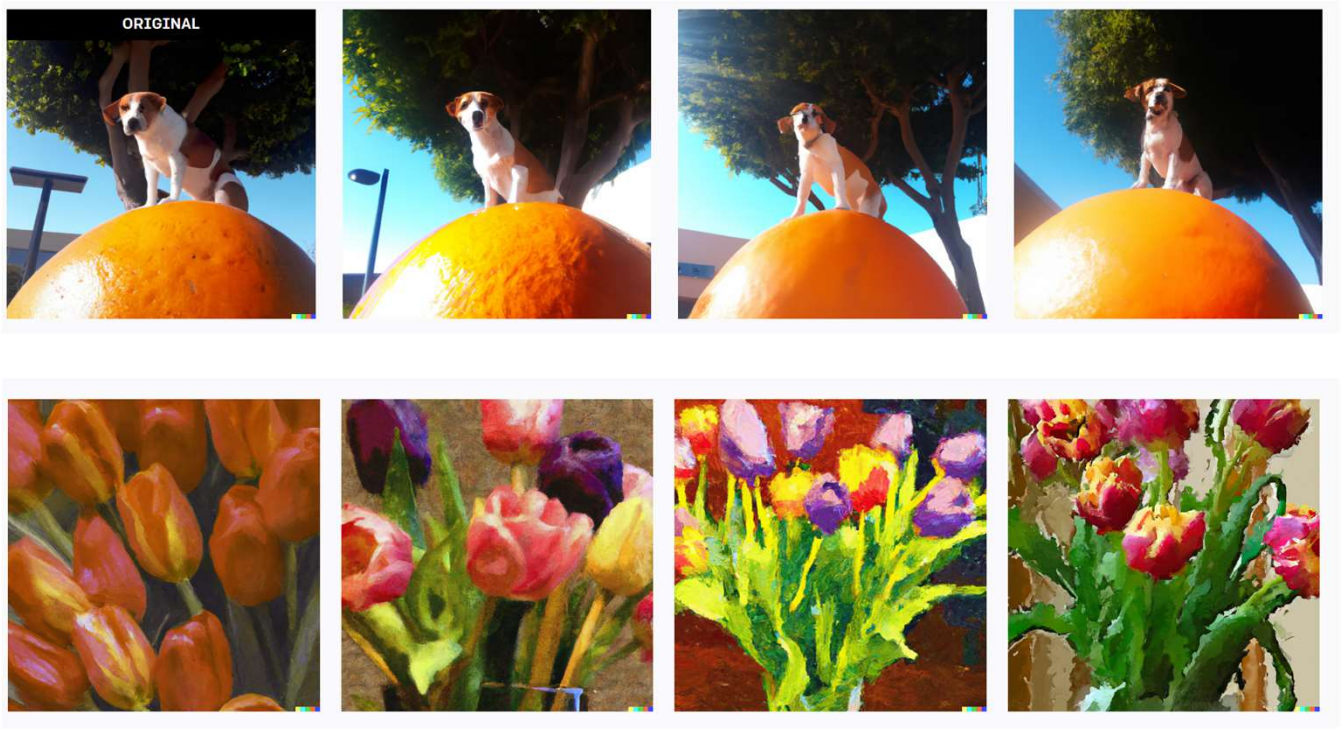
Generative Models

Lecture 1

Images

DAL-EE generated images with prompts ‘Dog sitting on a giant orange’ and ‘van gogh painting of a bunch of tulips’



Music

Generated with MusicGen with prompt ‘90s style with heavy lead guitar and heavy percussion’

Text

Let ChatGPT recommend the contents of the course

NI

What should be covered in a course on Generative AI

A comprehensive course on Generative AI should cover a range of topics that provide students with a solid understanding of the theory, algorithms, and applications of generative models. Here is an outline of what could be covered in such a course:

1. Introduction to Generative AI:
   - Definition and motivation behind generative models.
   - Comparison between generative and discriminative models.
   - Applications of generative AI in various domains.
2. Probability and Statistical Foundations:
   - Probability distributions and likelihood estimation.
   - Maximum likelihood estimation (MLE) and Bayesian inference.
   - Monte Carlo methods and Markov Chain Monte Carlo (MCMC) techniques.
3. Autoencoders and Variational Autoencoders (VAEs):
   - Understanding autoencoders as unsupervised learning tools.
   - Introduction to Variational Autoencoders (VAEs) and their latent space representation.

