Machine Learning Models

37741 Conor Cullen
37748 Mike Talbot

2nd November 2018

Question 1

Part 1:

A Gaussian Likelihood is used as it centres the variance to a normal value, with a controllable variance around that point. i.e any errors made in observations are more likely to end up around the true value than far from the true value.
Having a large number of observations would end up producing such a normal distribution, and therefore it is easier to calculate them as integrals rather than as a summation of each observation.

Part 2:

Having a spherical covariance matrix allows a probability distribution that is circularly symmetrical; changing a diagonal to another value that would change the circular distribution to look more like an ellipse where the long axis is vertical or horizontal. Having a spherical covariance means that we assume that there is no correlation between x and y.

Question 2

If we do not assume that the data points are independent then the likelihood would be in the form:

$$p(\mathbf{Y}|f,\mathbf{X}) = \Pi_{i=1}^{N} p(\mathbf{y_i}|\mathbf{y_{i-1}},...,\mathbf{y_1}|f,\mathbf{x_i}) \tag{1}$$

As y_i is dependant on y_{i-1}, which is dependant on y_{i-2}, etc. This forms a recursive definition of the probability of y_i as $y_i|y_{i-1},...,y_1$.

Question 3

The specific form of the likelihood is:

$$f(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \Pi_{n=1}^{N} \mathcal{N}(Y|W^T\Phi(x_n),\beta^{-1}) \tag{2}$$

Question 4

Calculating a conjugate prior saves us from having to calculate the constant of the formula $p(\theta|x) \propto p(x|\theta)p(\theta)d\theta$ where the constant used is as below:

$$konst = \frac{1}{\int p(x|\theta')p(\theta')d\theta'} \tag{3}$$

instead by using a conjugate prior we can continue evolving the model without ever calculating the constant.

Question 5

With a spherical covariance matrix, this encodes that there is no correlation between the two weights, allowing us to start from a blank slate and not let any prior beliefs sway an understanding in any direction. The quadratic form in the exponent of the covariance matrix encodes the Mahanolobis distance function.

Question 6

By using the conjugate prior we can calculate the posterior over **W**. In this specific case, the prior is in the form of a normal distribution so the posterior is calculated as:
Prior:

$$p(\mathbf{W}) = \mathcal{N}(\mathbf{W}|\mathbf{m_0},\mathbf{S_0}) \tag{4}$$

Likihood:

$$p(\mathbf{Y|W,X}) = \mathcal{N}(\mathbf{Y}|\mathbf{W}^T\phi(X), S) \tag{5}$$

Posterior:

$$p(\mathbf{W|X,Y}) = \mathcal{N}(\mathbf{W}|\mathbf{m_N}, \mathbf{S_N}) \tag{6}$$

$$\mathbf{m_N} = (\mathbf{S_0^{-1}} + \beta\phi(\mathbf{X}))^{-1}(S_0^{-1}\mathbf{m_0} + \beta\phi(\mathbf{X})^T\mathbf{t}) \tag{7}$$

$$\mathbf{S_N} = (\mathbf{S_0^{-1}} + \beta\phi(\mathbf{X}))^{-1}) \tag{8}$$

where $\mathbf{m_N}$ and $\mathbf{S_N}$ are the mean and covariance of the posterior after having seen N data points. β is the precision and ϕ is the basis function. The posterior is Gaussian due to the likelihood and prior both being Gaussian.

Question 7

A non-parametric model does not have the assumption of underlying statistical distributions in the data, unlike a parametric model. This means that the distribution is not defined by a finite set of parameters. The parameters control the complexity. The main aspects of non-parametric models are:

- The number of parameters are dependent on the number of data points. The model represents probabilistic predictive distributions.
- The way they interpret the data is that they don't assume anything and learn from the data.

Question 8

This prior represents a marginal distribution over functions, where f is the instantiation of the function at $\mathbf{X}$. It places structure on the space of functions by defining the Gaussian distribution over a finite set of points.

Question 9

The GP prior encodes only a subset of all possible functions. While it is a random process with an infinite domain, the prior is just a portion of a finite vector with a multivariate Gaussian distribution.

Question 10

The joint distribution of this model is derived from:

$$p(\mathbf{Y}, \mathbf{X}, f, \theta) = p(\mathbf{Y}|f)p(f|\mathbf{X}, \theta)p(\theta) \tag{9}$$

The full graphical model is shown in Fig. 1.

