Contents

1. What is cross-entropy?

There are different ways of understanding cross-entropy, one that I realized recently is to view it as entropy, but because of the cross in the name that suggests that the log probability should be the model distribution instead of the data distribution. The weighting is still from the data distribution. So it's like entropy

$$H(p) = -\sum_{x \in X} p(x) \log(p(x))$$

and when dealing with continuous values we get

$$H(p) = -\int_{x \in X} p(x) \log(p(x)) \, \mathrm{d}x$$

whereas cross-entropy is just

$$H(p, q) = -\int_{x \in X} p(x) \log(q(x)) \, \mathrm{d}x$$

and analogous with the discrete case. So we can see it only differs in the log probability distribution, where for the cross-entropy it constitutes the model probability distribution. Should mention that it can equivalently be defined in terms of expectation

$$H(p, q) = -\mathbb{E}_{x \sim p} \log q(x)$$

Furthermore, relating it to KL-divergence gives some more insight into interpreting it. While the information theoretic interpretation of entropy is

> Average number of bits needed to encode the data coming from a source p when we use model p, meaning same source.

and cross-entropy is

> The cross entropy is the average number of bits needed to encode data coming from a source with distribution p when we use model q.
>
> — Murphy

if we instead apply the KL-divergence interpretation we can get some more insights

$$D_{\mathrm{KL}}(p \parallel q) = \int_{x \sim p} \log \frac{p(x)}{q(x)} \, \mathrm{d}x = \mathbb{E}_{x \sim p}[\log p(x) - \log q(x)]$$

And since $p(x)$ is just fixed because we don't change the data it becomes a constant. Minimizing the KL-divergence is therefore the same thing as minimizing the cross-entropy

$$-\mathbb{E}_{x \sim p}[\log q(x)]$$

Interestingly, the minimum of KL-divergence is simply 0, meaning that the minimum value of cross-entropy is reached when $q(x) = p(x)$. This means that we get the best case scenario when our model can exactly model the data distribution, which is what we ideally want. However, in practise cross-entropy doesn't have a minimum value, which can cause issues if the model finds ways to cheat the cost function by getting infinite negative value. Regularization is one way to deal with this problem, which is addressed more in detail in chapter 8 in the book.

2. Connection between maximum likelihood and cross-entropy - section 6.2.1.1

I think this is well explained in chapter 5.5, but I will give a quick background on it in this section. MLE comes with log, it's just a common standard practise, because log helps with underflow problems and differentiation as it's easier to work with by skipping product rules where possible. So if we have a model p_{model}, then we can formulate the MLE as

$$\begin{aligned}\theta_{\text{ML}} &= \underset{\theta}{\operatorname{argmax}}\, p_{\text{model}}(X;\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \prod_i p_{\text{model}}(x_i;\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \sum_i \log p_{\text{model}}(x_i;\theta)\end{aligned}$$

where we want to find the optimal model parameters θ that maximizes the log likelihood of observing the data. Originally I like to think of the likelihood as coming from Bayes' theorem

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(x)}{p(x)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}} \propto p(x \mid \theta) = \text{likelihood}$$

where the likelihood is $p(x \mid \theta)$, even though it doesn't have to be of that form, just anything that is a function of the parameters θ while the data is fixed is called a likelihood function. In Figure 1 we can illustrate MLE as a point estimate, showing that it only focuses on finding one set of parameters, the optimal ones, that gives the best likelihood. If we use the Bayesian interpretation we can understand MLE as finding the model parameters that most likely generated the observed data X, and in DL lingo that would mean the parameters that best fit the data.

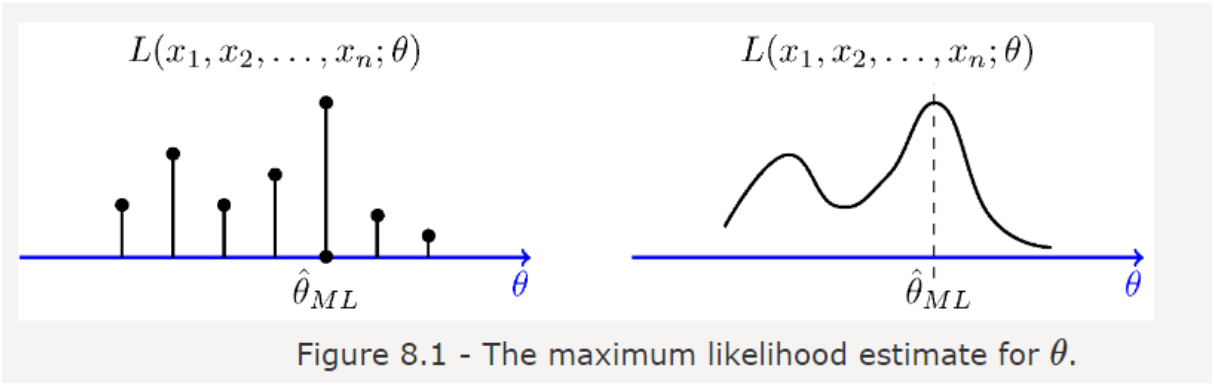

Figure 1: Maximum likelihood as a point estimate.

