Section 4: Gaussian processes

4.1 Non-linear functions

4.1.1 Regression view

So far, we've assumed our latent function is a linear function of our data – which is obviously limiting. One way of circumventing this is to project our inputs into some high-dimensional space using a set of basis functions $\phi : \mathbb{R}^d \to \mathbb{R}^N$, and then performing linear regression in that space, so that

$$y_i = \phi(x)^T \beta + \epsilon_i$$

For example, we could project x into the space of powers of x, i.e. $\phi(x) = (1, x, x^2, x^3 \dots)$ to obtain polynomial regression.

Exercise 4.1 *Let* $\mathbf{y}$ *and* $\mathbf{X}$ *be set of observations and corresponding covariates, and* y_* *be the unknown value we wish to predict at covariate* $\mathbf{x}_*$*. Assume that*

$$\begin{aligned} \beta \sim & N(0, \Sigma) \\ \begin{bmatrix} f_* \\ \mathbf{f} \end{bmatrix} = & \begin{bmatrix} \boldsymbol{\phi}_*^T \\ \boldsymbol{\Phi}^T \end{bmatrix} \beta \\ \begin{bmatrix} y_* \\ \mathbf{y} \end{bmatrix} \sim & N\left(\begin{bmatrix} f_* \\ \mathbf{f} \end{bmatrix}, \sigma^2 \mathbf{I} \right) \end{aligned}$$

where $\boldsymbol{\phi} := \phi(\mathbf{x})$ *and* $\boldsymbol{\Phi} := \phi(\mathbf{X})$.

What is the predictive distribution $p(f_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X})$*? Note: this is very similar to questions in Section 1.*

Solution: The posterior is obtained by completing the square of β

$$\begin{aligned} p(\beta|\mathbf{X}, \mathbf{y}) &\propto p(\mathbf{y}|\mathbf{X}, \beta) p(\beta) \\ &\propto \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y_i - \phi^T\beta)^2}{2\sigma^2} \right\} \cdot \exp\left(-\frac{1}{2} \beta^T \Sigma^{-1} \beta \right) \\ &\propto (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \Phi^T\beta)^T (\mathbf{y} - \Phi^T\beta) \right) \cdot \exp\left(-\frac{1}{2} \beta^T \Sigma^{-1} \beta \right) \\ &\propto \exp\left(-\frac{1}{2} (\beta - \bar{\beta})^T (\sigma^{-2} \Phi\Phi^T + \Sigma^{-1}) (\beta - \bar{\beta}) \right) \end{aligned}$$

where $\bar{\beta} = \sigma^{-2}(\sigma^{-2}\Phi\Phi^T + \Sigma^{-1})^{-1}\Phi\mathbf{y}$. Let $A = \sigma^{-2}\Phi\Phi^T + \Sigma^{-1}$, then the form of the posterior is Gaussian distribution with mean $\bar{\beta}$ and covariance matrix A^{-1}

$$p(\beta|\Phi, \mathbf{y}) \sim N(\bar{\beta} = \sigma^{-2} A^{-1} \Phi\mathbf{y}, A^{-1})$$

The predictive distribution for $f_* \triangleq f(\mathbf{x}_*)$ at $\mathbf{x}_*$ is given by averaging the output of all possible non-linear models w.r.t. the Gaussian posterior

$$\begin{aligned} p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \beta) p(\beta|\mathbf{X}, \mathbf{y}) d\beta \\ &\propto \int \exp\left(-\frac{1}{2\sigma^2}(y_* - \phi(\mathbf{x}_*)^T\beta)^2\right) \cdot \exp\left(-\frac{1}{2}(\beta - \bar{\beta})^T A^{-1}(\beta - \bar{\beta})\right) d\beta \end{aligned}$$

But instead of calculating the above integral, we will use the fact that the linear combination of a multivariate Gaussian distribution is a multivariate Gaussian distribution. Since $f_* = \phi(\mathbf{x}_*)^T\beta$ and x_* is independent of β, we have

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = p(\phi(\mathbf{x}_*)^T\beta|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \phi(\mathbf{x}_*)^T p(\beta|\mathbf{X}, \mathbf{y})$$

And also,

$$E(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = E(\phi(\mathbf{x}_*)^T\beta) = \phi(\mathbf{x}_*)^T E(\beta) = \frac{1}{\sigma^2}\phi(\mathbf{x}_*)^T A^{-1}\Phi\mathbf{y}$$

$$Cov(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = Cov(\phi(\mathbf{x}_*)^T\beta) = \phi(\mathbf{x}_*)^T Cov(\beta)\phi(\mathbf{x}_*) = \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)$$

