Gaussians

Contents

1 A Reminder in Probability **1**

2 Gaussian Distribution **4**

3 Fitting the Gaussian Distribution **8**

4 Gaussian Mixture Model (GMM) **10**

1 A Reminder in Probability

In this section we'll remind ourselves a bit of the basics of probability. Most of the definitions here will not be rigorous, and this is because it isn't the primary topic of... whatever this is.

1.1 Discrete Probability

The easiest place to begin is with discrete probability, however we will quickly depart into the realm of continuous probabilities, since that's where we'll dwell most of the time. In the discrete scheme, we assume that there is a space of "states" which we will call Ω for now. In this space of "states", let's assume that there are different discrete "states", i.e.:

$$\Omega = \{\omega_1, \omega_2, \omega_3, ...\}$$

The size of Ω may be finite or infinite, at this point it doesn't really matter.

The Probability Function

A *probability function* on Ω is any function $P : \Omega \to [0, 1]$ that satisfies the following[1]:

$$\sum_{\omega \in \Omega} P(\omega) = 1 \tag{1}$$

A very basic example would be a die; the state space would be $\Omega = \{1, 2, 3, 4, 5, 6\}$ and the probability function for $\omega \in \Omega$ is simply $P(\omega) = 1/6$, as you would expect.

Random Variables

This leads us to the next object of interest, the *random variable*. A random variable is a variable (that we define) that can take any value and is a placeholder for a state from Ω. This is confusing but intuitive if you have an example. Again, in the die example, we can define a random variable $X = \{1, 2, 3, 4, 5, 6\}$. Now, if we want to see how probable a singe value of X is, we can ask $P(X = \omega) =?$ which in this example will be like asking $P(\omega) =?$. This seems like a rather technical and unimportant point, but will help us later on. For now, we will abbreviate $P(X = \omega)$ into simply $P(\omega)$ - this is a typical abbreviation and is a bit more intuitive. A natural way to think about random variables is as a possible outcome of an experiment with the possible states Ω and the probability for each state $P(\omega)$.

[1] Note that, while not stated explicitly, $0 \leq P(\omega) \leq 1$ for any ω and there exists $\omega \in \Omega$ such that $P(\omega) > 0$. This is implied by the image of the function and eq. 1

Of course, we can have multiple random variables at once. For two random variables X and Y, we will say that they are independent if and only if:

$$P(X = x \text{ and } Y = y) \triangleq P(x, y) = P(X = x) P(Y = y)$$

The way to think about this is "the outcome of experiment X does not effect the outcome of experiment Y". This is not always true, of course.

Expected Value, Variance and Covariance

A useful quantity to get acquainted with is the *expected value* of a random variable (or its *expectation*):

$$\mathbb{E}[X] \triangleq \sum_{\omega \in \Omega} P(X = \omega) \cdot \omega \tag{2}$$

This quantity tells us what value will be the mean of an infinite number of experiments with the random variable x. The expected value is a linear function, in other words:

$$\mathbb{E}[X + aY] = \mathbb{E}[X] = a\mathbb{E}[Y]$$

This will prove to be extremely helpful in the near future.

A few more definitions that we will need in the near future are *variance* and *standard deviation*, *covariance*:

1. Variance:
$$\text{Var}[X] \triangleq \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}^2[X]$$
2. Standard deviation:
$$\sigma_X \triangleq \sqrt{\text{Var}[X]}$$
3. Covariance:
$$\begin{aligned}\text{Cov}[X, Y] &\triangleq \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \\ &= \text{Cov}[Y, X]\end{aligned}$$
In this case, if X is independent of Y, then $\text{Cov}[Y, X] = 0$

Conditionality

There are cases where if we know the outcome of one experiment, this greatly changes the possible outcomes of another experiment. This is called conditionality, and specifically if we know the value of a random variable Y and want to check the probability for any outcome of the random variable X, we will denote this probability as:

$$P(X = x | Y = y) \triangleq P(x|y)$$

Note that just from the definition we see that if X is independent from Y, then $P(x|y) = P(x)$. In general:

$$P(x|y) = \frac{P(x, y)}{P(y)}$$

Finally, from this we can derive the following formulas:

1. $P(x, y) = P(x|y) P(y)$
2. Bayes' Theorem: $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$
3. Law of total probability[2]: $P(x) = \sum_y P(x|y) P(y)$
4. Law of total expectation: $\mathbb{E}[X] = \sum_y \mathbb{E}[X|Y = y] P(y)$

And many other cases which will reveal themselves later on.

[2] This holds only if all possible states of Y partition Ω completely, but this will hold in most of the uses we'll see, so I've omitted it for now

1.2 Continuous Variables

Up until now, all of the definitions we saw were for discrete variables, but when we try to extend this to continuous variables the definitions fall flat. For one, if we assume that there are an infinite number of "states" with non-zero probability, then $\sum_{\omega} P(\omega) > 1$. Otherwise, if we assume that they all have 0 probability, then $\sum_{\omega} P(\omega) = 0$. Clearly, then, this definition for probability is not adequate. Indeed, we will look at a different function to determine how likely values of our continuous variables are - the *probability density function* (PDF). In a moment we'll see how this relates to the definitions we saw before, for now we'll define a PDF as any function $f : X \to \mathbb{R}$ such that $\forall x \; f(x) \geq 0$ that satisfies:

$$\int_{-\infty}^{\infty} f(x)\, dx = 1$$

This last condition is misleading as f can achieve values that are greater than 1 as long as the integration of all of its image is equal to 1. The reason this is called a density function is because we can now think of the probability that the function lies within a range of its values, i.e.:

$$P(a \leq X \leq b) = \int_{a}^{b} f(x)\, dx$$

So X is more likely to reside in areas where f is denser, but we still get $P(X = a) = \int_a^a f(x)\, dx = 0$, which makes sense in continuous variables. Note that f is not the probability function, only a middle man towards understanding the probability of finding the random variable in a range of values.