4. Generative Adversarial Networks (GANs):
    - Overview of GAN architecture and the adversarial learning framework.
    - Training GANs and the role of generator and discriminator networks.
    - Conditional GANs and applications in image-to-image translation.
5. Flow-Based Generative Models:
    - Introduction to normalizing flows and invertible transformations.
    - Architectures like RealNVP, Glow, and FFJORD for density estimation.
    - Applications of flow-based models in image generation and density modeling.
6. Sequence Generation Models:
    - Recurrent Neural Networks (RNNs) and their use in sequence generation.
    - Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures.
    - Attention mechanisms and Transformer models for sequence-to-sequence tasks.
7. Evaluation of Generative Models:
    - Evaluation metrics for assessing the quality of generated samples.
    - Perceptual metrics and comparisons with ground truth data.
    - Challenges in evaluating generative models and possible solutions.
8. Advanced Topics in Generative AI:
    - Denoising Diffusion Probabilistic Models (DDPM) and Energy-Based Models (EBMs).
    - Meta-learning and unsupervised representation learning.
    - Recent research in generative modeling and future directions.

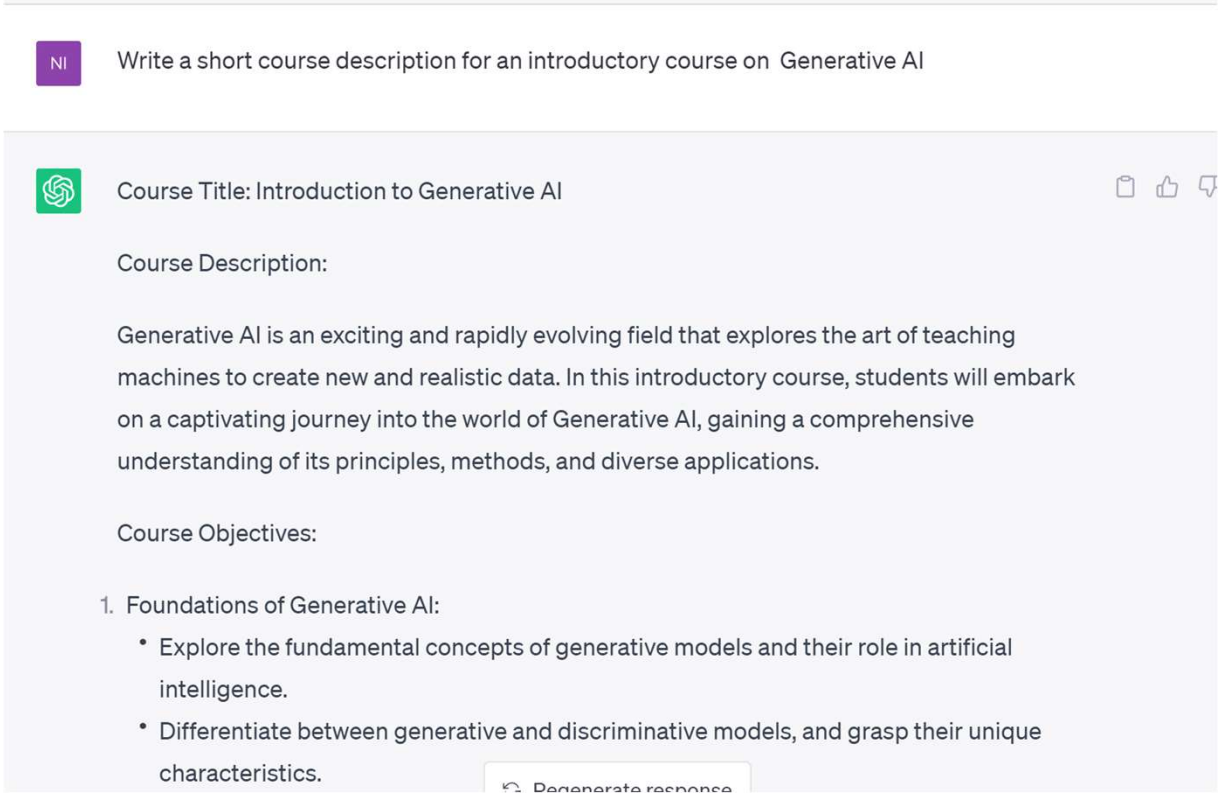
9. Applications of Generative AI:
    - Image synthesis and style transfer.
    - Text generation and language modeling.
    - Music generation and creative applications.
10. Ethical Considerations and Real-World Impact:

- Ethical implications of generative AI, including deepfakes and misinformation.
- Discussion on bias, fairness, and responsible AI in generative models.
- Case studies on the real-world impact of generative AI in different industries.

The course should include theoretical lectures, hands-on practical exercises, and project assignments to provide students with both the knowledge and practical skills to implement and apply generative models effectively. Additionally, it should cover the latest research and developments in the field, as generative AI is an active area of research with continuous advancements.

We will probably cover topics 1, 2, 3, 8 and 9

We can also let ChatGPT write the course description



characteristics.

2. Probabilistic Modeling and Inference:
   - Lay the groundwork with probability theory and statistical inference.
   - Learn how to model probability distributions and estimate parameters from data.
3. Variational Autoencoders (VAEs):
   - Delve into autoencoders as unsupervised learning tools.
   - Understand Variational Autoencoders (VAEs) and their applications in data generation.

