Using Gaussian Process Models to Predict for a Test Case

We have a training set of N cases, with inputs $x_1, \ldots, x_N$ and real-valued responses $y_1, \ldots, y_N$. We decide to use a Gaussian process model with mean zero and covariance function C for which $\text{Cov}(y_{i_1}, y_{i_2}) = C(x_{i_1}, x_{i_2})$.

How do we make a prediction for the response, y_*, in a test case with inputs x_*?

The Gaussian process prior assigns a multivariate Gaussian distribution to $y_1, \ldots, y_N, y_*$, with covariance determined by $x_1, \ldots, x_N, x_*$, which we know.

We also know $y_1, \ldots, y_N$. So we can find the conditional distribution for the response in the test case, $y_* \,|\, y_1, \ldots, y_N$.

Conditional distributions for multivariate Gaussians are Gaussian. So the predictive distribution for y_* is Gaussian with some mean and variance.
If we need to make a single-valued guess, we guess the mean.

It's also possible to randomly sample from the posterior distribution of function values at some set of point — this lets us see what the posterior distribution is like (in 1D or 2D).

Details of How to Predict

The details come from the general theory of multivariate Gaussian distributions.

Assume the GP has mean zero, and covariance function function C.

Create a matrix $\mathbf{C}$ of covariances for the training cases, with $C_{i_1,i_2} = C(x_{i_1}, x_{i_2})$.

Create a vector $\mathbf{k}$ of covariances of training cases with the test case, so that $k_i = C(x_i, x_*)$.

Compute $v = C(x_*, x_*)$, the prior variance of the response in the test case.

Then compute the mean and variance of y_* given $y_1, \ldots, y_N$ as

$$\begin{aligned} E(y_* \,|\, y_1, \ldots, y_N) &= \mathbf{k}^T \mathbf{C}^{-1} \mathbf{y} \\ \mathrm{Var}(y_* \,|\, y_1, \ldots, y_N) &= v \;-\; \mathbf{k}^T C^{-1} \mathbf{k} \end{aligned}$$

where $\mathbf{y} = [y_1, \ldots, y_N]^T$ is the vector of responses in training cases.

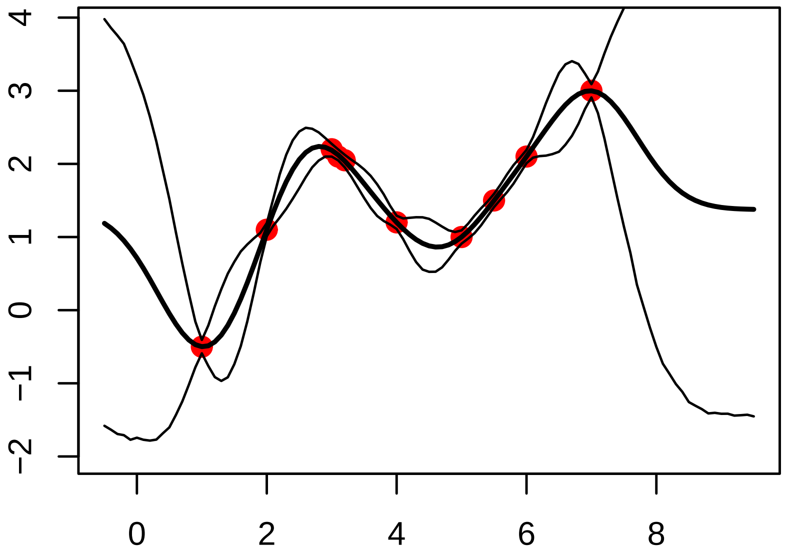
We can compute $\mathbf{C}^{-1}$ and $\mathbf{C}^{-1}\mathbf{y}$ once and then use them for many test cases. Predicting for a test case then takes time proportional to N if we just want the mean, and to N^2 if we want the variance too.

