Solution 1: Gaussian Posterior Process

(a) Prior distribution (assuming the same notation as in the lecture):

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K})$$

with $\boldsymbol{m} = m(\boldsymbol{x})$ and $\boldsymbol{K}$ defined by the entries $\boldsymbol{K}_{ij} = k(x_i, x_j)$. NB: Note the (in-)finite Gaussian property of a GP.

(b) Note that the posterior distribution $\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{x}$ in this case is different from the one of $\boldsymbol{f}_*|\boldsymbol{x}_*, \boldsymbol{x}, \boldsymbol{y}$ and also from the marginal distribution of $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K} + \sigma^2\boldsymbol{I})$! We have:

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{f}) \cdot p(\boldsymbol{f}) \\ &\propto \exp(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{f})^\top(\sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y}-\boldsymbol{f})) \cdot \exp(-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{m})^\top\boldsymbol{K}^{-1}(\boldsymbol{f}-\boldsymbol{m})) \\ &\propto \exp(-\frac{1}{2}\{\boldsymbol{f}^\top \underbrace{((\sigma^2\boldsymbol{I})^{-1} + \boldsymbol{K}^{-1})}_{=:\boldsymbol{K}_{post}^{-1}} \boldsymbol{f} - 2\boldsymbol{f}^\top \underbrace{((\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y} + \boldsymbol{K}^{-1}\boldsymbol{m})}_{=:\tilde{\boldsymbol{f}}}\}) \\ &\propto \exp(-\frac{1}{2}\{\boldsymbol{f}^\top\boldsymbol{K}_{post}^{-1}\boldsymbol{f} - 2\boldsymbol{f}^\top\tilde{\boldsymbol{f}}\}) \end{aligned} \tag{1}$$

by removing all constant factors that do not depend on $\boldsymbol{f}$ as we only need to know the density up to a constant of proportionality. By extending the proportionality, we can get a quadratic form in $\boldsymbol{f}$:

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{y}) &\propto \exp(-\frac{1}{2}\{\boldsymbol{f}^\top\boldsymbol{K}_{post}^{-1}\boldsymbol{f} - 2\boldsymbol{f}^\top\tilde{\boldsymbol{f}}\}) \\ &\propto \exp(-\frac{1}{2}\{\boldsymbol{f}^\top\boldsymbol{K}_{post}^{-1}\boldsymbol{f} - 2\boldsymbol{f}^\top\boldsymbol{K}_{post}^{-1}\underbrace{\boldsymbol{K}_{post}\tilde{\boldsymbol{f}}}_{:=\boldsymbol{f}_{post}}\}) \\ &\propto \exp(-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{f}_{post})^\top\boldsymbol{K}_{post}^{-1}(\boldsymbol{f}-\boldsymbol{f}_{post})) \end{aligned} \tag{2}$$

which is the so-called *kernel* of a multivariate normal distribution $\mathcal{N}(\boldsymbol{f}_{post}, \boldsymbol{K}_{post})$, i.e., $\boldsymbol{f}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{f}_{post}, \boldsymbol{K}_{post})$.

(c) In order to get the posterior predictive distribution for a new sample x_* from the same data-generating process, we could derive

$$p(y_*|x_*, \boldsymbol{y}, \boldsymbol{x}) = \int p(y_*|x_*, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{f}) \cdot p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{x})\, d\boldsymbol{f}.$$

This is feasible but cumbersome. Alternatively, we can make use of the fact that the joint distribution of $\boldsymbol{y}$ and y_* is known (cf. slides on noisy GP):

$$\begin{pmatrix} \boldsymbol{y} \\ y_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{m} \\ m_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K} + \sigma^2\boldsymbol{I} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^\top & K_{**} \end{pmatrix}\right),$$

with $m_* = m(x_*)$, $\boldsymbol{K}_* = k(x_*, \boldsymbol{x})$ and $K_{**} = k(x_*, x_*)$. The conditional distribution can then be derived using the rule of conditioning for Gaussian distributions:

$$y_*|x_*, \boldsymbol{x}, \boldsymbol{y} \sim \mathcal{N}(m_* + \boldsymbol{K}_*^\top(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{m}), K_{**} - \boldsymbol{K}_*^\top(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}_*).$$

(d) To implement a GP with squared exponential kernel and $\ell = 1$, we need the inverse of $\boldsymbol{K}$. $\boldsymbol{x}$ being a vector implies that we have only one feature and thus the entries of our matrix $\boldsymbol{K}$ are

$$\boldsymbol{K} = \begin{pmatrix} 1 & \exp(-0.5(x^{(1)} - x^{(2)})^2) \\ \exp(-0.5(x^{(2)} - x^{(1)})^2) & 1 \end{pmatrix}.$$