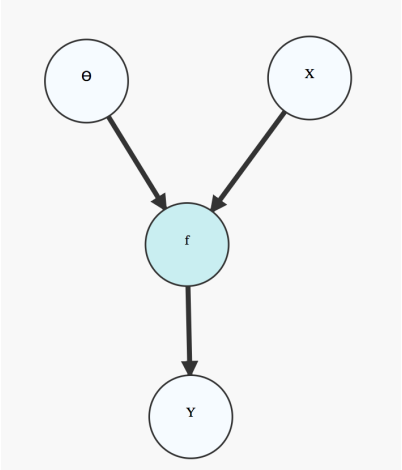


Figure 1: *Graphical Model of the Joint Distribution*

The assumptions that have been made in this model are as follows:

- There is zero mean and spherical covariance
- The data can be represented as a sample from a multivariate Gaussian distribution
- The term $p(f|\mathbf{X}, \theta)$ is Gaussian distributed

Question 11

The integral gives out all of the values of Y over the output f and multiplies it with the values of f given X and θ. As you have calculated f given X and θ, you are then able to replace this f in the Y given f. Uncertainty is filtered through this because the more data that is being observed, the greater our belief is in our function output. θ on the left-hand side implies that this is a variable that we are interested in so has not been marginalised out like f.

Question 12

With no data points, we can only assume a weak covariance around the weighting $\mathbf{W}$. Therefore a covariance matrix with a values of $\Sigma = \begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}$ was used as an arbitrary start point. This is a spherical covariance which will aim to produce little bias in the results as data points are added. We will start our prior at centring around the origin, as we have to assume no knowledge of the weighting; this then means that both weightings can vary with the same probability. The likelihood would be a multivariate normal distribution too as it comes in the

form:

$$\mathbf{y} = \mathbf{w}\mathbf{x} + \epsilon \tag{10}$$

This is where $\epsilon \sim \mathcal{N}(0, 0.3)$ rearranging we get:

$$y_i - \mathbf{w}x_i = \epsilon \sim \mathcal{N}(0, 0.3) \tag{11}$$

This gives us the likelihood function to be

$$p(y|w, x) = \mathcal{N}(y|xw, 0.3) \tag{12}$$

As both the likelihood and the prior are both multivariate normal distributions, we know that the posterior produced would also be a multivariate distribution.

Therefore the initial prior would look like this [Fig. 2 (top middle plot)]. Once getting the likelihood we can simply calculate the new posterior by using:

$$p(w|y, x) = \mathcal{N}(w|\frac{1}{0.3}(\frac{1}{0.3}x^T x + \Sigma^{-1})^{-1}x^T y, (\frac{1}{0.3}x^T x + \Sigma^{-1})^{-1}) \tag{13}$$

With the 0.3 being the variance of the noise and Σ being the covariance matrix of the prior.

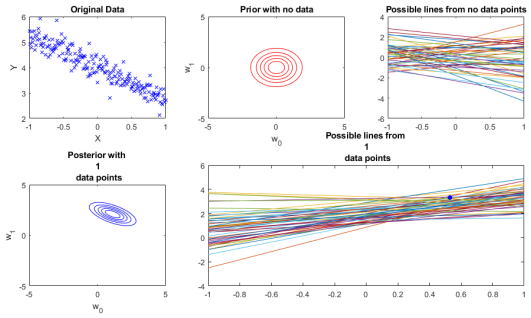


Figure 2: *Graphs of 1 Data Point, including how the prior looked before analysing the data points and how the posterior is changed once the data is shown. (top left: original data points using stated formula (y = wt + ϵ) ; top middle: Original prior, here it shows our mean at (0,0) and a spherical covariance; top right: examples of what 'possible' functions could fit our prior; bottom left: the new posterior after using the likelihood and combining it with the prior; bottom right: the new 'possible' functions after the the new posterior is calculated)*

As shown in the plots [Figs. 2,3,4,5] the prior for the weightings becomes more homed in on [-1.3, 4] and the covariance steadily decreases until it is large enough to only encompass the noise we generated. One thing to take note on is the first data point taken. This generated a w_0 with a positive value so all of the new predicted functions have a positive gradient unlike the actual data.

