Probabilistic Thinking (10/09/2026)

Notes by Paul Bouchaud.

Goal of generative modelling : Given iid samples $(x_1, x_2, ..., x_n) \sim p(x)$ learn to sample from $p(x)$.

Note: we write p interchangeably for the law, the probability or its density. This is not ideal, but everybody does it.

Fundamental Rules

Most of the computation we wil ldo in the class relies on the following 3 rules.

Bayes rule :
For two random variables X and Y, $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$.

Law of total probability :

$$p(x) = \int_y p(x,y)\,\mathrm{d}y = \int_y p(y)p(x|y)\,\mathrm{d}y$$

Jensen's inequality :
If f is a convex function, then for any random variable X, we have:

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$$

Level lines:

Visualize level lines of a function at https://mathurinm.github.io/levellines/

The Gaussian distribution :

This distribution is the most used and most important distribution in statistics and need to be know by heart. You cna play with the visualization at https://mathurinm.github.io/gaussian/

1D Gaussian distribution :
In 1D, the Gaussian distribution density function is given by:

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

Where $\mu \in \mathbb{R}$ is the mean and $\sigma^2 > 0$ is the variance.

Multivariate Gaussian distribution :
For a random vector $X \in \mathbb{R}^d$ with mean $\mu = \mathbb{E}[X]$, its covariance matrix is defined by:

$$\Sigma = \mathbb{E}[(X-\mu)(X-\mu)^T] \in \mathbb{R}^{d\times d}$$

This matrix is positive semidefinite (all its eigenvalues are greater or equal than 0). See https://mathurinm.github.io/assets/pdf/math_background.pdf

Reminder on covariance
Assume the components of X have finite second moments, and write $\mu_i = \mathbb{E}[X_i]$. The entry in row i, column j is the covariance between X_i and X_j:

$$\begin{aligned}\Sigma_{ij} &= \mathbb{E}\big[(X_i - \mu_i)(X_j - \mu_j)\big] \\ &= \mathbb{E}\big[X_i X_j\big] - \mu_j \mathbb{E}[X_i] - \mu_i \mathbb{E}\big[X_j\big] + \mu_i \mu_j \\ &= \mathbb{E}\big[X_i X_j\big] - \mathbb{E}[X_i]\mathbb{E}\big[X_j\big].\end{aligned}$$

In particular, on the diagonal of Σ lie the variances of the coordinates of X:

$$\Sigma_{ii} = \sigma_i^2 = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2.$$

The diagonal entries are the variances of the individual components; σ_i is their standard deviation. The off-diagonal entries describe how pairs of components vary together. Equivalently,

$$\mathbb{E}\big[X_i X_j\big] = \Sigma_{ij} + \mu_i \mu_j, \quad \mathbb{E}[X_i^2] = \sigma_i^2 + \mu_i^2.$$

These identities hold for any random vector with finite second moments, not only Gaussian vectors.

Multivariate Gaussian density

For $\mu \in \mathbb{R}^d$ and a positive definite matrix $\Sigma \in \mathbb{R}^{d \times d}$, we say that the random variable X follows a Gaussian distribution with mean μ and covariance Σ, and we write $X \sim \mathcal{N}(\mu, \Sigma)$, if its density is

$$p(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right), \quad x \in \mathbb{R}^d.$$

The factor before the exponential normalizes the density so that it integrates to one. The quadratic expression $(x-\mu)^T \Sigma^{-1}(x-\mu)$ measures squared distance from the mean after accounting for the variances and correlations.

A covariance matrix is always symmetric and positive semidefinite. The density above requires it to be positive definite, hence invertible. A singular covariance matrix describes a degenerate Gaussian supported on a lower-dimensional subspace, which has no density with respect to volume in $\mathbb{R}^d$.

Special case: Isotropic Gaussian distribution

An isotropic Gaussian is a Gaussian that has the same variance in every direction. Its covariance matrix is $\Sigma = \sigma^2 I_d$, where I_d is the identity matrix and $\sigma > 0$. Thus,

$$\Sigma_{ii} = \sigma^2, \quad \Sigma_{ij} = 0 \quad \text{for } i \neq j.$$

Its components are independent Gaussians with the same variance, though their means may differ. Since $\det(\Sigma) = \sigma^{2d}$ and $\Sigma^{-1} = \frac{1}{\sigma^2} I_d$, the density simplifies to

$$p(x) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left(-\frac{\|x-\mu\|^2}{2\sigma^2}\right).$$

This density depends only on the Euclidean distance from μ, so its contours are spheres (circles in two dimensions). The special case $\mu = 0$ and $\Sigma = I_d$ is the standard multivariate Gaussian $\mathcal{N}(0, I_d)$.

