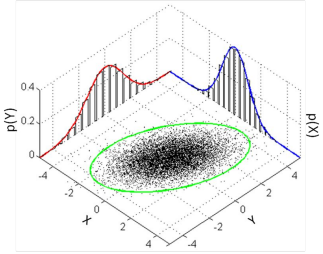
The central limit theorem was first discovered by the French mathematician Abraham de Moivre in the early 18th century.

"Pencü se, severler güzeli gencise"

However, the modern formulation of the theorem is attributed to the French mathematician Pierre-Simon Laplace in the late 18th and early 19th centuries.



Multivariate Gaussian Distribution



- In many real-world applications, we need to model data that has more than one dimension or feature.
- The multivariate Gaussian distribution is a generalization of the univariate Gaussian distribution to multiple dimensions.
- It is characterized by a mean vector (μ) and a covariance matrix (Σ) that describe the location and spread of the distribution in each dimension.
- The covariance matrix contains information about the correlations between the different features.

$$p(\boldsymbol{x}) = \frac{1}{\left|\sqrt{2\pi\boldsymbol{\Sigma}}\right|} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

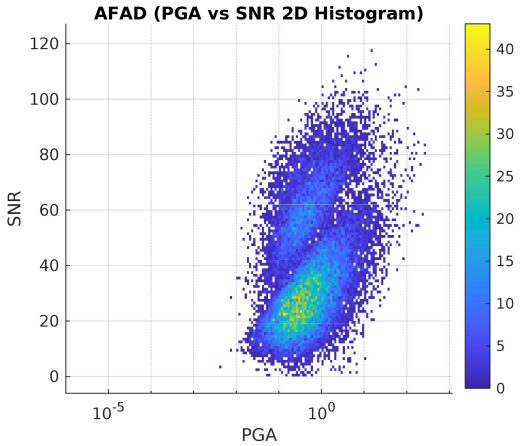
- Mean: $\boldsymbol{\mu}$ (vector 2x1)
- Covariance: $\boldsymbol{\Sigma}$ (matrix 2x2)

Modality

- Modality refers to the number of modes or peaks in a distribution.
- Unimodal distributions have a single mode or peak, while multimodal distributions have multiple modes or peaks.
- The number of modes in a distribution can be an important characteristic for understanding the underlying data.

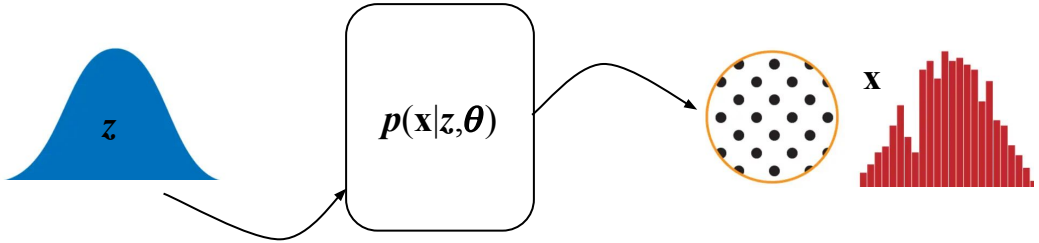
Peak ground acceleration (**PGA**) is equal to the maximum ground acceleration that occurred during earthquake shaking at a location

SNR is defined as the ratio of signal power to the noise power, often expressed in decibels



Generation and Distributions

- So, what is “generation” and why is it related to distribution?
- Is generation a stochastic process?
- If so, is the output of a generative function always a distribution?
- Are generative models always probability distributions?
- Crazy questions in my head…



The theorem is named after Reverend Thomas Bayes, an 18th-century British statistician and theologian.

"çok iyi adam"

Bayes Theorem

- Bayes' Theorem is a fundamental concept in probability theory that describes the probability of an event based on prior knowledge of related events.
- It provides a way to update our beliefs about the probability of an event as new evidence is obtained.

