Statistical Modeling and Inference – Problem Set #2

Niti Mishra · Miquel Torrens · Bálint Ván

October 18^{th}, 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 term C includes all constant terms not dependant on $\mathbf{w}$. Now we maximize with respect to $\mathbf{w}$ and set to zero:

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

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:

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

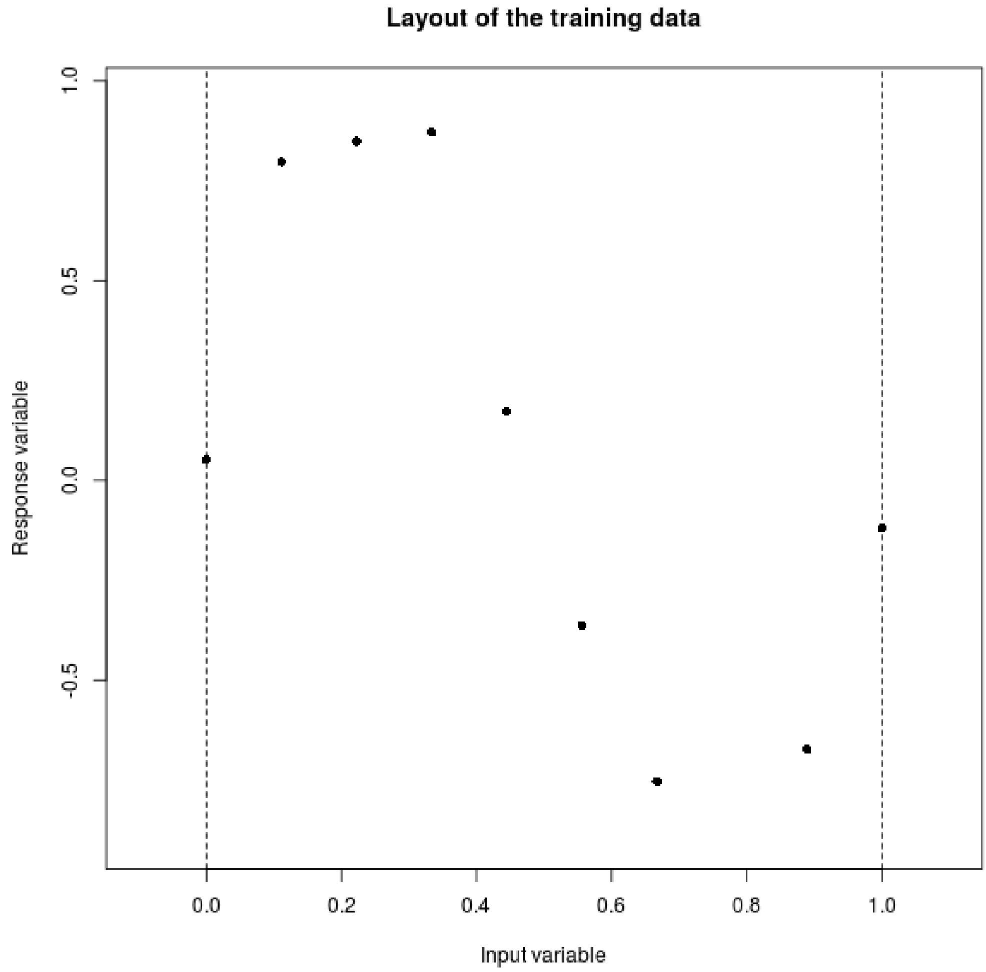
To finally obtain the normal equations:

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

Hence proved.