Before establishing the connection, recall that the definition of cross-entropy of p_{model} and p_{data} is

$$H(p_{\text{data}}, p_{\text{model}}) = -\mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(x;\theta)$$

Now going back to connecting MLE with cross-entropy, note that scaling a function doesn't change the optimum point of the function. So we can add a scaling term to turn it into something that starts to look like a cost function over some data

$$\underset{\theta}{\operatorname{argmax}} \frac{1}{n} \sum_i \log p_{\text{model}}(x_i;\theta) = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(x;\theta)$$

But maximizing is the same as minimizing the negative version of the expression, thus we have

$$\underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(x;\theta) = \underset{\theta}{\operatorname{argmin}}\, -\mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(x;\theta)$$

which is precisely the definition of cross-entropy over data x. The book went a bit further and showed that this can be interpreted as minimizing the KL-divergence as well, but I will skip for now to avoid clutter. The key idea is to fix the data as constant, then it will not affect the optimization and thus can be disregarded. Read chapter 5.5 if you want the full exposition. What has been shown here can also be easily extended to conditional distributions

$$\underset{\theta}{\operatorname{argmin}}\, -\mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(y \mid x,\theta)$$

where we haven't changed much other than converting $p_{\text{model}}(x;\theta)$ to $p_{\text{model}}(y \mid x,\theta)$ and they both are essentially the same. One added benefit of the probabilistic form is that we are now able to conveniently express it as finding all of the model parameters or only a subset of them if we decide for it. Also naturally, it's not more convenient to use the probabilistic form in a probabilistic setting.

The key idea of this particular section is that we simply flip the sign of the maximum log likelihood, change from argmax to argmin as a result and scale it with the number of data to get the cross-entropy, that's it. Otherwise the form of the cross-entropy is very similar to the general form of maximum likelihood.

3. Why mean squared error cost is the same as maximum likelihood on the mean of a gaussian distribution? - equation 6.13

Assume a function is described as

$$y = f(x;\theta) + \varepsilon$$

where $f(x;\theta)$ is any parametric model that produces real values and $\varepsilon \in N(0,\sigma^2)$ is gaussian distributed noise with a fixed variance σ^2. This is by the way a very sensible assumption for most parametric models, because of the central limit theorem that says that when the number of independent data from different distributions get large, the sum of these random variables tend to a gaussian distribution. Here we made the assumption that the sum of the noise from different distributions will tend towards a gaussian distribution when the number of data is large. We can denote $y \in N(f(x;\theta),\sigma^2)$ because of properties of gaussian distribution (you can prove this fact easily). Defining this probabilistically we get

$$p(y \mid x,\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(y_i - f(x;\theta))^2}{2\sigma^2}}$$

where we can see that our function now estimates the mean of the gaussian distribution. Note however, that our parametric model has another set of parameters of it's own θ, so by proxy of finding the best θ we will find the best μ. If we set up the maximum likelihood formulation of this function as a cost function, remembering that the variance σ^2 is fixed per the assumption and thus only wanting to find the mean $\mu = f(x;\theta)$ and therefore the θ, we get with cross-entropy (negative log likelihood)

