

Statistical Modeling and Inference – Problem Set #2

NITI MISHRA · MIQUEL TORRENS · BÁLINT VÁN

October 18th, 2015

Solution to proposed exercises.

Exercise 1

We need to solve:

$$\begin{aligned}\max_{\mathbf{w}} -2 \log p(\mathbf{w}|\mathbf{t}, \mathbf{X}) &= -2q\mathbf{t}^T \Phi \mathbf{w} + q\mathbf{w}^T \Phi^T \Phi \mathbf{w} + (\mathbf{w} - \mu)^T \mathbf{D}(\mathbf{w} - \mu) + C \\ &= -2q\mathbf{t}^T \Phi \mathbf{w} + q\mathbf{w}^T \Phi^T \Phi \mathbf{w} + \mathbf{w}^T \mathbf{D} \mathbf{w} - 2\mathbf{w}^T \mathbf{D} \mu + \mu^T \mathbf{D} \mu + C\end{aligned}$$

The term C includes all constant terms not dependant on \mathbf{w} . Now we maximize with respect to \mathbf{w} and set to zero:

$$-2q\mathbf{t}^T \Phi + q\mathbf{w}^T \left(\Phi^T \Phi + (\Phi^T \Phi)^T \right) + \mathbf{w}^T (\mathbf{D} + \mathbf{D}^T) - 2(\mathbf{D} \mu)^T = 0$$

During the derivation we will recurrently use two properties: $\mathbf{D} = \mathbf{D}^T$, as it is symmetric by construction, and $(\Phi^T \Phi)^T = \Phi^T \Phi$, which is a straightforward calculation. We just need to rearrange terms to reach the normal equations:

$$\begin{aligned}2\mathbf{w}^T \mathbf{D} - 2\mu^T \mathbf{D}^T - 2q\mathbf{t}^T \Phi + 2q\mathbf{w}^T \Phi^T \Phi &= 0 \\ \mathbf{w}^T (\mathbf{D} + q\Phi^T \Phi) &= q\mathbf{t}^T \Phi + (\mathbf{D} \mu)^T \\ (\mathbf{D} + q\Phi^T \Phi)^T \mathbf{w} &= q\Phi^T \mathbf{t} + (\mathbf{D} \mu)\end{aligned}$$

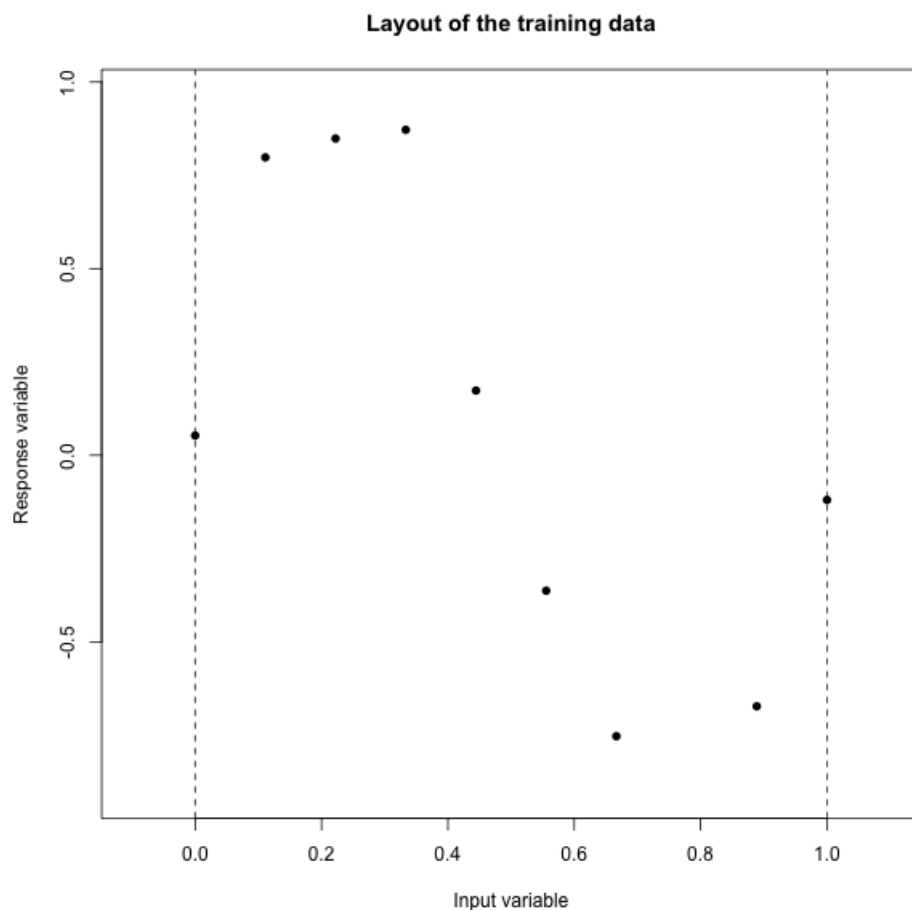
To finally obtain the normal equations:

$$(\mathbf{D} + q\Phi^T \Phi) \mathbf{w} = q\Phi^T \mathbf{t} + \mathbf{D} \mu$$

Hence proved.

Exercise 2

Part 1. Plotting the data:



Part 2. The `phix` function:

```
#####  
# Function "phix"  
phix <- function(x, M, basis) {  
#####  
  # Check correctness of input x  
  if (x < 0 || x > 1) {  
    stop('out-of-range values in the input vector "x".')  
  }  
  
  # Perform the calculations  
  if (basis == 'poly') {  
    out <- rep(NA, length = M + 1)  
    sapply(c(0, 1:M), function(i) {  
      out[i + 1] <-- x^i  
    })  
  } else if (basis == 'Gauss') {  
    #mus <- seq(0, 1, M ** (-1))  
    mus <- seq(0, 1, length.out = M)  
  }  
}
```

```

    out <- rep(NA, length = M)
    sapply(1:M, function(i) {
      out[i] <- exp(-(x - mus[i]) ** 2) / 0.1)
    })
    out <- c(1, out)
  } else {
    stop('specify a valid option for the parameter "basis".')
  }

  # Return the values
  return(out)
}

```

Part 3. The `post.params` function:

```

#####
# Function "post.params"
post.params <- function(tdata, M, basis, phix, delta, q) {
#####
  # Input data
  t <- tdata[, 't'] # Response variable
  x <- tdata[, 'x'] # Input variable

  # Initialize Phi matrix
  phi <- matrix(nrow = length(x), ncol = M + 1)
  sapply(1:length(x), function(i) {
    phi[i, ] <- phix(x = x[i], M = M, basis = basis)
  })

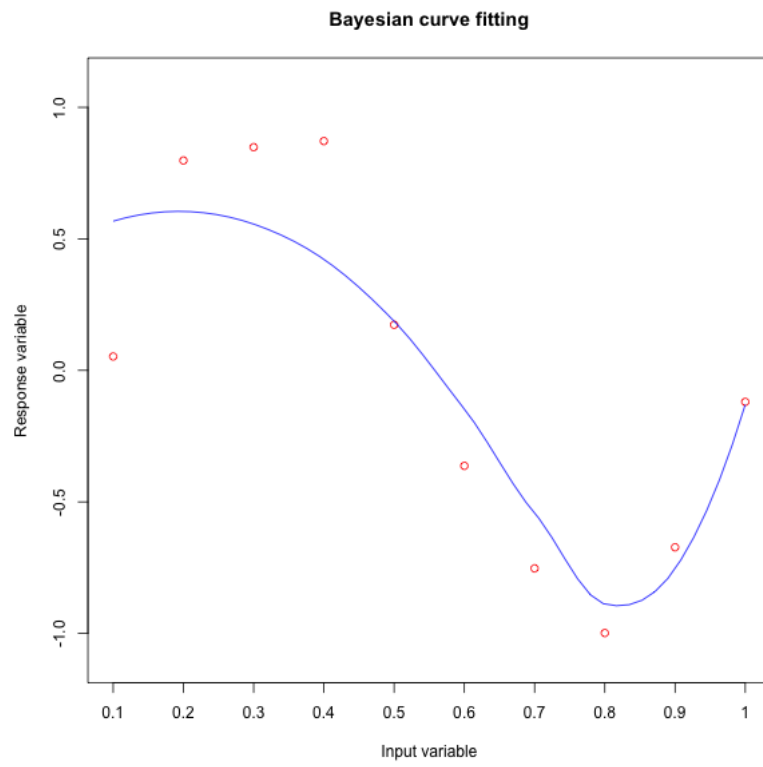
  # Function parameter
  lambda <- delta / q

  Q <- delta * diag(ncol(phi)) + q * t(phi) %*% phi
  w.bayes <- q * solve(Q) %*% t(phi) %*% t

  # Results
  return(list(w.bayes = w.bayes, Q = Q))
}

