

Bayesian Learning

Lab 2

Emil K Svensson and Rasmus Holm

2017-04-28

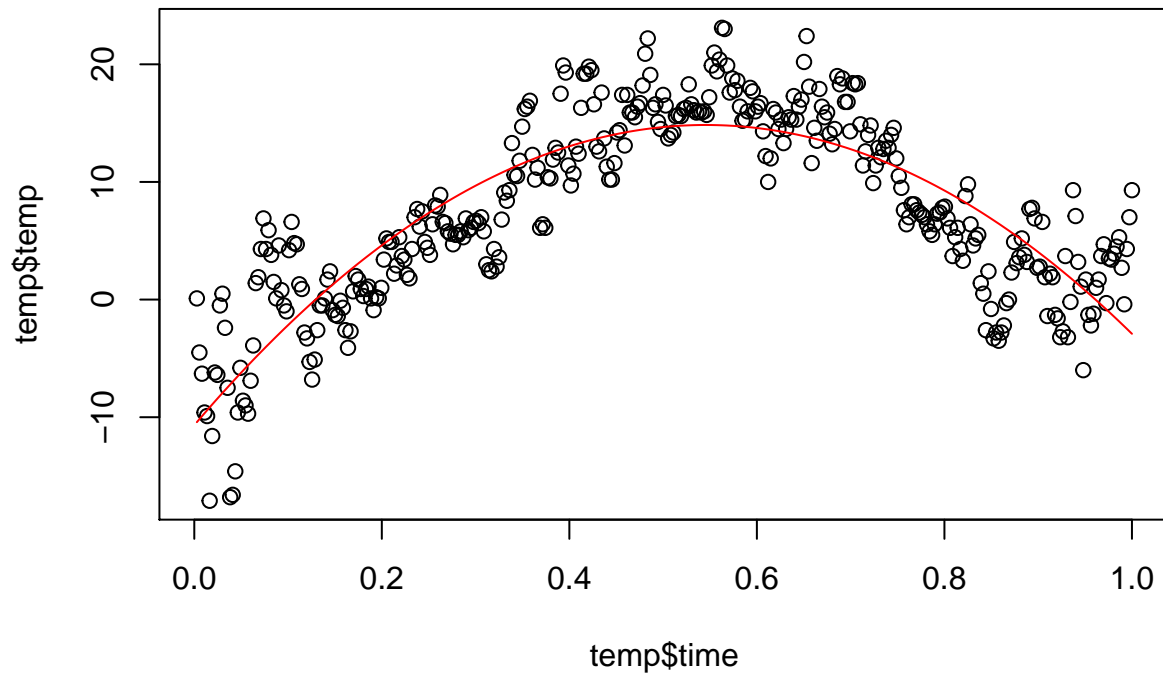
Question 1

```
temp <- read.table("../data/TempLinkoping2016.txt", header=T)

mod <- lm(temp ~ time + I(time^2), data=temp)

idx <- order(temp$time)
x <- temp$time[idx]
y <- fitted(mod)[idx]

plot(temp$time, temp$temp)
lines(x, y, col='red', type='l')
```



Prior

$$\sigma^2 \sim \text{Inv} - \chi^2(\nu_0, \sigma_0^2)$$

$$\beta | \sigma^2 \sim N(\mu_0, \sigma^2 \Omega_0^{-1})$$

Likelihood

$$\mathbf{y} | \beta, \sigma^2, \mathbf{X} \sim N(\mathbf{X}\beta, \sigma^2 I_n)$$

Posterior

$$\sigma^2 | \mathbf{y} \sim \text{Inv} - \chi^2(\nu_n, \sigma_n^2)$$

$$\beta | \sigma^2, \mathbf{y} \sim N(\mu_n, \sigma^2 \Omega_n^{-1})$$

where

$$\mu_n = (\mathbf{X}^\top \mathbf{X} + \Omega_0)^{-1} (\mathbf{X}^\top \mathbf{X} \hat{\beta} + \Omega_0 \mu_0)$$

$$\Omega_n = \mathbf{X}^\top \mathbf{X} + \Omega_0$$

$$\nu_n = \nu_0 + n$$

$$\nu_n \sigma_n^2 = \nu_0 \sigma_0^2 + (\mathbf{y}^\top \mathbf{y} + \mu_0^\top \Omega_0 \mu_0 - \mu_n^\top \Omega_n \mu_n)$$

a)

```
mu0 <- c(0, 0, 0)
omega0 <- diag(3) * 0.05
nu0 <- 1
sigmasq0 <- 20

hyperparams <- list(mu=mu0, omega=omega0, nu=nu0, sigmasq=sigmasq0)
```