Passage on the vizualization of the Gaussian distribution in 2D and 3D at the following link : (https://mathurinm.github.io/levellines/ and https://mathurinm.github.io/gaussian/)

What was discussed when the website was shown :

Definition level Lines : Things you see on a map that represent a moutains, it means that everything is the same height. Used to show 2D representation of a 3D object.

On the 2D representation of a Gaussian distribution. Increasing and decreasing the means of each of the two variables will move the center of the distribution. (Right/up if increase, left/down if decrease)

Increasing and decreasing the diagonal values of the matrix (the variances) will change the spread of the distribution. Increasing make it wider and decreasing make it thinner.

Increasing and decreasing the off-diagonal values of the matrix (the covariances) will change the orientation of the distribution.

If the non diagonal values are positive then the two dimension are positively corralated (they increase together), if they are negative then they are negatively correlated (one increases while the other decreases).

However, if the non diagonal values goes further than the squared root of the product of the diagonal values, then the covariance matrix is no longer positive definite and the distribution is no longer valid.

Another Tools for Gaussian !

Let $x_1, x_2, ..., x_n \in \mathbb{R}^d$ be independent and identically distributed observations from a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. The maximum likelihood estimator of the mean vector μ is the sample mean:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T.$$

This help when we want to find out the mean of a Gaussian distribution from a set of samples.

Mixture of K Gaussians (GMM)

Definition : A mixture of K gaussians is.a r.v whose density is

$p(x) = \sum_{k=1}^{K} \pi_k p_{\mathcal{N}}(x \mid \mu_k, \Sigma_k))$

Where π_k is the mixing coefficient of the k-th Gaussian distribution (sum to one), and $p_{\mathcal{N}}(x \mid \mu_k, \Sigma_k)$ is the density of the k-th Gaussian distribution with mean μ_k and covariance matrix Σ_k.

Why is MLE Gaussian useful for generative modelling ?

Because it allows us to estimate the parameters of a Gaussian distribution from a set of samples, which can then be used to generate new samples from the estimated distribution. This is particularly useful in generative modelling, where we want to learn the underlying distribution of the data and generate new data points that are similar to the original data.

BaD news, it's not the case for image. Except for very simple images, for exemple the number image does works with GMM (Ask professor for proof)

So what happened if we believe the data is a Gaussian Mixture ?

I look for theta = (pi_1, ..., pi_K, mu_1, ..., mu_K, Sigma_1, ..., Sigma_K) that maximize the likelihood of the data.

(Reminder MLE =

$$\arg\max \sum \log p(x_i \mid \theta)$$

)

But not computable in practice because we don't know which Gaussian generated which points

But we can approximate it ! It relies on the EM algorithm (Expectation Maximization)

Introduction of the EM algorithm :

The idea is to introduce a latent variable y_i for each observation x_i, which indicates which Gaussian generated the observation. Of course this variable is unknown (Definition of latent variable) after all if we know it, we would know which Gaussian generated which point and so just use the previous MLE formula.

If theta is known, estimating y_i (the law of) is easy. (If closer to one gaussian than the other, then it is more likely to be generated by this one)

$P(y = k \mid x = x_i) = \frac{\pi_k p_{\mathcal{N}}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j p_{\mathcal{N}}(x_i \mid \mu_j, \Sigma_j)}$

Proof :

$$p(y = k \mid x = x_i) = p(x = x_i \mid y = k)\frac{p(y=k)}{p(x=x_i)}$$

$$\begin{aligned} p(x) &= \sum p(x, y = k) \\ &= \sum p(x \mid y = k)p(y = k) \end{aligned}$$

Variational Auto Encoder (VAEs)

Auto Encoder (AE)

An autoencoder is a function from $\mathcal{R}^d$ to $\mathcal{R}^d$ as a neural network. You start by making smaller and smaller layers until you reach a botteneck layer, then you make the layer bigger and bigger until you get back to the original size. The idea is that the bottleneck layer will learn a compressed representation of the input data. This space is called the latent space.

What you want is that f is approximately the identity function so that the output is as close as possible to the input.

Training : $\min_\theta \sum_{\{i=1\}}^{n} \|x_i - D_e\left(E_{\theta(x_i)}\right)\|^2$

So we can use the encoder to decompresse a random distribution in the latent space to generate new samples.

However there is no proof that the generated samples will be similar to the original data.

