Bayesian Modelling

Farrel Zulkarnaen[1] and Lim Khai Xi[2]

[1] *fz16336*
[2] *kl16188*

March 22, 2019

Introduction

A model is a simplified framework of the real-world, with layers of abstraction to only account for the relevant; limiting the uncertainty around. Much as how humans learn, machines learn the same way. We make assumptions and build models of how the world works based on our beliefs of how it should be, but as we see new information our beliefs can change; adjusting our perspective - such is the process of learning. From a Bayesian point of view we can interpret probabilities as beliefs in a variable, allowing us to define probabilities over things that have not been observed. This interpretation allows us to put semantics onto the terms in Baye's Rule as *Posterior*, *Likelihood*, *Prior*, and *Evidence*.

1 The Prior

1.1 Question 1

In the real world, there will always be uncertainty in our data. The Gaussian Likelihood entails that such uncertainty (errors) are normally distributed. Another way of thinking about it, by the Central Limit Theorem, no matter what the distribution of the sample is, as you sample more and more from the population, the underlying resulting distribution would tend to be Gaussian. And since the noise is encoded into our Likelihood belief for our model, our Likelihood too, would be Gaussian.

If we choose a spherical covariance matrix, we are assuming that both x and y are not linearly dependent of one another. A change in x does not imply a change in y. This would mean that the covariance matrix, which is one of the parameters for a Gaussian multivariate, would be proportional to an Identity matrix, as the diagonal would represent equal variances in both axes, and zero covariance among each other. This is typically what we would assume first for our Prior belief before seeing any data, an equal spread of uncertainty in both variables. This is also what is known as an isotropic Gaussian distribution.

1.2 Question 2

If we assume that data points are not independent, then the covariance matrix of the contour of their joint probability would no longer be spherical. Instead, it would be elliptical as there would be greater variance in one of the axes. Thus, the principal axes of the elliptical spread would then be determine by the eigenvectors of the covariance matrix. To illustrate say scalar value σi^2 and σj^2 are the variances of x and y, and Σ is the covariance of x and y. And if $\sigma i^2 \neq \sigma j^2$, then we can determine the orientation of the ellipse, and inherently measure the gauge of the correlation between our variables and as well as our uncertainty.

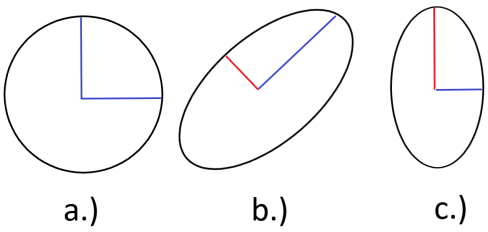


Figure 1: *a.)* $\sigma i^2 = \sigma j^2$ *b.)* $\sigma i^2 > \sigma j^2$, $\Sigma > 0$, *c.)* $\sigma i^2 < \sigma j^2$, $\Sigma = 0$

Hence, the covariances in the covariance matrix represent the orientation of the ellipse, whilst the variances determines the scale. It is also important to note that the contour would only represent the joint distribution of x and y. Therefore the shape of the contour

would also affect the marginal distribution of x and y. For instance, in the case of Fig.1 b.) $\sigma i^2 > \sigma j^2$ implies that there is greater variance in x which also means there is less uncertainty in y since the marginal probability of y would have less spread. And since $\Sigma > 0$ this also means that there is positive linear correlation between x and y. This is also a measure of 'similarity', since a positive covariance implies proportional change in both variables.

1.3 Question 3

A linear model of regression is like the following:

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + ... + w_D x_D \tag{1}$$

Therefore, the likelihood of the function is

$$P(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \mathbf{w}^T \mathbf{x_n}, \sigma I\right) \tag{2}$$

(3)

$$P(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}) = \Sigma_{n=1}^{N} \frac{1}{(2\pi\sigma^2 \boldsymbol{I})^{\frac{1}{2}}} exp(-\frac{1}{2}(y - wx_i)^T (\sigma^2 \boldsymbol{I})^{-1}(y - wx_i)) \tag{4}$$

(5)

1.4 Question 4

Conjugate prior allows us to avoid solving the integral representing the marginal distribution of our evidence. (show formula). As a result it would simplify the problem of solving the posterior significantly. Because the posterior follows the same functional form as the likelihood x prior. This way we can deduce that in order to maintain the proportionality of both sides of the equation, the conjugate prior too must have the same functional form as the likelihood.

Conjugate distributions also help in sequential learning. Let us assume that in the 1st stage of sequential learning, a posterior is computed from the product of its likelihood and prior. In the subsequent stage, the posterior distribution from the previous stage becomes the prior distribution, while a new posterior distribution becomes the output. The property of conjugacy would ensure that for many stages of Sequential Learning, the initial probability distribution holds and does not deviate.

1.5 Question 5

For a Gaussian distribution, the Euclidean distance is the distance-measure function commonly used. Euclidean distance is invariant under translation and rotation; however it is sensitive to scaling.

Consider the geometrical form of the Gaussian distribution. The functional dependence on $\boldsymbol{x}$ can be shown through the quadratic form

$$\Delta^2 = (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \tag{6}$$

whereby the factor Δ^2 is known as the Mahalanobis distance between μ and x. However, it changes to be the Euclidean distance when Σ is Identity matrix.

Therefore, in the case of a spherical Covariance matrix, whereby Σ is the Identity matrix, the distance function it encodes is the Euclidean distance.

1.6 Question 6

As mention previously, the role of using conjugacy is to avoid calculating the evidence of our model. As an update of our belief, the posterior must have the same functional form as the likelihood and the prior i.e left hand side of the equation must equate to the right hand side. The steps necessary to calculate the posterior is outline below.