Throughout the course, students will actively engage in hands-on exercises, practical demonstrations, and interactive discussions to solidify their understanding of Generative AI concepts. By the course's conclusion, students will possess the foundational knowledge to implement generative models and appreciate the transformative potential of Generative AI in diverse domains, inspiring further exploration in this dynamic and creative field.

NI

Tell me about Denoising Diffusion

Denoising Diffusion Probabilistic Models (DDPM) is a recent advancement in the field of generative modeling that has shown impressive results in image generation tasks. It was introduced in a paper titled "Denoising Diffusion Probabilistic Models" by Jonathan Ho, Ajay Jain, and Pieter Abbeel, published in 2020.

DDPM is based on the concept of diffusion processes, where a noise model is used to iteratively refine a noise-corrupted image into a high-quality target image. The core idea is to model the data likelihood using a diffusion process that starts from a noise distribution and gradually evolves towards the target data distribution by applying a series of conditional denoising steps.

The generative process of DDPM can be summarized as follows:

1. Initialization: Start with a noise sample (e.g., Gaussian noise) as the initial image.
2. Diffusion Steps: Iteratively update the noise-corrupted image using a series of denoising steps. Each denoising step is modeled by a neural network that takes the current noisy image and a noise level (or a learnable noise schedule) as inputs and outputs a refined version of the image. The noise level is annealed as the process advances.
3. Final Image: After a fixed number of diffusion steps, the process converges to a high-quality sample that resembles the target data distribution.

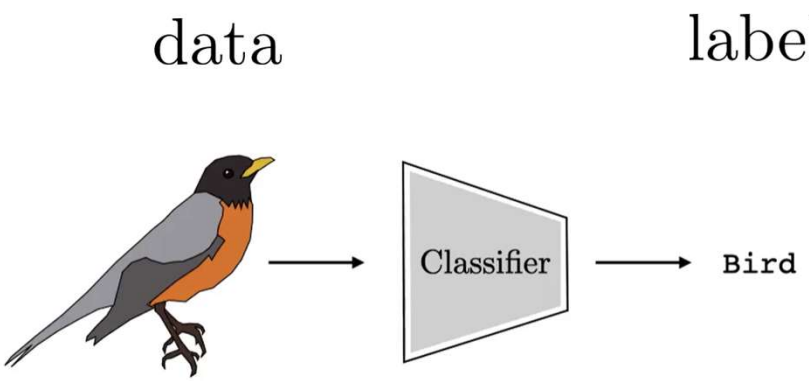
Two types of AI models

Supervised Learning

Used for classification i.e. labeling data points.

Trained on labeled datasets, this is called *supervised learning.*

Examples: identifying contents of images, character recognition, sentiment analysis



Supervised Learning

We are given a *data set* of *labeled data*

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$

The y's are labels in some finite set of labels

$$\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_K\}$$

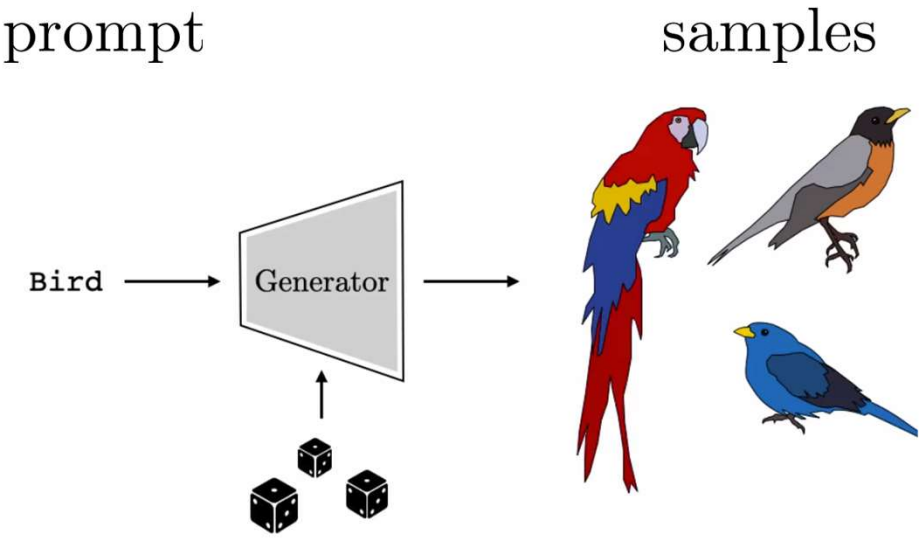
Goal

Learn a conditional probability distribution over the set of labels $\mathcal{L}$

$$p(\ell|\mathbf{x})$$

Generative Models (Unsupervised Learning)

Learns to generate new data samples



Unsupervised Learning

Data set of *unlabeled data*

$$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$$

samples from some unknown distribution p_{data}

Goal

Learn a distribution p_{model} 'close' to the unknown distribution p_{data} and generate samples that are close to samples from p_{data}

***Generative Models* are trained on *unlabeled* datasets**

Dataset of vectors

$$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$$

.