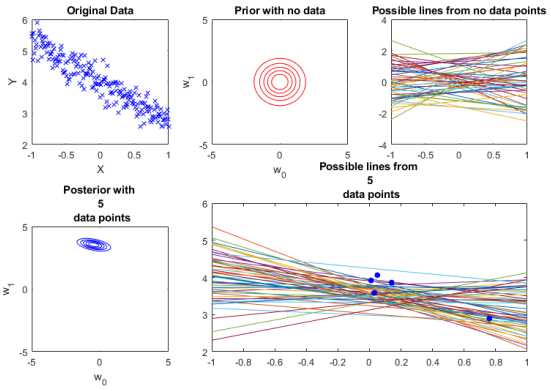


Figure 3: *Graphs of 5 Data Points*

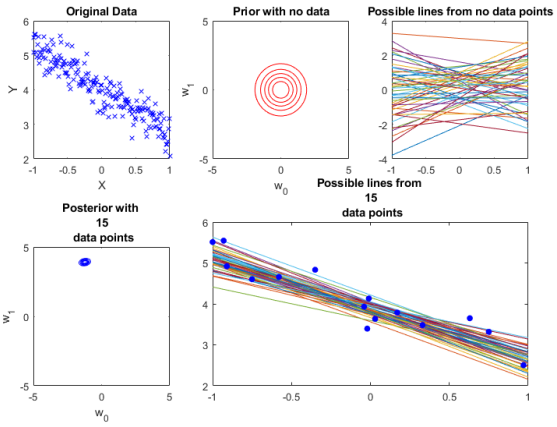


Figure 4: *Graphs of 15 Data Points*

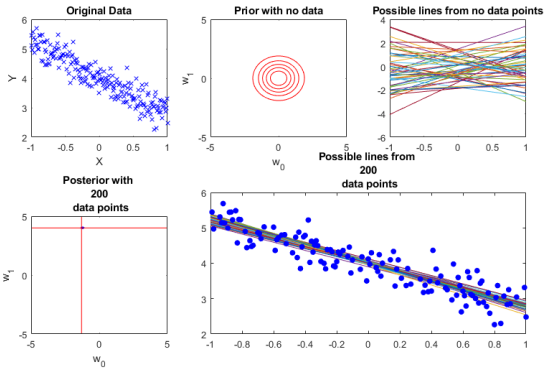


Figure 5: *Graphs of 200 Data Points (in the bottom left figure, the intercept of the red lines indicate the true mean of w_0 and w_1 before noise, showing how close to the true value the posterior has become*

This is why having a substantial amount of data at hand is vital.

Another thing to note is in Fig 3, here the samples (blue points on the bottom right figure) are very bunched together in the centre around x = 0. This ends up causing a spread more varied than the one from just one data point. This is due to the fact that a function will always be far more accurate if you are interpolating between two points than extrapolating off the ends of points. Since all the points are bunched together, there is very little interpolation between the points, therefore many functions can fulfil hitting all the points with little error.

As the number of data points is steadily increased in Fig. 4 the image of the data becomes clearer, here no predicted function have any positive gradients, and the posterior is now in the area of [-1.3, 4] our initial value. By increasing the number of data points sampled now the posteriors means now stay steady and the covaraiance shrinks down until it becomes once again spherical around the mean. This spherical behaviour is expected because w_0 has no direct correlation to w_1, they are both effected just by the noise generated right at the start, so there would be no covariance between them.

Question 13

The Gaussian Process is a prior over functions $p(f_i)$ used for Bayesian regression, where every function f_i is assumed to follow a Gaussian distribution. The prior for a set of inputs $\mathbf{X} = x_1, x_2, ..., x_N$ corresponding to a set of random function variables $\mathbf{f} = f_1, f_2, ..., f_N$ is given by:

$$\begin{bmatrix} \mathbf{f_1} \\ \mathbf{f_2} \\ \vdots \\ \mathbf{f_N} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{x_1}) \\ \mu(\mathbf{x_2}) \\ \vdots \\ \mu(\mathbf{x_N}) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x_1}, \mathbf{x_1}) & \dots & k(\mathbf{x_1}, \mathbf{x_N}) \\ \vdots & \vdots & \vdots \\ k(\mathbf{x_N}, \mathbf{x_1}) & \dots & k(\mathbf{x_N}, \mathbf{x_N}) \end{bmatrix} \right) \tag{14}$$

where there is a squared exponential covariance function of each combination of data points. Using the GP-prior above, we can draw 5 samples from the resulting Gaussian:

