SDS 383D: Modeling II **Spring 2017**

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*

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

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 we did in Section 1.*

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 $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}^T \Sigma \phi(\mathbf{x}')$. 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 refered 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 *and* σ^2*, 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.*

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

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

. 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 $m(x)$.

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 datapoints 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 lengthscale of the kernel?*

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).*

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 if you're not sure what to pick), and plot the posterior mean plus a 95% credible interval on a grid of 200 inputs between 0 and 100, overlaying the actual data.*

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

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

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.*

Answer: The optimal vector of hyperparameters, $\theta = \{\alpha^2, \ell^2, \sigma^2\}$ was found using a gradient-based optimization method. The partial derivatives wrt each hyperparameter are given below.

$$\frac{\partial}{\partial \alpha^2} = exp\left[\frac{-1}{2\ell^2}|x - x'|^2\right] \tag{4.1}$$

$$\frac{\partial}{\partial \ell^2} = \frac{\alpha^2 exp\left[\frac{-1}{2\ell^2}|x - x'|^2\right]|x - x'|^2}{2\ell^4} \tag{4.2}$$

$$\frac{\partial}{\partial \sigma^2} = \delta_{x-x'} \tag{4.3}$$

The MATLAB function `fitgrp` was used with the kernel function k(x,x'), and predicted values were calculated using `predict`, where it is recommended to input the training data. The functions `fitgrp` and `predict` are great for fitting Gaussian Processes, however, the training and test data need to be the same size, which can lead to less accurate fits for small datasets, given that the data must be split in half.

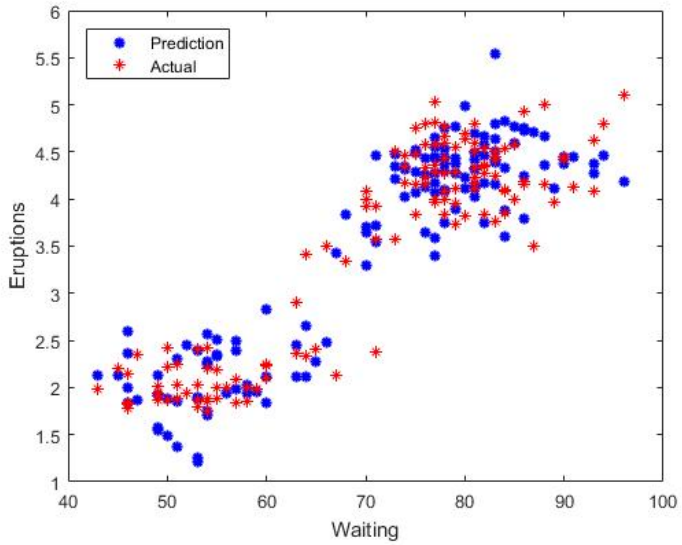

Figure 4.1: Plot showing the actual and predicted values using test data.

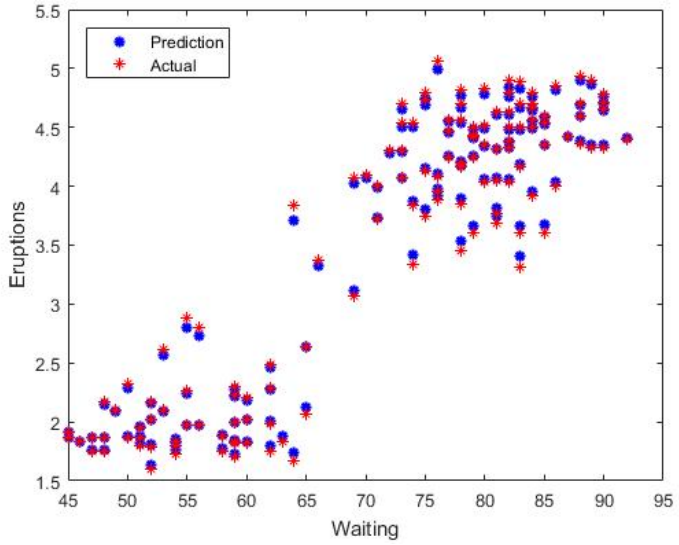

Figure 4.2: Plot showing the actual and predicted values using training data.

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.*