We know that the formula for the posterior is given by:

$$p(\mathbf{W} \mid \mathbf{X}, \mathbf{Y}) = \frac{1}{Z} p(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}) p(\mathbf{W}) \tag{7}$$

Therefore the process of calculating the posterior involves the convolution of 2 Gaussian distributions. Let us consider the exponential part of the convolution.

$$\begin{aligned} &\frac{1}{\sigma^2}(y - wx_n)^T(y - wx_n) + \frac{1}{\tau^2}(w - w_0)^T(w - w_0) \\ &= \frac{1}{\sigma^2} y_n^T y_n + \frac{1}{\tau^2} w_0^T w_0 + (\frac{1}{\sigma^2} x_n^T x_n + \frac{1}{\tau^2}) w^T w + \\ &(\frac{1}{\sigma^2} x_n^T y_n + \frac{1}{\tau^2} w_0) w^T + (\frac{1}{\sigma^2} y_n^T x_n + \frac{1}{\tau^2} w_0^T) w \end{aligned} \tag{8}$$

We are seeking for the covariance matrix in this case, given by the equation in Question 5. We are looking for a matrix that can be sandwiched between a matrix and its transpose. It is observed that the 3rd term on the LHS from the above equation fits this profile nicely. Hence, we shall now define our covariance, S, as:

$$\boldsymbol{S}^{-1} = (\frac{1}{\sigma^2} x_n^T x_n + \frac{1}{\tau^2}) \tag{9}$$

The final posterior over parameters $\boldsymbol{W}$ is:

$$\begin{aligned} p(\mathbf{W} \mid \mathbf{X}, \mathbf{Y}) = &\prod_{n=1}^{N} \mathcal{N}(w \mid (\frac{1}{\sigma^2} x_n^T x_n + \frac{1}{\tau^2})^{-1} (\frac{1}{\sigma^2} x_n^T y_n + \frac{1}{\tau^2} w_0)), \\ &(\frac{1}{\sigma^2} x_n^T x_n + \frac{1}{\tau^2})^{-1}) \end{aligned} \tag{10}$$

1.7 Question 7

Parametric methods are specific probabilistic functions that are controlled by a number of small adaptive parameters, such as mean and variance. However, parametric models are limited by the fact that a chosen probabilistic function might be an inadequate choice for to model some data, resulting in inadequate results. As an example. if a specific set of data is of a multimodal form, a Gaussian distribution, which is unimodal, will never accurately represent the data.

Nonparametric methods are methods where the form of a distribution depends on the size of the data, where parameters control the complexity of the model instead of the type of function. For example, k-means clustering is a non parametric method, where a dataset is categorised into a number of central points within the data. This number is decided by the number k, which represents the complexity of the model at the same time.

1.8 Question 8

A GP Prior is the generalisation of the normally distributed random variables from the real continuous space to the space of functions. The hyperparameters of a GP Prior are the mean function $\mathbb{M}$, which encodes our belief that the mapping would focus at a given point with respect to time, and the kernel/covariance function $\mathbb{K}$, encodes our belief on the smoothness of the function. Hence, the GP prior determines the structure on the space of functions as it gives higher probability desnity for functions adhering to the parameters of the GP.

1.9 Question 9

Therefore, the GP Prior encodes the space of all possible functions that can fit the mapping of our data. Additionally, every finite subset of the space functions of the GP are multivariate Gaussian distribution, i.e. an instane of a Gaussian process is a Gaussian distribution.

1.10 Question 10

In order to find the joint distribution p($\boldsymbol{Y}$) acting over the range of input functions $(f_1, f_2, ...f_N)^T$. Since we know the linear regression to be

$$y_i = f_i + \epsilon \tag{11}$$

and we know that

$$P(f \mid \mathbf{X}, \theta) = \mathcal{N}(0, k(\boldsymbol{X}, \boldsymbol{X})) \tag{12}$$

thus, the marginal distribution p($\boldsymbol{Y}$) has to be a Gaussian.

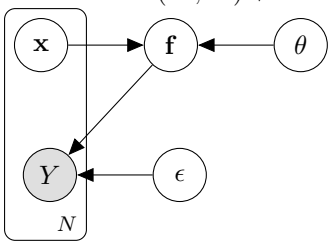
We know that $E(y) = E(f) + E(\epsilon)$, which results in $E(y) = 0$. The variance however, will be calculated as follows:

$$\begin{aligned} E(yy^T) =& E[(f_n + \epsilon)(f_n + \epsilon)^T] \\ =& E(f_n {f_n}^T) + E(\epsilon\epsilon^T) \\ =& k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 I \end{aligned} \tag{13}$$

Therefore we can say that the joint distribution of p($\boldsymbol{Y}$) is can be written as:

$$p(\boldsymbol{Y}) = \mathcal{N}(0, \boldsymbol{C}) \tag{14}$$

where $\boldsymbol{C} = k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 I$.



1.11 Question 11

The marginalisation has marginalised out f. By removing f, the probability function forms a direct connection from X to Y. The fact that θ is left on the LHS after marginalisation means that θ has not been marginalised out.

1.12 Question 12

To generalize our model for linear regressions, say t is a target vector acting as an estimate for some dependent variable y. Given by:

$$\mathbf{Y} = f(\mathbf{X}, \mathbf{W}) + \epsilon \tag{15}$$

$$t = y_i = w_0 x_i + w_1 + \epsilon \tag{16}$$

Where t is a target estimate for the dependent variable y of our i^{th} model, with our input space defined as the set $\mathbf{X} = [x_i, ..., x_n]$ with 2 unknown parameters $\mathbf{W} = [w_0, w_1]$. And of course our model is encoded with some uncertainty which we assume to be normally distributed, i.e. $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$

Now essentially, a linear mapping entails only two parameters, a gradient and a height, which are defined as w's. A Linear Regression is just a model that seeks to explain a linear mapping between two variables. With that being said, those two parameters are what tunes our linear model to best fit our data. Commonly, Linear Regressions are often solved with the method of Ordinary Least Squares (OLS), where you aim to minimise the sum of distances between your estimated best fit line and the observed data points. However, in the context of learning, we shall try to show the importance of parameters and uncertainty when it comes to optimising your model.