Independent Events

$$P(X \cap Y) = P(X) \cdot P(Y)$$

Dependent Events

$$P(X \cap Y) = P(Y) \cdot P(X \mid Y)$$

LIKELIHOOD
The probability of "B" being True, given "A" is True

PRIOR
The probability "A" being True. This is the knowledge.

$$P(A \mid B) = \frac{P(B \mid A).P(A)}{P(B)}$$

POSTERIOR
The probability of "A" being True, given "B" is True

MARGINALIZATION
The probability "B" being True.
also called the "Evidence"

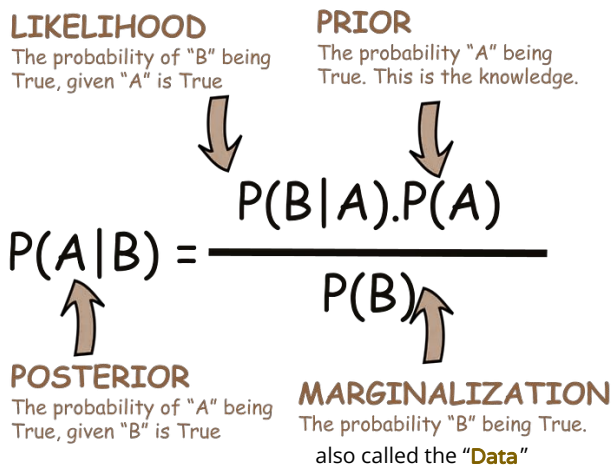
The theorem is named after Reverend Thomas Bayes, an 18th-century British statistician and theologian.

"çok iyi adam"

Bayes Theorem

- Bayes Theorem plays a crucial role in generative models, which are used to learn the underlying probability distributions of a given dataset.

- Bayes Theorem provides a way to update our prior beliefs about the parameters of a probability distribution in light of new evidence (i.e., data).
- In the context of generative models, the theorem is used to estimate the parameters of the underlying distribution that generated the data.
- Specifically, it helps us to update our prior beliefs about the parameters of the distribution based on the observed data.

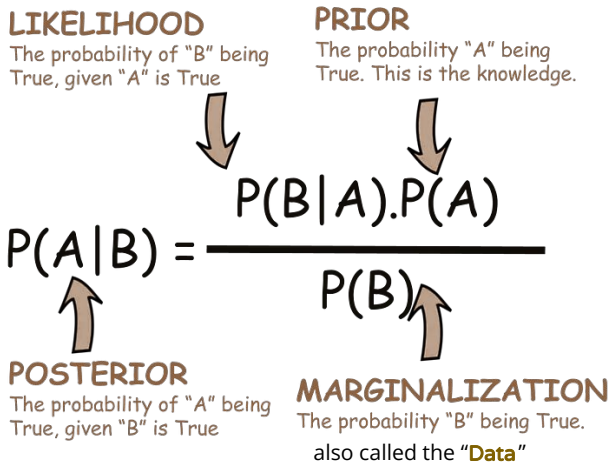


The theorem is named after Reverend Thomas Bayes, an 18th-century British statistician and theologian.

"çok iyi adam"

Bayes Theorem

- Bayes Theorem plays a crucial role in generative models, which are used to learn the underlying probability distributions of a given dataset.
- Bayes' Theorem is widely used in deep learning-based generative models, such as GANs, VAEs, and Bayesian neural networks.
- In GANs, for example, the discriminator network can be seen as an approximate likelihood function, and the generator network is used to generate samples from the learned posterior distribution over the latent variables.