Therefore, the predictive distribution becomes

$$f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim N(\frac{1}{\sigma^2}\phi(\mathbf{x}_*)^T A^{-1}\Phi\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*))$$

with $\Phi = \Phi(\mathbf{X})$ and $A = \sigma^{-2}\Phi\Phi^T + \Sigma^{-1}$. Let $\phi_*(\mathbf{x}) = \phi_*$ and $K = \Phi^T\Sigma\Phi$, we can rewrite the above formula in the following way

$$f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim N(\phi_*^T\Sigma\Phi(K + \sigma^2 I)^{-1}\mathbf{y}, \phi_*^T\Sigma\phi_* - \phi_*^T\Sigma\Phi(K + \sigma^2 I)^{-1}\Phi^T\Sigma\phi_*)$$

Note that, in the solution to Exercise 1, we only ever see ϕ or Φ in a form such as $\Phi^T\Sigma\Phi$. We will define $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\Sigma\phi(\mathbf{x}')$, where x and x' are in either the training or the test sets. Since Σ is positive definite, we can write:

$$k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^T\psi(\mathbf{x})$$

where $\psi(\mathbf{x}) = \phi(\mathbf{x})\Sigma^{1/2}$.

If (as here) we only ever access ψ via this inner product, we can choose to work instead with $k(\cdot, \cdot)$. This may be very convenient if the dimensionality of $\psi(x)$ is very high (or even infinite, see later). $k(\cdot, \cdot)$ is often referred to as the kernel, and this replacement is referred to as the kernel trick.

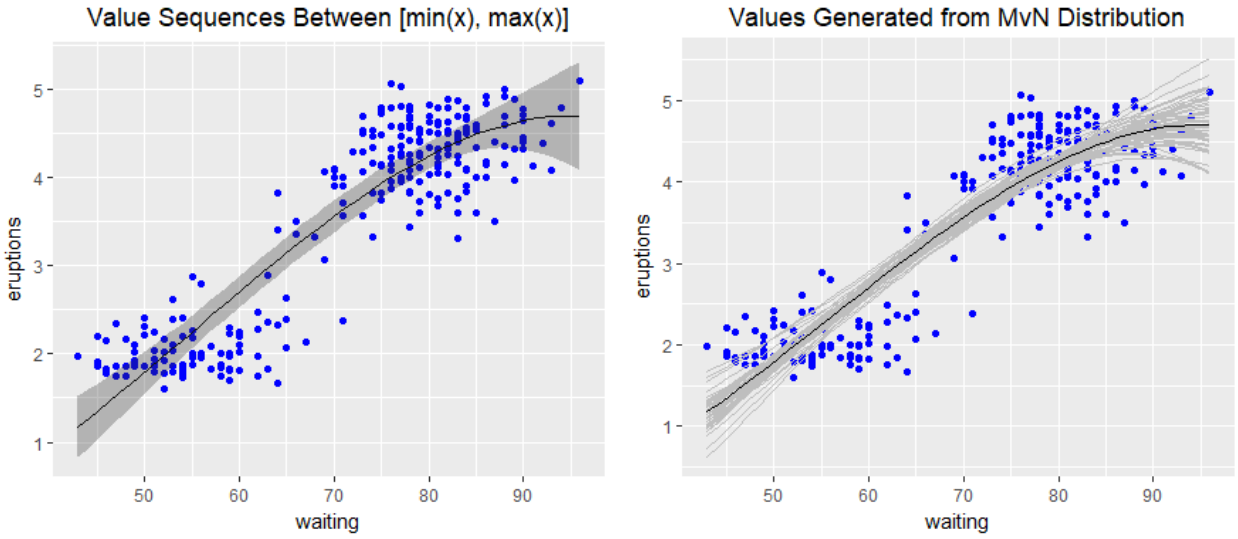
Exercise 4.2 *Let's look at a concrete example, using the old faithful dataset on R*

- `data("faithful",package="datasets")` *in R*
- *or available as* `faithful.csv` *on github if you're not using R.*

Let $\phi(x) = (1, x, x^2, x^3)$. Using appropriate priors on β, obtain a posterior distribution over $f := \phi(x)^T\beta$. Plot the function (with a 95% credible interval) by evaluating this on a grid of values.