The idea is that these vectors are samples from some unknown complicated distribution that we seek to approximate.

A Generative Model is trained to generate *samples* from the unknown distribution

Generative Models

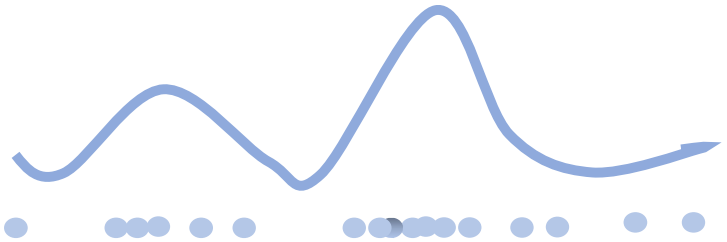
Goals:

Density Estimation:

Learn a *probability distribution* that describes the data

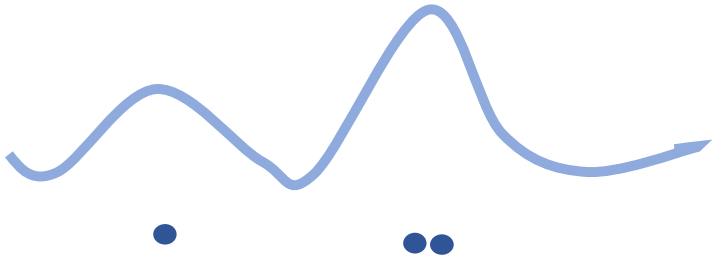
density p_X

data X



Sample Generation:

Generate new samples from the distribution



Generated samples $\sim p_X$

Reminder about Random Variables and Distributions

A *Random Variable* (with values in $\mathbb{R}^d$) is a function

$$\mathbf{X} : \Omega \to \mathbb{R}^d$$

where $(\Omega, \mathbb{P})$ is some fixed probability space (it doesn't really matter what it is)

We can express properties of this abstract function in terms of usual real functions:

- The *Distribution Function* $d_{\mathbf{X}} : \mathbb{R}^d \to [0,1]$

  $d_{\mathbf{X}}(a_1, a_2, \ldots, a_d) = \mathbb{P}(\{\omega \in \Omega \,|\, \mathbf{X}(\omega) \in \{(t_1, t_2, \ldots, t_d) \in \mathbb{R}^d \,|\, t_1 < a_1, t_2 < a_2, \ldots, t_d < a_d\}\}$

  In other words it is the probability that a random ω in Ω satisfies

  $$\mathbf{X}_1(\omega) < a_1, \mathbf{X}_2(\omega) < a_2, \ldots, \mathbf{X}_d(\omega) < a_d$$

  where

  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d)$ are the coordinate functions

- The *Density Function* $p_{\mathbf{X}} : \mathbb{R}^d \to \mathbb{R}_+$

  $$p_{\mathbf{X}} = \frac{\partial}{\partial t_1} \frac{\partial}{\partial t_2} \cdots \frac{\partial}{\partial t_d} d_{\mathbf{X}}$$

If $A \subset \mathbb{R}^d$ is a reasonable (= Borel measurable) subset then we can express the probability that a value of $\mathbf{X}$ lands in A, as a usual d-dimensional integral

$$\mathbb{P}(\mathbf{X}^{-1}(A)) = \int_A p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$$

$$\int_{\mathbb{R}^d} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = 1$$

The *mean* $\mu_{\mathbf{X}}$ (which is a vector in $\mathbb{R}^d$) is

$$\begin{aligned}\mu_{\mathbf{X}} &= (\mu_{\mathbf{X}_1}, \mu_{\mathbf{X}_2}, \ldots, \mu_{\mathbf{X}_d}) \\ &= \left(\int_{\mathbb{R}^d} x_1 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \int_{\mathbb{R}^d} x_2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \ldots, \int_{\mathbb{R}^d} x_d p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \right)\end{aligned}$$

Covariance Matrix

$$\Sigma_{ij} = \int (x_i - \mu_{\mathbf{X}_i})(x_j - \mu_{\mathbf{X}_j}) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{X})$$

Σ is a $d \times d$ symmetric, positive definite matrix

Expectation

If $f : \mathbb{R}^d \to \mathbb{R}$ is a function. Then

$$f(\mathbf{X}) : \Omega \to \mathbb{R}$$

is a Random Variable with values in $\mathbb{R}$.