We can now extend the definition of the expected value to continuous variables as well:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \cdot f(x)\, dx \quad \text{"}\approx\text{"} \sum_x xP(x)$$

and can even extend the definition to functions of the random variable:

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)\, f(x)\, dx$$

All of the other definitions we have seen above follow from this scheme and all of the properties still hold.

One more thing we should define is the *joint probability*; the PDF of two random variables x and y is defined as the function $f : \mathbb{R}^2 \to \mathbb{R}$ such that $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, dxdy = 1$ and is non-negative. If x and y are independent, then

$$f(x, y) = f(x)\, f(y)$$

This is exactly as we saw with independent discrete variables, only now we have integrals everywhere (but we'll get used to them).

1.3 Using a Lot of Variables

There are some cases where we want to look at the probability of a lot of random variables at the same time. This could be, as a simple example, throwing n dice together and looking at the probability that each of them lands on a specific number. The way we defined the probability space above, we would need to look at the function $P(x_1, x_2, ..., x_n)$ which might be cumbersome to write. Instead, we can define a *random vector*. This is a vector that is comprised of random variables that is

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

where $x_1, x_2, ..., x_n$ are all random variables. In essence, this is nothing but a new way to notate what is actually going on. In this case, we will write

$$P(x_1, x_2, x_3, ..., x_n) \triangleq P(\boldsymbol{x})$$

so that we'll have a more concise way to denote the joint probability. This is of course the same for discrete and continuous variables.

Using this notation, we can define the *covariance matrix* Σ as follows:

$$\Sigma_{ij} = \mathrm{Cov}[x_i, x_j]$$

There are several things to note here, from this definition. First, the diagonal of Σ is actually the variance of each of the random variables, that is $\Sigma_{ii} = \mathrm{Var}[x_i]$. Second, notice that Σ must be symmetrical. Another important aspect is that if $\Sigma_{ij} = 0$ then x_i and x_j are independent. Finally, the covariance matrix is a *positive semi-definite* (PSD) matrix.

Positive Definitiveness of the Covariance Matrix

A PSD is matrix A has the following important quality:

$$\forall x \neq 0 \quad x^T A x \geq 0$$

This is as a result of the definition of the covariance. While this is always true, we will only think of the instances where the covariance matrix is a *positive definite* (PD) matrix, i.e. when:

$$\forall x \neq 0 \quad x^T A x > 0$$

The reason behind this is that if there exists a non-trivial vector x such that $x^T A x = 0$, this means that at least on of the values of the diagonal of A is equal to 0, i.e. there's a random variable with no variance - it is constant. This means that there is a random variable we can "drop", which will make our lives easier. Easier how? If a matrix is PD, it is invertible.

Further properties that PD matrices have are:

- All of the eigenvalues of the matrix are positive $\forall i \quad \lambda_i > 0$ (for PSD matrices they are non-negative)
- All PSD matrices have a root $A = RR^T$. If the matrix is PD, this root is unique. In both cases, the root can R is a lower triangular matrix (and the decomposition $A = RR^T$ is called the Cholesky decomposition)

1.4 Change of Variables

The final part in our probability chapter will be devoted to looking into the case where we want to change from one random vector to another. We can think of this as a sort of change of variable.

Recall that from the definition of our probability and PDF, they are defined on a specific random variable/set of random variables. Suppose that we have two (different) random vectors x and y with respective PDFs f_x and f_y, and that we have a function $g : X \to Y$ such that $y = g(x)$ that is one-to-one and differentiable. It would be ideal if we could somehow represent the probability of the random variable y using the same PDF as x. In this case, the following equation will be useful:

$$f_y(y) = f_x\left(g^{-1}(y)\right) \left| J\left(g^{-1}(y)\right) \right| \tag{3}$$

where $|J(g(y))|$ is the determinant of the Jacobian of the function g at the point y. This all seems rather difficult, but will help us later on in proving some basic properties in Gaussian distributions.

2 Gaussian Distribution

The Gaussian distribution is a distribution that widely used since it has certain favorable aspects that most distributions do not. For instance, the marginals of a Gaussian is a Gaussian, the product of i.i.d. Gaussians is a Gaussian, sampling is relatively easy, a linear transformation of a Gaussian is a Gaussian and on and on. The easiest place to start is in a single dimension.

2.1 1D Gaussian Distribution

The PDF of the Gaussian distribution is defined as:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{4}$$

For any $\sigma \geq 0$ and $\mu \in \mathbb{R}$. Since this will come up a lot, we will rewrite the above as:

$$p(x) \triangleq \mathcal{N}\left(x;\, \mu, \sigma^2\right) \Leftrightarrow x \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

So anytime you see something like $\mathcal{N}\left(\mu, \sigma^2\right)$, it will be a placeholder for eq. 4. The two most important properties of the Gaussian distribution are it's mean and standard deviation:

$$\mathbb{E}[x] = \mu$$
$$\sqrt{\mathrm{Var}[x]} = \sigma$$

and these are the only parameters needed to define the distribution. Further, we can easily see that the distribution is symmetrical and centered around μ and is uni-modal - it has only one mode. In this case that means that:

$$\mathbb{E}[x] = \arg\max_x p(x) = \mathrm{median}(x)$$

which are all useful properties for a distribution.

Add a visualization of the Gaussian distribution here.

A special case of the Gaussian distribution is called the *normal distribution*, which you have probably heard of. The normal distribution is

$$z \sim \mathcal{N}(0, 1)$$

and in fact every other Gaussian distribution is a linear transformation of the normal distribution (we will see this later on).

2.2 Multivariate Gaussian Distribution

The multivariate Gaussian distribution, also called Multivariate Normal Distribution (MVN) sometimes, is a generalization of the 1D case of the Gaussian distribution. This distribution is also completely defined by the mean and the variance, but since we are in the territory of multivariate variables (in other words, random vectors), we will need to be careful of how we define everything.