To start with, we generated 100 data points by setting the range of our input space to be from -1 to 1 and

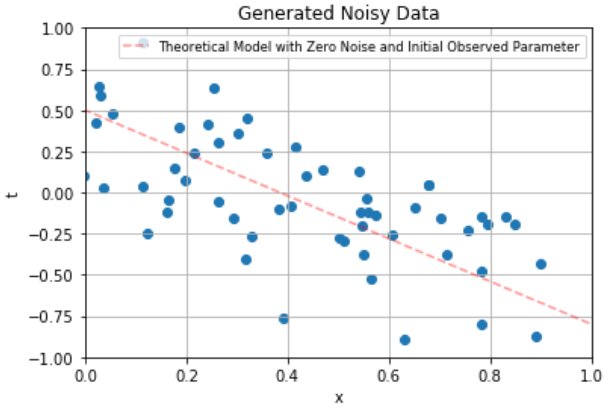


Figure 2: *Plot of 100 generated data points in blue with best fit according to our generating parameters.*

our corresponding y value to be normally distributed with mean zero, and standard deviation 0.3 as seen in Figure 2. We then set our generating parameters to be weights $\mathbf{W} = [-1.3, 0.5]$. Afterwards we 'threw away' our parameters, and then 'recover' them as we update our beliefs of the model, all in an attempt to show how parameters inherently define the tuning of the underlying mapping we are seeking to understand.

Initially, in the absence of any data we belief that there are no correlation between our two parameters. As such, the joint probability distribution of our weights would be spherical, implying equal spread of uncertainty in w_0 and w_1. Another way of thinking about it, is that with zero data, the mean vector of the joint probability of our weights would be (0,0), and if both our parameters are zero that means our model is zero everywhere for all x. This make sense, since if we have no data, we cannot model anything. Refer to Figure 3.

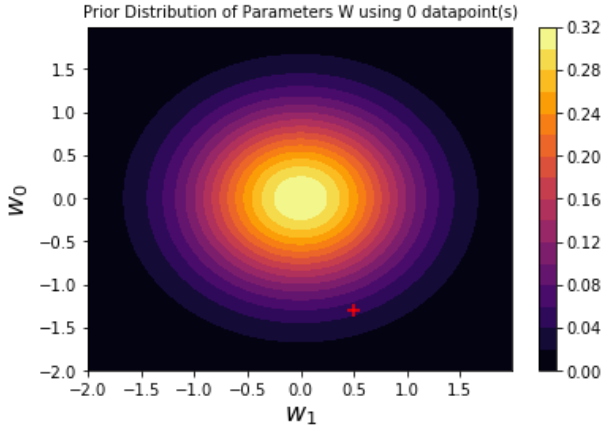


Figure 3: *Prior distribution of weights with no generated data. Initial weights are marked as the red cross.*

Now, we start by taking samples of data from 2. Let the number of samples be 5. The data points will be marked as blue dots, and the model will try to fit a line through the number of data sampled, They are depicted as the lighter blue lines, whereas the red dashed line was our initial model based on the generating parameter. Note that what we mean by 'throwing away' the parameters is that we are essentially ignoring our initial model. It is only still shown in the graph to illustrate how the blue lines will converge to our initial model as we generate more data and update our belief. We start with 2 samples, 10 samples, then all 100.

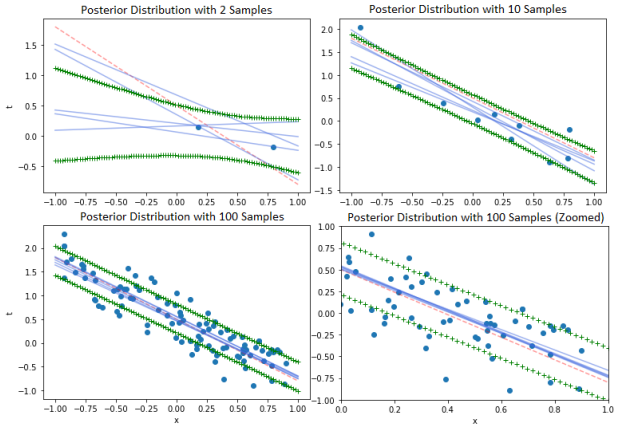


Figure 4: *4 plots depicting how the posterior distribution changes as more data are incorporated*

The generated data are plotted alongside the posterior distribution showing how our beliefs are being updated. The blue lines are converging to the to true value of our initial parameters, and by sample size of 100 the blue lines are accurately models our initial assumption that $\mathbf{W} = [-1.3, 0.5]$. It is also worth noting that we have also plotted confidence intervals marked with a green cross-lines, and they too will converge to around the expected value of our model. This illustrate how the red-line is actually expected value/means of all possible sample models as we generate more data.

$$\mathbb{E}\{t|\mathbf{x}\} = \int tp(t|\mathbf{x})dt \tag{17}$$

The confidence intervals depict the spread of possibility for weights from its average value. This implies that as the number of our data increases the variance of our weights diminishes, meaning we are becoming more certain of its true value being $\mathbf{W} = [-1.3, 0.5]$. As you can see in the bottom right of Figure 5, the joint probability of w have become so small that it has converged to out initial value (the red-cross). If you were to plot the marginal distribution of w_0 and w_1 respectively you will see a low variance of both weights with high probability around the updated mean value of our parameters.