$$
\begin{aligned}
\underset{\theta}{\operatorname{argmax}} \log p(y \mid x, \mu, \sigma^2) &= \underset{\theta}{\operatorname{argmin}} - \log p(y| \ x, \mu, \sigma^2) \\
&= \underset{\theta}{\operatorname{argmin}} - \log \prod_i p(y_i \mid x_i, \mu, \sigma^2) \\
&= \underset{\theta}{\operatorname{argmin}} - \log \prod_i \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{\left(y_i - f(x_i;\theta)\right)^2}{2\sigma^2}} \right] \\
&= \underset{\theta}{\operatorname{argmin}} - \sum_i \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{\left(y_i - f(x_i;\theta)\right)^2}{2\sigma^2}} \right] \\
&= \underset{\theta}{\operatorname{argmin}} - \sum_i \left[\log \frac{1}{\sqrt{2\pi\sigma^2}} + \log \exp^{-\frac{\left(y_i - f(x_i;\theta)\right)^2}{2\sigma^2}} \right] \\
&= \underset{\theta}{\operatorname{argmin}} - \sum_i \left[\log(1) - \log\left(\sqrt{2\pi\sigma^2}\right) - \frac{\left(y_i - f(x_i;\theta)\right)^2}{2\sigma^2} \right] \\
&= \underset{\theta}{\operatorname{argmin}} - \sum_i \left[0 - \frac{1}{2}\log(2\pi\sigma^2) - \frac{\left(y_i - f(x_i;\theta)\right)^2}{2\sigma^2} \right] \\
&= \underset{\theta}{\operatorname{argmin}} - \frac{1}{2}\sum_i \left[- \log(2\pi\sigma^2) - \frac{\left(y_i - f(x_i;\theta)\right)^2}{\sigma^2} \right] \\
&= \underset{\theta}{\operatorname{argmin}} - \sum_i \left[-\left(y_i - f(x_i;\theta)\right)^2 \right] \\
&= \underset{\theta}{\operatorname{argmin}} \sum_i \left[\left(y_i - f(x_i;\theta)\right)^2 \right]
\end{aligned}
$$

where we have dropped the variance σ terms and any scaling term because they don't contain the mean and don't affect the optimum point of the mean μ. We can see at the last line that we get the squared error, but once again, since any scaling done to the expression doesn't change the optimum point of the mean, we can easily add a scaling term back that gives the mean squared error like so

$$
\underset{\theta}{\operatorname{argmin}} \sum_i \left[\left(y_i - f(x_i;\theta)\right)^2 \right] \Rightarrow \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_i \left[\left(y_i - f(x_i;\theta)\right)^2 \right]
$$

where n is the size of the data x.

And there we have it, we have shown that a gaussian distribution with fixed variance under the maximum likelihood framework will produce the same optimization problem as the squared mean error. The **main take-away** is that anytime you use mean squared error on a function you are essentially applying maximum likelihood on a model that is assumed to have gaussian noise with zero mean and fixed variance, which effectively turns the problem into applying MLE on a gaussian distribution where we want to find the optimal mean of that gaussian distribution that is given by the model $\mu = f(x;\theta)$.

4. Why sigmoid saturation is bad for gradient?

Firstly anything that saturates is bad simply because the derivative becomes zero, so it's not limited to sigmoids. Regardless, I thought maybe it would be instructive to show why the derivative

becomes zero at these regions for sigmoid. Just take the derivative and it will be clear. The sigmoid function is given by

$$\sigma(z) = \frac{1}{1 + \exp^{-z}}$$

And the curve is illustrated in Figure 2. We can see that it saturates to 1 for large positive values and saturates to 0 for large negative values. Eyeballing it we see that the curve is completely flat at these saturated regions, meaning that the derivative is zero.

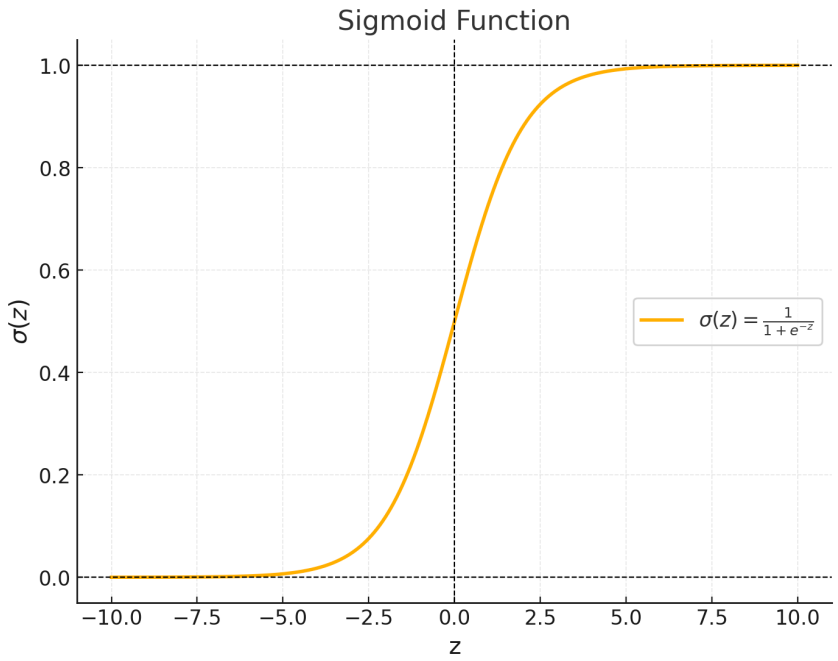

Figure 2: Sigmoid function curve.