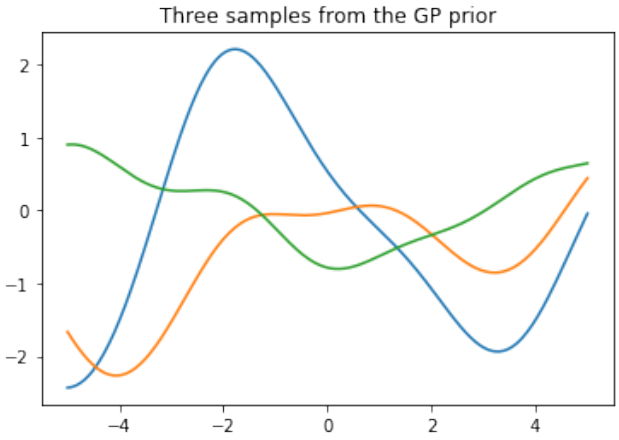
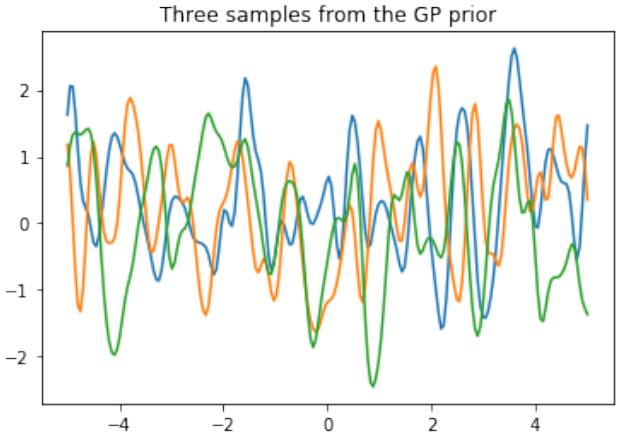
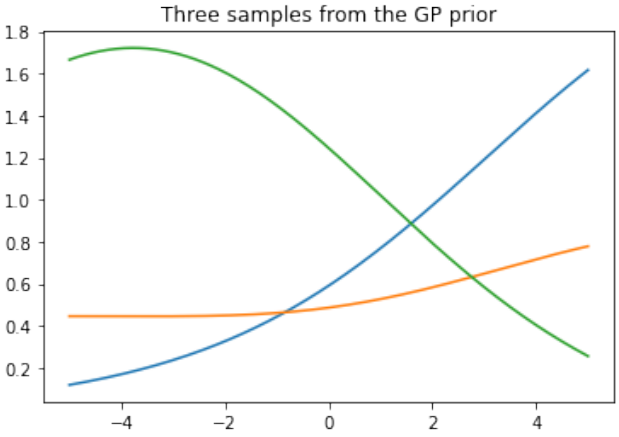


Figure 6: *Gaussian Prior Plots*

The lengthscale specifies the width of the kernel and therefore describes how smooth the function is. A small lengthscale value means that function values can change quickly, whereas large values correspond to values that change slowly:



(a) *Scale length = 0.1*



(b) *Scale Length = 5*

The covariance function encodes all assumptions about the form of the function we are modelling. The length-scale encodes the assumption that the distribution over the function parameters is Gaussian, rather than specifying the function directly.

Question 14

The predictive posterior distribution (also Gaussian) of the model is the joint probability of the outcome values, and is given by:

$$\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) & k(\mathbf{X}, \mathbf{x}_*) \\ k(\mathbf{x}_*, \mathbf{X}) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \tag{15}$$

where $\mathbf{f}$ are functions we have observed and f_* are functions we have not observed. With the inclusion of noise $\epsilon \sim \mathcal{N}(0, 0.5)$ in the model, this posterior becomes:

$$\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & k(\mathbf{X}, \mathbf{x}_*) \\ k(\mathbf{x}_*, \mathbf{X}) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \tag{16}$$