Solution: The plots of the function with a 95% credible interval are shown below. The 50 values of the first panel are sequences between $[min(x), max(x)]$, while the 50 values in the second panel are generated from

multivariate normal distribution whose parameters are estimated in exercise 4.1 (the mean and covariance of posterior distribution of β)



4.1.2 Function space view

Look back at the plot from Exercise 2. We specified a prior distribution over regression parameters, which we can use to obtain a posterior distribution over those regression parameters. But, what we calculated (and plotted) was a posterior distribution over *functions*. Similarly, we can think of our prior on β as specifying a prior distribution on the space of cubic functions. Evaluated at a finite number of input locations – as you did in Exercise 2 – this posterior distribution is multivariate Gaussian. This is in fact the definition of a Gaussian process: A distribution over functions, such that the marginal distribution evaluated at any finite set of points is multivariate Gaussian.

A priori, the covariance of f is given by

$$\text{cov}(x, x') = E[(f(x) - m(x))(f(x^T) - m(x^T))] = k(x, x')$$

For this reason, our kernel k is often referred to as the covariance function (note, it is a function since we can evaluate it for any pairs x, x'). In the above example, where β had zero mean, the mean of f is zero; more generally, we will assume some mean function $m(x)$. The Gaussian Process is written as

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

Rather than putting a prior distribution over β, we can specify a covariance function – remember that our covariance function can be written in terms of the prior covariance of β. For example, we might let

$$k(x, x') = \alpha^2 \exp\left\{-\frac{1}{2\ell^2}|x - x'|^2\right\}$$

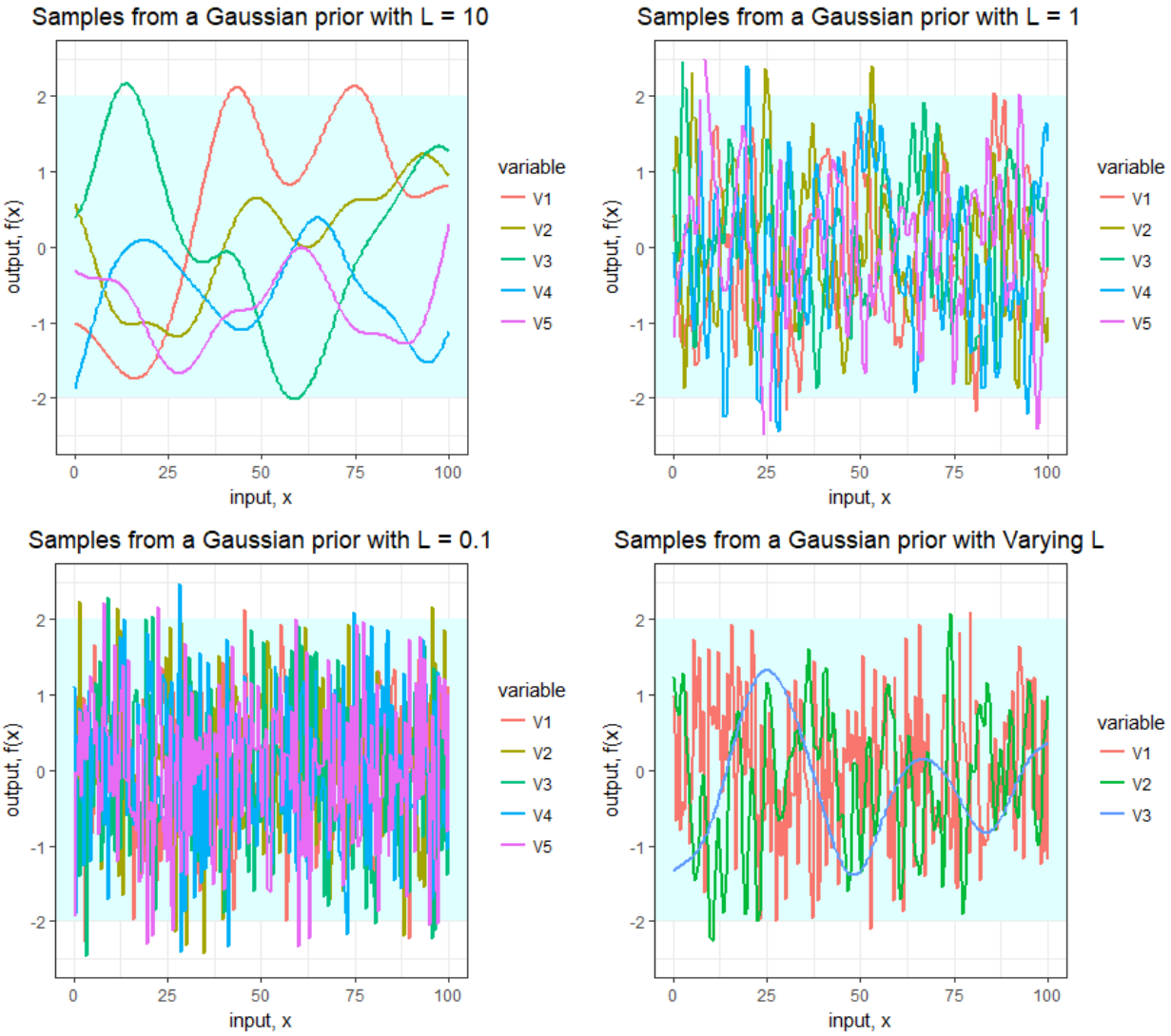
This is known as a squared exponential covariance function, for obvious reasons. This prior encodes the following assumptions:

- The covariance between two data points decreases monotonically as the distance between them increases.

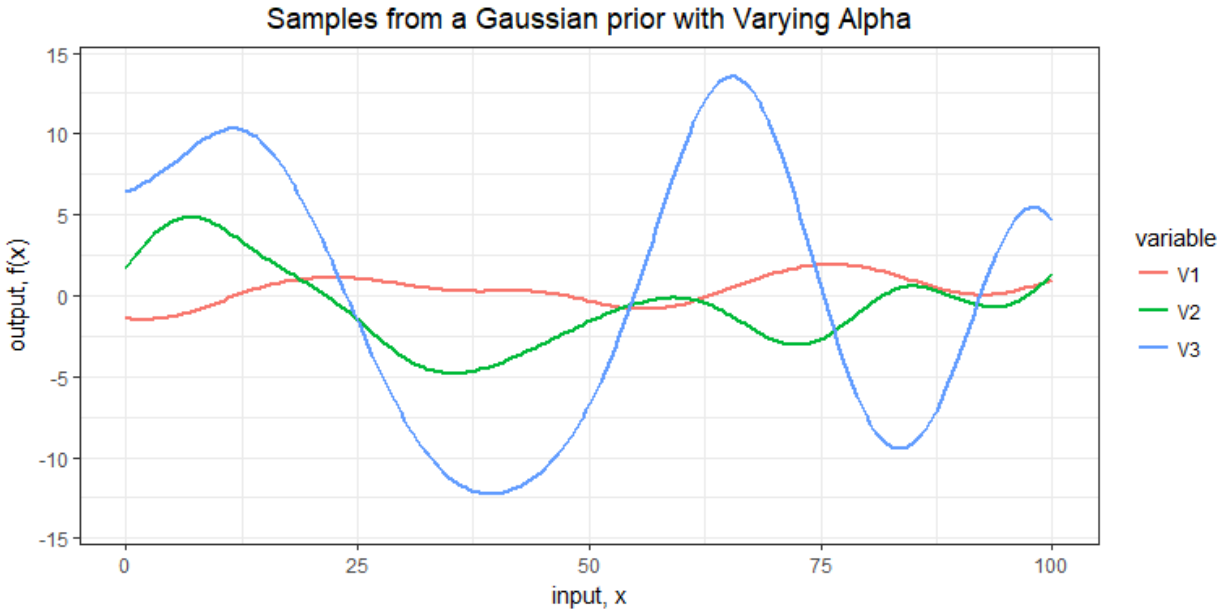
- The covariance function is stationary – it only depends on the distance between x and x', not their locations.
- Even more than being stationary, it is isotropic: It depends only on $|x - x'|$.

Exercise 4.3 *Let's explore the resulting distribution over functions. Write some code to sample from a Gaussian process prior with squared exponential covariance function, evaluated on a grid of 200 inputs between 0 and 100. For $\ell = 1$, sample 5 functions and plot them on the same plot. Repeat for $\ell = 0.1$ and $\ell = 10$. Why do we call ℓ the length-scale of the kernel?*

Solution:The squared exponential covariance functions seem to have a characteristic length-scale ℓ which informally can be thought of as roughly the distance we have to move in input space before the function value can change significantly. I have sampled 5 functions respectively from Gaussian prior with varying $\ell = 10, 1, 0.1$. The following first three plots shows that the distance narrows down as the length-scale decreases. When ℓ takes larger value (≥ 10), the plot is smoother than that of smaller ℓ. The fourth plot compares the sampled functions with different length-scale.