Exercise 2

Part 1. Plotting the data:



Part 2. The `phix` function:

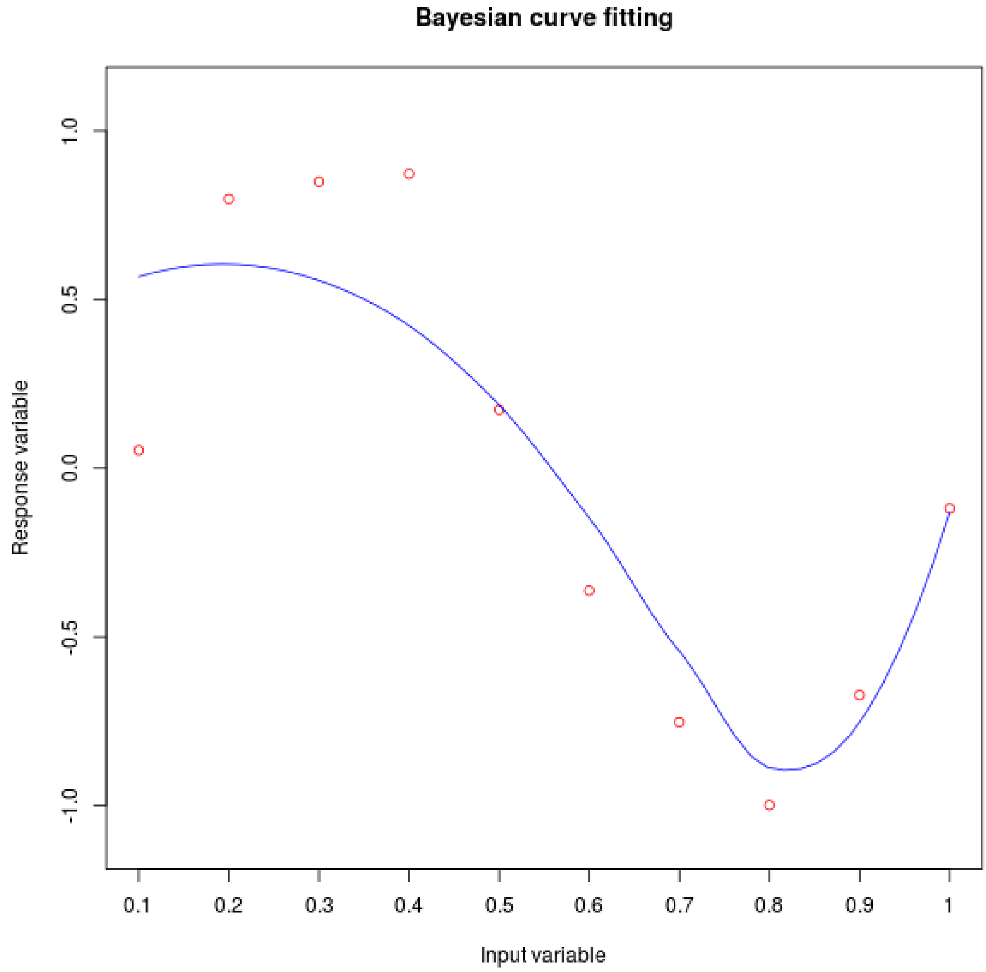
```
> ####################################################################################
> # Function "phix"
> phix <- function(x, M, basis) {
+ ####################################################################################
+   # Check correctness of input x
+   if (x < 0 || x > 1) {
+     stop('out-of-range 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, 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)