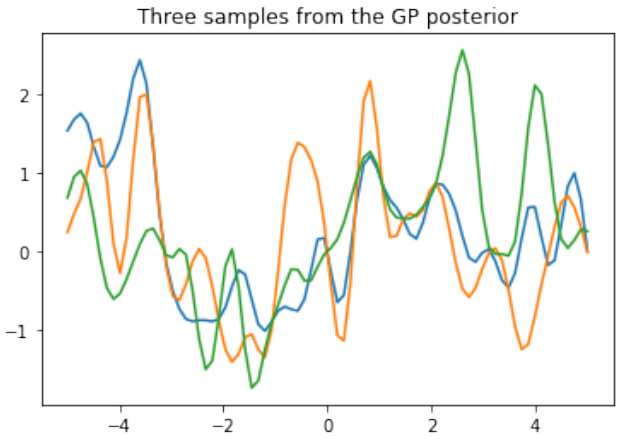


Figure 8: *3 samples taken from the Gaussian Posterior*

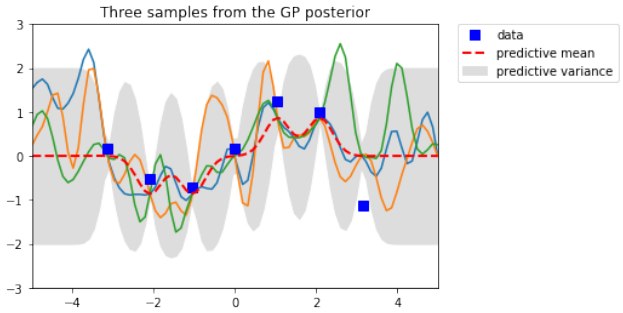


Figure 9: *3 samples from the Gaussian Posterior with noisy data, predictive mean and predictive variance ($\pm$ 2 standard deviations)*

From Fig.9, we see that the samples taken from the posterior all pass exactly through the data points (without noise of course; the figure shows the noisy data). The samples being closer together means there is less uncertainty compared to samples from the prior distribution. We see that the predictive variance starts to increase outside the data points. In order to decrease our uncertainty we need to see data over these areas as well.

This is desirable behaviour as we have combined our Gaussian prior with data, which strengthens our belief in the prior. If you were to add a diagonal co-variance matrix to the squared exponential matrix, the samples will move closer to the data points, meaning more samples see the data. This gives us less uncertainty as the predictive variance of the data is smaller.

Question 15

We need to make assumptions to learn from our data and then formulate our belief over the prior.

- Assumptions: These have a little backing from real world evidence but generally are formed from having a lack of data. When forming an assumption, you must aim to create it with as little bias as possible. For example, in Question 12 a prior had to be created. This prior had no knowledge of the system beforehand so arbitrary values had to be picked for the distribution. Generally, we pick zero for the mean, and ensure that the covariance is spherical so that we do not also add the assumption that the two weights were related in any way.

- Beliefs: These are generated by our assumptions, so in the example before our assumptions were a zero mean and a spherical covariance. These evolved into the multivariate normal. The prior created from this is our belief. The belief can be updated as data is input into the likelihood. So once we update our belief, the posterior becomes our new updated belief. This new belief allows us to be able to take random values that are probable to the 'true value'.

- Preferences: This is how far we are willing to make assumptions until we have uncertainty in our beliefs. Our preference is what we encode to show what assumptions we can make without making the model unrealistic.

Question 16

In the prior $p(\mathbf{x}) = \mathcal{N}(\mathbf{0}, I)$ a spherical covariance with a zero mean is used. This encodes a probability that decreases due to the Euclidean distance from the centre point. This also gives the assumption that the items in $\mathbf{x}$ have no correlation between each other. Normally a covariance matrix from a normal distribution would encode a Mahalanobis distance. However, the spherical covariance matrix forms the special case where the Mahalanobis is equivalent to the Euclidean distance.

Question 17

Marginalising the likelihood of the above prior across all of X. In order to get the marginal of the data, we first need to calculate the likelihood $p(\mathbf{Y}|\mathbf{W}, \mathbf{X})$.

We assume that the observed variable $\mathbf{y}$ is defined by a linear transformation of the latent variable $\mathbf{x}$ plus Gaussian noise ϵ that has zero mean and covariance $\sigma^2\mathbf{I}$:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mu + \epsilon \tag{17}$$

We want to determine the values of the parameters $\mathbf{W}$, μ and σ^2 using maximum likelihood. The marginal distribution of the observed variable is given by:

$$p(\mathbf{Y}|\mathbf{W}, \mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{X})d\mathbf{X} \tag{18}$$

This marginal distribution is also Gaussian, as our prior over the latent variable $\mathbf{x}$ is also Gaussian:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu, \mathbf{C}) \tag{19}$$

where $\mathbf{C}$ is the covariance matrix defined by:

$$\mathbf{C} = \mathbf{W}\mathbf{W}^\mathbf{T} + \sigma^2\mathbf{I} \tag{20}$$

This is derived by evaluating the mean and covariance by exploiting the linear relationship between $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{W}\mathbf{x} + \mu + \epsilon] = \mu \tag{21}$$

$$cov(\mathbf{y}) = \mathbb{E}[(\mathbf{W}\mathbf{x} + \epsilon)(\mathbf{W}\mathbf{x} + \epsilon)^\mathbf{T}] \tag{22}$$

$$= \mathbb{E}[\mathbf{W}\mathbf{x}\mathbf{x}^\mathbf{T}\mathbf{W}^\mathbf{T}] + \mathbb{E}[\epsilon\epsilon^\mathbf{T}] \tag{23}$$