Let's start in the simplest generalization we can. Suppose that we have n different, **independent**, Gaussian random variables:

$$x_1, x_2, ..., x_n \sim \mathcal{N}(\mu_1, \sigma_1), \mathcal{N}(\mu_2, \sigma_2), ..., \mathcal{N}(\mu_n, \sigma_n)$$

We can, very simply, stack them together into a vector:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

But, how will this vector's distribution look? Let's check:

$$\begin{aligned} p(x) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2}\frac{(x_i-\mu_i)^2}{\sigma_i^2}\right] \\ &= \frac{1}{(2\pi)^{n/2}} \frac{1}{\prod_i \sigma_i} \exp\left[-\frac{1}{2}\sum_i \frac{(x_i-\mu_i)^2}{\sigma_i^2}\right] \end{aligned}$$

This doesn't seem to help us all that much... However, notice that if we define:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \sigma_n^2 \end{bmatrix} = I \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_n^2 \end{bmatrix}$$

as well as:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}$$

then we get:

$$\sum_i \frac{(x_i - \mu_i)^2}{\sigma_i^2} = (x-\mu)^T \Sigma^{-1} (x-\mu)$$

This greatly decreases the amount we need to write. Also, notice how:

$$\prod_i \sigma_i = \sqrt{\prod_i \sigma_i^2} = \sqrt{|\Sigma|}$$

Suddenly, what we have before can be written in very simply matrix algebra as:

$$p(x) = \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right] \tag{5}$$

And this is the definition of the MVN!

In fact, eq. 5 is also the equation for a general MVN, even when the x_is are not independent, with one key difference: Σ becomes the covariance matrix. We will not explicitly derive this, just recall that the covariance matrix is defined as $\Sigma_{ij} = \mathrm{Cov}[x_i, x_j]$, that it is symmetrical and PD, meaning it is invertible. We can now change the notation again, so that it will become something slightly more compact:

$$p(x) = \mathcal{N}(x;\ \mu, \Sigma) \Leftrightarrow x \sim \mathcal{N}(\mu, \Sigma)$$

And here we really see why we constrained ourselves to PD covariance matrices - if it was not, the expression in eq. 5 would be undefined as the inverse of Σ would not be defined if it were not PD but only PSD (i.e., there exists an eigenvalue that equals 0).

Just to drive the point home, of course for any $x \sim \mathcal{N}(\mu, \Sigma)$ the following holds:

$$\mathbb{E}[x] = \mu \qquad \mathrm{Cov}[x] = \Sigma$$

and if is symmetrical. Since the PDF is once again symmetrical, we can try and think about the shape that would be the result of choosing all of the points with a probability higher than some value, i.e. looking at the shape defined by $P(x \geq a)$ for some a. In the multivariate case this shape becomes an ellipsoid, and in 2D it is an ellipse. In the special case when

$$\Sigma = \sigma^2 I$$

in other words when all of the random variables are independent and have the same variance, this becomes a ball.

Mahalanobis Distance

In the above section we have derived the PDF of the MVN and stumbled across a distance metric that is useful to know. This distance metric is the Mahalanobis distance and is defined as:

$$D_M(x; y) = \sqrt{(x - \mathbb{E}[y])^T \mathrm{Cov}[y]^{-1} (x - \mathbb{E}[y])}$$

where y is some random variable. This is a distance metric that tries to check how far the point x is from the distribution of y. In fact, the square of this distance is exactly how we will check what the likelihood of a point x is to be part of a Gaussian y, since it is what appears in the exponent of $p(y)$.

2.3 Sampling from the Gaussian Distribution

To show how to sample new points from an MVN, we will first look at an interesting property of the Gaussian distribution - a linear transformation of a Gaussian is, itself, a Gaussian.

Linear Transformations of the Gaussian Distribution

Given $x \sim \mathcal{N}(\mu, \Sigma)$, we want to know what the distribution of $y = Ax + b$ is, for any invertible A and any b. Recall from eq. 3 that if $y = f(x)$ and f is invertible, then we will be able to quantify the PDF of y in terms of the PDF of x. In this case, $f(x) = Ax + b$ so the inverse can be easily found:

$$\begin{aligned} y &= Ax + b \\ \Rightarrow y - b &= Ax \\ \Rightarrow A^{-1}(y - b) &= x \\ \Rightarrow f^{-1}(y) &= A^{-1}(y - b) \end{aligned}$$

We also need to check what the Jacobian of f^{-1} is[3]:

$$\frac{\partial}{\partial y} A^{-1}(y - b) = A^{-1}$$

Now we can find the distribution of y in the terms of the distribution of x:

$$\begin{aligned} p_y(y) &= \left|J\left(f^{-1}(y)\right)\right| p_x\left(f^{-1}(y)\right) \\ &= \left|J\left(A^{-1}(y - b)\right)\right| p_x\left(A^{-1}(y - b)\right) \\ &= \frac{1}{|A|}\mathcal{N}\left(A^{-1}(y - b);\quad \mu, \Sigma\right) \end{aligned}$$

Here I have used the fact that $\left|A^{-1}\right| = |A|^{-1}$. Okay, now we have to actually show that this PDF becomes a Gaussian distribution. Let's only look at the term that will be in the exponent for a moment:

$$\left(A^{-1}(y - b) - \mu\right)^T \Sigma^{-1}\left(A^{-1}(y - b) - \mu\right) = (y - b - A\mu)^T A^{-1T}\Sigma^{-1}A^{-1}(y - b - A\mu)$$

Let's denote $b + A\mu \equiv \eta$. At the same time, notice that since $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$, then $A^{-1T}\Sigma^{-1}A^{-1} = \left(A\Sigma A^T\right)^{-1}$, so the term in the exponent becomes:

$$= (y - \eta)^T\left(A\Sigma A^T\right)^{-1}(y - \eta)$$

Suddenly this seems very close to our original Gaussian distribution, right? Since this is the term in the exponent, we know have to show that the covariance in the normalizing term is also $A\Sigma A^T$:

$$\begin{aligned} \frac{1}{|A|} \cdot \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} &= \frac{1}{|A|^{1/2}|A^T|^{1/2}} \cdot \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \\ &= \frac{1}{\sqrt{(2\pi)^n |A||\Sigma||A^T|}} \\ &= \frac{1}{\sqrt{(2\pi)^n |A\Sigma A^T|}} \end{aligned}$$

In these equations have used the identities of $|A| = \left|A^T\right|$, as well as the fact that $|A|$ is a scalar, so I can move it about freely. Finally, I used the identity $|AB| = |A||B|$.