The *Expectation*

$$\mathbb{E}_{p_{\mathbf{X}}}(f(\mathbf{X})) = \int f(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$$

Marginalization

$\mathbf{X} : \Omega \to \mathbb{R}^{d_1}, \mathbf{Y} : \Omega \to \mathbb{R}^{d_2}$ random variables

Joint random variable $\mathbf{X}, \mathbf{Y} : \Omega \to \mathbb{R}^{d_1+d_2}$

Densities

$p_{\mathbf{X}}, p_{\mathbf{Y}}$ and joint density $p_{\mathbf{X},\mathbf{Y}}$

$$p_{\mathbf{X}}(\mathbf{x}) = \int_{\mathbb{R}^{d_2}} p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

$$p_{\mathbf{Y}}(\mathbf{y}) = \int_{\mathbb{R}^{d_1}} p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) d\mathbf{x}$$

Example

The Multi-Variate Normal density of dimension d

$$\mathcal{N}(\mathbf{x}; \mu_{\mathbf{X}}, \Sigma) = p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_{\mathbf{X}})\,\Sigma^{-1}\,(\mathbf{x} - \mu_{\mathbf{X}})^T\right)$$

Example

Energy Based Density function (Boltzmann Distribution)

$f : \mathbb{R}^d \to \mathbb{R}$ function such that

$$\int_{\mathbb{R}^d} \exp(-f(\mathbf{t}))d\mathbf{t}$$

is finite, then

$$p(\mathbf{x}) = \frac{\exp(-f(\mathbf{x}))}{\int \exp(-f(\mathbf{t}))d\mathbf{t}}$$

is a density function

Conditional Density

If $\mathbf{X}$ and $\mathbf{Y}$ are random variables then the conditional density function of $\mathbf{X}$ conditional on $\mathbf{Y}$ is

$$p_{\mathbf{X}|\mathbf{Y}} = \frac{p_{\mathbf{X},\mathbf{Y}}}{p_{\mathbf{Y}}}$$

Bayes' formula

$$p_{\mathbf{Y}|\mathbf{X}} = \frac{p_{\mathbf{X}|\mathbf{Y}}}{p_{\mathbf{X}}} p_{\mathbf{Y}}$$

Without further assumptions the general problem of finding the distribution of the dataset $\mathcal{D}$, is intractable.

Parametrized Densities

Instead we will focus on families of distributions $\{p_\theta\}_{\theta \in \Theta}$ (like Gaussians) where each individual distribution is specified by parameters θ. These parameters are chosen from a set of all possible parameters Θ, for the family

For instance for Gaussians $\theta = (\mu, \Sigma)$ and Θ will be the set of all pairs of a d-dimensional vector μ and a symmetric, positive definite matrix Σ

Finding Optimal Parameters

For each $\theta \in \Theta$ we have a density p_θ from our chosen family

We want to find the θ that gives the best "fit" to the dataset

One way to measure the fit is to look at the *log-likelihood*

The log-likelihood of a point $\mathbf{x}$ with respect to p_θ is $\log p_\theta(\mathbf{x})$

If $\mathbf{X}_\theta$ is a random variable such that

$$p\mathbf{x}_\theta = p_\theta$$

we want to find θ that maximizes the probability that the data set $\mathcal{D}$ are samples from $\mathbf{X}_\theta$

For each $\mathbf{x} \in \mathcal{D}$ we choose a small neighborhood $\mathcal{N}_{\mathbf{x}}$ around $\mathbf{x}$

The probability that $\mathbf{X}_\theta(\omega) \in \mathcal{N}_{\mathbf{x}}$ is

$$\mathbb{P}(X_\theta^{-1}(\mathcal{N}_{\mathbf{x}})) = \int_{\mathcal{N}_{\mathbf{x}}} p_\theta(\mathbf{t}) d\mathbf{t} \approx p_\theta(\mathbf{x}) vol(\mathcal{N}_{\mathbf{x}})$$

Given a sequence of elements $\omega_1, \omega_2, \ldots, \omega_N$ the probability that

$$\mathbf{X}_\theta(\omega_i) \approx \mathbf{x}_i \text{ for } i = 1, 2, \ldots, N$$

is $\approx \prod_i p_\theta(\mathbf{x}_i) vol(\mathcal{N}_{\mathbf{x}_i})$ so we want to maximize $\prod_{\mathbf{x} \in \mathcal{D}} p_\theta(\mathbf{x})$

Maximum Likelihood

To maximize $\prod_{\mathbf{x}\in\mathcal{D}} p_\theta(\mathbf{x})$ we may as well take logs

$$LL(\theta) = \sum_{\mathbf{x}\in\mathcal{D}} \log p_\theta(\mathbf{x})$$

So we want to maximize $LL(\theta)$ or equivalently, minimize $NLL(\theta) = -LL(\theta)$

So we would like to solve

$$\nabla_\theta NLL(\theta) = 0$$

Except for a small number of cases it is not possible to to find an exact solution for θ