```

Part 4. Plotting the estimated linear predictor:



Exercise 3

Part 1. The function that returns the mean and the precision of the predictive distribution at each of the inputs:

```
#####
bayesian.precision <- function(x) {
#####
  # Parameters
  M <- 9
  delta <- 1L
  q <- 0.1 ** (-2)

  # Initialize Phi matrix (THINK post.params)
  phi <- matrix(nrow = length(x), ncol = M + 1)
  sapply(1:length(x), function(i) {
    phi[i, ] <-- phix(x = x[i], M = 9, basis = 'Gauss')
  })

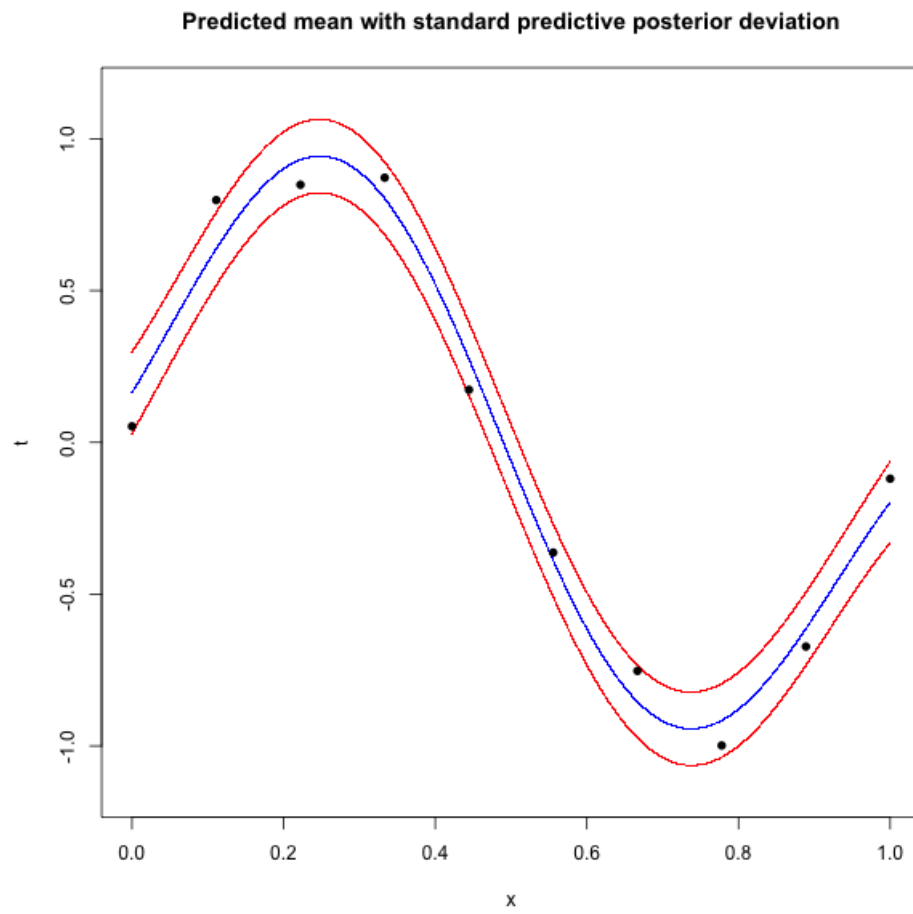
  # Execute the function with the specified parameters
  params <- post.params(cd, M = 9, basis = 'Gauss', phix,