One can formally show this by taking the derivative of sigmoid

$$\begin{aligned}
\frac{\partial \sigma(z)}{\partial z} &= \frac{\exp^{-z}}{(1 + \exp^{-z})^2} \\
&= \frac{1}{1 + \exp^{-z}} \cdot \frac{\exp^{-z}}{1 + \exp^{-z}} \\
&= \frac{1}{1 + \exp^{-z}} \cdot \frac{(1 + \exp^{-z}) - 1}{1 + \exp^{-z}} \\
&= \frac{1}{1 + \exp^{-z}} \cdot \left[\frac{1 + \exp^{-z}}{1 + \exp^{-z}} - \frac{1}{1 + \exp^{-z}}\right] \\
&= \sigma(z) \cdot (1 - \sigma(z))
\end{aligned}$$

And clearly we see that the gradient becomes zero when either $\sigma(z) = 0$ or $\sigma(z) = 1$, the same thing we saw when eyeballing the graph. The reason saturation is bad is because when a gradient becomes 0 then multiplying with that gradient will just give zero back, so anything that is immediately interacting with this gradient through multiplication will be set to 0. In worst case if the entire chain of functions is just a multiplication, or if the important parts of the chain is just multiplication that is affected by this zero gradient, then that chain will be killed off not contributing anymore to the overall gradient in the backprop. However, 0 gradient is not the only issue, very

small gradients close to zero are also an issue, because if they affect the important parts of the backprop, then the training will be slowed down significantly and possibly die.

5. Why multivariate gaussian restricts covariance to be symmetric positive definite - page 182 close to equation 6.17?

Recall the definition of multivariate gaussian

$$N(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}}|\Sigma|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right\}$$

Now, the simple answer is because it requires the covariance to be invertible, meaning it needs to have full rank, otherwise there is no inverse in the traditional sense. There's pseudoinverse, but I think in this chapter we are only concerned with non-degenerate cases for gaussian distributions, meaning matrices that have full rank. To prove why this requirement exists we will need to show three things:

1. Covariance matrix is symmetric
2. Covariance matrix is positive definite
3. Positive definite matrix is always invertible

Consider the definition of covariance matrix and note that we are taking expectation w.r.t a random vector X and not to μ and remember that $E[X] = \mu$ then we have

$$\begin{aligned}
\Sigma &= \mathbb{E}[(X-\mu)(X-\mu)^T] \\
&= E[XX^T - X\mu^T - \mu X^T + \mu\mu^T] \\
&= \mathbb{E}[XX^T] - \mathbb{E}[X]\mu^T - \mu\mathbb{E}[X^T] + \mu\mu^T \\
&= \mathbb{E}[XX^T] - \mu\mu^T - \mu\mu^T + \mu\mu^T \\
&= \mathbb{E}[XX^T] - \mu\mu^T
\end{aligned}$$

Moreover, we can show that it's symmetric by using the familiar rules of tranposition on matrix and vector products

$$\left(\mathbb{E}[XX^T] - \mu\mu^T\right)^T = E\left[\left(XX^T\right)^T\right] - \left(\mu\mu^T\right)^T = \mathbb{E}[XX^T] - \mu\mu^T$$

Now to show that the covariance matrix Σ is positive definite there needs to be $\forall z \neq 0$ such that

$$z^T\Sigma z > 0$$

Substituting the covariance matrix into the expression gives

$$\begin{aligned}
z^T\Sigma z &= z^T\mathbb{E}[(X-\mu)(X-\mu)^T]z \\
&= \mathbb{E}[z^T(X-\mu)(X-\mu)^Tz] \\
&= \mathbb{E}\left[\left((X-\mu)^Tz\right)^T\left((X-\mu)^Tz\right)\right] \\
&= \{\text{Notice that } (X-\mu)^T \text{ is a vector and } z \text{ is a vector as well, so } (X-\mu)^Tz \text{ gives a scalar}\} \\
&= \{\text{If we set } b = (X-\mu)^Tz \text{ then we get } b^T\cdot b = b\cdot b > 0 \text{ because same sign * sign = positive}\} \\
&= \mathbb{E}\left[\left((X-\mu)^Tz\right)^T\left((X-\mu)^Tz\right)\right] = \mathbb{E}[b^T\cdot b] = b^T\cdot b = b\cdot b > 0
\end{aligned}$$

This is true because $b\cdot b$ is a scalar which the expectation will treat as a constant and therefore not change the value. $b^T = b$ because it's a scalar as well, so transposing a scalar doesn't change the

value. Furthermore, we can push z^T and z into the expectation on the first line because they are constants in relation to X (we are taking expectation w.r.t X).