I also compare the plots of varying $\alpha = 1(red), 5(green), 10(blue)$. It changes the scale of the output values. The larger the α, the larger the range of the output values.



Exercise 4.4 *Let* $\mathbf{f}_* := f(\mathbf{X}_*)$ *be the function* f *evaluated at test covariate locations* $\mathbf{X}_*$. *Derive the posterior distribution* $p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y})$, *where* $\mathbf{y}$ *and* $\mathbf{X}$ *comprise our training set. (You can start from the answer to Exercise 1 if you'd like).*

Solution: Gaussian Process Regression is non-parametric model that assumes

$$\mathbf{f} \equiv [f(x_1), f(x_2), \cdots, f(x_n)]^T \sim \mathcal{N}(0, K)$$

where K is the covariance matrix whose (i, j)th element is given by kernel function. And

$$(\mathbf{y}|\mathbf{f}) = [(y_1, y_2, \cdots, y_n|\mathbf{f})]^T \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

Then the prior on the noisy observation is $Cov(\mathbf{y}) = K(X, X) + \sigma^2 I$. The joint distribution of the observed target values and the function values at the test locations under the prior is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

Following standard Bayesian approach, we write

$$P(\mathbf{f}^*|X, \mathbf{y}, X^*) = \int P(\mathbf{f}^*, \mathbf{f}|X, \mathbf{y}, X^*) d\mathbf{f} = \int P(\mathbf{f}^*|\mathbf{f}, X, \mathbf{y}, X^*) p(\mathbf{f}|X, \mathbf{y}, X^*) d\mathbf{f}$$

The derivation of the conditional distribution of $P(f^*|\mathbf{f}, X, \mathbf{y}, X^*)$ and $p(\mathbf{f}|X, \mathbf{y}, X^*)$ can be found in the following link: `https://www.csie.ntu.edu.tw/~cjlin/mlgroup/tutorials/gpr.pdf`. After deriving the conditional distribution, we have the key predictive equation for Gaussian process regression

$$\mathbf{f}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, Cov(\mathbf{f}_*))$$

where

$$\bar{\mathbf{f}}_* = E(\mathbf{f}_*|X, \mathbf{y}, X_*) = K(X_*, X)[K(X, X) + \sigma^2 I]^{-1}\mathbf{y}$$
$$Cov(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma^2 I]^{-1}K(X, X_*)$$

Exercise 4.5 *Return to the faithful dataset. Evaluate the posterior predictive distribution* $p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y})$*, for some reasonable choices of parameters (perhaps explore a few length scales), and plot the posterior mean plus a 95% credible interval on a grid of 200 inputs between 0 and 100, overlaying the actual data.*