$$= \mathbf{W}\mathbf{W}^\mathbf{T} + \sigma^2\mathbf{I} \tag{24}$$

Question 18

- The difference between ML and MAP is the inclusion of the prior distribution over $\mathbf{W}$, p($\mathbf{W}$), in MAP. The ML estimate of a parameter is the value that maximises the likelihood, whereas the MAP estimate is the value that maximises the posterior distribution. Type-II Maximum Likelihood is where we believe the posterior distribution over θ to be well concentrated over many training examples. This allows us to choose a hyperprior over $\mathbf{W}$ and marginalise out any unknown hyperparameters. This cannot be resolved using maximum likelihood.
- When we observe more data, ML tends to overfit the data, meaning that the parameters we are measuring become sensitive to random variations in the data. However, with MAP our prior beliefs about the parameters assume that they are drawn from some random process. So if the prior beliefs are strong, the observation of more data won't have as much impact on the parameter estimates as with ML.
- Integrating a MAP would give the probability of x given data, therefore if we were to integrate the entire function the value must come to 1 as there is a 100% chance of something happening and you statistically cannot get a probability higher than 1.

Question 19

From Question 17, we know that $\mathbf{X}$ has a multivariate Gaussian distribution with mean μ and covariance $\mathbf{C}$. The objective function for the D-dimensional Y variable for N data points is therefore:

$$\mathcal{L}(\mathbf{W}) = \log p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) \tag{25}$$

$$= \Sigma_{i=1}^N \log p(x_i|\mu, \mathbf{W}, \sigma^2) \tag{26}$$

$$= -\frac{ND}{2}\log 2\pi - \frac{N}{2}|\mathbf{W}\mathbf{W}^\mathbf{T} + \sigma^2\mathbf{I}| - \frac{1}{2}\Sigma_{i=1}^N(x_i - \mu)^T(\mathbf{W}\mathbf{W}^\mathbf{T} + \sigma^2\mathbf{I})^{-1}(x_i - \mu) \tag{27}$$

$$= -\frac{ND}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}(\mathbf{W})| - \frac{1}{2}\Sigma_{i=1}^N y_i^T(\mathbf{C}(\mathbf{W}))^{-1}y_i \tag{28}$$

The gradients of the objective with respect to the parameters are: By Differentiating $\mathcal{L}$ w.r.t. $\mathbf{W}$ on the left hand side we need to derive $\mathbf{C}$ w.r.t. $\mathbf{W}$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}} = -\frac{\partial}{\partial\mathbf{W}}\frac{ND}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}(\mathbf{W})| - \frac{1}{2}\Sigma_{i=1}^N y_i^T(\mathbf{C}(\mathbf{W}))^{-1}y_i \tag{29}$$

Before handling the derivative, the summation needs to be put into a more handleable format. Luckily the sum

is of $y_i^T(\mathbf{C}(\mathbf{W}))^{-1}y_i$, as both y vectors have the index i being summed we can use the 'Trace' function.

$$m = \Sigma_{i=1}^{N}\mathbf{X} = tr\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^T\right) \tag{30}$$

Therefore using this rule we are able to rewrite the objective function as:

$$\mathcal{L}(\mathbf{W}) = -\frac{ND}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}(\mathbf{W})| - \frac{1}{2}tr\left(Y\mathbf{C}(\mathbf{W}))^{-1}Y^T\right) \tag{31}$$

In the derivative the first term will vanish, leaving the last two terms containing **C**. The derivative of **C** w.r.t. $\mathbf{W}_{ij}$ can be rewritten as below and then rearranged using product rule to produce Eq. 32:

$$\frac{\partial\mathbf{C}}{\partial\mathbf{W}_{ij}} = \frac{\partial\mathbf{W}\mathbf{W}^T}{\partial\mathbf{W}_{ij}} = \mathbf{W}\frac{\partial\mathbf{W}^T}{\partial\mathbf{W}_{ij}} + \frac{\partial\mathbf{W}}{\partial\mathbf{W}_{ij}}\mathbf{W}^T = \mathbf{W}\mathbf{J}_{ij} + \mathbf{J}_{ji}\mathbf{W}^T \tag{32}$$

Where $(\mathbf{J}_{ij})_{ij} = 1$ and $(\mathbf{J}_{ij})$ is zero everywhere else.

