

# Lecture 3 Exercises

**1. For a Bayesian linear regression model with prior on  $p(w) = N(w|\mu, D^{-1})$  and  $q$  known, show that  $w_{Bayes}$  solves  $(D + q\phi^T\phi)w_{Bayes} = q\phi^T t + D\mu$**  To solve for bayes we take the known posterior probability:

$$-2\log p(w|t, X) = -2qt^T\phi w + qw^T\phi^T\phi w + (w - \mu)^T D(w - \mu) + const$$

To get the best estimate of  $w$ , we take the derivative of the posterior probability and set it equal to zero:

$$\partial - 2\log p(w|t, X)/\partial w = 0$$

To take the derivative of the first term:

$$\partial - 2qt^T\phi w/\partial w = -2qt^T\phi$$

The derivative of the second term uses the property  $\partial x^T A x/\partial x = 2Ax$

$$\partial qw^T\phi^T\phi w/\partial w = 2q\phi^T\phi w$$

Expanding the third term  $(w - \mu)^T D(w - \mu)$  gives:

$$= (w^T D - \mu^T D)(w - \mu)$$

$$= w^T D w - \mu^T D w - w^T D \mu + \mu^T D \mu$$

$$\partial(w^T D w - \mu^T D w - w^T D \mu + \mu^T D \mu)/\partial w = 2w^T D - 2\mu^T D$$

So we have:

$$0 = -2qt^T\phi + 2qw^T\phi^T\phi + 2w^T D - 2\mu^T D$$

Move negative terms to the LHS and divide both sides by 2:

$$qt^T\phi + \mu^T D = qw^T\phi^T\phi + w^T D$$

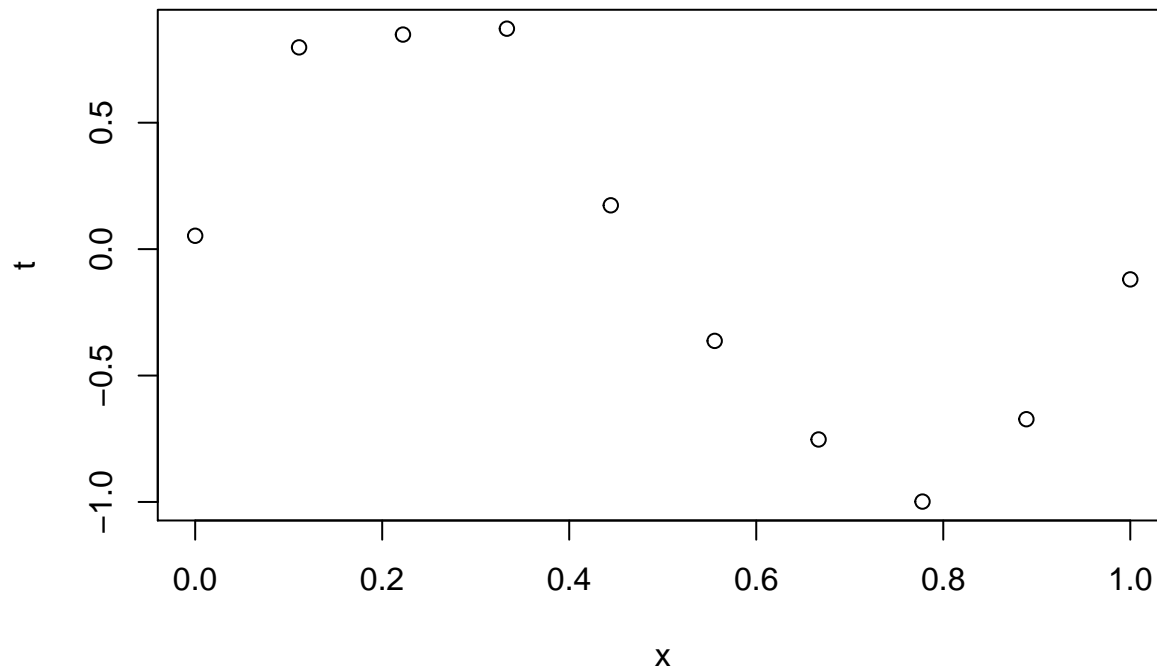
$$qw^T\phi^T\phi + w^T D = qt^T\phi + \mu^T D$$

Take the transpose of both sides (note:  $D^T = D$ ):

$$q\phi^T\phi w + Dw = q\phi^T t + D\mu$$

$$(D + q\phi^T\phi)w_{Bayes} = q\phi^T t + D\mu$$

**2. Curve fitting (pt1)** The aim is to learn a smooth function from the cloud of points stored in `curve_data.txt` using Bayesian linear regression models. In all that follows the prior  $p(w) = N(w|0, \delta^{-1}I)$  is used and  $q, \delta$  are constants specified by the user. **2.1 Plot the data**



**2.2** Write a function in R, called `phix` that takes as input a scalar  $x$  (the input in curve fitting), with values in  $[0, 1]$ ,  $M$  the number of bases functions, and a categorical variable that specifies the type of basis used, and returns the vector of basis functions evaluated at  $x$ . Hence a call of the function `phix(0.3,4,"poly")` should return `c(1.0000, 0.3000, 0.0900, 0.0270, 0.0081)`. Code it up so that the option "poly" gives the polynomial bases and "Gauss" the Gaussian kernels with means  $\mu_i$  equally spaced in  $[0, 1]$ , with  $\mu = 0$  and  $\mu_M = 1$ .

```
phix <- function(x, M, option) {
  phi <- rep(0,M)
  if (option == "poly") {
    for (i in 1:(M)) {
      phi[i] <- x**i
    }
  }
  if (option == "Gauss") {
    for (i in 0:(M-1)) {
      phi[i] <- exp(-((x-i/(M))**2)/0.1)
    }
  }
  phi
}
```

**2.3** Write a function in R, called `post.params`, that takes as input the training data,  $M$ , the type of basis, the function `phix`,  $\delta$  and  $q$  and returns the parameters of the posterior distribution,  $w_{Bayes}$  and  $Q$ .

```
post.params <- function(data, M, option, delta, q) {
  phi = phix(data$x[1],M, option)
  for (i in 2:length(data$x)) {
    phi_ <- phix(data$x[i], M, option)
    phi = rbind(phi, phi_)
  }
}
```

```

}
phi = cbind(rep(1,10),phi)
Q = diag(delta,M+1) + q*t(phi)%*%phi
w = solve(Q)%*(q*t(phi)%*%data$t)
t <- phi%*%w

return(list(Q, w, t))
}

```

**2.4** Plot the estimated linear predictor, by plugging in  $w_{Bayes}$ , and superimpose the training data; use  $q = (1/0.1)^2$ ,  $\delta = 2.0$  and  $M = 9$