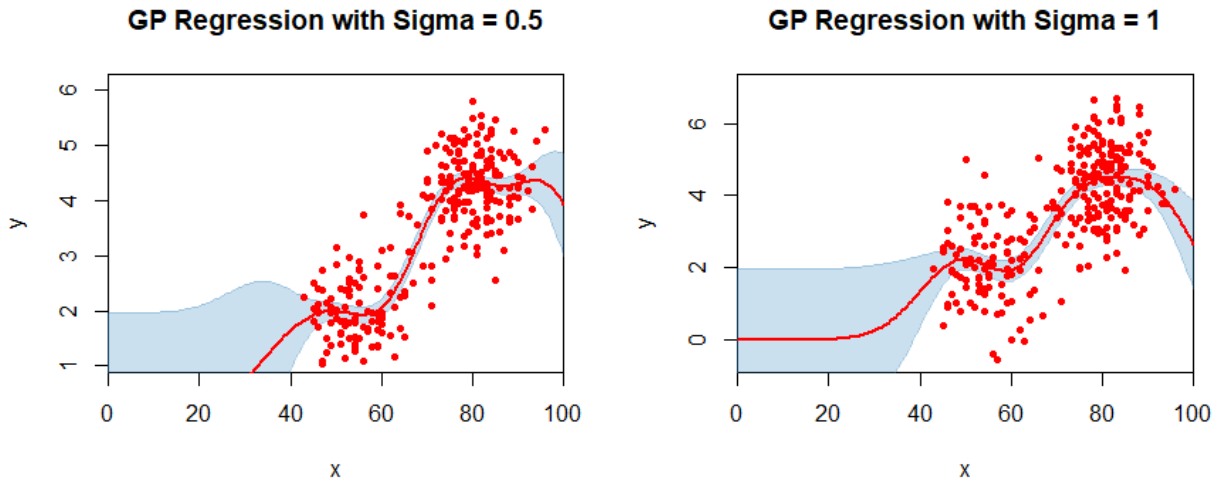


Figure 1: Gaussian Process Regression with $\sigma^2 = 0.25, 1$ and $\ell = 10$; inputs $x \in [0, 100]$

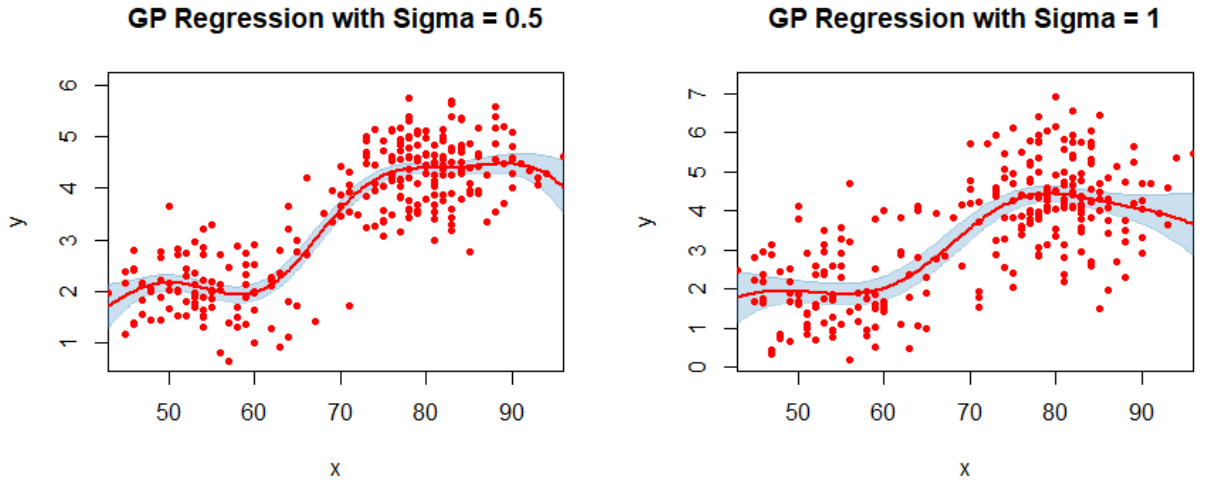


Figure 2: Gaussian Process Regression with $\sigma^2 = 0.25, 1$ and $\ell = 10$; inputs $faithful\$x \in [\min(x), \max(x)]$

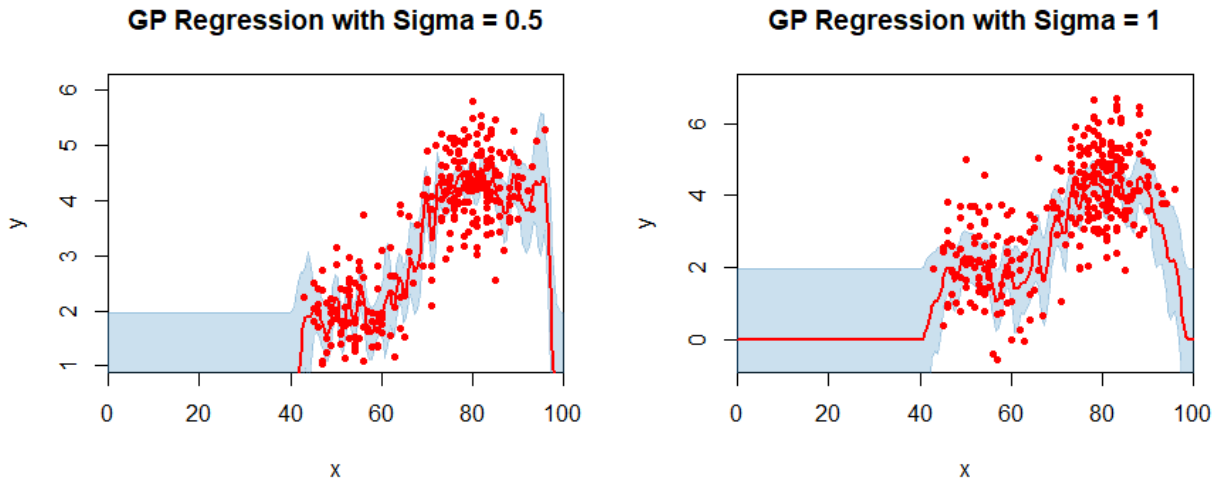


Figure 3: Gaussian Process Regression with $\sigma^2 = 0.25, 1$ and $\ell = 1$; inputs $x \in [0, 100]$