1.13 Question 13

In contrast to the parametric based model above. Gaussian processes (GP) are what we called 'non-parametric.' However, GP is the procedure that generalises regression by theoretically allowing infinitely

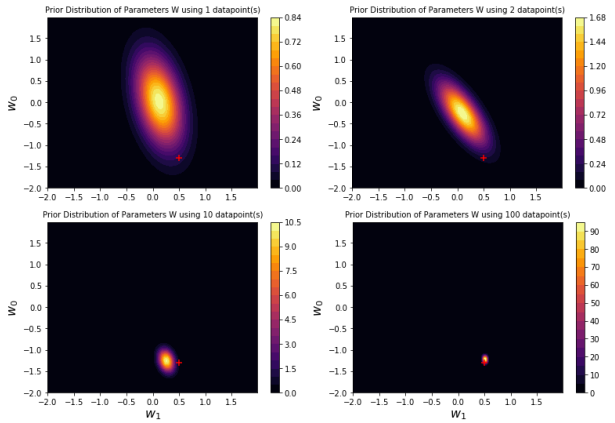


Figure 5: *Prior distributions of W with increasing data points. Initial weight value is marked as a red-cross.*

possible parameters, and that is what 'non-parametric' actually means. Take for instance in the case of a Linear Regression. We have two parameters for our function. But say we observe some data and see that they are correlated in some way, but not linearly. Say if we plot x and y, and they are curved, depicting a quadratic behaviour. In such cases, 2 parameters would not be enough, we need 3 parameters to act as our now polynomial function, and so on. Therefore, this implies that instead of assuming what kind of mapping our model would have, we model the uncertainty over a distribution of all possible function that fits our data. Thus, the advantage of GP is that enables us to tune and model the uncertainty present.

To start with, we need a prior, a belief in the absence of data. But how can we have a belief if we don't even know what is our underlying function. Because we cant model every single possible functions. We limit our view 'locally' by only considering a smaller subspace. Since GP defines a prior belief over a distribution of functions, within finite domain $X = \{x_i, ..., x_n\}$. GP assumes that the joint probability of all $f(x_i)$ is normally distributed with parameters $\mu(X)$ and $\Sigma(X)$ where $\Sigma_{ij} = K(x, x')$. where K is a positive definite function, which forces the Covariance matrix to have positive linear correlation. And this is what acts as the smoothing function of our model. The Kernel function used for GP is the the Squared Exponential function.

$$K(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')}{L^2}) \qquad (18)$$

Where variance σ^2 and length-scale L are the hyper parameters that define our distribution of functions.

As you can see in Figure 6, in the case of a 2D Gaussian (which means 2 parameters), we sample 2 points and connect them through a line. But as the number of dimension increases, and subsequently the number of points, it becomes harder to fit a straight line with more than 2 variables. That is why the samples are 'jagged' with noise. In conclusion, as the number of parameters

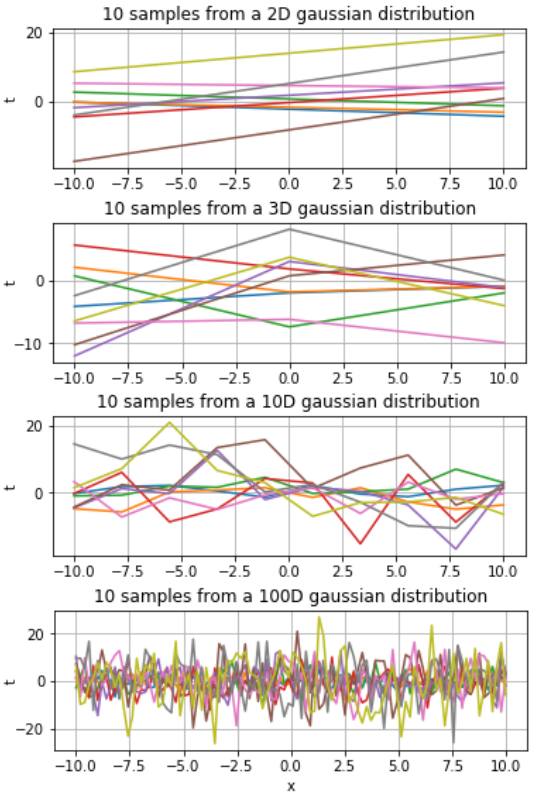


Figure 6: *Prior distribution of our GP-model*

increases the more noise (uncertainty) is being incorporated into our data. So, in order to smooth our belief we use a Kernel Function that measures 'similarity' in the input space.

The theory behind it is that if we take in similar input values (values high Covariance) say x and x' and use that to estimate y', then the output values would also be close to each other. As a result, this would smoothen the graph and effectively reduce our uncertainty.

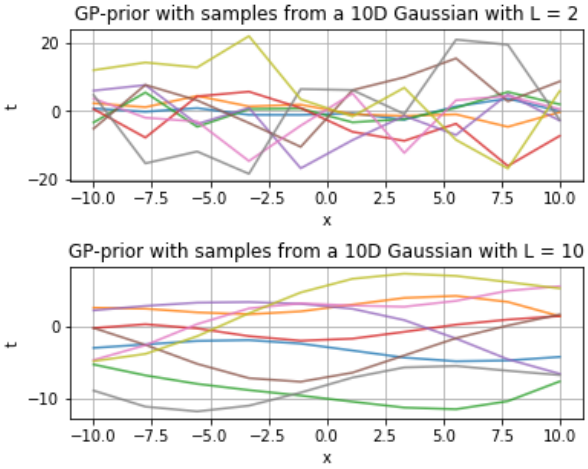


Figure 7: *Smoothen GP-prior withe the squared exponential kernel function*

One interesting thing to note is that as the length-

scale L of the Kernel increases, the 'smoothness' of the functions increases exponentially. However, the 'smoothness' of the Kernel works faster for distributions on higher dimensions, since there are more points to sample from. Comparing Figure 7 and Figure 8, a 10D Gaussian are still noisy with $L = 2$, whereas for a 100D Gaussian even with $L = 2$ the distributions are much smoother already. This implies that our hyperparameter L is what determines the smoothness of our functions. In fact σ^2 defines the vertical variation of our distribution, whereas L modifies the scale. i.e. σ^2 squishes the distribution, whereas L stretches it.