Let's write everything together:

$$p_y(y) = \frac{1}{\sqrt{(2\pi)^n |A\Sigma A^T|}} \exp\left[-\frac{1}{2}(y - \eta)^T\left(A\Sigma A^T\right)^{-1}(y - \eta)\right]$$

This is very clearly a Gaussian distribution, and in fact we can now write $y \sim \mathcal{N}\left(\eta, A\Sigma A^T\right) = \mathcal{N}\left(A\mu + b, A\Sigma A^T\right)$. Now we know that for any invertible A and b, we can transform one Gaussian distriubtion to another. We will show an even stronger bond - every Gaussian distribution is a linear transformation of the normal distribution.

[3] You should check this yourself. Finding the derivatives of matrices is, in general, troublesome. However, many of them are recorded in the wikipedia page Matrix Calculus or can be found in the Matrix Cookbook.

Transformation of the Normal Distribution

To show that any Gaussian distribution is a linear transformation of the normal distribution, we have to show that $x = Az + b$ where $x \sim \mathcal{N}(\mu, \Sigma)$ and $z \sim \mathcal{N}(0, I)$. Let's rewrite this transformation explicitly:

$$x = Az + b \Rightarrow x \sim \mathcal{N}\left(A0 + b, AIA^T\right) = \mathcal{N}\left(b, AA^T\right)$$

Then very simply we see that $b = \mu$. Σ is a bit more complicated, but not by much. Remember how, in 1.3 we said that Σ is a PD matrix? Well, any PD matrix has a root, also called the Cholesky decomposition so that:

$$\Sigma = RR^T$$

Now it is clear that $A = R$ must hold. In addition, for PD matrices this root is unique. Finally, we can say that any Gaussian distribution x is a linear transformation of the normal distribution:

$$\begin{aligned} z &\sim \mathcal{N}(0, I) \\ x \sim \mathcal{N}(\mu, \Sigma) &= \mathcal{N}\left(\mu, RR^T\right) \\ &\Rightarrow x = Rz + \mu \end{aligned} \tag{6}$$

How does this help us? Well, notice that the multivariate case of the normal distribution is just like sampling n times from $\mathcal{N}(0, 1)$. So under the assumption that you know how to sample numbers from $\mathcal{N}(0, 1)$, sampling from $\mathcal{N}(0, I)$ is easy. This means that you can now actually sample from any Gaussian distribution as long as you know what the root of the covariance is, using the following simple formula:

$$\begin{aligned} \epsilon &\leftarrow \begin{bmatrix} \epsilon_1 \sim \mathcal{N}(0, 1) \\ \epsilon_2 \sim \mathcal{N}(0, 1) \\ \vdots \\ \epsilon_n \sim \mathcal{N}(0, 1) \end{bmatrix} \\ x &\leftarrow R\epsilon + \mu \end{aligned}$$

3 Fitting the Gaussian Distribution

While it is neat to know all that we saw about Gaussian distributions, what we actually want to use it for is to fit such a distribution to a set of data points.

3.1 Maximum Likelihood Estimation (MLE)

MLE is a method for fitting data to a model. Specifically, what we are trying to do is to find a model that says that the given data points have very high probability of appearing.

First, let us define what *likelihood* even is. Given a model with parameters θ (for instance, μ and Σ for a Gaussian), we will notate the PDF slightly differently to how we have done so far:

$$p_\theta(x) \triangleq p(x; \theta)$$

So, for instance, if we want a model with a Gaussian distribution, we could write:

$$p(x; \mu, \Sigma) = \mathcal{N}(x;\ \mu, \Sigma)$$

Why this notation? Well, the probability of x depends on the values of the parameters θ. What this notation is actually trying to say is "the probability of x conditional on the parameters θ". So, if we have different parameters, the probability of a data point may change. This notation is a bit weird, as stated, but it will later on help in reminding us what the parameters are and what the random variables are. The likelihood of a set of data points $x = \{x_i\}_{i=1}^N$ is defined (and notated) as:

$$L(x; \theta) \triangleq \prod_{i=1}^{N} p(x_i; \theta)$$

Again, let us use the Gaussian distribution as an example:

$$L\left(x;\ \mu,\Sigma\right)=\prod_{i=1}^{N}p\left(x_i;\ \mu,\Sigma\right)=\prod_{i=1}^{N}\mathcal{N}\left(x_i;\ \mu,\Sigma\right)$$

Great, so now we know what the likelihood of a set of data points is. In general, it will be easier to look at the *log-likelihood* instead of the likelihood:

$$\ell\left(x;\ \theta\right)\triangleq\log\left(L\left(x;\ \theta\right)\right)=\sum_{i=1}^{N}\log\left(p\left(x_i;\ \theta\right)\right)$$

As you can see, we made now have the sum of many elements instead of the product of those elements, which will help in the algebra later on.

By now you can probably guess what MLE will try to do. What we are going to do is to try and find:

$$\hat{\theta}=\arg\max_{\theta}L\left(x;\ \theta\right)=\arg\max_{\theta}\ell\left(x;\ \theta\right)$$

This seems very intuitive when you think about it. As said before, we are trying to find a model that gives the maximal likelihood on the set of points x.

