

Solution 1:

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

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

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

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

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

by removing all constant factors that do not depend on \mathbf{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 \mathbf{f} :

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

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

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

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

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

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

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

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

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

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

The inverse of \mathbf{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 $\sigma^2 \mathbf{I}_2$ to \mathbf{K} with resulting inverse

$$\mathbf{K}_y^{-1} = \frac{1}{\sigma^4 - \exp(-(x^{(1)} - x^{(2)})^2)} \begin{pmatrix} \sigma^2 & -\exp(-0.5(x^{(1)} - x^{(2)})^2) \\ -\exp(-0.5(x^{(2)} - x^{(1)})^2) & \sigma^2 \end{pmatrix}.$$

Assuming a zero mean GP, we can derive $\frac{\partial \mathbf{K}_y}{\partial \theta}$ 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

$$\text{tr}(\mathbf{K}_y^{-1} \mathbf{y} \mathbf{y}^\top \mathbf{K}_y^{-1} - \mathbf{K}_y^{-1}) = 0.$$

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)

## Warning: package 'kernlab' was built under R version 4.0.3

# 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)
K_y <- K + diag(rep(sigma^2,2))
(y <- t(mvtnorm::rmvnorm(1, sigma = K_y)))

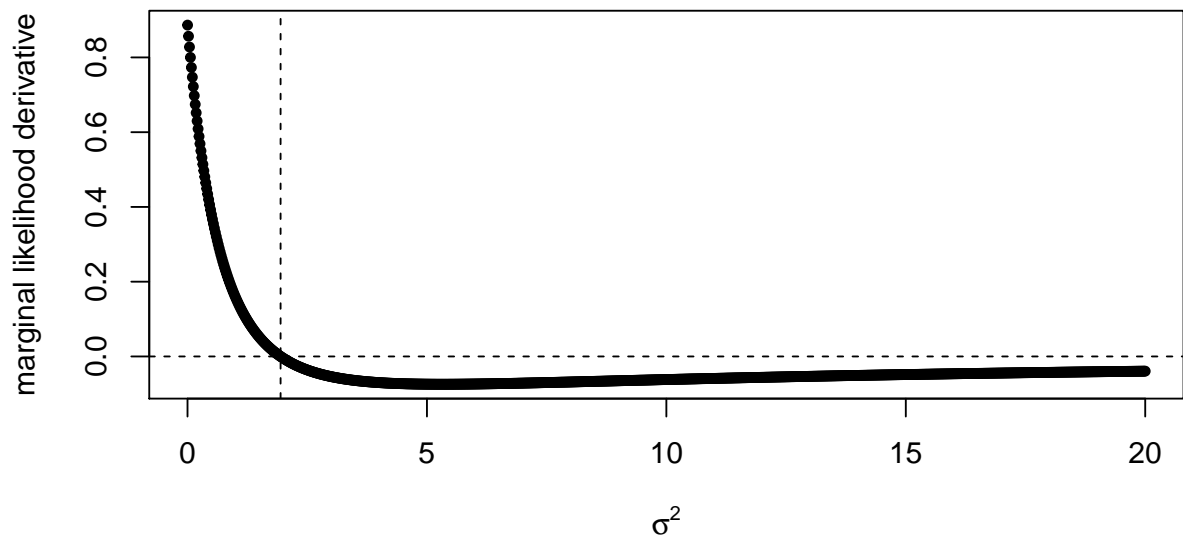
##           [,1]
## [1,] 2.012317
## [2,] 1.866819

# 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

## [1] 1.943684

# plot the optimization problem and best sigma
possible_signals <- seq(0.001,20,l=1000)
plot(possible_signals, sapply(possible_signals, root_fun),
  xlab = expression(sigma^2), ylab = "marginal likelihood derivative",
  pch = 20)
abline(h=0, lty=2)
abline(v=bestSigmaSq$root, lty=2)
```



```
# 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,
                           y = y, x = x, xstar = 0)

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

Histogram of samples_posterior