Solution: The above 6 plots show the posterior mean plus 95% credible interval with different choices of parameters, i.e., $\sigma^2 = 0.25, 1$ and $\ell = 1, 10$ (which illustrates the smoothness of the plots). As we can see, Figure 1 shows the plots generated from predictive multivariate normal distribution. Since the range of the datasets is $[43, 96]$, the observations and posterior mean concentrate on the right of the plot. To make it reasonable, I set the range of $x \in [43, 96]$ and have the plots in Figure 2, which is easy to interpret. I also generated the plots with $\ell = 1$, as shown in Figure 3, comparing with Figure 1. There are more spikes when plotting the posterior mean. To get smoother plots, we will choose larger length-scale. In this case, $\ell = 10$ is much better.

4.2 Model selection

As we saw in the previous section, the choice of hyperparameters (for the squared exponential case, the length scale ℓ) effects the properties of the resulting function. Rather than pick a specific value for the hyperparameter, we can specify the model in a hierarchical manner—just like we did in the linear case.

For example, in the squared exponential setting, we could specify our model as

$$\begin{aligned}
\ell^2 &\sim \text{Inv-Gamma}(a_\ell, b_\ell) \\
\alpha^2 &\sim \text{Inv-Gamma}(a_\alpha, b_\alpha) \\
\sigma^2 &\sim \text{Inv-Gamma}(a_\sigma, b_\sigma) \\
k(x, x') &= \alpha^2 \exp\left\{-\frac{1}{2\ell^2}|x - x'|^2\right\} + \sigma^2 \delta_{x-x'} \\
y|X &\sim N(0, \tilde{K})
\end{aligned}$$

where K is the covariance function evaluated at the input locations X. Note that we have integrated out f and placed our prior directly on y, incorporating the Gaussian likelihood into the covariance. We can then infer the posterior distribution over ℓ using Bayes' Law:

$$p(\ell|y, X) = \frac{p(y|X, \ell)p(\ell)}{\int_0^\infty p(y|X, \ell)p(\ell)d\ell}$$

Unfortunately, we typically do not have an analytical form for this posterior, so we must resort to either optimization, or MCMC-based inference.

4.2.1 Optimization

In practice, a common approach is to find the ML estimate for the hyperparameters. Let's assume a generic setting, where the log likelihood is parametrized by some vector of parameters θ. The log likelihood is given by

$$logp(y|X, \theta) = -\frac{1}{2}y^T K^{-1}y - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi$$

Taking partial derivatives, we see that

$$\begin{aligned}\frac{\partial}{d\partial\theta_j}\log p(y|X, \theta) =& \frac{1}{2}y^T K^{-1}\frac{\partial K}{\partial\theta_j}K^{-1}y - \frac{1}{2}\text{tr}\left(K^{-1}\frac{\partial K}{\partial\theta_j}\right) \\ =& \frac{1}{2}\text{tr}\left((\alpha\alpha^T - K^{-1})\frac{\partial K}{\partial\theta_j}\right)\end{aligned}$$

where $\alpha = K^{-1}y$. We can use these partial derivatives to find the ML estimate of θ, using a gradient-based optimization method.

Exercise 4.6 *Calculate the appropriate derivatives for the one-dimensional, squared exponential case used for the* `faithful` *dataset. Use these gradient to find the optimizing value of ℓ^2, α^2 and σ^2. Plot the resulting fit.*

Exercise 4.7 *Repeat the previous exercise, but this time only use the first 10 data points from the faithful dataset. Repeat the optimization several times, using different initializations/random seeds. You will likely see widely different results – sometimes ℓ is big, sometimes σ^2 is big. Why is this? Discuss why this is a problem here, but wasn't in the previous setting. You may find it helpful to look at the corresponding scatter plot, or plot the log likelihood for certain values of σ^2 and ℓ.*

4.2.2 MCMC

Optimization is typically pretty quick, which is why it is commonly used in practice. However, we have no guarantee that our optimization surface is convex. An alternative approach is to sample from the posterior distribution over our hyperparameters.