3.2 MLE on the Gaussian Distribution

Suppose you are given a set of points $x=\{x_i\}_{i=1}^{N}$ and you want to find the Gaussian that fits these data points the best. We can do this using MLE, and because the Gaussian distribution is concave it is even quite simple. Whenever you are trying to find the maxima of a function, the easiest way to do it is by finding the point whose derivative is equal to 0, of course. We will do exactly this, for each parameter of the Gaussian distribution (there are two). First, let's see what the log-likelihood according to the Gaussian distribution even is:

$$\begin{aligned}\ell\left(x;\ \mu,\Sigma\right)&=\sum_{i}\log\mathcal{N}\left(x_i;\ \mu,\Sigma\right)\\&=\sum_{i}\left[-\frac{d}{2}\log\left(2\pi\right)-\frac{1}{2}\log\left|\Sigma\right|-\frac{1}{2}\left(x_i-\mu\right)^{T}\Sigma^{-1}\left(x_i-\mu\right)\right]\\&=-N\frac{d}{2}\log\left(2\pi\right)-\frac{N}{2}\log\left|\Sigma\right|-\frac{1}{2}\sum_{i}\left(x_i-\mu\right)^{T}\Sigma^{-1}\left(x_i-\mu\right)\end{aligned}$$

Now, let us derive by μ:

$$\begin{aligned}\frac{\partial}{\partial\mu}\ell\left(x;\ \mu,\Sigma\right)&=-\frac{1}{2}\frac{\partial}{\partial\mu}\sum_{i}\left(x_i-\mu\right)^{T}\Sigma^{-1}\left(x_i-\mu\right)\\&=-\frac{1}{2}\sum_{i}\frac{\partial}{\partial\mu}\left(x_i-\mu\right)^{T}\Sigma^{-1}\left(x_i-\mu\right)\\&=-\frac{1}{2}\sum_{i}\frac{\partial}{\partial\mu}\left[\mu^{T}\Sigma^{-1}\mu-2x_i^{T}\Sigma^{-1}\mu\right]\\&=-\frac{1}{2}\sum_{i}\left[2\Sigma^{-1}\mu-2\Sigma^{-1}x_i\right]\\&=-\Sigma^{-1}\sum_{i}\left[\mu-x_i\right]\end{aligned}$$

Since we want to find the maximum, we will equate this to 0:

$$
\begin{aligned}
-\Sigma^{-1} \sum_i [\mu - x_i] &\stackrel{!}{=} 0 \\
\sum_i [\mu - x_i] &= 0 \\
\sum_i \mu &= \sum_i x_i \\
N\mu &= \sum_i x_i \\
\Rightarrow \hat{\mu} &= \frac{\sum_i x_i}{N}
\end{aligned}
$$

Unsurprisingly, the MLE of μ is the empirical mean of all of the data points. Let's do the same for Σ:

$$
\begin{aligned}
\frac{\partial}{\partial \Sigma} -\frac{N}{2} \log |\Sigma| &= \frac{N}{2} \Sigma \\
\frac{\partial}{\partial \Sigma} -\frac{1}{2} \sum_i (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) &= -\frac{1}{2} \sum_i (x_i - \mu)(x_i - \mu)^T \\
\frac{\partial}{\partial \Sigma} \ell(x;\ \mu, \Sigma) &= \frac{N}{2} \Sigma - \frac{1}{2} \sum_i (x_i - \mu)(x_i - \mu)^T
\end{aligned}
$$

The term $(x_i - \mu)(x_i - \mu)^T$ is the outer product of the two vectors, which is a matrix. Again, equating to 0:

$$
\begin{aligned}
\frac{N}{2} \Sigma - \frac{1}{2} \sum_i (x_i - \mu)(x_i - \mu)^T &\stackrel{!}{=} 0 \\
N\Sigma &= \sum_i (x_i - \mu)(x_i - \mu)^T \\
\Rightarrow \hat{\Sigma} &= \frac{\sum_i (x_i - \mu)(x_i - \mu)^T}{N}
\end{aligned}
$$

Once more, unsurprisingly, the MLE of Σ is the empirical covariance of all of the data points.

So, the MLE solution for a Gaussian distribution given a set of points is:

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{N} \sum_i x_i \\
\hat{\Sigma} &= \frac{1}{N} \sum_i (x_i - \hat{\mu})(x_i - \hat{\mu})^T
\end{aligned}
\tag{7}
$$

As you can see, the result isn't very surprising, but we still needed to go through the steps to make sure we get to the right update rule.

4 Gaussian Mixture Model (GMM)

While Gaussians are nice, they are very limited. In this chapter we will look at more general distributions called GMMs. As suggested by the name, these distributions are just a combination of a bunch of Gaussians and are very versatile. Of course, the downside of using a more complicated model is that optimizing it is also much more difficult, but we'll go into details regarding that later.

4.1 Intro to GMMs

The PDF of a Gaussian mixture model is defined as follows:

$$
p(x;\ \theta) = \sum_{k=1}^{K} p_k \mathcal{N}(x;\ \mu_k, \Sigma_k) \qquad s.t. \quad \sum_{k=1}^{K} p_k = 1 \tag{8}
$$

where K is a positive integer that we can think of as a hyper-parameter of the model. Notice that when $K = 1$, this distribution is simply a Gaussian (as you would expect). The parameters of this model are $\theta = \{p_k, \mu_k, \Sigma_k\}_{k=1}^K$ and we will denote them using θ for the sake of brevity.

The important thing to note is that the p_ks are essentially weights that each Gaussian receives, and are themselves a discrete multinomial probability. In fact, we can rewrite the above with k as a random variable as well:

$$p(x, k;\ \theta) = p_k \mathcal{N}(x;\ \mu_k, \Sigma_k)$$

and this would read as "the probability that x came from the kth Gaussian". Of course, summing out the random variable k from this joint distribution, we get:

$$p(x;\ \theta) = \sum_k p(x, k;\ \theta) = \sum_{k=1}^K p_k \mathcal{N}(x;\ \mu_k, \Sigma_k)$$

which coincides with the initial definition exactly. Just to confuse everyone, these k Gaussians are sometimes called *clusters* or *modes*, so please don't be surprised if I suddenly use one of these names.

Since we will want to fit a GMM to a data set, the place to start is by checking what the log-likelihood of the distribution is:

$$\begin{aligned}
\ell(x;\ \theta) &= \sum_{i=1}^N \log\left(p(x_i;\ \theta)\right) \\
&= \sum_i \log\left(\sum_k p_k \mathcal{N}(x_i;\ \mu_k, \Sigma_k)\right)
\end{aligned}$$

... and we're in trouble. The sum inside the logarithm will really make our lives miserable now, as simply deriving this expression will be a disturbing affair. Let's contrast this expression to what we had for a single Gaussian:

$$\ell(x;\ \mu, \Sigma) = \sum_i \log \mathcal{N}(x_i;\ \mu, \Sigma)$$

... yup, that's much easier. So we will now acknowledge the fact that we have a problem. We will overcome it, but first let's look at a simplified version of this problem.