Finally, to show that a positive definite matrix is always invertible recall that a positive definite matrix has all it's eigenvalues non-zero positive $\lambda_i > 0$. This means that

$$Ax = \lambda x \neq 0 \cdot x$$

when $x \neq 0$, which implies that the equation system $Ax = 0$ in $Ax = \lambda x = 0$ can only have the trivial solution $x = 0$, as the eigenvalues are all non-zero positive. Therefore A must be invertible (a very basic rule in linear algebra that if Ax=0 only has trivial solution then A is invertible). Another way to prove this is to show that the product of eigenvalues of A is the determinant of A that is non zero

$$\det(A) = \prod_i \lambda_i \neq 0$$

because once again all eigenvalues are non-zero positive, and since the determinant of the positive definite matrix A is non-zero, then it's invertible (another very basic rule in linear algebra that if $\det(A) \neq 0$ then A is invertible). Therefore, we can conclude that all positive definite matrices are invertible.

This explains why the covariance matrix is restricted to being symmetric positive definite for multivariate gaussian distributions, because it acquires these properties as a consequence of the definition of covariance.

6. Maximum likelihood estimation on variance of gaussian distribution with fixed variance - page 188

6.1. Univariate gaussian

This will be for the univariate gaussian. I don't know how to derive it for the multivariate (I tried and failed). Recall the definition of univariate gaussian

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

Assume the variance is fixed, only we don't know what the optimal fixed variance is. Then applying MLE and as a result cross-entropy on it's log variant gives

$$-\log(p(y \mid x, \theta)) = -\sum_{i=1}^{N} \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right) = \frac{N}{2}\log(2\pi) + \frac{N}{2}\log(\sigma^2) + \sum_{i=1}^{N} \frac{(x-\mu)^2}{2\sigma^2}$$

Taking the derivative w.r.t σ to obtain the optimal variance gives

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\sigma}\log(p(y \mid x,\theta)) &= n\frac{2\sigma}{2\sigma^2} - \sum_{i=1}^{N}\frac{(x_i-\mu)^2}{\sigma^3} \\
&= \frac{n}{\sigma} - \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\mu)^2 \\
&= 0 \\
&\Rightarrow -\frac{n}{\sigma}\cdot -\sigma^3 = \sum_{i=1}^{N}(x_i-\mu)^2 \\
&\Rightarrow \sigma^2 = \frac{1}{n}\sum_{i=1}^{N}(x_i-\mu)^2
\end{aligned}
$$

which is what the author claimed on page 188.

6.2. Multivariate gaussian

Recall the definition of multivariate gaussian

$$
N(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}}|\Sigma|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right\}
$$

7. Show BB^T is positive definite - page 189

B is assumed to be a square matrix. For BB^T to be positive definite we need $\forall(x) \neq 0$ such that $x^TBB^Tx > 0$, which we can show by applying a transpose trick

$$
x^TBB^Tx = (x^TB)(B^Tx) = (B^Tx)^T(B^Tx) = v^Tv > 0
$$

where $v = B^Tx$ is a vector and naturally applying dot product between vector and itself will always be positive because $v^Tv = v_1^2 + ... + v_n^2 > 0$. Therefore BB^T is positive definite.

8. Gaussian mixture models

This is taken from one of my notebooks, which in turn has taken inspiration from Bishop and Prince. This section is mostly to give an intuition for gaussian mixture models.

The simplest way of thinking about mixture of gaussians is to think of them as a weighted sum of gaussians, with each of their own parameters. To demonstrate the power of gaussian mixture models (gmm) let's look at data collected from a geyser activity in Yellowstone