Maximum Likelihood for Gaussians

$$\log p_\theta(\mathbf{x}) = \log p_{\mu,\Sigma}(\mathbf{x}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log\det\Sigma - \frac{1}{2}(\mathbf{x}-\mu)\,\Sigma^{-1}\,(\mathbf{x}-\mu)^T)$$

$$NLL(\mu,\Sigma) = \sum_{\mathbf{x}\in\mathcal{D}} \frac{n}{2}\log 2\pi + \frac{1}{2}\log\det\Sigma + \frac{1}{2}(\mathbf{x}-\mu)\,\Sigma^{-1}\,(\mathbf{x}-\mu)^T)$$

$$\nabla_\mu NLL(\mu,\Sigma) = \Sigma^{-1}\sum_{\mathbf{x}\in\mathcal{D}}(\mathbf{x}-\mu)$$

Solving $\Sigma^{-1}\sum_{\mathbf{x}\in\mathcal{D}}(\mathbf{x}-\mu) = 0$ for μ

$$\mu = \frac{1}{N}\sum_{\mathbf{x}\in\mathcal{D}}\mathbf{x}$$

Finding Σ

Formulas for derivatives of matrices

- $\nabla_\Sigma \log \det \Sigma = \Sigma^{-1}$
- $\nabla_\Sigma (a \Sigma^{-1} a^T) = -\Sigma^{-1} a^T a \Sigma^{-1}$

$$\nabla_{\Sigma} NLL(\theta) = \sum_{\mathbf{x} \in \mathcal{D}} (-\Sigma^{-1} + \Sigma^{-1}(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1})$$

$$\nabla_{\Sigma} NLL(\theta) = 0 \implies \sum_{\mathbf{x} \in \mathcal{D}} \left(-\Sigma + (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T\right) = 0$$

$$\Sigma = \frac{1}{N} \sum_{\mathbf{x} \in \mathcal{D}} (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T$$

Latent Variable Models

Assume we have a Random Variable $\mathbf{X}$ where the density $p_{\mathbf{X}}$ is intractable.

It may be that there is another Random Variable $\mathbf{Z}$, called a *latent* or *hidden* variable, such that $\mathbf{X}$ depends on $\mathbf{Z}$ in a way that we can deal with i.e. the *conditional density* $p_{\mathbf{X}|\mathbf{Z}}$ is computable

For instance if $\mathbf{Z}$ is a given Random Variable (and Σ is a given covariance matrix).

Let

$$p_{\mathbf{X}} = \mathcal{N}(\mathbf{Z}, \Sigma)$$

This is a complicated (intractable) distribution

In this case

$$p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{z} = \mathbf{Z}(\omega), \Sigma)$$

So conditional on $\mathbf{Z} = \mathbf{z}$, $\mathbf{X}$ is simply a Gaussian where the mean is a sample from the Random Variable $\mathbf{Z}$

$$\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu \sim \mathbf{Z}$$

How could we find the density $p_{\mathbf{X}}$?

If we know $p_{\mathbf{Z}}$ and $p_{\mathbf{X}|\mathbf{Z}}$ we can theoretically compute $p_{\mathbf{X}}$

$$p_{\mathbf{X}}(\mathbf{x}) = \int p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{p_{\mathbf{Z}}(\mathbf{z})}(p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{z}))$$

Remark that in the present case, for a fixed $\mathbf{x}$,

$$\mathcal{N}(\mathbf{x}|\mathbf{z}, \Sigma) = \mathcal{N}(\mathbf{z}|\mathbf{x}, \Sigma)$$

so in this case

$$p_{\mathbf{X}}(\mathbf{x}) = \int \mathcal{N}(\mathbf{z}|\mathbf{x}, \Sigma) p_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z}$$

Inference from a dataset

$$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$$

Given a sample

$$\mathbf{z} = \mathbf{Z}(\omega)$$

we can calculate

$$p_{\mathbf{X}}(\mathcal{D}|\mathbf{Z} = \mathbf{z}) = \prod_{\mathbf{x} \in \mathcal{D}} p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{Z} = \mathbf{z})$$

This is known as the *likelihood*

If we have a *prior* distribution $p_{\mathbf{Z}}$ we can use Bayes' rule

$$p(\mathbf{z}|\mathcal{D}) = \frac{p_{\mathbf{X}|\mathbf{Z}}(\mathcal{D}|\mathbf{z})}{p_{\mathbf{X}}(\mathcal{D})} p_{\mathbf{Z}}(\mathbf{z})$$

$$p_{\mathbf{X}}(\mathcal{D}) = \int p_{\mathbf{X}|\mathbf{Z}}(\mathcal{D}|\mathbf{z}) p_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z} = \int \prod_{\mathbf{x}\in\mathcal{D}} p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z}$$

This integral is in general intractable and so we can't actually compute the *posterior distribution* $p(\mathbf{z}|\mathcal{D})$

So we have to rely on approximation methods.

If we can't compute the posterior distribution can we at least approximate it with some tractable distribution?

This is where *Variational Inference* comes in

First of all, what does it mean to approximate a probability distribution by another probability distribution?