4.2 Supervised GMM

Let's look at a special case of the problem we had before. Suppose that instead of just getting the data points $x = \{x_i\}_{i=1}^N$ we also got know to which Gaussian each of these points belongs to, i.e. we also get $\kappa = \{k_i\}_{i=1}^N$. Using this information, let's look at the joint probability of both of these variables for a moment:

$$p(x_i, k_i;\ \theta) = p_{k_i} \mathcal{N}(x_i;\ \mu_{k_i}, \Sigma_{k_i})$$

This is exactly like the joint probability we wrote before. However, now we can write the log-likelihood of the whole data set a bit differently:

$$\begin{aligned}
\ell(x, \kappa;\ \theta) &= \sum_i \log\left(p(x_i, k_i;\quad \theta)\right) \\
&= \sum_i \log\left(p_{k_i} \mathcal{N}(x_i;\ \mu_{k_i}, \Sigma_{k_i})\right) \\
&= \sum_i \left[\log p_{k_i} + \log \mathcal{N}(x_i;\ \mu_{k_i}, \Sigma_{k_i})\right]
\end{aligned}$$

Unlike before, this term is **really** easy to derive. Actually, let's rewrite this in a more complicated manner. Suppose that instead of k_i, we get the vectors $z_i \in \mathbb{R}^K$ whose coordinates are defined as:

$$z_{ik} = \begin{cases} 1 & k = k_i \\ 0 & k \neq k_i \end{cases}$$

If we get these, we are getting exactly the same information as the k_is, just in a more complicated manner. Now, we can write the log-likelihood as (defining $z = \{z_i\}_{i=1}^N$):

$$\begin{aligned}
\ell(x, z;\ \theta) &= \sum_i \log\left[\sum_k z_{ik} p(x_i, k;\ \theta)\right] \\
&= \sum_i \log\left[z_{ik_i} p(x_i, k_i;\ \theta)\right] \\
&= \sum_i z_{ik_i} \log p(x_i, k_i;\ \theta) \\
&= \sum_i \sum_k z_{ik} \log p(x_i, k;\ \theta) \\
&= \sum_i \sum_k z_{ik} \left[\log p_k + \log \mathcal{N}(x_i;\ \mu_k, \Sigma_k)\right]
\end{aligned}$$

In all of the above steps I used the definition of z_{ik} to my advantage. But why did we write it in this manner? Well, later on this will help us to understand what is going on, although I agree that for now it's just a more convoluted way to write the same thing.

MLE for Supervised GMM

Let's find the MLE solution for this distribution (later on we will consider how to do this if we don't have the z_is). As we have already written the log-likelihood, let's directly go on to the deriving stage. I'll write $\#_k = \sum_i z_{ik}$ as the number of points x_i such that $k_i = k$, since we will need this soon.

Starting with p_k, since we have the restriction $\sum_k p_k = 1$, we will use Lagrange multipliers to find the maximizing values. First, let's define the Lagrangian:

$$\mathcal{L}(x, z;\ \theta) = \ell(x, z;\ \theta) - \lambda\left(\sum_k p_k - 1\right)$$

Now, we will derive the Lagrangian according to p_k and λ is order to find the MLE of p_k:

$$\begin{aligned}
\frac{\partial}{\partial p_k}\mathcal{L}(x, z;\ \theta) &= \frac{\partial}{\partial p_k}\ell(x, z;\ \theta) - \lambda\frac{\partial}{\partial p_k}\left(\sum_k p_k - 1\right) \\
&= \sum_i z_{ik}\frac{1}{p_k} - \lambda \overset{!}{=} 0 \\
\Rightarrow p_k &= \frac{\sum_i z_{ik}}{\lambda}
\end{aligned}$$

Actually, we don't even need to derive by λ:

$$\begin{aligned}
&\sum_k p_k = \sum_k \frac{\sum_i z_{ik}}{\lambda} = \frac{1}{\lambda}\sum_k\sum_i z_{ik} = \frac{N}{\lambda} = 1 \\
&\Rightarrow \lambda = N
\end{aligned}$$

So, the MLE for p_k is:

$$\hat{p_k} = \frac{1}{N}\sum_i z_{ik} = \frac{\#_k}{N}$$

This makes a lot of sense - the weight of the kth Gaussian is simply equal to the portion of data points that arrived from it.

Let's continue with μ_k:

$$\begin{aligned}\frac{\partial}{\partial \mu_k}\ell\left(x,z;\ \theta\right) &= -\frac{1}{2}\sum_i z_{ik}\frac{\partial}{\partial \mu_k}\left(x_i-\mu_k\right)^T\Sigma_k^{-1}\left(x_i-\mu_k\right)\\ &= -\sum_i z_{ik}\Sigma_k\left(\mu_k-x_i\right)\stackrel{!}{=}0\\ \Rightarrow &\mu_k\sum_i z_{ik} = \sum_i z_{ik}x_i\\ \Rightarrow &\hat{\mu}_k = \frac{1}{\#_k}\sum_i z_{ik}x_i\end{aligned}$$

Again, this makes a lot of sense, since the only points that should affect the kth Gaussian are those that are part of it, so we get the empirical mean of the points that originated from the kth Gaussian.

Finally, let's look at Σ_k:

$$\begin{aligned}\frac{\partial}{\partial \Sigma_k}\ell\left(x,z;\ \theta\right) &= -\frac{1}{2}\sum_i z_{ik}\frac{\partial}{\partial \Sigma_k}\left[\left|\Sigma_k\right|+\left(x_i-\mu_k\right)^T\Sigma_k^{-1}\left(x_i-\mu_k\right)\right]\\ &= -\sum_i z_{ik}\left[\Sigma_k-\left(x_i-\mu_k\right)\left(x_i-\mu_k\right)^T\right]\stackrel{!}{=}0\\ \Rightarrow &\Sigma_k\sum_i z_{ik} = \sum_i z_{ik}\left(x_i-\mu_k\right)\left(x_i-\mu_k\right)^T\\ \Rightarrow &\hat{\Sigma}_k = \frac{1}{\#_k}\sum_i z_{ik}\left(x_i-\hat{\mu}_k\right)\left(x_i-\hat{\mu}_k\right)^T\end{aligned}$$

And again, this makes a lot of sense.