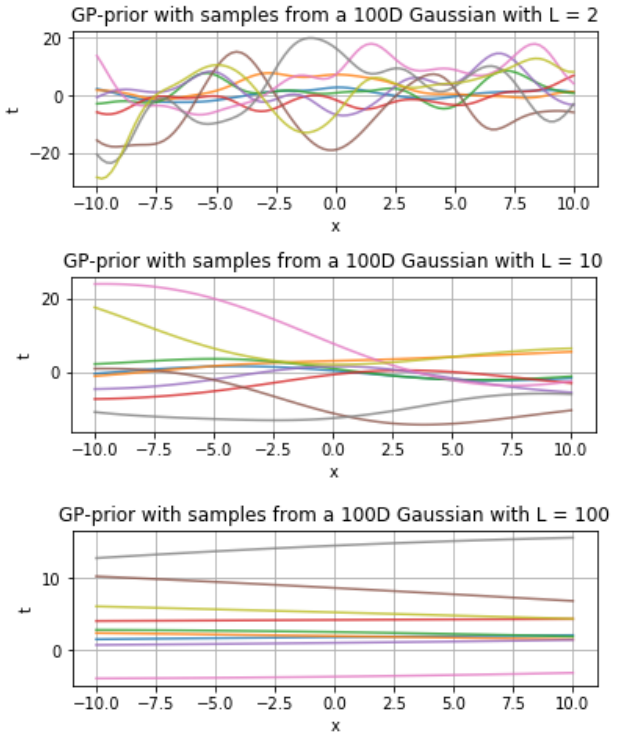


Figure 8: *Smoothen GP-prior withe the squared exponential kernel function*

To elaborate, from Figure 8, as L tends to infinity, the functions would tend to straight line. This is because as L tends to infinity, the exponent of our kernel would tend to zero, which is essentially just one. Making the Covariance matrix to be a diagonal Identity matrix with some scalar constant σ^2, turning the encoded Mahalanobis Distance measure to a Euclidean one.

1.14 Question 14

To further elaborate the Gaussian Process, let us compute a predictive posterior over range $x = (-\pi, \pi)$. GP updates the posterior belief by computing the conditional probability of $y = f(x)$ over all observed values within the specified input domain. Our objective is to then to estimate y_* given observed values of y.

Essentially we want to find $P(f_*|x_*, x, f)$, while assuming that f_* and f are normally distributed. To do so let us defined the Covariance Matrix with elements k,k_*,k_{**}.

$$\mathbf{Y} = \left(\begin{bmatrix} f \\ f_* \end{bmatrix} \right) \sim \mathcal{N} \left(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_{*\mathsf{T}} & K_{**} \end{bmatrix} \right)$$

k is the Kernel function applied to our observed data and is the K_{11} index of the matrix. k_* represents the kernel function applied to an observed value and estimated value, which means the covariance of the matrix and thus the K_{12} and K_{21} respectively. And k_{**} signifies the kernel function applied to our estimated values i.e index K_{22} of the matrix.

Now to simplify the model, just as we can standardise a normal distribution in the single variable case, i.e. $\mathcal{N}(\mu, \sigma^2) \rightarrow \mathcal{N}(0, 1)$. We can do the same for multivariate normals. Or in standard form $x \sim \mu + \sigma\mathcal{N}(0, 1)$.

$$f_* \sim \mu + \mathbf{B}\mathcal{N}(0, 1) \tag{19}$$

Where $\mathbf{B}\mathbf{B}^T = \Sigma_*$ and Σ_* represents the squared root of our Covariance matrix. We use the Cholesky decomposition in our data to solve this problem.

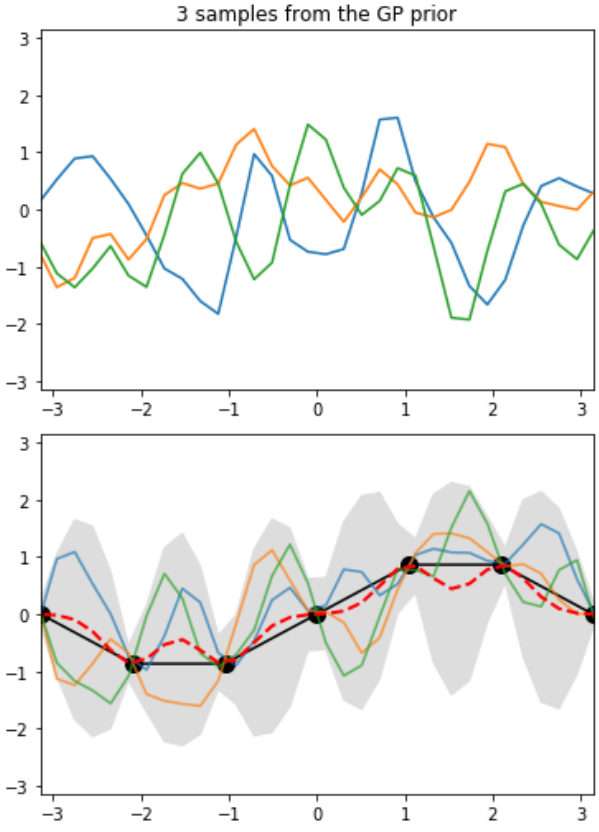


Figure 9: *Predicted GP Posterior*

After sampling seven observed points we plot them and show that how the distribution converges to the mean value marked in red-dashed. The black points are the observed data, whilst the uncertainty interval are essentially silhouette of the uncertainty range of the GP-Prior shown above it.

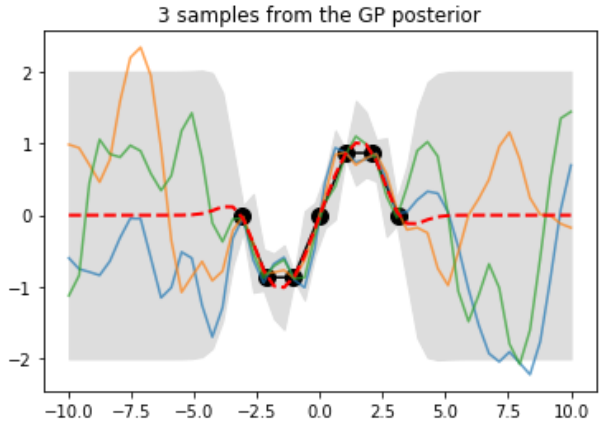


Figure 10: *Zoomed out Posterior*

As you may notice in Figure 10, after sampling 7 points from our observed domain, our uncertainty converges locally as they passes through each of the sampled points. However, outside of the sampled domain our uncertainty is still large and will remain like so until we observe more data there (expand our range). Therefore, to globally-optimise this procedure is the essence of Bayesian Optimisation.