                        delta = 1L, q = 0.1 ** (-2))

  # Resulting parameters
  Q <- params[[2]]
  w.bayes <- params[[1]]

  # Predicted values
  means <- phi %*% w.bayes
  vars <- phi %*% solve(Q) %*% t(phi) + q ** (-1)

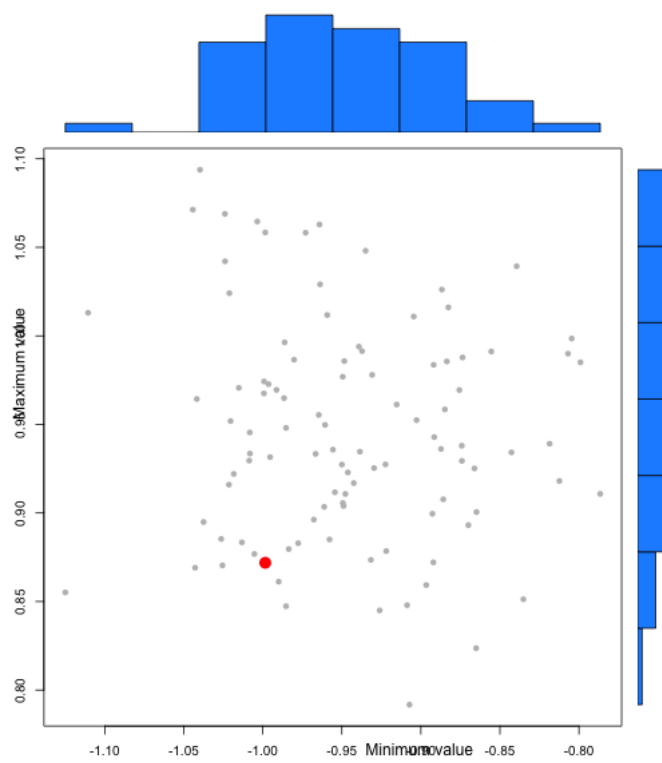
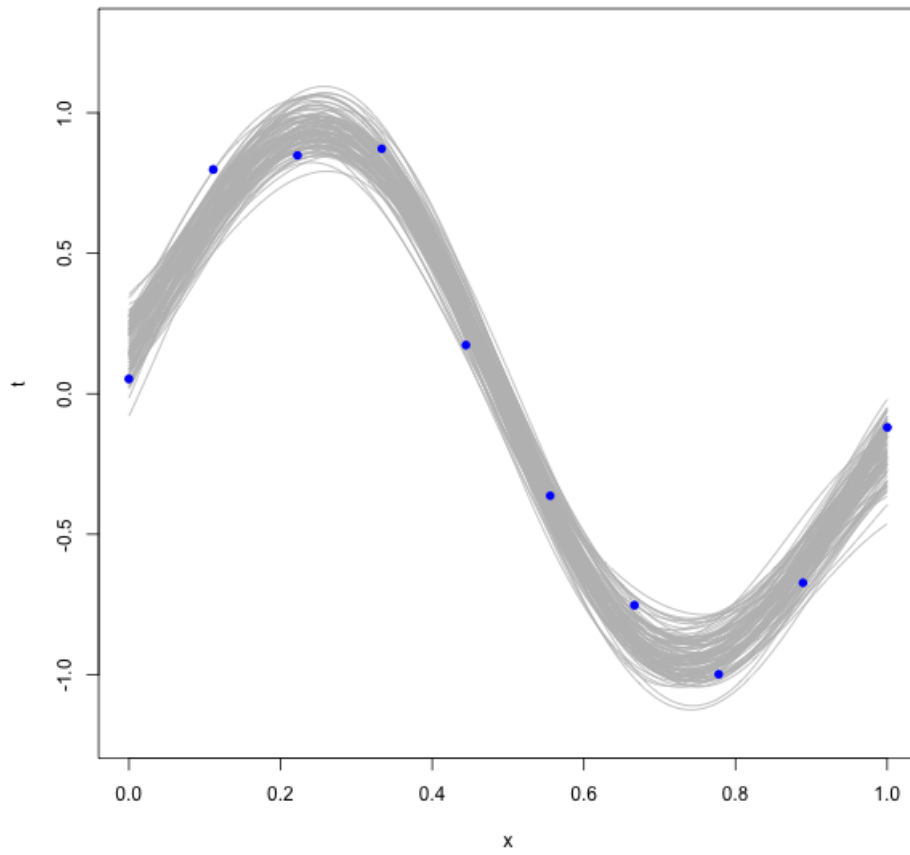
  # Return
  return(list(means = means, vars = diag(vars)))
}
```

Part 2. Plotting the predicted mean with its standard predictive posterior deviation:



Part 3. Replicating the plot of the slides:

Simulation of Bayesian functions given inputs



Exercise 4