This requires that we have some measure of 'closenes' between distributions

One such measure is the *Total Variation* between distributions

$$TV(q,p) = \sup_A \int_A |p-q|$$

This is in fact a distance measure on the space of distributions over the same space

Suppose we want to approximate p by some parametrized distribution q_ϕ.

So we want to find

$$\operatorname*{argmin}_\phi TV(q_\phi, p)$$

The problem is that $TV(q_\phi, p)$ is not differentiable with respect to ϕ so we can't use derivatives to find a minimum

Kullbach-Liebler Divergence

Instead we use another measure: the Kullbach-Liebler Divergence, defined by

$$D_{KL}(q||p) = -\int \log \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} = -\mathbb{E}_q(\log \frac{p}{q})$$

While the KL-divergence is certainly differentiable with respect to parameters, it is not a true distance function e.g. it is not symmetric so in general

$$D_{KL}(q||p) \neq D_{KL}(p||q)$$

Some Properties of the KL-Divergence

$D_{KL}(q||p) \geq 0$

This follows from *Jensen's Theorem*:

If f is a concave function then

$$\mathbb{E}(f(\mathbf{X})) \leq f(\mathbb{E}(\mathbf{X}))$$

Since log is a concave function we get

$$\mathbb{E}_q(\log \frac{p}{q}) \leq \log \mathbb{E}_q(\frac{p}{q}) = \log \int \frac{p}{q} q = \log \int p = \log 1 = 0$$

So

$$D_{KL}(q||p) = -\mathbb{E}_q(\log \frac{p}{q}) \geq 0$$

$D_{KL}(q||p) = 0$ if and only if $p = q$ outside a set of measure 0

Clearly

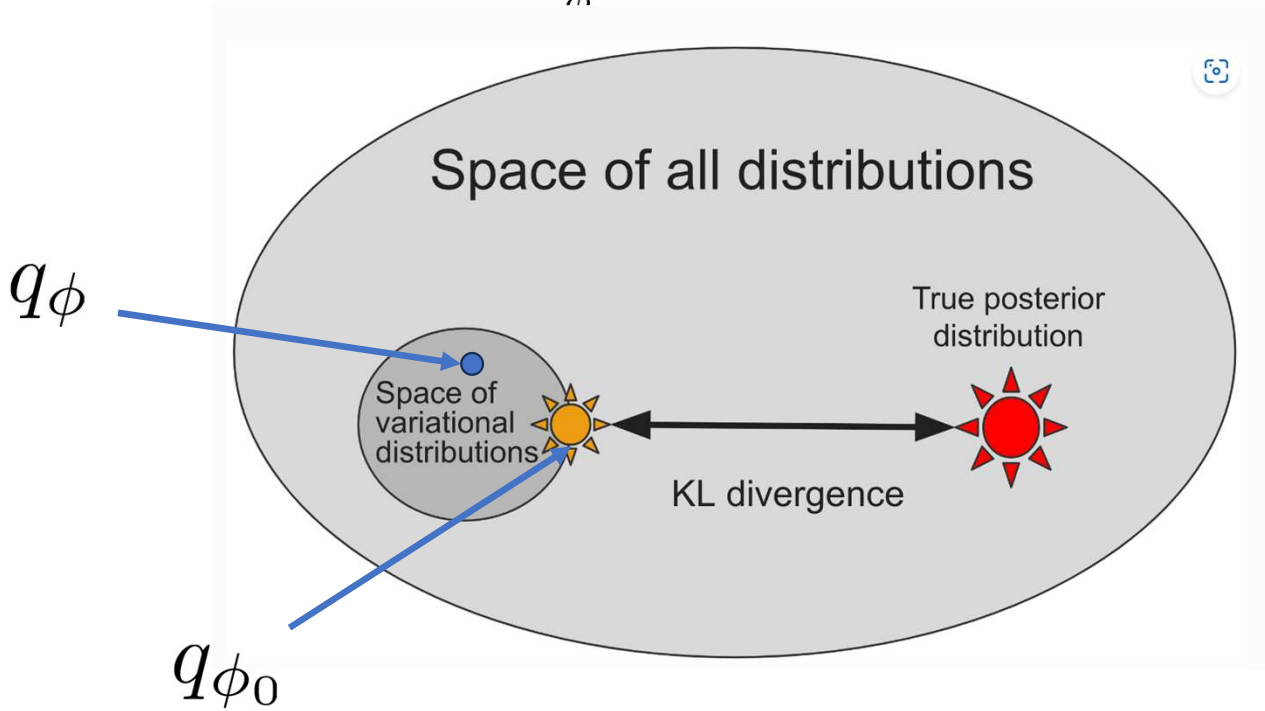
$$D_{KL}(q||q) = 0$$

The other direction follows from *Pinsker's Inequality*

$$TV(q, p) \leq D_{KL}(q||p))$$

The idea of Variational Inference is to choose a distribution from a family of parametrized distributions $\{q_\phi\}_{\phi \in \Phi}$ and find

$$\phi_0 = \underset{\phi}{\operatorname{argmin}}\ D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathcal{D}))$$