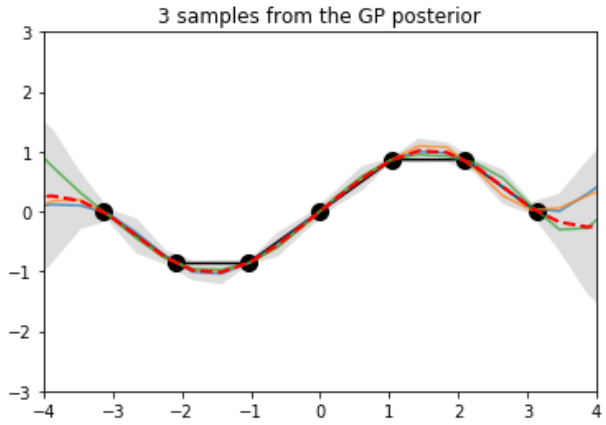


Figure 11: *Posterior with increases length-scale*

Just like before, the length-scale of our Kernel acts as a hyper parameter that increases the smoothness of our distribution. While also at the same time minimizing the uncertainty of the distribution of functions exponentially, see Figure 11

2 Posterior

2.1 Question 15

Assumptions are our underlying foundation for learning. Without assumptions we cannot initialise any model, as we have nothing to based on. If we do not have the slightest idea on whatever we our trying to estimate, and then clearly no matter the outcome, we cannot make any inference from the model as we do not understand anything. Preference, is what we are hoping to achieve, or rather how to archive our belief. It is our own personal motivation on what we ought the model should be like. To elaborate, another way of thinking about it is how far are we willing to generate more assumptions to match our belief. For instance, we would prefer our model to be continuous and linear, and if we can make such assumptions we would. It simply matters and reduce complexity so that our model would be simpler, and easier to compute. Note though, preference in this setting means that we are willing to make new assumptions based on some sensible idea of the parameters, it does not mean that one should make any assumptions just for the sake of computational ease.

2.2 Question 16

The parameters of a Gaussian distribution are the means and Covariance Matrix. And in the case of a spherical Gaussian where the Covariance matrix is and an Identity matrix. The assumption/preference we have made is that with x is non-linearly correlated amongs all other x values.

2.3 Question 17

we know the linear regression to be

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X} + \epsilon \tag{20}$$

and we know that

$$P(x) = \mathcal{N}(0, \boldsymbol{I}) \tag{21}$$

thus, the marginal distribution p($\boldsymbol{Y}$) has to be a Gaussian.

We know that $E(\boldsymbol{Y}) = E(\boldsymbol{X}) + E(\epsilon)$, which results in $EE(\boldsymbol{Y}) = 0$. The variance however, will be calculated as follows:

$$\begin{aligned} E(yy^T) =& E[(\boldsymbol{W}\boldsymbol{X} + \epsilon)(\boldsymbol{W}\boldsymbol{X} + \epsilon)^T] \\ =& E[\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}\boldsymbol{W}]^T + E[\epsilon\epsilon^T] \\ =& \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I \end{aligned} \tag{22}$$

Therefore we can say that the marginal distribution of p($\boldsymbol{Y}$) can be written as:

$$p(\boldsymbol{Y}) = \mathcal{N}(0, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I) \tag{23}$$

2.4 Question 18

Maximum Likelihood (ML) differs from Maximum-a-Posteriori (MAP) only in that MAP is essentially ML times the prior distribution. Therefore they are both equal for the case of a uniform prior distribution. Whereas for a type 2 ML, it maximises the marginal likelihood by summing out one parameter and maximising over the other.

2.5 Question 19

The Objective function with respect to W can be written in the form of a negative log function

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}) &= -log(p(\boldsymbol{Y}|\boldsymbol{W})) \\ &= \frac{ND}{2}ln(2\pi) + \frac{N}{2}ln|\boldsymbol{C}| + \frac{1}{2}\sum_{n=1}^{N} y_n {y_n}^T \boldsymbol{C}^{-1} \\ &= \frac{ND}{2}ln(2\pi) + \frac{N}{2}ln|\boldsymbol{C}| + tr(\boldsymbol{Y}\boldsymbol{C}^{-1}\boldsymbol{Y})\end{aligned} \tag{24}$$

We would want to optimise this objective function, whereby we would need to first write out the gradients of the $\mathcal{L}(\boldsymbol{W})$ function with respect to W. We know that the 1st term on the RHS, which is a constant, reduces to 0 under derivation, hence all we need is the derivation of the 2nd and 3rd terms.

First we would need to find the derivative of $\boldsymbol{C}$ with respect to $\boldsymbol{W}$. Keep in mind that $\boldsymbol{C}$ is a 10 $\times$10 matrix, whereas $\boldsymbol{W}$ is a 10 $\times$2 matrix. The derivation is shown below:

$$\begin{aligned}\frac{\delta \boldsymbol{C}}{\delta \boldsymbol{W}_{ij}} &= \frac{\delta \boldsymbol{W}\boldsymbol{W}^{\boldsymbol{T}}}{\delta \boldsymbol{W}_{ij}} \\ &= \boldsymbol{W}\boldsymbol{J}_{ij} + \boldsymbol{J}_{ji}\boldsymbol{W}\end{aligned} \tag{25}$$

where $\boldsymbol{J}$ is a matrix where all of its entries are 0 and $(\boldsymbol{J_{ij}})_{ij} = 1$