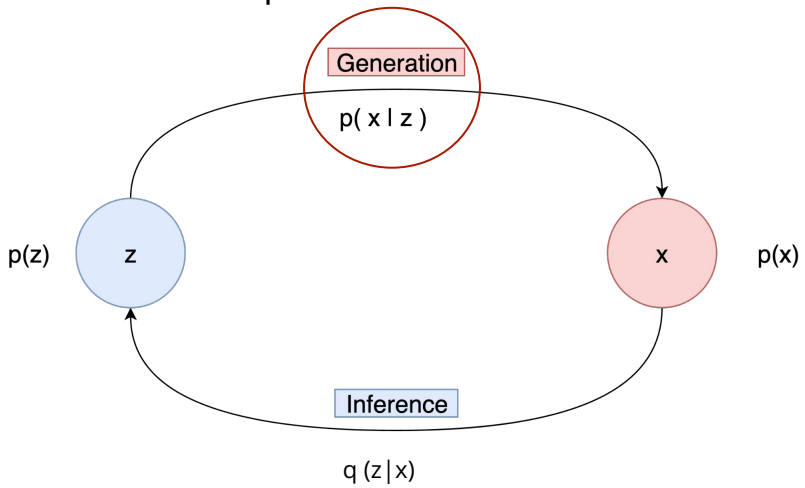


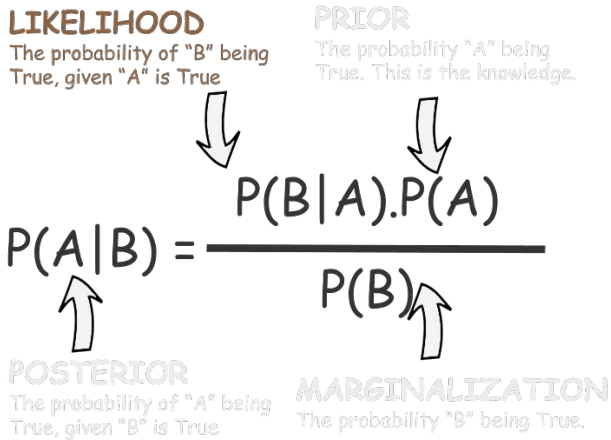
previous week's slide

Likelihood?

In the context of generative models, the likelihood refers to **the probability of observing a given set of data points under the assumed probability distribution of the generative model.**

In other words, the likelihood measures how well the generative model can explain the observed data.





Maximizing the Likelihood

- Maximizing the likelihood involves finding the set of model parameters θ that maximize the likelihood function (i.e. the Generator)

  For example, by using the log-likelihood loss function in a deep generative model to maximize the likelihood of the model parameters given the data.

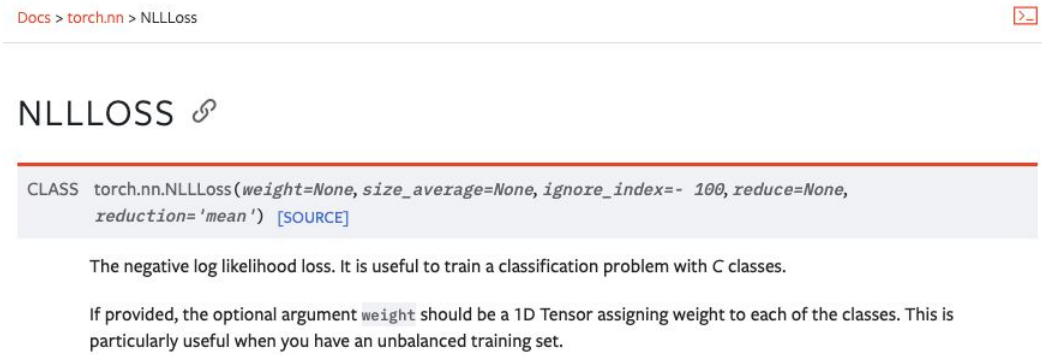
- In deep generative models, the likelihood function is often intractable or difficult to optimize directly.

  For example, in VAEs, a lower bound on the likelihood is optimized instead, while in GANs, a game-theoretic objective is used to implicitly maximize the likelihood.