Overall, the MLE updates for this problem are:

$$\begin{aligned}\hat{p}_k &= \frac{\#_k}{N}\\ \hat{\mu}_k &= \frac{1}{\#_k}\sum_i z_{ik}x_i\\ \hat{\Sigma}_k &= \frac{1}{\#_k}\sum_i z_{ik}\left(x_i-\hat{\mu}_k\right)\left(x_i-\hat{\mu}_k\right)^T\end{aligned}$$

This reads very simply as fitting a single Gaussian, separately, to all of the points for each k, and then gluing them together with p_k which is simply equal to the size of the cluster.

Classification for Supervised GMM

The model, as described, can be used for classification. Suppose that after learning the optimal parameters θ, you are given a new point y, and want to know what the probability of it belonging to cluster k is. We can use Bayes Theorem for this:

$$\begin{aligned}p\left(k|y;\ \theta\right) &= \frac{p\left(y,k;\ \theta\right)}{p\left(y;\ \theta\right)}\\ &= \frac{p\left(y,k;\ \theta\right)}{\sum_k p\left(y,k;\ \theta\right)}\\ &= \frac{p_k\mathcal{N}\left(y;\ \mu_k,\Sigma_k\right)}{\sum_{k'}p_{k'}\mathcal{N}\left(y;\ \mu_{k'},\Sigma_{k'}\right)}\end{aligned}$$

So we have a pretty simple expression for classifying as well as for fitting.

Sampling from a Supervised GMM

Something even nicer we can do using this distribution is to sample new points according to the parameters we have learned. To sample we want to create a point y that will have the probability

$p(y;\ \theta)$. Let's see how we can do this. First, notice that if we decide on some k, then sampling from the kth Gaussian is straightforward (using eq. 6). In fact what we want to do is to sample y and k, and we can rewrite this as:

$$p(y, k;\ \theta) = p(k;\ \theta)\, p(y|k;\ \theta)$$

That is, first we sample k proportionally to the value of p_k, and then we sample y from the kth Gaussian.

In fact the manner of sampling from an unsupervised GMM is exactly the same, as the equations are the same. So, after we fit the model, we can essentially generate new points so that the original distribution is still the same as it was.

4.3 Expectation Maximization (EM)

The reason we could use MLE directly on the unsupervised GMM is because there was a variable that was "hidden" from us - the z_is. As we saw, if we do know the z_is, then the solution is rather straightforward and can be calculated analytically. In general, this is not the case.

A popular optimization method for problems with this exact setup is *Expectation Maximization*. In EM, what we will do is to try and "fill in" the gaps in our knowledge iteratively as best as we can given the parameters of the current iteration, and update the parameters according to the what we would "expect" to find in those hidden variables. This sounds very intuitive, but the math can be much less intuitive, as we all know.

EM Algorithm

The algorithm is an iterative algorithm and has 2 steps in each iteration. First, for the setup, suppose we are trying to maximize the log-likelihood of a certain model $\ell(x;\ \theta) = \sum_i \log p(x_i;\ \theta)$, but don't know how to maximize it directly. Now, suppose that there is some variable z, that if you would have had it's value, you could maximize the joint log-likelihood quite easily. So $\ell(x;\ \theta)$ is hard to maximize, but $\ell(x, z;\ \theta)$ is easy to maximize. The EM algorithm is defined as follows:

Algorithm 1 EM Algorithm

θ^0 =some parameter initialization
for t=1, ..., T:

E step: $Q\left(\theta|\theta^{t-1}\right) \triangleq \mathbb{E}_{z|x;\ \theta^{t-1}}\left[\ell(x, z;\ \theta)\right]$
M step: $\theta^t = \arg\max_\theta Q\left(\theta|\theta^{t-1}\right)$

return θ^T

To do the above, we need to recall that:

$$\text{Discrete:}\quad \mathbb{E}_{x|y}\left[f(x)\right] = \sum_x f(x)\, p(x|y)$$

$$\text{Continuous:}\quad \mathbb{E}_{x|y}\left[f(x)\right] = \int_{-\infty}^{\infty} f(x)\, p(x|y)\, dx$$

But except for this, assuming that $\ell(x, z;\ \theta)$ is easy to maximize, should be easy to solve.

A nice example for the (almost) EM algorithm is the k-means algorithm. In k-means, in each step you update the cluster only according to the points that are closer to that cluster than other clusters. This is like filling in gaps regarding which cluster the data point belongs to, and then using this information that is initially hidden from you to update the parameters. Of course, this isn't exactly what the EM algorithm does, but it's very similar.

An important note about the EM algorithm: it is not promised that the maxima is reached, only that each iteration improves the log-likelihood. So it is quite probable that the algorithm will get stuck in a local maxima. The EM algorithm is also very sensitive to the initial conditions. The usual ways to try and get around this can be used here, for instance many different initializations for the model

parameters, keeping only the model that performed the best. Another way is to choose the initial conditions in a "smart" manner.

4.4 GMM

Finally we have all the pieces in order to find out how to update the parameters of the general GMM. First, let's write the full (joint) log-likelihood from 4.2:

$$\ell\left(x, z;\ \theta\right) = \sum_i \sum_k z_{ik} \left(\log p_k + \log \mathcal{N}\left(x_i;\ \mu_k, \Sigma_k\right)\right)$$

Now, recall that the z_is are our hidden variables in this case.

