Gaussian Processes

Robin Aldridge-Sutton

07/05/2022

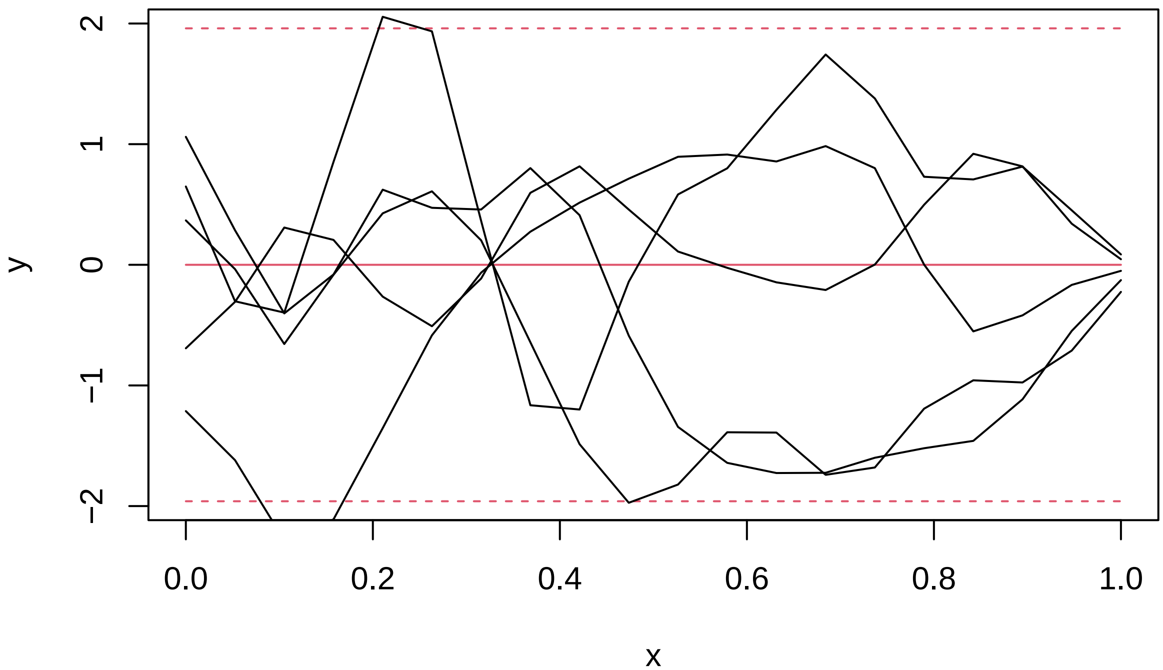
```r
# Functions to sample from and predict values of a Gaussian process.
source("GP funcs.R")
```

A Gaussian process (GP) is stochastic process (a distribution over functions) such that for any finite set of input values the function values have a multivariate Gaussian distribution.

A GP can be used as a functional prior, e.g.

$$
\begin{gathered}
f(\mathbf{x}) \sim N(\mu(\mathbf{x}) = \mathbf{0}, \Sigma = K(\mathbf{x}, \mathbf{x})), \\
K(\mathbf{x}, \mathbf{x}')_{i,j} = \sigma_f^2 \exp\left(\frac{(x_i - x_j')^2}{2l^2}\right).
\end{gathered}
$$

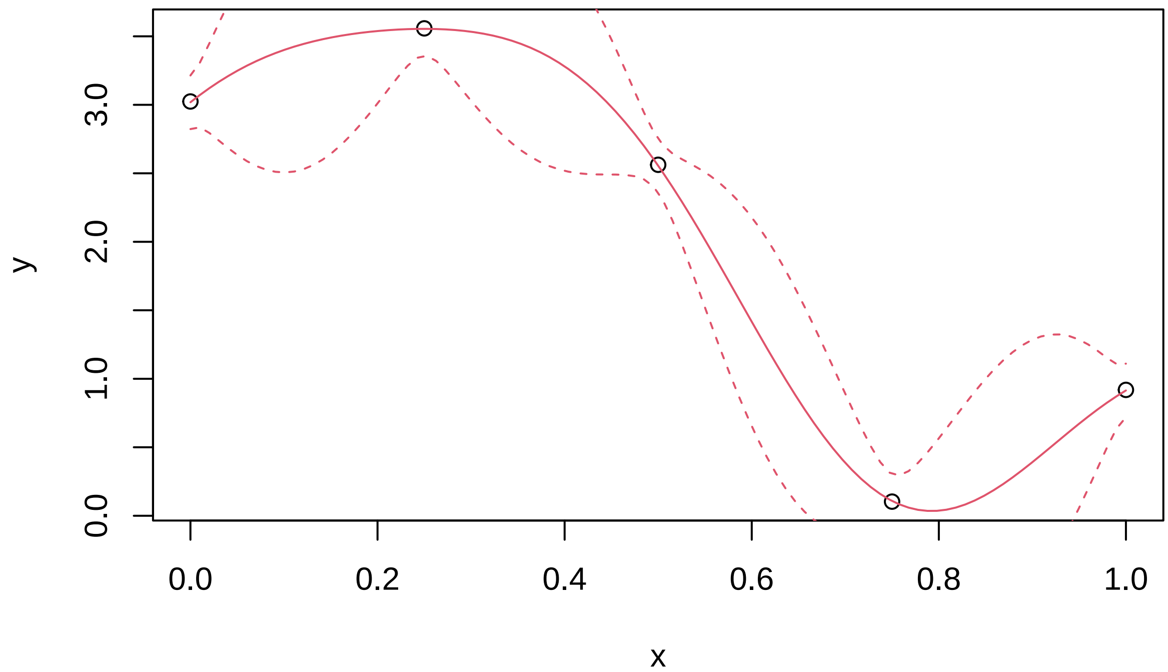
```r
# Plot samples from a GP. Too many samples gives an error in the Cholesky
# factorization of the covariance matrix, but why?? Smaller length-scales allow
# larger samples... Function sd makes no difference.
plot_samp(
  n_samp = 20, # Number of samples per realisation
  n_real = 5, # Number of realisations
  l = 0.1, # Length scale
  sigma_f = 1 # Function standard deviation
)
```