+   })
+
+   # Parameter estimation
+   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:

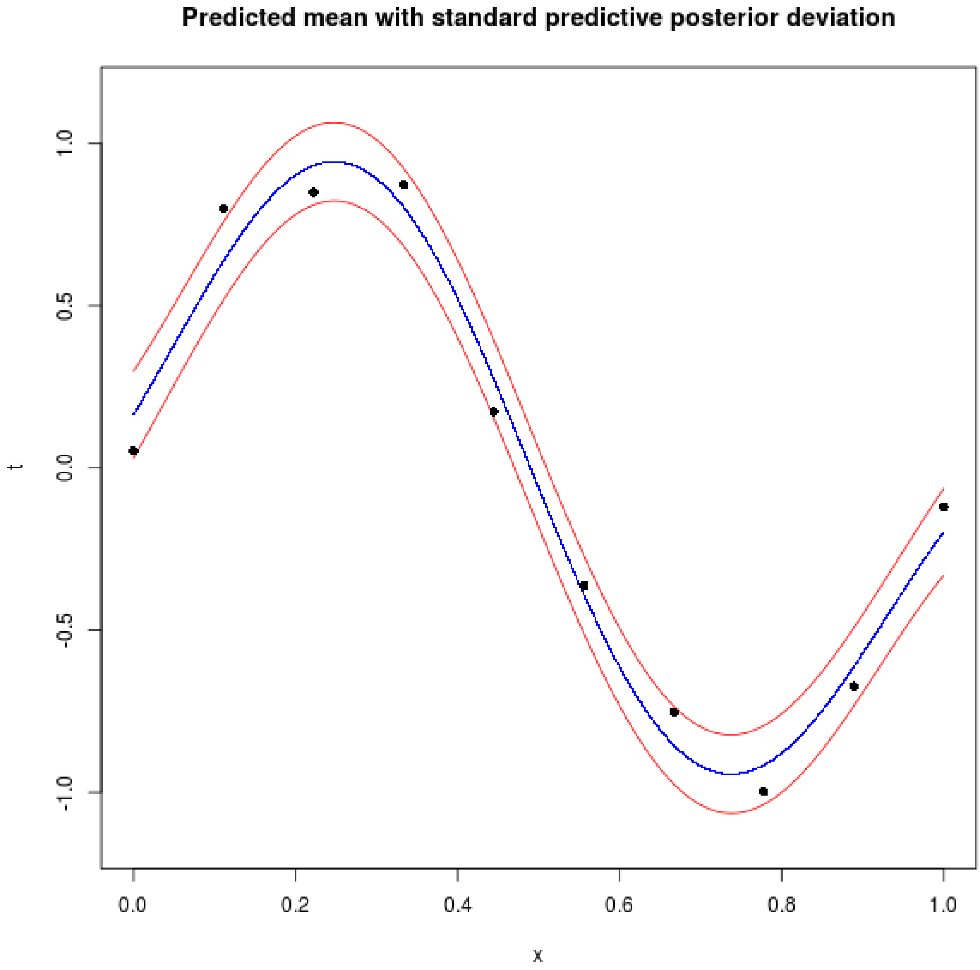

Bayesian curve fitting
Response variable
Input variable
1.0
0.5
0.0
-0.5
-1.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Exercise 3

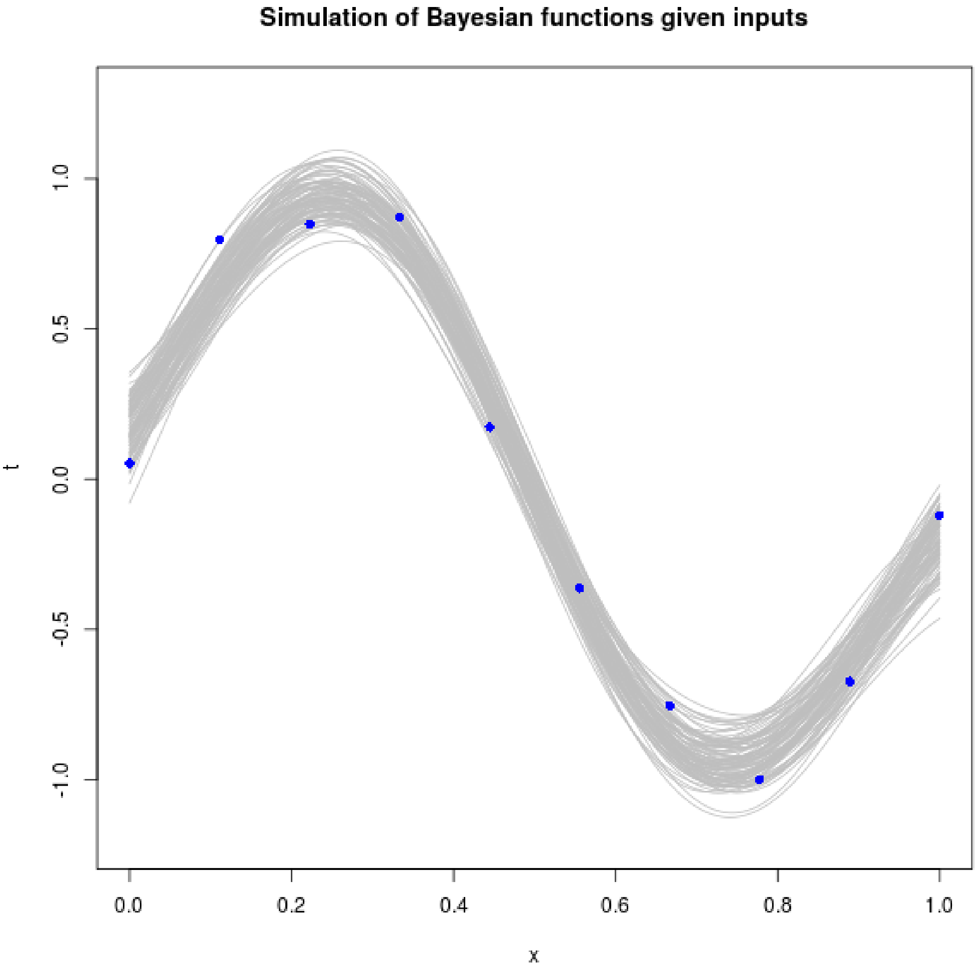
Part 1. The function that returns the mean and the precision of the predictive distribution at each of the inputs:

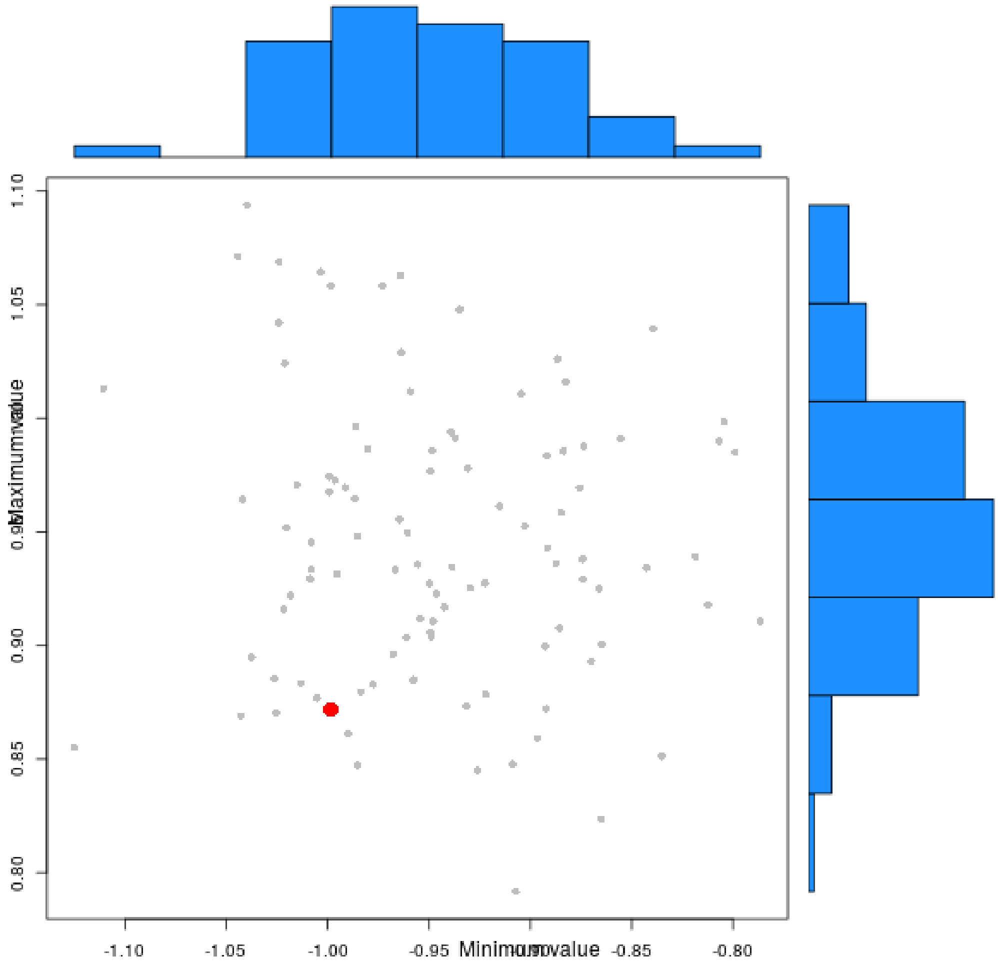
```
> #################################################################################
> bayesian.precision <- function(x) {
+ #################################################################################
+   # Parameters
+   M <- 9
+   delta <- 1L
+   q <- 0.1 ** (-2)
+
+   # Initialize Phi matrix
+   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
1.0
0.5
0.0
-0.5
-1.0
t
0.0
0.2
0.4
0.6
0.8
1.0
x