Variational Auto Encoder (VAE)

This is where the variational autoencoder comes in. We assume that an observation $x \in \mathbb{R}^d$ is generated from a latent variable $z \in \mathbb{R}^k$. We choose a standard Gaussian prior and a Gaussian distribution for x given z:

$$z \sim \mathcal{N}(0, I_k), \quad x \mid z \sim \mathcal{N}\big(D_{\theta(z)}, I_d\big).$$

Here, D_θ is the decoder network. The **posterior** $p_{\theta(z \mid x)}$ describes the possible latent values given an observation, while the **evidence** $p_{\theta(x)}$ is the marginal density of that observation. Choosing the prior and the likelihood determines both, through the law of total probability and Bayes' rule.

How do we find good parameters θ? We want to maximize the log likelihood of the data:

$$\hat{\theta} = \arg\max_\theta \sum_{i=1}^{n} \log p_{\theta(x_i)},$$

where

$$\log p_{\theta(x)} = \log\left(\int_z p_{\theta(x \mid z)} p(z)\, \mathrm{d}z\right).$$

This integral is generally **intractable**: computing it directly is too difficult in practice. This also makes the true posterior $p_{\theta(z \mid x)}$ difficult to compute.

We therefore introduce an **approximate posterior** $q_{\varphi(z \mid x)}$, represented by the encoder. We choose it to be Gaussian:

$$q_{\varphi(z \mid x)} = \mathcal{N}\big(z; \mu_{\varphi(x)}, \Sigma_{\varphi(x)}\big).$$

Here, the Gaussian density is evaluated at z, and the encoder outputs its mean $\mu_{\varphi(x)}$ and covariance $\Sigma_{\varphi(x)}$. The parameters φ belong to the encoder, while θ belongs to the decoder.

Using Jensen's inequality, we obtain a lower bound on the log likelihood:

$$\log p_{\theta(x)} \geq \mathbb{E}_{q_{\varphi(z \mid x)}}\left[\log\left(\frac{p_{\theta(x,z)}}{q_{\varphi(z \mid x)}}\right)\right] = \mathcal{L}(\theta, \varphi; x).$$

This is the **ELBO** (Evidence Lower Bound). Instead of maximizing the log likelihood directly, we maximize this bound over both θ and φ, summed over the observations, to learn the decoder and encoder together.

Jensen's inequality for a probability density
Let $K \subset \mathbb{R}^d$ and let p be a probability density on K, so that

$$p(x) \geq 0 \quad \wedge \quad \int_K p(x)\,\mathrm{d}x = 1.$$

For every convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$ and every integrable function $f : K \to \mathbb{R}^d$, Jensen's inequality gives

$$\varphi\left(\int_K f(x)p(x)\,\mathrm{d}x\right) \leq \int_K \varphi(f(x))p(x)\,\mathrm{d}x.$$

Equivalently, if X has density p, then

$$\varphi(\mathbb{E}[f(X)]) \leq \mathbb{E}[\varphi(f(X))].$$

Proof of the ELBO
For any approximate posterior $q_{\varphi(z \mid x)}$, we can rewrite the evidence as

$$\log p_{\theta(x)} = \log\left(\int_z q_{\varphi(z \mid x)} \frac{p_{\theta(x,z)}}{q_{\varphi(z \mid x)}}\,\mathrm{d}z\right) \setminus \geq \int_z q_{\varphi(z \mid x)} \log\left(\frac{p_{\theta(x,z)}}{q_{\varphi(z \mid x)}}\right)\mathrm{d}z.$$

The inequality follows from Jensen's inequality, because the logarithm is concave. The right-hand side is the ELBO:

$$\mathcal{L}(\theta, \varphi; x) = \mathbb{E}_{q_{\varphi(z \mid x)}}\left[\log\left(\frac{p_{\theta(x,z)}}{q_{\varphi(z \mid x)}}\right)\right]$$

Kullback–Leibler divergence
For probability densities q and p, the KL divergence is

$$\mathrm{KL}(q \parallel p) = \int q(z)\log\left(\frac{q(z)}{p(z)}\right)\mathrm{d}z \geq 0.$$

Using Bayes' rule, the log evidence can be decomposed as

$$\log p_{\theta(x)} = \mathcal{L}(\theta, \varphi; x) + \mathrm{KL}\Big(q_{\varphi(z \mid x)} \parallel p_{\theta(z \mid x)}\Big).$$

Therefore,

$$\mathcal{L}(\theta, \varphi; x) = \log p_{\theta(x)} - \mathrm{KL}\Big(q_{\varphi(z \mid x)} \parallel p_{\theta(z \mid x)}\Big),$$

which explains why the ELBO is a lower bound on the log evidence.

We now only need to maximize the ELBO over both θ and φ, summed over the observations, to learn the decoder and encoder together.