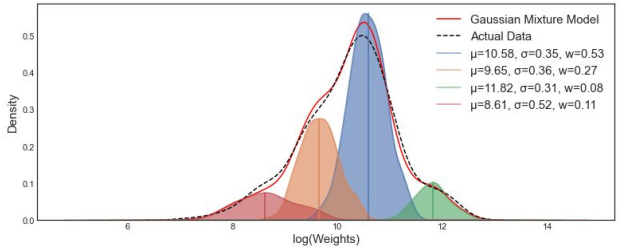
- Maximizing likelihood is a common objective in deep generative models.

Log-likelihood Loss function

- Log-likelihood is a measure of how well a statistical model fits the data it is given. It is commonly used in maximum likelihood estimation (MLE), where the goal is to find the parameter values that maximize the likelihood of the data.
- In deep generative models, the log-likelihood is used as a loss function to ...es the difference between ...stribution of the data. The ... the model's parameters.



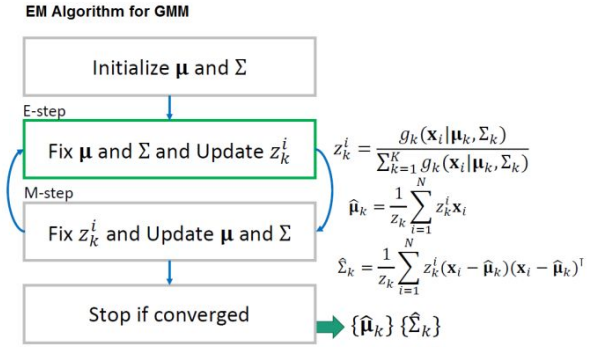
Gaussian Mixture Models (GMM)

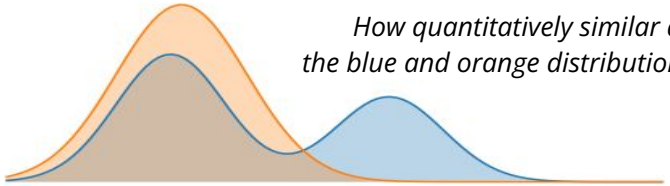


- A Gaussian mixture model (GMM) is a generative probabilistic model that consists of multiple Gaussian distributions.
- It is used to model complex data that cannot be represented by a single Gaussian distribution.
- GMMs are commonly used in clustering and density estimation tasks.
- The parameters of a GMM include the number of mixture components, the mean and standard deviation of each component, and the mixing coefficients that determine the weight of each component.

Expectation-Maximization Algo.

- Expectation Maximization (EM) is an iterative algorithm used to estimate the parameters of a statistical model when some of the variables are unobserved.
- EM is commonly used to train Gaussian Mixture Models (GMMs), which are generative models that approximate a probability distribution as a weighted sum of Gaussian distributions.
- In GMMs, each Gaussian distribution represents a cluster in the data, and the weights represent the relative importance of each cluster.
- EM is a powerful and widely used algorithm for unsupervised learning and can be applied to many other types of models beyond GMMs.





Distribution Distances

- Why do we need to compare distributions?
  - In generative modeling, it is important to compare different probability distributions to determine how well our model is performing.
  - For instance, we need to be able to evaluate the similarity between the true data distribution and the distribution learned by our generative model.
- “Divergence”
  - is a way to measure the distance between two probability distributions.
  - measures quantify how much one distribution differs from another in terms of their shapes, locations, or other characteristics.

Divergence (first)

- Not all distance measures between two distributions are “divergence” measures, but we will start with them first.
- Some of which, we may benefit from in this course, are:
  - Kullback-Leibler (KL) divergence
  - Jensen-Shannon (JS) divergence
  - Total Variation (TV) distance
  - Hellinger distance

Kullback-Leibler (KL) Divergence

- The KL divergence is a measure of the difference between two probability distributions, P and Q.
- It is defined as the expected value of the logarithmic difference between P and Q, where the expectation is taken with respect to P. The KL divergence is denoted as D(P||Q).