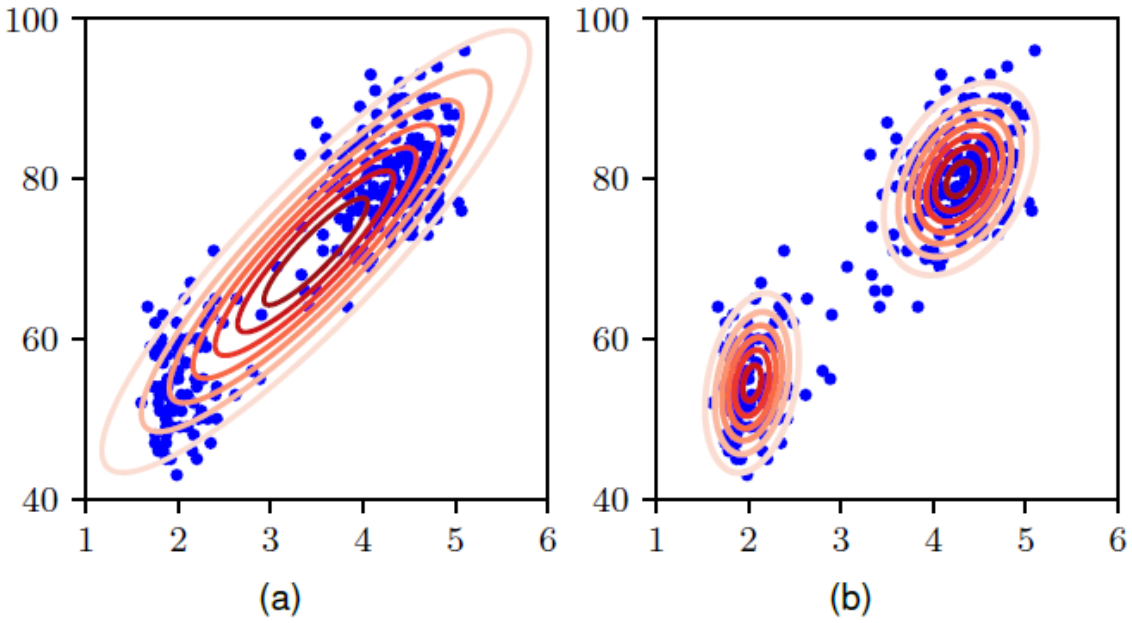

Figure 3: Yellow stone geyser activity captured by unimodal and multimodal gaussians.

x-axis is duration in minutes for an eruption and y-axis is minutes until next eruption. These are contour plots of the gaussian distributions. Blue dots are the data. Here we can see the limitations of a gaussian distribution that is only unimodal, simply because the data is distributed in a way that an unimodal gaussian would not be able to capture, since the peak in the middle (the very middle part of the gaussian distribution) is not representative of the peaks that otherwise would be on the very clustered areas (with lots of blue dots). However, a gmm is perfectly capable of capturing this distribution as it looks like there are two different gaussian distributions laying over the data with peaks at the right areas. Another one to show the intuition of combining gaussians is this 1D example

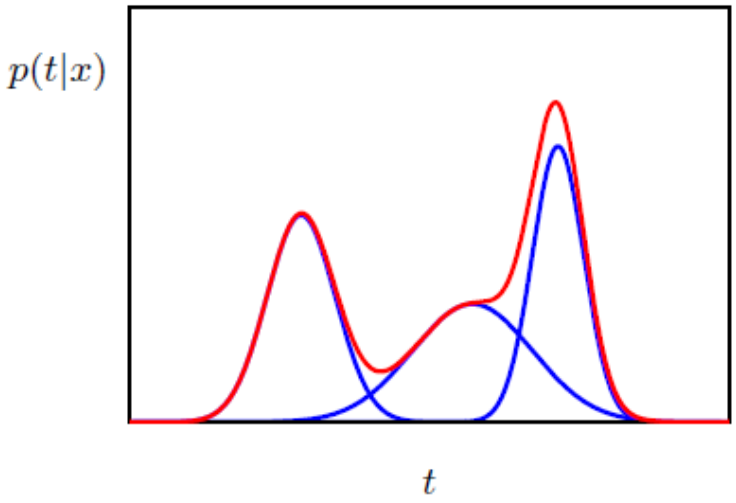

Figure 4: Combining 3 univariate gaussians.

Here we have 3 unimodal gaussians that represents the blue curves and the red is the gmm that is a weighted sum of the three gaussians. It's interesting to see that the gmm is almost a perfect merge of

the three gaussians, although in general that doesn't have to be the case. For a 2D case where we merge 3 different multivariate gaussians we have