For the 2nd term of $\mathcal{L}(\boldsymbol{W})$, we know that:

$$\begin{aligned}\frac{\delta ln|\boldsymbol{C}|}{\delta \boldsymbol{W}_{ij}} &= tr[\boldsymbol{C}^{-1}\frac{\delta \boldsymbol{C}}{\delta \boldsymbol{W}_{ij}}] \\ &= tr[\boldsymbol{C}^{-1}(\boldsymbol{W}\boldsymbol{J}_{ij} + \boldsymbol{J}_{ji}\boldsymbol{W})]\end{aligned} \tag{26}$$

whereas for the 3rd term of $\mathcal{L}(\boldsymbol{W})$, we know that:

$$\begin{aligned}\delta\frac{tr(\boldsymbol{Y}\boldsymbol{C}^{-1}\boldsymbol{Y})}{\boldsymbol{W}_{ij}} &= tr[\boldsymbol{Y}\boldsymbol{Y}^{\boldsymbol{T}}(-\boldsymbol{C}^{-\boldsymbol{1}}\frac{\delta \boldsymbol{C}}{\delta \boldsymbol{W}_{ij}}\boldsymbol{C})] \\ &= tr[\boldsymbol{Y}\boldsymbol{Y}^{\boldsymbol{T}}[-\boldsymbol{C}^{-1}(\boldsymbol{W}\boldsymbol{J}_{ij} + \boldsymbol{J}_{ji}\boldsymbol{W})\boldsymbol{C}]]\end{aligned} \tag{27}$$

Adding the 2 terms above yields the gradients of the objective function.

2.6 Question 20

When we want to marginalize a variable, what we are effectively doing is "connecting" the marginalized variable with our output. In this case, were comparing between "connecting" $\boldsymbol{X}$ with output $\boldsymbol{Y}$ and "connecting" f with output $\boldsymbol{Y}$. As our graphical model in Question 10 shows, there is a direct connection between f and $\boldsymbol{Y}$, in the form of $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})$, however there isn't a direct connection between $\boldsymbol{X}$ with output $\boldsymbol{Y}$. Therefore, in general, it is much easier to marginalize variables that have a direct connection.

2.7 Question 21

This practical began with the generation of data. We had to generate a data set, $\boldsymbol{Y}$, which has a 100$\times$10 dimension, from a 1 dimensional generating parameter $\boldsymbol{x}$, with a 1$\times$N dimension. The aim of this practical is that given $\boldsymbol{Y}$, we need to recover $\boldsymbol{x}$.

Firstly, we defined $\boldsymbol{x}$ as an array from 0 to 4π, with 100 points in between. $\boldsymbol{Y}$ will be generated via 2 functions, a non linear function and a linear function. The 1st function to be parsed is a non linear function, whereby:

$$f_{non-lin}(x_i) = [x_i sin(x_i), x_i cos(x_i)] \tag{28}$$

Coding this with $\boldsymbol{x}$ returns a 2 $\times$100 matrix. Let us call this matrix $\boldsymbol{x}$'. The subsequent function is a linear function that will take in $\boldsymbol{x}$' and return our $\boldsymbol{Y}$ dataset.

$$f_{lin}(x') = \boldsymbol{A}\boldsymbol{x'} \tag{29}$$

where $\boldsymbol{A}$ is a 10$\times$2 matrix and given by the normal distribution $A_{ij} \sim \mathcal{N}(0, 1)$. As a result of our matrix multiplcation, we obtain the 10$\times$100 $\boldsymbol{Y}$ dataset. If we now refer to the objective function in Question 19, which is essentially the probabilistic distribution of $\boldsymbol{Y}$. We use a logarithmic function in this case as it is monotonically increasing function and will not alter the location of the extremes of the function. Our aim is therefore to optimize the function using the gradients of the functions with respect to $\boldsymbol{W}$, which is essentially $\boldsymbol{A}$. We would do this by using a scipy optimiser function to minimize a function using a nonlinear conjugate gradient algorithm. The optimised vector (which would effectively act as the Inverse of vector $\boldsymbol{W}$) would then be converted to the 10$\times$2 matrix, and multiplied with the initial $\boldsymbol{Y}$ dataset to obtain a 2$\times$100 matrix for the recovered dataset $\boldsymbol{x}$. The representation would then be obtained by plotting 1 dimension of $\boldsymbol{x}$ against the other.

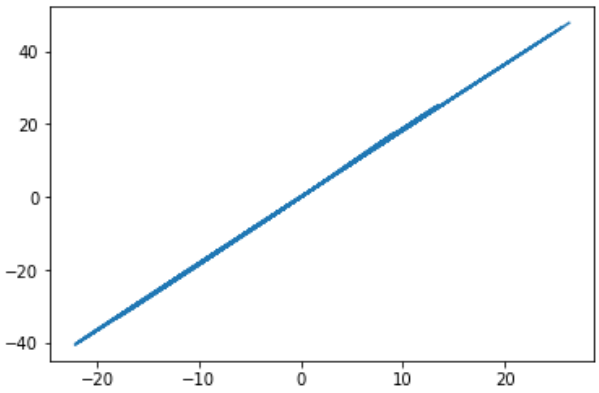


Figure 12: *Linear representation of recovered* $\boldsymbol{x}$

Instinctively, we might assume that when we recover $\boldsymbol{x}$', we will obtain a straight line, as $_{lin}(x')$ indicates.

However, we have to remember the fact that $\boldsymbol{x}$' is actually a 2 $\times$100 matrix of sines and cosines, which represents polar functions. Hence, our recovery of the plot of $\boldsymbol{x}$' should be in the form of a spiral.

However, our results plotted a straight line recovery. It was only after a lot of trials that we obtained some form of resemblance to a helix shaped representation, which could be chalked up to a random error as after all, we are dealing with probabilistic functions. The error could have been due to the optimiser functions finding the wrong local minimas throughout the dataset $\boldsymbol{Y}$. Nevertheless, we cannot avoid the fact that the polar nature of the $\boldsymbol{x}$' matrix was not recovered, and all we recovered was essentially a linear regression.