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

Kullback-Leibler (KL) Divergence

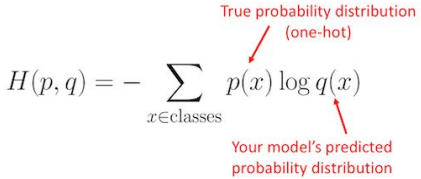
- The KL divergence is always non-negative, and it is zero if and only if the two distributions P and Q are identical.
- The KL divergence is not symmetric, meaning that D(P||Q) is not necessarily equal to D(Q||P).

$$D_{KL}(P||Q) \neq D_{KL}(P||Q)$$

- The KL divergence can be interpreted **as the amount of information lost when using Q to approximate P** (or vice versa). It measures the additional number of bits of information needed to specify P instead of Q.

Kullback-Leibler (KL) Divergence

- The KL divergence is commonly used in generative modeling to measure the similarity between the true data distribution and the distribution learned by a generative model.
- It is often used as a loss function to train generative models, such as Variational Autoencoders (VAEs), which aim to learn a lower-dimensional representation of the data that can be used to generate new samples.
- Did KL remind you of something?
    - Cross-entropy maybe?



KL vs CE

- Cross-entropy is a measure of the dissimilarity between two probability distributions, typically between a true distribution and an estimated distribution.
- KL divergence measures the divergence between two probability distributions, P and Q, by measuring the additional number of bits of information needed to specify P instead of Q.

$$H(P,Q) = -\sum_{x} P(x) \log Q(x)$$

$$KL(P|Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

KL vs CE

- Both KL divergence and cross-entropy are used to measure the similarity or dissimilarity between two probability distributions.
- Both measures are commonly used in generative modeling to evaluate the performance of generative models and to optimize their parameters.
- Both measures are non-negative and minimize t zero if and only if the two distributions are identical.
- Both measures are asymmetrical.

$$H(P,Q) = -\sum_{x} P(x) \log Q(x)$$

$$KL(P|Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

KL vs CE

- KL Divergence focuses on the additional information needed to be accurate about the true distribution when starting with an approximation. It emphasizes the "gap" between the true distribution and the approximation.
- Cross Entropy is more about the efficiency of encoding events from the true distribution when using the code optimized for an approximation. It looks at the average number of bits needed and emphasizes the cost of using the code optimized for

$$H(P,Q) = -\sum_{x} P(x) \log Q(x)$$

$$KL(P|Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

Jensen-Shannon (JS) Divergence

- The JS divergence is a symmetric measure of the difference between two probability distributions.
- It is a smoothed version of the KL divergence, which can be used to compare two probability distributions that may have disjoint support.
-

$$D_{JS}(p||q) = \frac{1}{2}D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q||\frac{p+q}{2})$$

Jensen-Shannon (JS) Divergence

- The JS divergence is a symmetric measure of the difference between two probability distributions.
- It is a smoothed version of the KL divergence, which can be used to compare two probability distributions that may have disjoint support.
- The JS divergence is bounded by 0 and 1, and is equal to 0 if and only if the two distributions are identical.

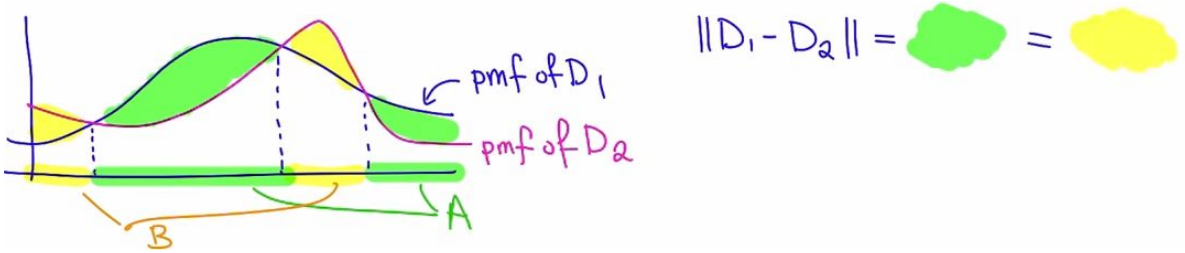
$$0 \leq \mathrm{JSD}(P \parallel Q) \leq 1$$

- The JS divergence are used as a loss function to train generative models, such as Generative Adversarial Networks (GANs).