Exercise 4.8 *Since the posterior is non-conjugate, we can't use a Gibbs sampler. We won't go into the details of appropriate sampling methods since this isn't an MCMC course, but we will explore using black-box samplers. In the R folder, there are three files:* `faithful_data.R`, `gp_regression.stan` *and* `run_gp_regression.R`. *Use these to sample from the model and produce 95% credible intervals for α, ℓ and σ, and 95% predictive intervals for t. Go through the code and make sure you understand what is going on.*

Exercise 4.9 *Let's now look at a dataset with multiple predictors. Download the dataset weather.csv – this contains latitude and longitude data for 147 weather stations, plus a response "temperature", which is the difference between the forecasted and actual temperature for each station.*

How should we extend our kernel to multiple dimensions? (There is more than one option here). Should we use the same lengthscale for latitude and longitude? Construct an appropriate parametrized kernel, and learn the parameters either via optimization or using MCMC by editing the Stan code (Note: If you go for the stan code, you will need to implement your new kernel).

Using an appropriate visualization tool, plot the mean function (try imshow or contourf in matlab or matplotlib (for python), or image or filled.contour for R).

4.3 Beyond regression: non-conjugate likelihoods

So far, we've focused on Gaussian processes in a regression context. We can however use them as the basis of a non-Gaussian regression... using exactly the same techniques as we used for the regression setting! For example, in Section 3, we dealt with non-Gaussian data by transforming our regression output:

$$y_i|\beta, x_i \sim f(g^{-1}(x_i^T\beta))$$

where f is an appropriate likelihood model (e.g. Bernoulli for binary data, Poisson for count data) and g^{-1} was a function that maps the real-valued $x_i^T\beta$ to an appropriate space for that likelihood.

We can do exactly the same here, by letting

$$\begin{aligned} f \sim& \text{GP}(0, k) \\ y_i \sim& f(g^{-1}(f(x_i))) \end{aligned}$$

Let's start by considering a binary example. In the GLM setting, we looked at both probit and logit regression. We can use the same approaches here!

Exercise 4.10 *Describe (including pseudo-code) how we could implement probit Gaussian process regression, using an auxiliary variable method analogous to that used in Exercise 3.1.*

For the logit case, we can again use a Laplace approximation to approximate our posterior. In the GLM setting, we used the Laplace approximation to approximate the posterior over β. Here, we will work directly with our function f evaluated at our training locations, and approximate $p(f|X, y, \theta)$, where θ are the parameters of our covariance function.

Let $P^*(f) \propto p(f|X, y, \theta)$ be our unnormalized posterior, so that $\log P^*(f) = \log p(y|f) + logp(f|X) = \log p(y|f) - \frac{1}{2} f^T K^{-1} f - \frac{1}{2}\log|K| + \text{const}$.

Exercise 4.11 *Derive the Hessian of* $\log P^*(f)$*, when* $y_i \sim$ *Bernoulli* $\left(\frac{1}{1+e^{-f_i}}\right)$

Exercise 4.12 *The dataset iris.csv contains details of 150 flowers from three species. Pick two of them (your choice) as your regression dataset. Find the MAP of* f*, for some reasonable choice of hyperparameters and squared exponential kernel. Visualize the corresponding class probabilities on a series of 2d plots.*

Exercise 4.13 *Evaluate the Hessian using the same dataset, and visualize uncertainty in your plots from the previous exercise (e.g. by creating a contour plot of the marginal standard deviations).*

Exercise 4.14 *In a multi-class setting, an appropriate likelihood is the multinomial, which is parametrized by a simplex-valued vector* $\pi = (\pi_1, \ldots, \pi_K)$*. We can map a real-valued vector* y *to the simplex using the softmax transformation:*

$$\pi_i = \frac{e^{y_i}}{\sum_{j=1}^{K} e^{y_k}}$$

To use this transformation, we will have to have one Gaussian process for each class.[1] *Practically, we can think of this as using a single Gaussian process, but with a block-diagonal covariance matrix with* K $N \times N$ *blocks. Using a Laplace approximation to the posterior distribution, repeat the previous three exercises using all three types of iris.*

[1] Technically, we could use $K - 1$ GPs plus a constant reference value for the third class, but let's use K for now.