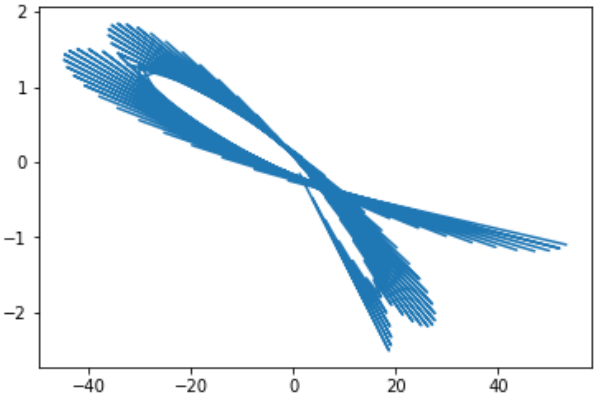


Figure 13: *Helix representation of recovered* $\boldsymbol{x}$

2.8 Question 22

This part of the section asked for a random subspace, which basically meant that we needed to find the product of a random inverse $\boldsymbol{W}$ and the dataset $\boldsymbol{Y}$. Now, if our representation worked the 1st time, and a spiral representation was obtained, we would assume that the representation would return a spiral shape as well. However, we shall try to spot whether there are any differences when it comes to a linearly recovered graph.

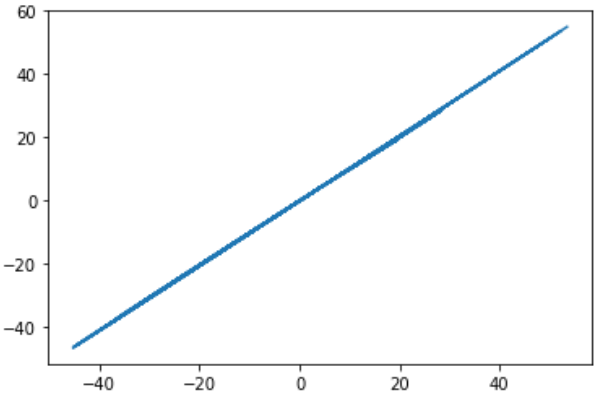


Figure 14: *Linear representation of recovered* $\boldsymbol{x}$ *from random subspace* $\boldsymbol{W}$

As the results show, there wasn;t much of a difference when we utilised a random $\boldsymbol{W}$. This is because we didnt really change the linearity of x, thus the core representation could still be recovered, as all we changed was the parameters $\boldsymbol{W}$.

3 Evidence

3.1 Question 23

We have the model evidence in the form of

$$p(\mathcal{D}|M_0, \boldsymbol{\theta_0}) = \frac{1}{512} \tag{30}$$

This assumption firstly, implies that the probability of generating the dataset D from this model is constant for the entire dataset D. It is the simplest possible model as it takes in no parameters, hence its value is the same for the entire dataset.

On the other hand, one can argue that this is the most complex model as its distribution covers the entire dataset D. If we follow the logic that a simpler model would cover a narrow range of D, and more complex models cover wider ranges of D, then we can argue that M_0, which covers the entire dataset D, is the most complex.

3.2 Question 24

The model M_0 returns an equal probability for all datasets. Model M_1 is a logistic regression without the bias weight θ_3 and the 2nd dimension of $\boldsymbol{x}$. Model M_2 is Model M_1 plus the 2nd dimension of $\boldsymbol{x}$. Model M_3 is standard logistic regression, adding the bias weight to Model M_2. Model M_1 is more flexible from Model M_0 in the sense that Model M_1 can take in different parameters to return a distribution of data, whereas Model M_0 returns constant data. However, it can also be argued that Model M_1 is less flexible as its range of data it acts upon is limited to the center of the dataset, whereas Model M_0 acts on all data. Model M_1 is a simple model that focuses on the data region which is near to the center of the plot of evidence vs dataset D.

The choices of the different models above simply allow us to choose the best model that acts upon a particular range of data. Imagine that models are probabilistic functions with different variance. For example, if we choose a dataset that is close to the mean, Model M_1 will be the best choice as it has the highest probability of evidence with a smaller variance. However, Model M_3 will be the best choice for data that is very extreme. Model M_1 will also have a narrower distribution of the data, whereas Model M_3 will have a wider coverage of the dataset. When we choose a model, want to choose a more complex model as it can describe more of the data. However, there is also a penalising factor, which penalises higher complexity

models when they produce the same level of evidence with lower complexity models.

To visualise this, let us have a 3 $\times$3 dataset consisting of only 1 or -1. If all of the entries of the matrix is 1, then Model M_3 is the best model as it has a bias term to account for this. If one column of the matrix is 1, the Model M_1 which is a simple model that captures decision boundaries would have the highest probability, thereby being the best model. However, if the matrix is transposed, (i.e. one row of matrix is 1), then Model M_2 as Model M_1 can only capture data from 1 dimension as per its formula. Model M_0 comes in when there are no sharp linear boundaries for the matrix (i.e. 2 corners of the matrix are 1). As no other model can cover this, thus the model with the uniform distribution is chosen. (**murray2005note**)

As we can see, higher complexity models (with the exception of Model M_0) are more flexible as they allow a wider range of the dataset to be sampled. However, in a sense, one can argue that higher complexity models actually restrict us in terms of the level of evidence as it is usually quite low compared to lower complexity models. This means that often, with higher complexity models, we will be considering data that have a very low probability of occurrence and should not be taken into account.

One can also say that in terms of uncertainty, if we choose a model with lower complexity, we are rather certain about our data, whereas if we choose a model with high complexity, we are very uncertain about our data.

3.3 Question 25

The choice of this prior means the probability of the parameters given the model follows a normal distribution. The choice of a very large variance ensures that the we are measuring the distribution of the θ with a high complexity model. In essence, this model implies that we are very unsure about what the parameter's distribution is, and we would not want to lose data, despite it being potentially irrelevant.

Conclusion

There is a trade off between accuracy and precision. As such this is what we mean by the 'No-free Lunch Theorem.' There is always an opportunity cost when generating a model, and that is the key take-way in model selection, to establish a reasonable prior. In the words of George E.P. Box, "*All models are wrong, but some are useful.*"

References

[1] Murray, Iain; Ghahramani, Zoubin. *A note on the evidence and Bayesian Occam's razor* 2005

[2] Osborne, Michael; *A note on Cholesky Decomposition: http://gpss.cc/lfm13/talks/Sheffield-Workshop2013-Osborne.pdf*