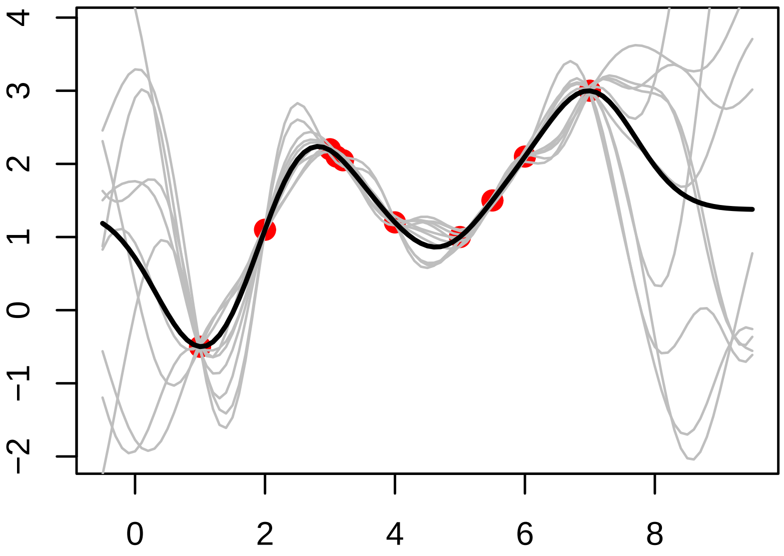
Example of a Posterior Distribution From a GP Model

This 1D example uses a covariance function with a constant term that allows the overall mean of the function to be non-zero, and an exponential term producing smooth functions. The final term is for noise with standard deviation 0.05.

$$C(x_{i_1}, x_{i_2}) \;=\; 5^2 \;+\; 2^2 \exp(-(x_{i_1} - x_{i_2})^2) \;+\; \delta_{i_1,i_2} 0.05^2$$

Ten training cases are shown below as red dots. The left plot shows mean predictions for tests cases from -0.5 to 9.5 and 10% and 90% quantiles of the predictive distribution. The right plot shows ten functions from the posterior.



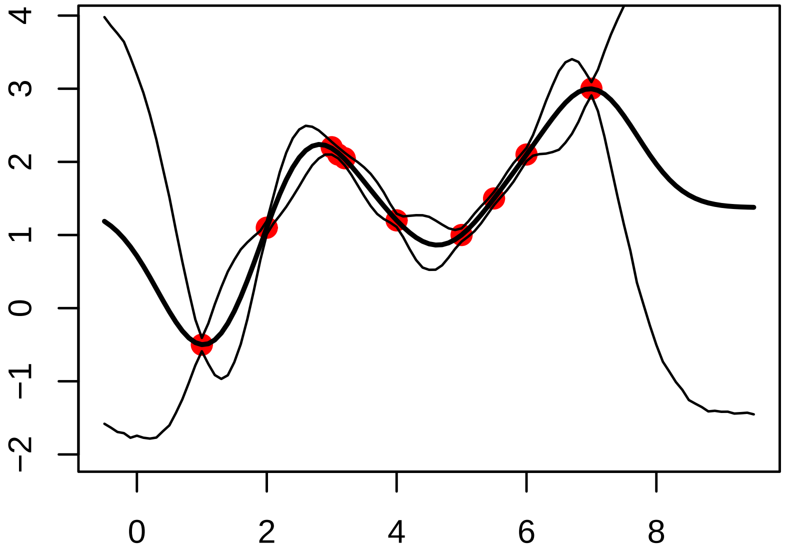


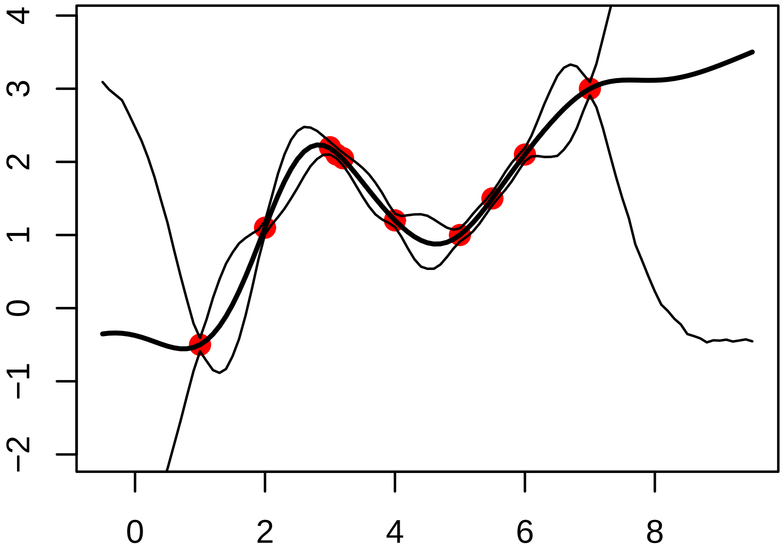
Effect of Allowing for a Linear Trend

We can compare the results with those using a covariance function that also has an $x_{i_1} x_{i_2}$ term corresponding to a linear function:

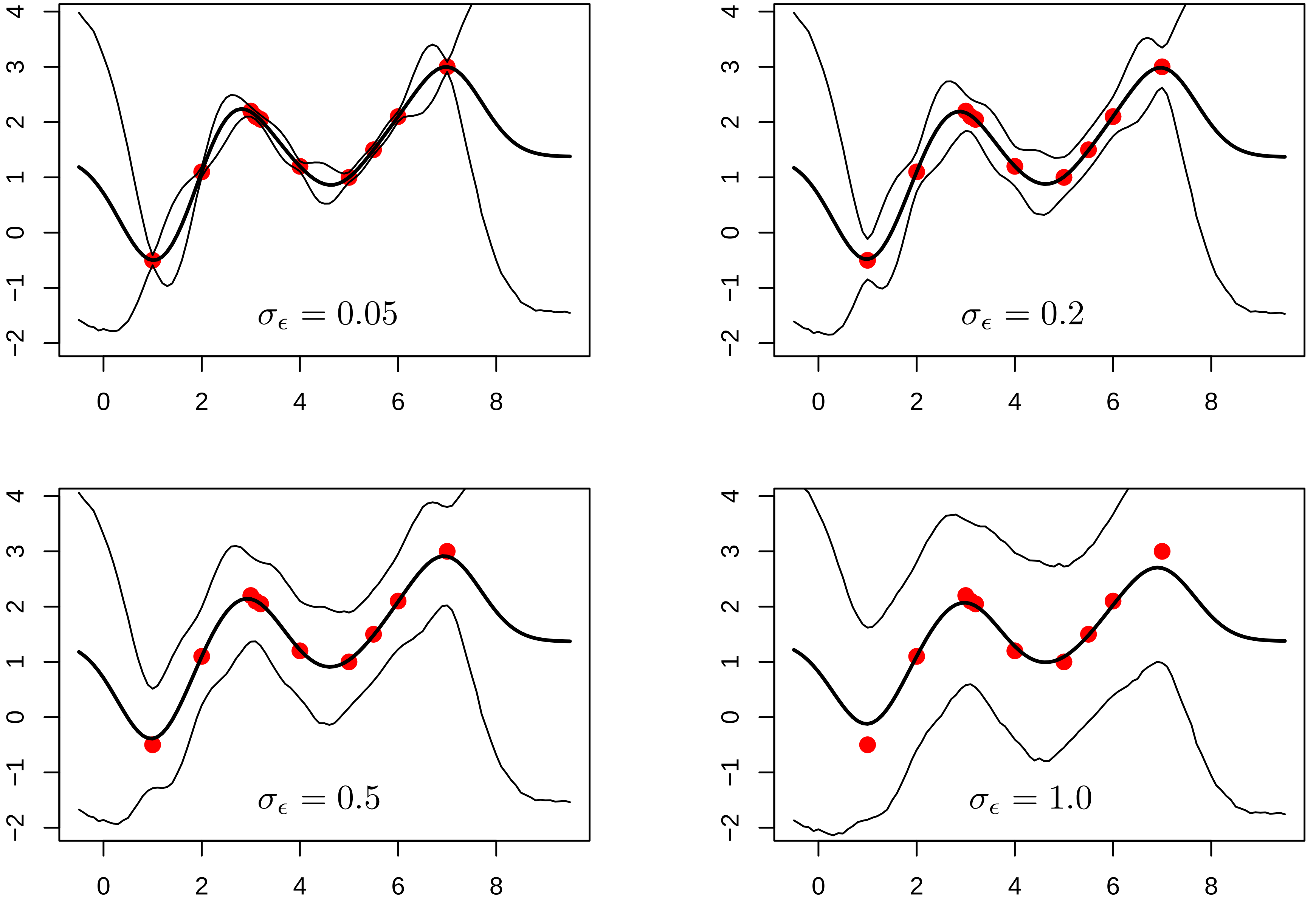
$$C(x_{i_1}, x_{i_2}) \;=\; 5^2 \,+\, x_{i_1} x_{i_2} \,+\, 2^2 \exp(-(x_{i_1} - x_{i_2})^2) \,+\, \delta_{i_1,i_2} 0.05^2$$

The left plot shows the mean and 10% and 90% quantiles for the previous covariance function; the right plot shows the same for the new covariance function:

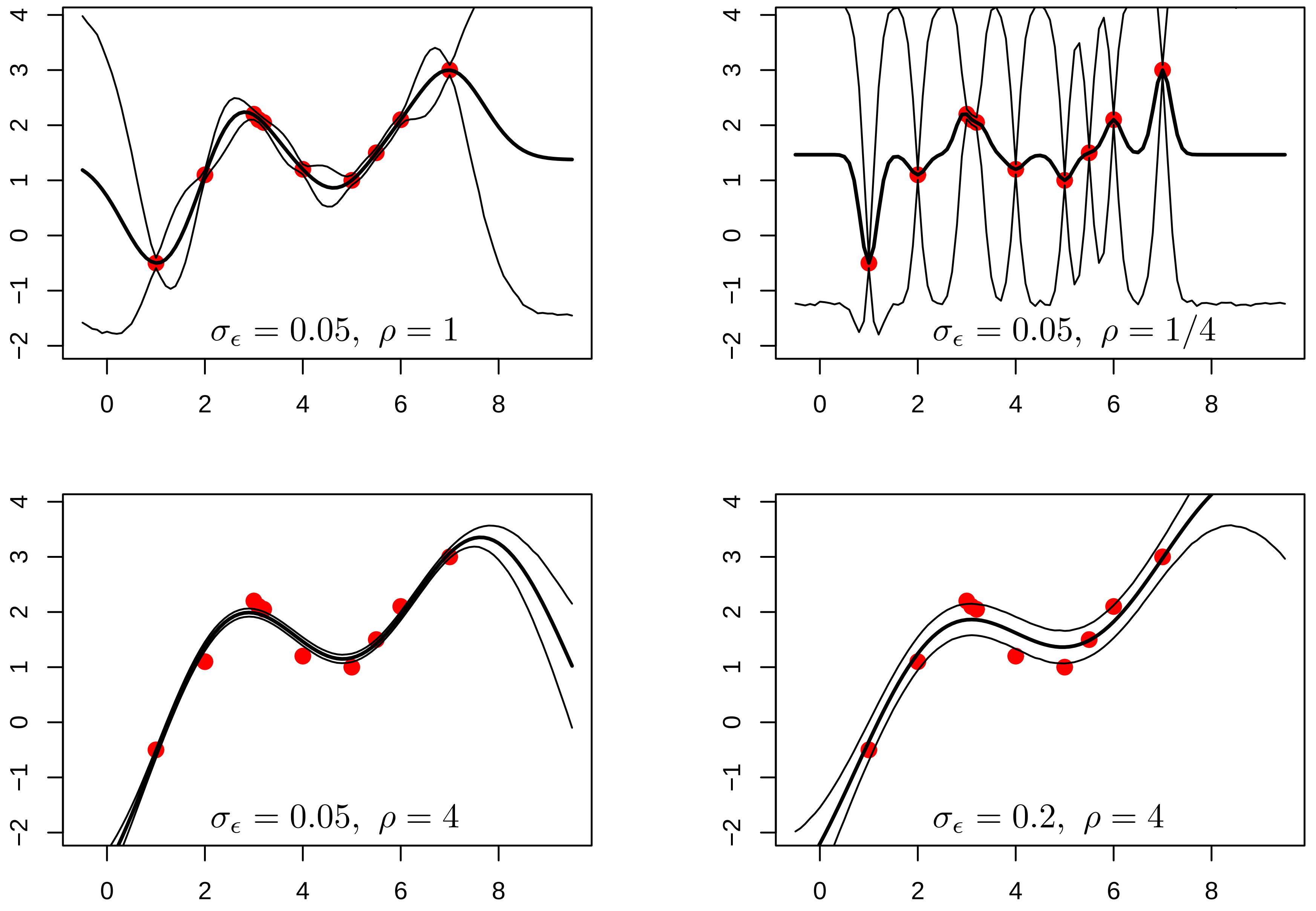




Effect of Changing the Noise Standard Deviation



Effect of Changing Noise and Length Scale



Choosing Values for Hyperparameters

In Bayesian terminology, η, ρ, and σ_ϵ are often called *hyperparameters*, because they control overall characteristics of the function — as opposed to the specific details, as for the parameters β of a linear regression model.

They play roles similar to the penalty parameter λ in ridge regression or splines.

Choosing the right values for hyperparameters is crucial to getting good results. How should we do this?

- We could use cross validation, as is commonly done for penalties.
- But the Bayesian method is to choose values that maximize the probability of the data — or better, to *average* over many values of the hyperparameters, based on the probability they give to the data and their prior probability.

For Gaussian process models, it's easy to compute the probability of the data for given values of the hyperparameters. The hyperparameters determine the covariance matrix, $\mathbf{C}$, of the N training responses, $\mathbf{y}$. We then just compute the mutivariate Gaussian density for y:

$$(2\pi)^{-N/2}\,\det(C)^{-1/2}\,\exp(-\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}/2)$$