Now to solve the derivative we take the the two terms separately:
First Term:

$$\frac{\partial}{\partial\mathbf{W}_{ij}}log|\mathbf{C}| = tr\left(\mathbf{C}^{-1}\frac{\partial\mathbf{C}}{\partial\mathbf{W}_{ij}}\right) \tag{33}$$

Second Term:

$$\frac{\partial}{\partial\mathbf{W}_{ij}}\left(tr\left(\mathbf{Y}\mathbf{C}^{-1}\mathbf{Y}^T\right)\right) = tr\left(\frac{\partial}{\partial\mathbf{W}_{ij}}\left(\mathbf{Y}\mathbf{C}^{-1}\mathbf{Y}^T\right)\right) \tag{34}$$

$$= tr\left(\left(\mathbf{Y}\mathbf{Y}^T\right)\frac{\partial\mathbf{C}^{-1}}{\partial\mathbf{W}_{ij}}\right) = tr\left(\mathbf{Y}^T\mathbf{Y}\left(-\mathbf{C}^{-1}\frac{\partial\mathbf{C}}{\partial\mathbf{W}_{ij}}\mathbf{C}^{-1}\right)\right) \tag{35}$$

Question 20

The purpose of marginalising out a variable is to connect that variable with the observed data. The relationship between f and **Y** is stronger than that of **X** and **Y**. We can visualise this using the graphical model of $P(\mathbf{Y}|\mathbf{X},\theta)$(shown in Fig. 10):

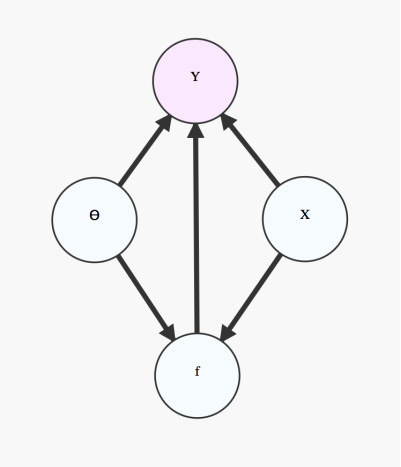


Figure 10: *Graphical Model of the Posterior*

Here we see that the relationship between f and **Y** is a lot easier to connect as opposed to that between **X** and **Y**, as both f and **Y** have observed the same data **X** and θ. This means the best choice is to marginalise over f as it is a simpler relationship to construct. In a general model, you want to marginalise the variable that has observed the most data as this is the variable that we are not interested in. It is variables that haven't seen data that we have an interest in because these tell us the likelihood with the most information.

Question 21

The aim of representation learning in this case is to recover the data set **X**. We can't learn the exact values of **X** as the non-linear function is impossible to recover. However, we can use the objective function and gradients to learn the linear function in terms of **A**. We have that **A** is two basis vectors in a 10D space. Our aim is to learn the values of the basis vectors, $\mathbf{A_0}$ and $\mathbf{A_1}$, and generate the data values of **X** given that we have $\mathbf{Y} = \mathbf{A^T x}$. After this, we repeat the process until an optimal result is achieved, which will give accurate parameters that have generated the function **Y**. The objective function and its gradients are formulated using our results from Question 19. In this case, $f_{lin}(x)$ can be translated to $\mathcal{L}(\mathbf{A})$ and use Eqs 33 and 34 to calculate the gradients.

We look at the projection of the 10D data in the basis space (i.e. $\mathbf{A_0}$ and $\mathbf{A_1}$, our recovered latent representation), optimised using gradient descent, and plotted **X** as a 2D representation from our learned parameters:

