Portfolio 7

Rachel Wood

2023-04-20

Task 1

1. A function k is a kernel if it is semi-definite (by Mercer's Theorem), i.e.

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

   for all points $x_1, \dots, x_n \in \mathcal{X}$ and $c_1, \dots, c_n \in \mathbb{R}$.

For $k(x, x') = g(x)g(x')$ for $g : \mathcal{X} \to \mathbb{R}$ we can see:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j g(x_i) g(x_j) = \left(\sum_{i=1}^{n} c_i g(x_i)\right)\left(\sum_{j=1}^{n} c_j g(x_j)\right) = \left(\sum_{i=1}^{n} c_i g(x_i)\right)^2 \geq 0$$

so k is a kernel.

2. If we set $g(x) = \sqrt{a}$ for all $x \in \mathcal{X}$ in the above example, then $k(x, x') = g(x)g(x') = a$ is a kernel.
3. For kernels $\{k_l\}_{l=1}^m$ and constants $\{b_l\}_{l=1}^m$, $k = \sum_{l=1}^m b_l k_l$ satisfies:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \sum_{l=1}^{m} b_l k_l(x_i, x_j) = \sum_{l=1}^{m} b_l \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k_l(x_i, x_j) \geq 0$$

   as $\sum_{j=1}^n c_i c_j k_l(x_i, x_j) \geq 0$ due to k_l being a kernel and all b_l are non-negative.
4. Since k is a kernel on $\mathbb{R}^p$, we must have

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

   For any $c_1, \dots, c_n \in \mathbb{R}$, we can then set $c_i' = \mathbf{1}_{\mathcal{X}}(x_i) c_i \in \mathbb{R}$, then the above becomes

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{1}_{\mathcal{X}}(x_i)\mathbf{1}_{\mathcal{X}}(x_j) c_i c_j k(x_i, x_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j)\mathbf{1}_{\mathcal{X}\times\mathcal{X}}(x_i, x_j) \geq 0$$

so $k(x, x')\mathbf{1}_{\mathcal{X}\times\mathcal{X}}(x, x')$ is a kernel. # Task 2

For this task we use the bone mineral density dataset obtained from the Elements of Statistical learning

```
data <- read.csv("spnbmd.csv", sep = "\t")
head(data)
```

```
##   idnum   age gender      spnbmd
## 1     1 11.70   male 0.018080670
## 2     1 12.70   male 0.060109290
## 3     1 13.75   male 0.005857545
## 4     2 13.25   male 0.010263930
## 5     2 14.30   male 0.210526300
## 6     2 15.30   male 0.040843210
```

For this task, we will be implementing a Gaussian kernel ## Empirical Bayes

We compute the posterior using an empirical Bayes approach, where we choose $(\lambda, \psi) = (\lambda_n, \psi_n)$ given by maximisers of the marginal likelihood:

$$(\lambda_n, \psi_n) \in \arg\max_{\lambda, \psi} \left(-\frac{1}{2} \log |\mathbf{K}_n + \lambda \mathbf{I}_n| - \frac{1}{2} y^0_{1:n} (\mathbf{K}_n + \lambda \mathbf{I}_n)^{-1} y^0_{1:n} \right)$$

We first create a function to compute the value of the objective function:

```
marginal_likelihood <- function(par, x, y){
  n <- length(x)
  lambda <- par[1]
  psi <- par[2]

  K <- kernelMatrix(rbfdot(psi), x)
  Klam <- K + lambda * diag(n)
  alpha <- solve(Klam) %*% y
  out <- -0.5*(log( det(Klam) ) + t(y) %*% alpha)
  if(is.finite(out)){
    return(out)
  } else {
    return(-100000 + sum(log(par)))
  }
}
```

We can now extract the predictor and response, as well as define the negative marginal likelihood function (since `optim` minimises)

```
library(Matrix)
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
y <- as.vector(data$spnbmd)
x <- as.vector(data$age)


negml <- function(par, x, y)-marginal_likelihood(par, x, y)
```

We now use the `optim` function to find the empirical bayes estimator of (λ, ψ)

```
guess <- c(0.01, 0.5)
opt <- optim(guess, negml, x = x, y = y)
lambda <- opt$par[1]
psi <- opt$par[2]
```

Computing Posterior and Credible Interval

The posterior is given by $f|y_{1:n} \sim GP(f_n, k_n)$ where

$$f_n(x) = k_n(x)^T \left(\mathbf{K}_n + \lambda \mathbf{I}_n\right)^{-1} y_{1:n} k_n(x, x') = k(x, x') - k(x)^T \left(\mathbf{K}_n + \lambda \mathbf{I}_n\right)^{-1} k(x')$$

We code these functions below:

```
f_n <- function(x_new, x, y, lambda, psi){
  n <- length(x)

  K <- kernelMatrix(rbfdot(psi), x)

  kn <- kernelMatrix(rbfdot(psi), x, x_new)
  f <- t(kn) %*% solve(K + lambda* diag(n)) %*% y
  return(f)
}

k_n <- function(x0, x1, x, y,lambda,psi){
  n <- length(x)
  K <- kernelMatrix(rbfdot(psi), x)
  k <- kernelMatrix(rbfdot(psi), x0,x1)

  k_x0 <- kernelMatrix(rbfdot(psi), x ,x0)
  k_x1 <- kernelMatrix(rbfdot(psi), x ,x1)

  out <- k - t(k_x0) %*% solve(K + lambda * diag(n)) %*% k_x1
  return(out)
}
```