Maximum value
Minimum value
1.10
1.05
0.95
0.90
0.85
0.80
-1.10
-1.05
-1.00
-0.95
-0.90
-0.85
-0.80

Exercise 4

Part 1.

We can prove this using the following rule:

$$\phi(\mathbf{x})^T \mathbf{w}_B = \phi(\mathbf{x})^T q \mathbf{Q}^{-1} \phi(\mathbf{x}) \mathbf{t} = \phi(\mathbf{x})^T q \mathbf{Q}^{-1} \sum_{n=1}^{N} \phi(\mathbf{x}_n) \mathbf{t}_n = \sum_{n=1}^{N} q \phi(\mathbf{x})^T \mathbf{Q}^{-1} \phi(\mathbf{x}_n) t_n$$

Where $\mathbf{w}_B$ are the Bayesian parameters. We derive the fact that $\phi(\mathbf{x})\mathbf{t} = \sum_{n=1}^{N} \phi(\mathbf{x}_n)\mathbf{t}_n$ by noticing that this product is the inner product between them.

Part 2.

We define:

$$k(\mathbf{x}, \mathbf{y}) = q \phi(\mathbf{x})^T \mathbf{Q}^{-1} \phi(\mathbf{y})$$

Then:

$$\phi(\mathbf{x})^T \mathbf{w}_B = \sum_{n=1}^{N} q \phi(\mathbf{x})^T \mathbf{Q}^{-1} \phi(\mathbf{x}_n) t_n = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n$$

By this definition $k(\mathbf{x}, \mathbf{x}_n)$ becomes the weight of t_n when computing the mean of the predictive distribution $\phi(\mathbf{x})^T \mathbf{w}_B$ at the input location $\mathbf{x}$.

Part 3.

We use the following derivation:

$$\begin{aligned}
\mathbf{K} &= \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_K) \\ & \ddots & \\ \vdots & k(\mathbf{x}_n, \mathbf{x}_k) & \vdots \\ & \ddots & \\ k(\mathbf{x}_N, \mathbf{x}_1) & \cdots & k(\mathbf{x}_N, \mathbf{x}_K) \end{pmatrix} \\
&= \begin{pmatrix} q\phi(\mathbf{x}_1)^T \mathbf{Q}^{-1} \phi(\mathbf{x}_1) & \cdots & q\phi(\mathbf{x}_1)^T \mathbf{Q}^{-1} \phi(\mathbf{x}_K) \\ & \ddots & \\ \vdots & q\phi(\mathbf{x}_n)^T \mathbf{Q}^{-1} \phi(\mathbf{x}_k) & \vdots \\ & \ddots & \\ q\phi(\mathbf{x}_N)^T \mathbf{Q}^{-1} \phi(\mathbf{x}_1) & \cdots & q\phi(\mathbf{x}_N)^T \mathbf{Q}^{-1} \phi(\mathbf{x}_K) \end{pmatrix} \\
&= q \begin{pmatrix} \phi(\mathbf{x}_1)^T \\ \vdots \\ \phi(\mathbf{x}_N)^T \end{pmatrix} \mathbf{Q}^{-1} \begin{pmatrix} \phi(\mathbf{x}_1) \\ \vdots \\ \phi(\mathbf{x}_K) \end{pmatrix} \\
&= q \mathbf{\Phi} \mathbf{Q}^{-1} \mathbf{\Phi}^T.
\end{aligned}$$

Hence proved.

Part 4.

Given $\lambda = 0$, the proof is the following:

$$\mathbf{K} = q\mathbf{\Phi}(\delta\mathbf{I} + q\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T = \mathbf{\Phi}(\lambda\mathbf{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T = \mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T = \mathbf{H}.$$