This may seem impossible since we obviously don't know $p(\mathbf{z}|\mathcal{D})$

We can write

$$D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}||\mathcal{D})) = -\mathbb{E}_{q_\phi}\left(\log \frac{p(\mathbf{z}|\mathcal{D})}{q_\phi(\mathbf{z})}\right)$$

and using Bayes' formula

$$\begin{aligned}\log \frac{p(\mathbf{z}|\mathcal{D})}{q_\phi(\mathbf{z})} &= \log \frac{p(\mathcal{D}|\mathbf{z})p_{prior}(\mathbf{z})}{p(\mathcal{D})q_\phi(\mathbf{z})} \\ &= \log p(\mathcal{D}|\mathbf{z}) + \log p_{prior}(\mathbf{z}) - \log p(\mathcal{D}) - \log q_\phi(\mathbf{z})\end{aligned}$$

Taking the negative expectation

$$
\begin{aligned}
& D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathcal{D})) \\
& = -\mathbb{E}_{q_\phi}(\log p(\mathcal{D}|\mathbf{z}) + \log p_{prior}(\mathbf{z}) - \log p(\mathcal{D}) - \log q_\phi(\mathbf{z})) \\
& = -\mathbb{E}_{q_\phi}(\log p(\mathcal{D}|\mathbf{z})) + \mathbb{E}_{q_\phi}(\log p(\mathcal{D})) - \mathbb{E}_{q_\phi}\left(\log \frac{p_{prior}(\mathbf{z})}{q_\phi(\mathbf{z})}\right)
\end{aligned}
$$

Now $p(\mathcal{D})$ does not depend on q_ϕ so

$$\mathbb{E}_{q_\phi}(\log p(\mathcal{D})) = \log p(\mathcal{D})$$

It follows that

$$D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathcal{D})) = -\mathbb{E}_{q_\phi}\left(\log p(\mathcal{D}|\mathbf{z})\right) + \log p(\mathcal{D}) + D_{KL}(q_\phi(\mathbf{z})||p_{prior}(\mathbf{z}))$$

and since $\log p(\mathcal{D})$ does not depend on θ we are reduced to minimizing

$$-\mathbb{E}_{q_\phi}\left(\log p(\mathcal{D}|\mathbf{z})\right) + D_{KL}(q_\phi(\mathbf{z})||p_{prior}(\mathbf{z}))$$

The expression

$$\mathbb{E}_{q_\phi}\left(\log p(\mathcal{D}|\mathbf{z})\right) - D_{KL}(q_\phi(\mathbf{z})||p_{prior}(\mathbf{z}))$$

is known as the $ELBO = \mathbf{E}vidence\ \mathbf{L}ower\ \mathbf{BO}und$

Rearranging we get

$$\log p(\mathcal{D}) = \underbrace{\mathbb{E}_{q_\phi}\left(\log p(\mathcal{D}|\mathbf{z})\right) - D_{KL}(q_\phi(\mathbf{z})||p_{prior}(\mathbf{z}))}_{ELBO} + D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathcal{D}))$$

Since $D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathcal{D})) \geq 0$ we get that

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q_\phi}\left(\log p(\mathcal{D}|\mathbf{z})\right) - D_{KL}(q_\phi(\mathbf{z})||p_{prior}(\mathbf{z})) = ELBO$$

so the $ELBO$ is indeed a lower bound for the *log-likelihood* $\log p(\mathcal{D})$ (which is also known as the *Evidence*)

The variational posterior can be used, in principle, to compute the probability of new data points, conditional on the observed data

If $\mathbf{x}^*$ is a new point in the data space, we can write

$$p(\mathbf{x}^*|\mathcal{D}) = \int p(\mathbf{x}^*, \mathbf{z}|\mathcal{D}) d\mathbf{z} = \int \frac{p(\mathbf{x}^*, \mathbf{z}, \mathcal{D})}{p(\mathcal{D})} d\mathbf{z} = \int p(\mathbf{x}^*|\mathbf{z}, \mathcal{D}) p(\mathbf{z}|\mathcal{D}) d\mathbf{z}$$

Our assumption is that the distribution conditional on the latent variable $\mathbf{z}$ is a known distribution $p(\mathbf{x}^*|\mathbf{z})$ independent of $\mathcal{D}$ so

$$p(\mathbf{x}^*|\mathbf{z}, \mathcal{D}) = p(\mathbf{x}^*|\mathbf{z})$$

Since

$$p(\mathbf{z}|\mathcal{D}) \approx p_{\theta_0}(\mathbf{z})$$

we get

$$p(\mathbf{x}^*|\mathcal{D}) \approx \int p(\mathbf{x}^*|\mathbf{z}) p_{\theta_0}(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{p_{\theta_0}}(p(\mathbf{x}^*|\mathbf{z}))$$

The expectation can be estimated by Monte Carlo approximation