The posterior is then the conditional distribution of functions given observed function values at a set of input values. This has a simple analytical form.

$$
\begin{gathered}
f(\mathbf{x})|f(\mathbf{x}') \sim N(\mu(\mathbf{x}), \Sigma), \\
\mu(\mathbf{x}) = K(\mathbf{x}', \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}f(\mathbf{x}'), \\
\Sigma = K(\mathbf{x}', \mathbf{x}') - K(\mathbf{x}', \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}K(\mathbf{x}, \mathbf{x}').
\end{gathered}
$$

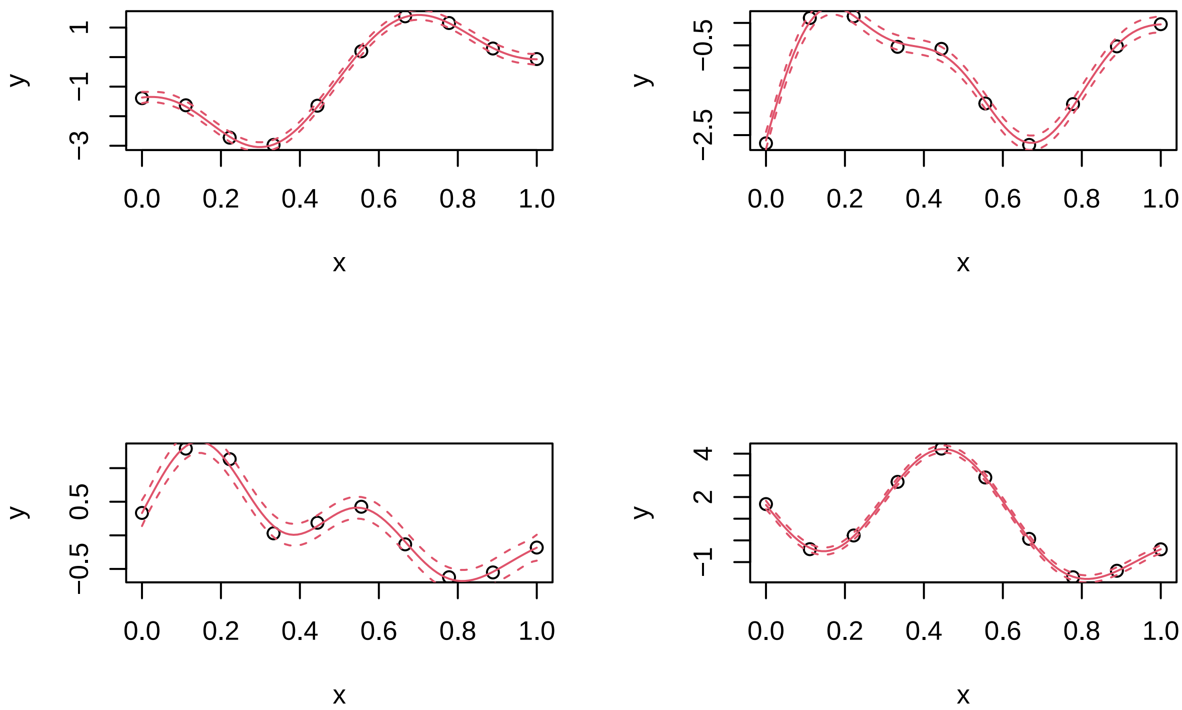
```r
plot_GP(
  n_samp = 5, # Number of points to sample
  n_pred = 100, # Number of data points to predict
  l = 0.2, # Length scale
  sigma_f = 2, # Function standard deviation
  # sigma_n = 0.1 # Noise standard deviation
  sigma_n = 0.1 # Noise standard deviation
)
```



$$\mathbf{y} = f(\mathbf{x}) + \epsilon,$$
$$\epsilon \sim N(0, \sigma_n^2 I_d)$$
$$\mathbf{x} \in \mathbb{R}^d$$

```
par(mfrow = c(2, 2))

for (i in 1:4)
  plot_GP(
    n_samp = 10, # Number of points to sample
    n_pred = 100, # Number of data points to predict
    l = 0.2, # Length scale
    sigma_f = 2, # Function standard deviation
    sigma_n = 0.1 # Noise standard deviation
  )
```


y
1
−1
−3
0.0
0.2
0.4
0.6
0.8
1.0
x
y
−0.5
−2.5
0.0
0.2
0.4
0.6
0.8
1.0
x
y
0.5
−0.5
0.0
0.2
0.4
0.6
0.8
1.0
x
y
4
2
−1
0.0
0.2
0.4
0.6
0.8
1.0
x