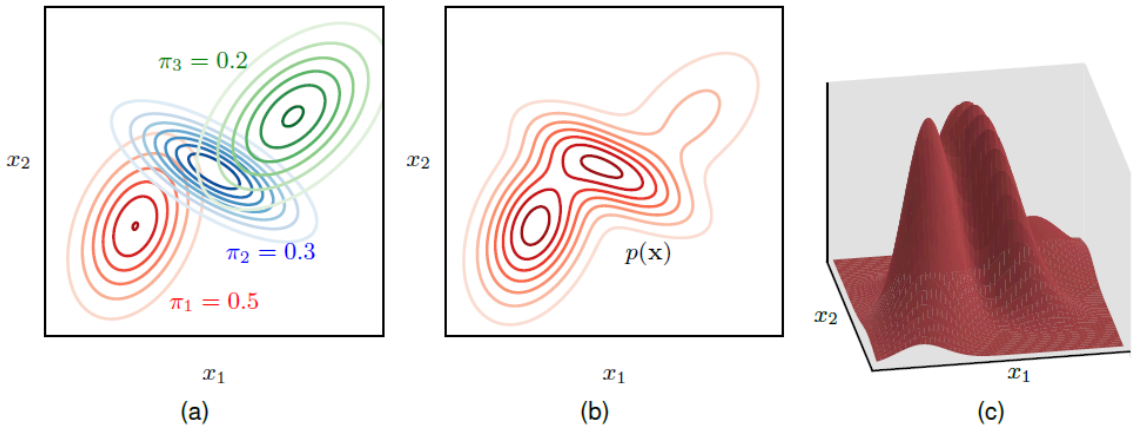

Figure 5: Combining 3 multivariate gaussians.

where a) shows the 3 gaussians, b) the gmm of the three with some weighting and this looks almost exactly like a merged version of the previous one and c) the 3D plot of the gmm.

Moving on to the definition of gmm

$$P(x) = \sum_{k \in K} \pi_k N(x|\mu_k, \Sigma_k)$$

where π_k is the weighting of distribution k, i.e. the higher weighting the more important it is and therefore will have more influence on the appearance of the gmm. Also, π_k has to fulfill $\sum_k \pi_k = 1$. Another constraint is that the covariance matrix has to be positive definite meaning given a column vector x we have the relation $x^T \Sigma_k x > 0$. To set the parameters one could perform MLE on the log likelihood of the data

$$\log P(X|\theta) = \sum_{n=1}^{N} \log \left[\sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k) \right]$$

where $\theta = \{\pi, \mu, \Sigma\}$. Bishop notes that taking the derivative of the term inside log and setting it to zero doesn't have a closed-form, so we can't directly solve the MLE. One way to do remedy this is to use EM short for expectation maximization that is out of scope for this document, so I will skip covering it. Notice that the above equation can be interpreted as a marginalization of the gmms, in which prince provides a nice illustration of. If we assume that we instead introduce hidden variables that we marginalize out to get the marginal of the data then we have the same situation as above, but with the added benefit of being able to relate the terms to the hidden variables h. If we assume the weighting terms are categorically distributed, because that's what they are and the conditional $P(x|h, \theta)$ is multivariate normally distributed

$$P(x|h, \theta) = N[\mu_h, \Sigma_h]$$

$$P(h|\theta) = \text{Cat}_h[\lambda]$$

then we can relate the gmm in the marginal equation above to the following equation

$$
\begin{aligned}
P(x|\theta) &= \sum_{k=1}^{K} P(x, h=k|\theta) \\
&= \sum_{k=1}^{K} P(h=k|\theta) P(x|\ h=k, \theta) \\
&= \sum_{k=1}^{K} \lambda_k N[\mu_k, \Sigma_k]
\end{aligned}
$$

Here we can see that the weighting is represented by the density of λ_k and the gaussian distributions are the density of the data given the hidden variables $P(x|h=k,\theta)$. It has a nice interpretation, in order to sample x from the joint we start by sampling the hidden variable h and then given that hidden variable we sample x from the gaussian distribution. Because sampling x from the gaussian distribution is conditioned on the hidden variable h we can interpret the hidden variable as assigning how much the gaussian distribution is responsible for generating the sampled x. Furthermore, by adding up this contribution of all distributions and thus marginalizing out the hidden variables we get the overall likelihood of sampling that specific datapoint x. Two illustrations by Prince makes this very clear

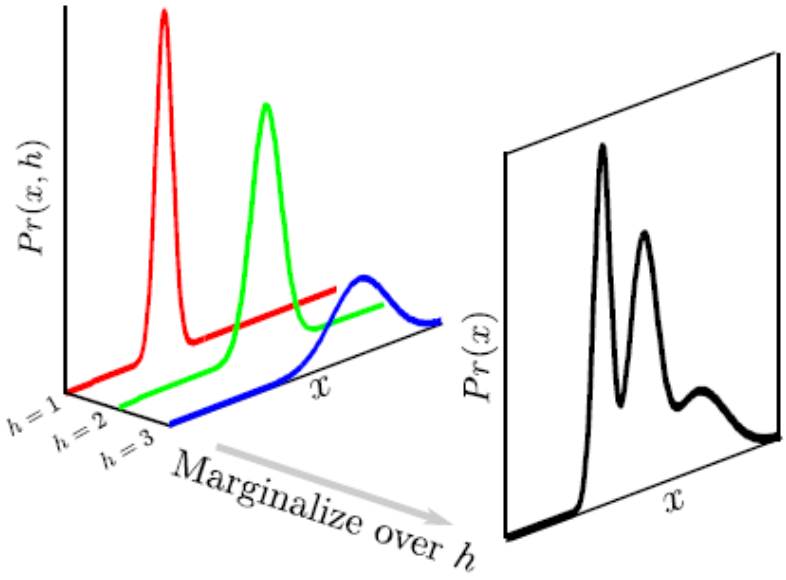

Figure 6: Marginalize gmm over h.