Total Variation (TV) Distance

- In probability theory, the total variation distance is a distance measure for probability distributions.
- It is an example of a statistical distance metric, and is sometimes called the statistical distance, statistical difference or variational distance.
-

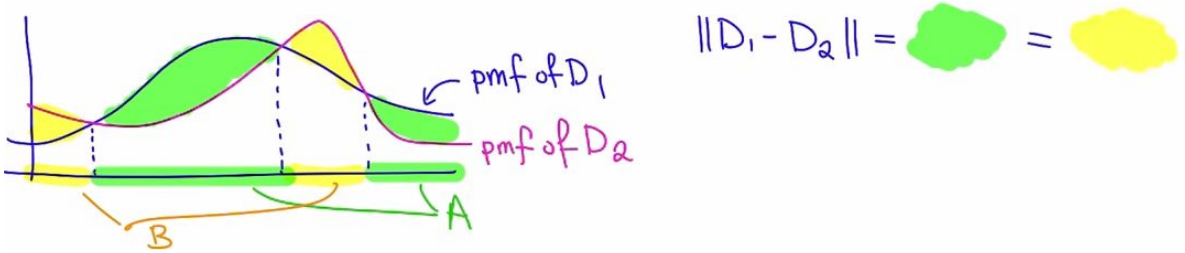
$$||D_1 - D_2|| = \frac{1}{2}\sum_{s \in S} |D_1(s) - D_2(s)|$$



Total Variation (TV) Distance

- The TV distance is defined as half the L1 norm of the difference between the CDFs of the two distributions.
- The TV distance is bounded by 0 and 1, and is equal to 0 if and only if the two distributions are identical.

$$||D_1 - D_2|| = \frac{1}{2}\sum_{s \in S} |D_1(s) - D_2(s)|$$



Hellinger Distance

- Hellinger Distance is defined as the square root of half the sum of the squared differences between the square roots of the probability density functions of the two distributions.
- The Hellinger distance is symmetric, non-negative, and bounded between 0 and 1. It is 0 if and only if the two distributions are identical, and 1 if and only if the supports of the two distributions are disjoint.

For two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$, their Hellinger distance is defined as

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2},$$

Hellinger Distance

- It is less sensitive to outliers and heavy tails, and can capture differences in both the shape and the spread of the distributions.
- It is also computationally efficient to compute, and has a closed-form expression for many common distributions.
- However, the Hellinger distance is not a true metric, since it does not satisfy the triangle inequality.
- It also has a weaker convergence property than the KL divergence, which can lead to slower convergence in certain optimization problems.

Information Theory

- Information theory was initially developed to understand and improve communication systems, especially in the context of telegraphy and radio.
- However, its scope has expanded to various other areas such as data compression, cryptography, and more recently, deep learning and generative models.
- In the context of deep generative models, information theory provides a theoretical framework for understanding and designing models that can generate high-quality and diverse samples from complex distributions.