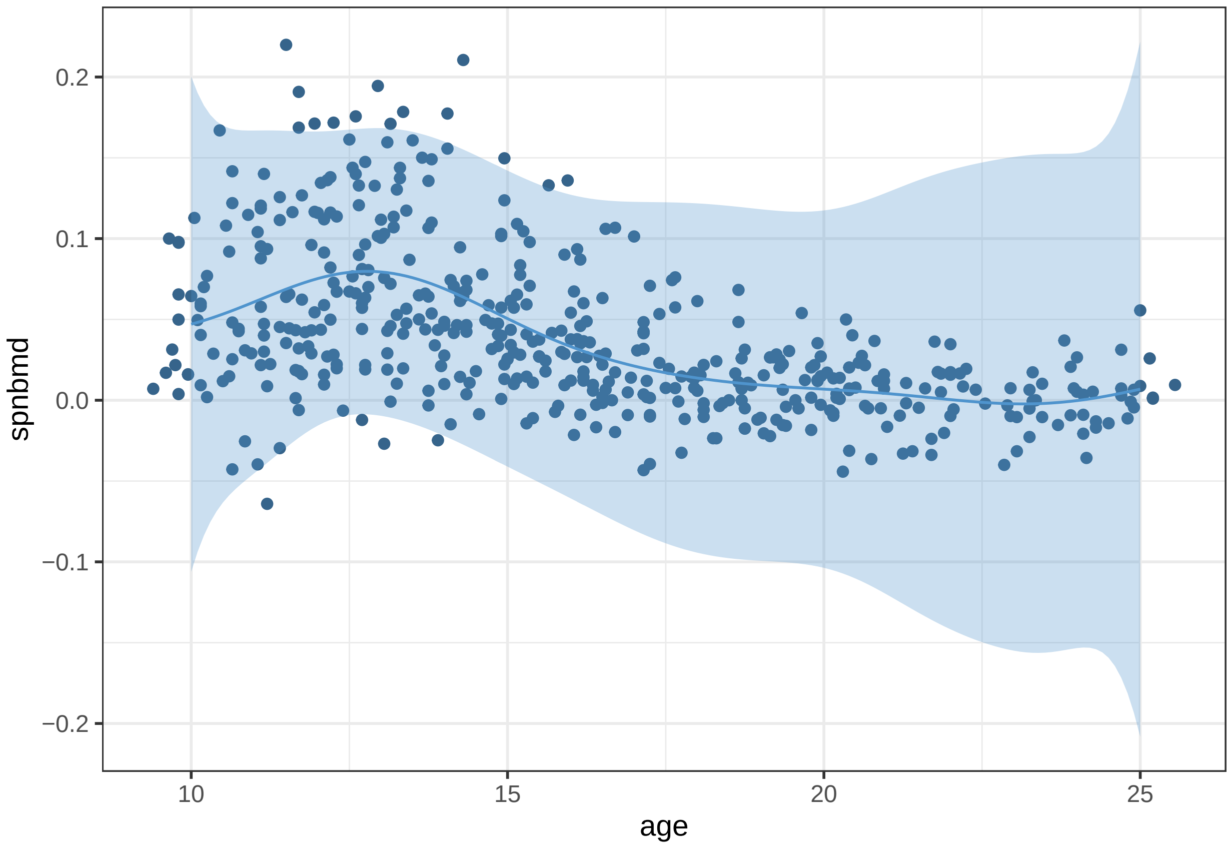
Further the credible interval for the posterior is given by:

$$C_\alpha(x) = \left[f_n(x) - z_{1-\alpha/2}\sqrt{k_n(x,x)},\ f_n(x) + z_{1-\alpha/2}\sqrt{k_n(x,x)}\ \right]$$

and so we can now compute and plot the mean function and the credible interval for a new `x` vector with the empirical bayes (λ, ψ):

```r
credible_int <- function(x_new, x, y, lambda, psi, alpha = 0.05){
  mean <- f_n(x_new,x, y, lambda, psi)

  k_n_xx <- diag(k_n(x_new, x_new, x, y, lambda, psi))

  ci_lower <- mean - qnorm(alpha/2,lower.tail = FALSE) * sqrt(k_n_xx)
  ci_upper <- mean + qnorm(alpha/2,lower.tail = FALSE) * sqrt(k_n_xx)

  return(cbind(ci_lower, ci_upper))
}
```

```r
library(dplyr)
library(ggplot2)

x_seq <- seq(10, 25, 0.1)

mean_f <- as.vector(f_n(x_seq,x, y, lambda, psi ))

ci <- credible_int(x_seq, x, y, lambda, psi)

plot_dat <- tibble(x = x_seq, mean = mean_f, lower = ci[,1], upper = ci[,2])

data <- as_tibble(data)

ggplot(data = data, aes(x = age, y = spnbmd)) +
  geom_point(colour = "steelblue4") +
  geom_line(data = plot_dat, mapping = aes(x = x, y = mean), colour = "steelblue3") +
  geom_ribbon(data = plot_dat, inherit.aes = FALSE, mapping =  aes(x = x, ymin = lower, ymax = upper), a
```


spnbmd
age
0.2
0.1
0.0
−0.1
−0.2
10
15
20
25