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:

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

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

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

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

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

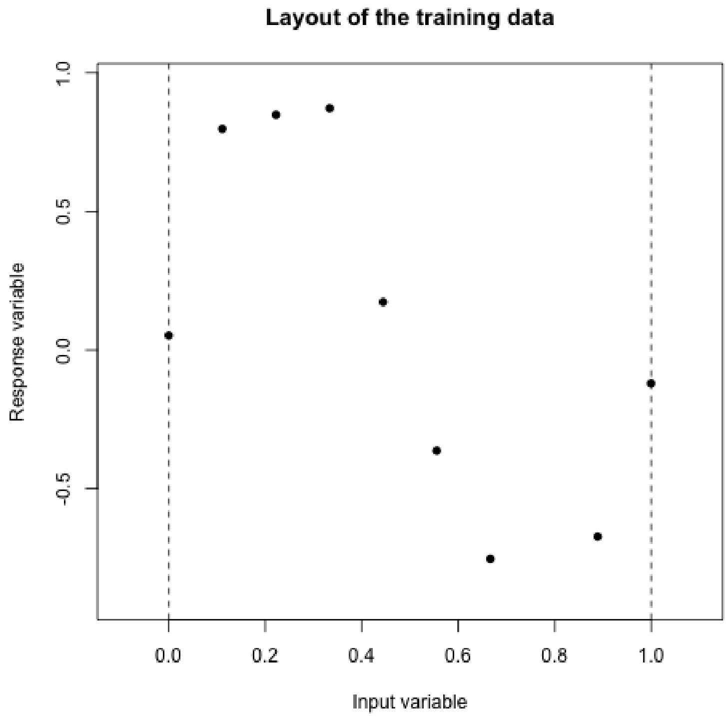
To finally obtain the normal equations:

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

Hence proved.

Exercise 2

Part 1. Plotting the data:



Part 2. The `phix` function:

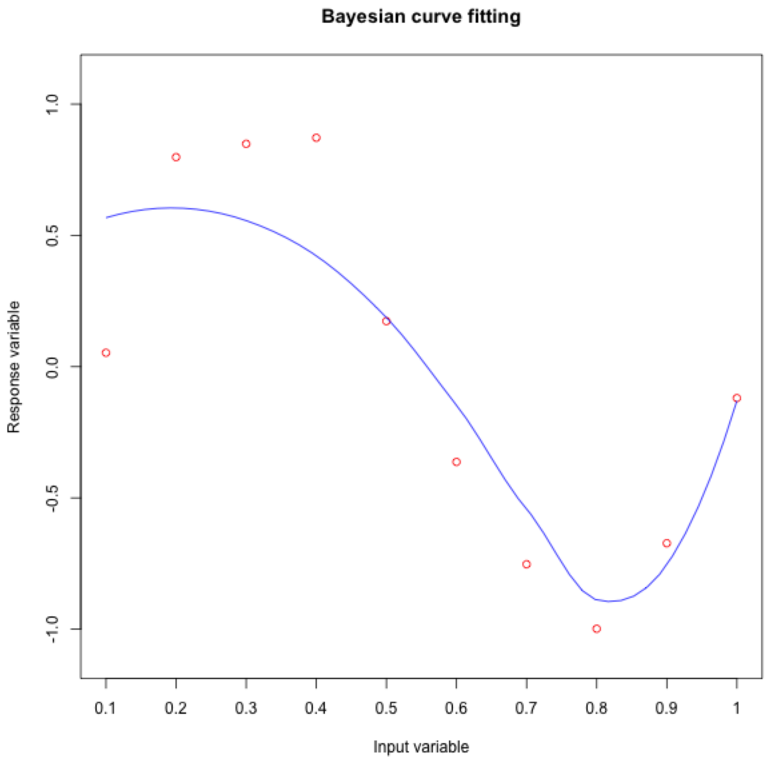
```
#################################################################################
# Function "phix"
phix <- function(x, M, basis) {
#################################################################################
  # Check correctness of input x
  if (x < 0 || x > 1) {
    stop('out-of-range values 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))
    out <- rep(NA, length = M + 1)
    sapply(1:(M + 1), function(i) {
      out[i] <<- exp((-(x - mus[i]) ** 2) / 0.1)
    })
  } else {
    stop('specify a valid option for the parameter "basis".')
  }
```

```r
  # Return the values
  return(out)
}
```

Part 3. The `post.params` function:

```r
##############################################################################
# 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

  # D matrix
  D <- lambda * diag(ncol(phi))

  # Response parameters
  w.bayes <- solve(D + t(phi) %*% phi) %*% t(phi) %*% t
  Q <- D + q * t(phi) %*% phi

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

Part 4. Plotting the estimated linear predictor:


Bayesian curve fitting
Response variable
Input variable
1.0
0.5
0.0
-0.5
-1.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Exercise 3

Exercise 4