The inverse of $\boldsymbol{K}$ is then given by

$$\frac{1}{1-\exp(-(x^{(1)}-x^{(2)})^2)}\begin{pmatrix} 1 & -\exp(-0.5(x^{(1)}-x^{(2)})^2) \\ -\exp(-0.5(x^{(2)}-x^{(1)})^2) & 1 \end{pmatrix}.$$

If we have a noisy GP, we would have to add $\sigma^2 \boldsymbol{I}_2$ to $\boldsymbol{K}$ with resulting inverse

$$\boldsymbol{K}_y^{-1} = \frac{1}{(1+\sigma^2)^2-\exp(-(x^{(1)}-x^{(2)})^2)}\begin{pmatrix} 1+\sigma^2 & -\exp(-0.5(x^{(1)}-x^{(2)})^2) \\ -\exp(-0.5(x^{(2)}-x^{(1)})^2) & 1+\sigma^2 \end{pmatrix}.$$

Assuming a zero mean GP, we can derive $\frac{\partial \boldsymbol{K}_y}{\partial \theta}$ with $\theta = \sigma^2$, which gives us the identity matrix. We can thus maximize the marginal likelihood (slide on *Gaussian Process Training*), by finding σ^2 that yields

$$\operatorname{tr}\left(\boldsymbol{K}_y^{-1}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{K}_y^{-1} - \boldsymbol{K}_y^{-1}\right) = 0.$$

This can be solved analytically (though quite tedious). We will use a root-finding function for this. For the posterior predictive distribution we can make use of the results from the previous exercise.

```
library(kernlab)

## Error in library(kernlab):  there is no package called 'kernlab'

# set seed, define n, true (unknown) sigma
set.seed(4212)
n <- 2
sigma <- 1

# define kernel with l = 1
kernel_fun <- function(x)
  kernelMatrix(kernel = rbfdot(sigma = 1/2),
               x = x)
kernel_fun_pred <- function(x,y)
  kernelMatrix(kernel = rbfdot(sigma = 1/2),
               x = x, y = y)

# draw data according to the generating process:
x <- rnorm(n)
K <- kernel_fun(x)

## Error in kernelMatrix(kernel = rbfdot(sigma = 1/2), x = x):  could not find function
"kernelMatrix"

K_y <- K + diag(rep(sigma^2,2))

## Error in eval(expr, envir, enclos):  object 'K' not found

(y <- t(mvtnorm::rmvnorm(1, sigma = K_y)))

## Error in isSymmetric(sigma, tol = sqrt(.Machine$double.eps), check.attributes = FALSE):
object 'K_y' not found

# function to find the best sigma^2
root_fun <- function(sigmaSq){
  K_y_inv <- solve(K + diag(rep(sigmaSq,2)))
  0.5*sum(diag(K_y_inv%*%y%*%t(y)%*%K_y_inv - K_y_inv))
}

# get the best sigma
(bestSigmaSq <- uniroot(f = root_fun, interval = c(0,20)))$root
```

```
## Error in solve(K + diag(rep(sigmaSq, 2))):  object 'K' not found

# plot the optimization problem and best sigma
possible_sigvals <- seq(0.001,20,l=1000)
plot(possible_sigvals, sapply(possible_sigvals, root_fun),
     xlab = expression(sigma^2), ylab = "marginal likelihood derivative",
     pch = 20)

## Error in solve(K + diag(rep(sigmaSq, 2))):  object 'K' not found

abline(h=0, lty=2)

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  plot.new has not
been called yet

abline(v=bestSigmaSq$root, lty=2)

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  object
'bestSigmaSq' not found

# function to draw samples from the predictive posterior
draw_from_pred_posterior <- function(number_samples, y, x, xstar, sigmaSq = 1)
{

  # invert noisy K
  K_y_inv <- solve(kernel_fun(x) + diag(rep(sigmaSq,2)))
  # get the other K's for new data
  Kstar <- kernel_fun_pred(x,xstar)
  Kstarstar <- kernel_fun(xstar)
  # draw samples according to Ex. (d)
  rnorm(number_samples,
        mean = as.numeric(t(Kstar) %*% K_y_inv %*% y),
        sd = sqrt(as.numeric(Kstarstar - t(Kstar) %*% K_y_inv %*% Kstar))
  )

}

# draw enough samples to get a feeling for the distribution
samples_posterior <-
       draw_from_pred_posterior(number_samples = 1000, sigmaSq = bestSigmaSq$root,
                                y = y, x = x, xstar = 0)

## Error in kernelMatrix(kernel = rbfdot(sigma = 1/2), x = x):  could not find function
"kernelMatrix"

# plot the distribution
hist(samples_posterior, breaks=50, xlab=expression(y["*"]^b))

## Error in hist(samples_posterior, breaks = 50, xlab = expression(y["*"]^b)):  object
'samples_posterior' not found
```