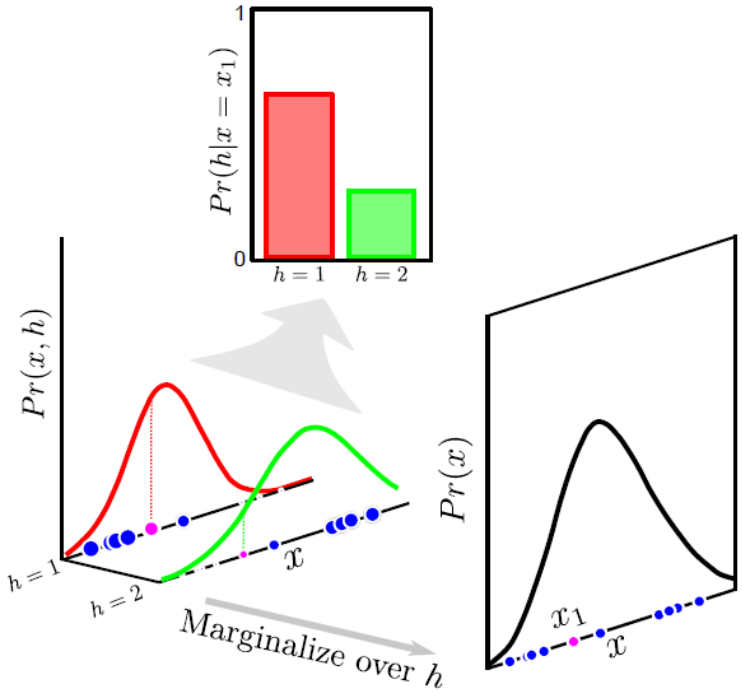

Figure 7: Gmm responsibilities.

First one shows the process of marginalizing out all the hidden variables from the joint $P(x, h)$ to produce the marginal $P(x)$, which is just a weighted sum of all the gaussian distributions where we have taking into account all of their contribution to produce the data x. Second image illustrates the idea of the weighting constituting the responsibility each gaussian distribution has on each datapoint x being sampled from it. In the binary case $P(h|x = x_1)$ we can see in the example that distribution $k = 1$ has a bigger responsibility (weighting) on generating the sampled datapoint (purple) than distribution $k = 2$, as can be also clearly seen in their probability distribution where red has higher probability than green for the purple datapoint. As a result of this, the first distribution will have a bigger influence on the appearance of the marginal $P(x)$ at that specific datapoint.

9. Derivative of relu

Small section to show the derivative of relu. The definition of relu is

$$\text{relu}(x) = \max(0, x) = \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

Taking the derivative we obtain

$$\frac{\mathrm{d}}{\mathrm{d}x} \text{relu}(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

This shows the simplicity of relu as an activation function for learning. We like linear-like functions, because they are easy to compute the gradients of and optimize.

10. Tanh expressed as sigmoid

Tanh is defined as

$$\tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$

There are some non obvious tricks to get it in form of sigmoid, basically factor out $\exp(z)$ and add 1 and subtract by 1, then rewrite the positive 1 as a fraction using the term in the denominator after the cancellation of $\exp(z)$ in tanh, simplify and you will see that you can express in terms of sigmoid function.

$$\begin{aligned}
\tanh(z) &= \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \\
&= \frac{\exp(z)(1 - \exp(-2z))}{\exp(z)(1 + \exp(-2z))} \\
&= \frac{1 - \exp(-2z)}{1 + \exp(-2z)} + 1 - 1 \\
&= \frac{1 - \exp(-2z)}{1 + \exp(-2z)} + \left(\frac{1 + \exp(-2z)}{1 + \exp(-2z)}\right) - 1 \\
&= \frac{1 - \exp(-2z) + 1 + \exp(-2z)}{1 + \exp(-2z)} - 1 \\
&= \left(\frac{2}{1 + \exp(-2z)}\right) - 1 \\
&= 2\left(\frac{1}{1 + \exp(-2z)}\right) - 1 \\
&= 2\sigma(2z) - 1
\end{aligned}$$