E Step

Let's check what the E-Step looks like for arbitrary parameters θ:

$$\mathbb{E}_{z|x;\,\theta}\left[\ell\left(x, z;\ \theta\right)\right] = \sum_i \sum_k \mathbb{E}_{z_i|x_i;\,\theta}\left[z_{ik}\right] \left(\log p_k + \log \mathcal{N}\left(x_i;\ \mu_k, \Sigma_k\right)\right)$$

Here I used the fact that the expected value is a linear function as well as the fact that all the x_is are independent from each other, as well as all the z_is from their peers. Let's look more closely at the term in the middle there:

$$\mathbb{E}_{z_i|x_i;\,\theta}\left[z_{ik}\right] = \sum_{k'} z_{ik} p\left(z_{ik}|x_i;\quad \theta\right)$$

Remember that $z_{ik} = 1$ if $k' = k$, otherwise $z_{ik} = 0$, so the above term becomes:

$$\begin{aligned}\mathbb{E}_{z_i|x_i;\,\theta}\left[z_{ik}\right] &= p\left(z_{ik}|x_i;\quad \theta\right) = p\left(k|x_i;\quad \theta\right) \\ &= \frac{p_k \mathcal{N}\left(x_i;\ \mu_k, \Sigma_k\right)}{\sum_k p_k \mathcal{N}\left(x_i;\ \mu_k, \Sigma_k\right)} \triangleq c_{ik}\end{aligned}$$

So, at a certain iteration t, we have:

$$c_{ik} = p\left(k|x_i;\quad \theta^t\right)$$

and:

$$Q\left(\theta|\theta^t\right) = \sum_i \sum_k c_{ik} \left(\log p_k + \log \mathcal{N}\left(x_i;\ \mu_k, \Sigma_k\right)\right)$$

These c_{ik} are actually the weights the data point gives each cluster and are usually called the *responsibilities*. In this case, the responsibilities have a very clear role - if the data point is very close to the center of the kth cluster, it will give that cluster the highest weight and in turn the cluster will be affected by it quite a bit.

M Step

Once we hold $Q\left(\theta|\theta^t\right)$ we are tasked with finding $\hat{\theta} = \arg\max_\theta Q\left(\theta|\theta^t\right)$. In this case, we can (as usual) use MLE to find the optimal solution (for the iteration).

Starting with p_k, we must once again use Lagrange multipliers:

$$\mathcal{L} = Q\left(\theta|\theta^t\right) - \lambda\left(\sum_k p_k - 1\right)$$

$$\frac{\partial}{\partial p_k}\mathcal{L} = \sum_i c_{ik}\frac{1}{p_k} - \lambda \Rightarrow \hat{p_k} = \frac{1}{\lambda}\sum_i c_{ik}$$

$$\sum_k p_k = \frac{1}{\lambda}\sum_k \sum_i c_{ik} = \frac{1}{\lambda}N \overset{!}{=} 1 \Rightarrow \lambda = N$$

$$\Rightarrow \hat{p_k} = \frac{1}{N} \sum_i c_{ik}$$

Now the update for μ_k:

$$\begin{aligned}
\frac{\partial}{\partial \mu_k} Q\left(\theta|\theta^t\right) &= -\frac{1}{2} \sum_i c_{ik} \frac{\partial}{\partial \mu_k} \left(x_i - \mu_k\right)^T \Sigma_k^{-1} \left(x_i - \mu_k\right) \\
&= -\sum_i c_{ik} \Sigma_k \left(\mu_k - x_i\right) \stackrel{!}{=} 0 \\
\Rightarrow \mu_k \sum_i c_{ik} &= \sum_i c_{ik} x_i \\
\Rightarrow \hat{\mu}_k &= \frac{\sum_i c_{ik} x_i}{\sum_i c_{ik}}
\end{aligned}$$

Finally, the update for Σ_k:

$$\begin{aligned}
\frac{\partial}{\partial \Sigma_k} Q\left(\theta|\theta^t\right) &= -\frac{1}{2} \sum_i c_{ik} \frac{\partial}{\partial \Sigma_k} \left[|\Sigma_k| + \left(x_i - \mu_k\right)^T \Sigma_k^{-1} \left(x_i - \mu_k\right)\right] \\
&= -\sum_i c_{ik} \left[\Sigma_k - \left(x_i - \mu_k\right)\left(x_i - \mu_k\right)^T\right] \stackrel{!}{=} 0 \\
\Rightarrow \Sigma_k \sum_i c_{ik} &= \sum_i c_{ik} \left(x_i - \mu_k\right)\left(x_i - \mu_k\right)^T \\
\Rightarrow \hat{\Sigma}_k &= \frac{\sum_i c_{ik} \left(x_i - \hat{\mu}_k\right)\left(x_i - \hat{\mu}_k\right)^T}{\sum_i c_{ik}}
\end{aligned}$$

All of these updates make a lot of sense - they are the weighted mean/covariance of the data points, where their weights are the probability that they even belonged to that cluster to begin with. Let's put them all together:

$$\begin{aligned}
\hat{p_k} &= \frac{1}{N} \sum_i c_{ik} \\
\hat{\mu}_k &= \frac{\sum_i c_{ik} x_i}{\sum_i c_{ik}} \\
\hat{\Sigma}_k &= \frac{\sum_i c_{ik} \left(x_i - \hat{\mu}_k\right)\left(x_i - \hat{\mu}_k\right)^T}{\sum_i c_{ik}}
\end{aligned}$$

And, actually, this occasion is special enough that we will even write the full algorithm:

Algorithm 2 EM Algorithm for GMM

`for` $k = 1, ..., K$`:`

$p_k = \frac{1}{K}$
$\mu_k \sim \mathcal{N}\left(0, I\epsilon\right)$
$\Sigma_k = I$

`for` t=1, ..., T`:`

$\forall i, k \quad c_{ik} \leftarrow \frac{p_k \mathcal{N}(x_i;\, \mu_k, \Sigma_k)}{\sum_k p_k \mathcal{N}(x_i;\, \mu_k, \Sigma_k)}$
`for` $k = 1, ..., K$`:`

$p_k \leftarrow \frac{1}{N} \sum_i c_{ik}$
$\mu_k \leftarrow \frac{\sum_i c_{ik} x_i}{\sum_i c_{ik}}$
$\Sigma_k \leftarrow \frac{\sum_i c_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_i c_{ik}}$

`return` $\{p_k, \mu_k, \Sigma_k\}_{k=1}^K$