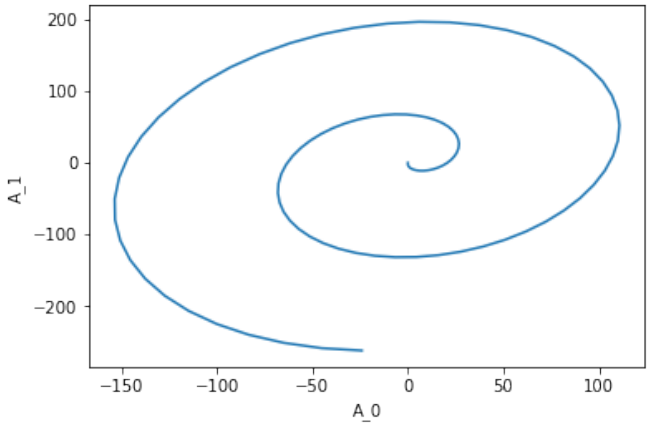


Figure 11: *X plotted before optimisation*

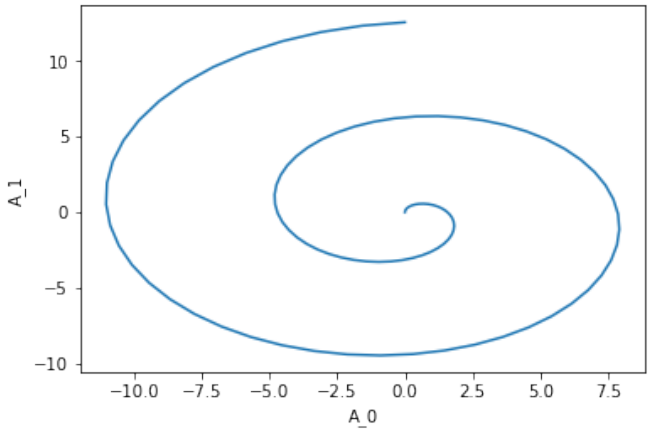


Figure 12: *X plotted after optimisation*

What we have done here is optimised the values of the linear function and substituted this into the non-linear function in order to recover **X**. Plotting the two basis vectors against each other in Fig. 11 gives us an initial non-optimal representation of **X** which is far off from the values of **X** that we wish to recover. Still, there is more learning to be done. The optimised version is plotted after we have optimised the objective function and learned a good representation of **X**, which can be seen in comparison to a plot of **Y** given the actual values of **X**. After each iteration, the parameters being generated become closer to what we are trying to represent here, so the basis space rotates as we are changing the two vectors in our 10D space.

This is the result we expected as from optimising the objective function and its gradients we have generated parameters that accurately recover the dataset **X** given only **A** and **Y**.

Question 22

This is a 10D to 2D mapping which is different to the result we learnt in the previous question as the 2D subspace is random as opposed to the mapping chosen by initialising **A**.

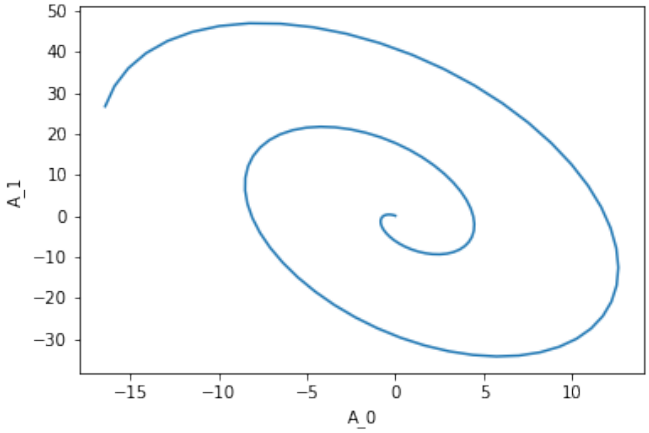


Figure 13: *Mapping from 10D to 2D*

Visually, the plot of the basis vectors becomes more diagonal. Compared to the subspace that we learnt, choosing a random subspace means choosing a new mapping of **A** as an input into f_{lin}, hence we are mapping from a 10D space to a 2D space. This is why the result looks different to that in Fig. 12, as the basis space is not as restricted as opposed to the classic linear representation learning problem in Question 21.

Question 30

Model 1 showed us the importance of making assumptions in our models. For our example we had to assume normal distributions of error. Any assumptions made must not have any bias in them. We did not want to influence the model in a way to produce a different result, for example when generating a prior for Q1-12 it was important that it was spherical so that any covariances between the weights were 'learnt' by the code and not instructed by us. The mapping from linear regression allowed us to make a continuous function of y=f(x) allowing new data points to be made with any value of x. Moving forward from that we used GP-priors where changing the parameters allowed us to change the complexity of the model, whereas in the last model only a y = mx + c would be generated (unless deeper changes in the code were made). Finally we aimed to recover the values of x only from the dataset **Y** which has previously been calculated. This dataset was a 10D problem which had to be boiled down to 2D so that it was conceivable.

This was done by using our objective function and taking its derivative, as calculated in Question 19. By boiling this 10D to 2D we were able to regenerate the vector $\mathbf{x}$ as we know the linear function of $\mathbf{Y} = \mathbf{A^T x}$, as $\mathbf{A}$ is a $2\times$ N sized matrix, where N is the number of data points in $\mathbf{x}$. In general, Machine Learning wasn't exactly what we expected when we first walked in with all of its stats and nonsense, but we can definitely see its use in the world around us. It's still nonsense though...