Entropy & Information

- Entropy is a measure of uncertainty or disorder in a random variable, while information is the reduction of uncertainty or surprise gained from an event.
- In deep generative models,
  - the entropy of the output distribution can be used to measure the complexity of the generated samples,
  - while information can be used to measure the amount of information captured in the learned representation.
- **They are formulated measures!**

Shannon's Entropy

- Introduced by Claude Shannon in 1948 as a measure of uncertainty or information content in a random variable or probability distribution, defined as the expected value of the information contained in each possible outcome, given the probability distribution:

$$H(X) = -\sum_{i} P(x_i) \log P(x_i)$$

- maximized when all outcomes are equally likely (i.e., maximum uncertainty)
- minimized when there is only one possible outcome (i.e., no uncertainty)

Conditional Entropy

- is the amount of uncertainty remaining in a random variable given that another random variable has been observed or known.

$$\boldsymbol{H(X) = -\sum_i P(x_i) \log P(x_i)} \longrightarrow \mathrm{H}(Y|X) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)}$$

- Conditional entropy tells us how much information we gain about Y by observing X. If conditional entropy is high, it means that observing X gives us little information about Y, and vice versa.

Entropy and Generative Models

- Entropy measures such as Shannon entropy and differential entropy are used to quantify the uncertainty or randomness in the generated samples.
- In a well-trained generative model, the generated samples should have high entropy, indicating that the model is able to produce a diverse set of samples that capture the variability in the training data.
    - Example: In a generative adversarial network (GAN), the generator tries to generate samples that fool the discriminator. The entropy of the generated samples can be used to measure the diversity and quality of the samples generated by the GAN.

Shannon's "Self-Information"

- The amount of information gained by an event with probability p is defined as:

$$\mathrm{I}(x) := -\log_b \left[\Pr\left(x\right)\right] = -\log_b \left(P\right).$$

- Shannon's definition of self-information meets several axioms:
  - An event with probability 100% is perfectly unsurprising and yields no information.
  - The less probable an event is, the more surprising it is and the more information it yields.
  - If two independent events are measured separately, the total amount of information is the sum of the self-informations of the individual events

Mutual Information

- is a measure of the amount of information that two variables share.
- MI quantifies the reduction in uncertainty about one variable given knowledge of the other variable, and can be represented using the Entropy:

$$
\begin{aligned}
\mathrm{I}(X;Y) &\equiv \mathrm{H}(X) - \mathrm{H}(X \mid Y) \\
&\equiv \mathrm{H}(Y) - \mathrm{H}(Y \mid X) \\
&\equiv \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X, Y) \\
&\equiv \mathrm{H}(X, Y) - \mathrm{H}(X \mid Y) - \mathrm{H}(Y \mid X)
\end{aligned}
$$

  - Example: MI can be used as a regularizer to encourage disentanglement of the latent variables in the learned representation.

Information and Generative Models

- Information measures such as mutual information and conditional entropy are used to evaluate the ability of the generative model to capture the underlying structure of the data.
- In a well-trained generative model, the mutual information between the generated samples and the training data should be low, indicating that the generated samples are not duplicating the training data.
    - Example: In a variational autoencoder (VAE), the encoder tries to compress the input data into a low-dimensional latent space. The mutual information between the latent space and the input data can be used to measure the amount of information that is preserved in the latent space (like a regularizer to encourage disentanglement of the latent variables in the learned representation)

What to do until next week?

You may

- Go over your probability notes, remember what likelihood, Bayes Theorem, etc were…
- Go over the STEAD dataset, get familiar with it, prepare some questions to ask, so that, when the time comes, you will be ready for your project….

What will we do next week?

- Starting with next week…
  - *Time Series, Autoregressive Models, Deep Autoregressive networks, Recurrent Neural Networks for Generation*
- Following weeks
  - *Energy-based models, latent space, etc…*

Additional Reading & References

- https://pytorch.org/docs/stable/generated/torch.nn.NLLLoss.html
- https://sandipanweb.wordpress.com/2016/07/02/using-expectation-maximization-algorithm-for-the-gaussian-mixture-models-to-detect-outliers/
- https://www.youtube.com/watch?v=-EGA7S-qcwY
- https://www.youtube.com/watch?v=SxGYPqCgJWM
- https://www.youtube.com/watch?v=LJwtEaP2xKA
- https://www.youtube.com/watch?v=aJP0Bj6jVkI