b)

```
library(geoR)
library(MASS)

time <- data.frame(rep(1, nrow(temp)), temp$time, temp$time^2)
mtime <- as.matrix(time)
mtemp <- matrix(temp$temp, ncol = 1)

prior_estimate <- function(data, params) {
  sigmasq <- rinvchisq( n = 1, df = params$nu, scale = params$sigmasq)
  betacoef <- mvrnorm(n = 1, mu = params$mu, Sigma = sigmasq * solve(params$omega) )

  data %*% betacoef
}
```

```

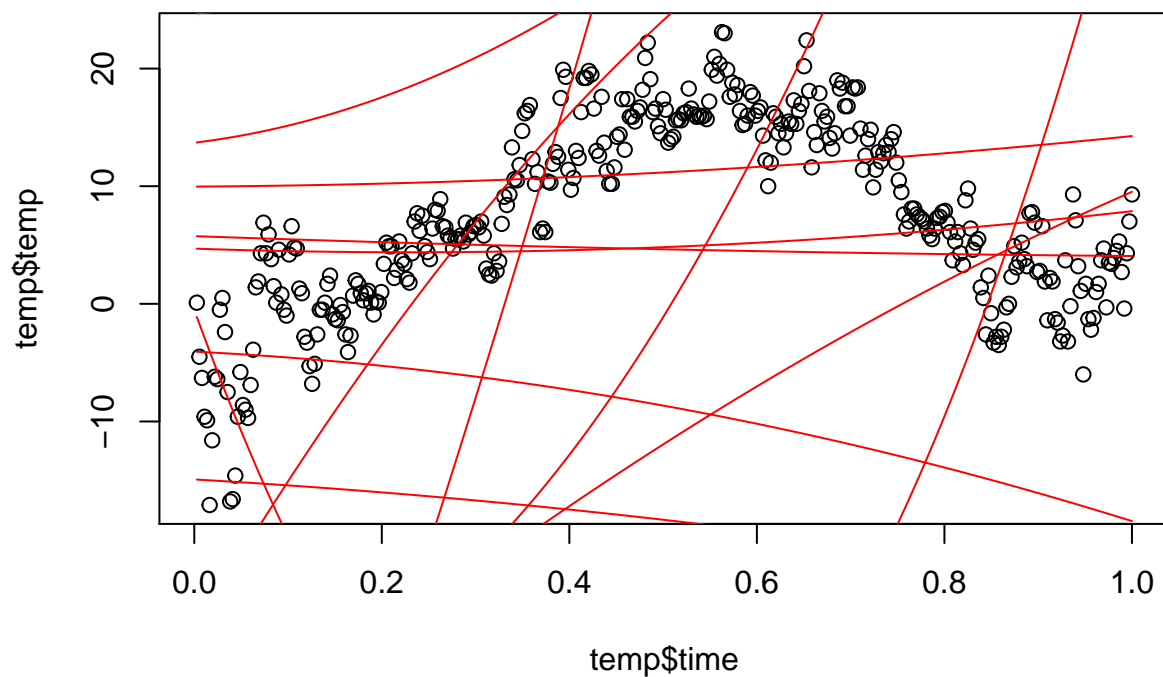
plot(temp$time, temp$temp)

x <- sort(temp$time)

set.seed(12345)

for (i in 1:20){
  y <- prior_estimate(mtime, hyperparams)[order(temp$time)]
  lines(x, y, col='red', type='l')
}

```



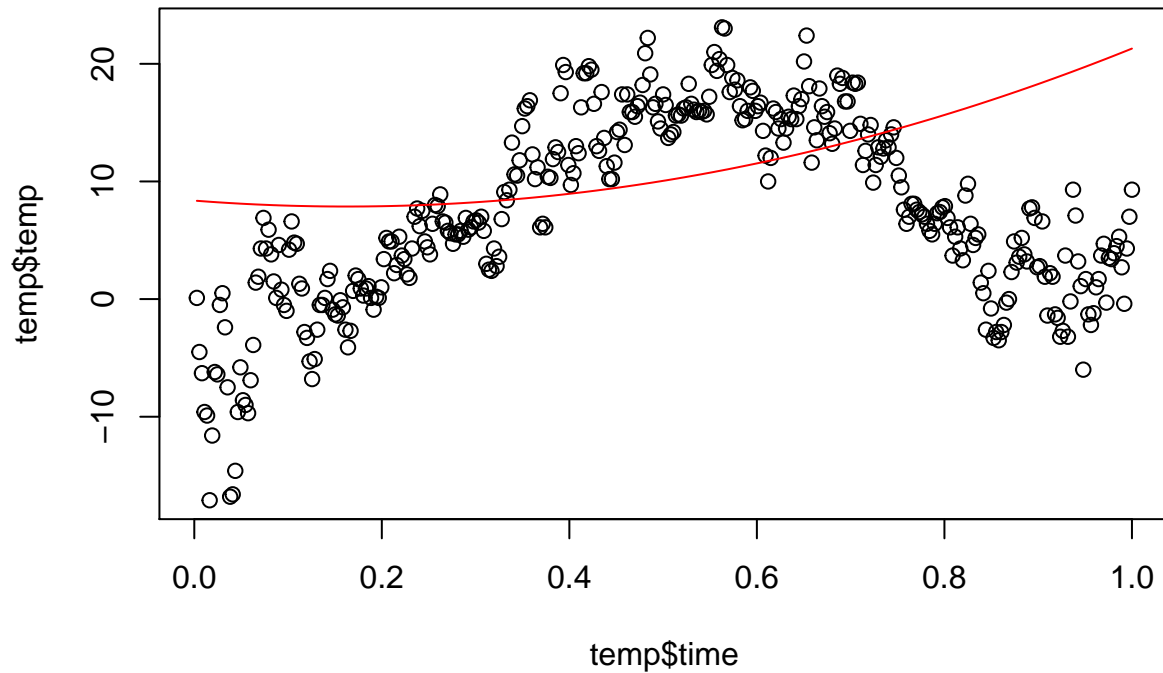
```

set.seed(12345)

x <- sort(temp$time)
y <- rowMeans(sapply(1:1000, function(x) prior_estimate(mtime, hyperparams)[order(temp$time)])))

plot(temp$time, temp$temp)
lines(x, y, col='red', type='l')

```



c)

```
posterior_param_sample <- function(X, y, hyperparams){
  XX <- t(X) %*% X

  betahat <- solve(XX) %*% t(X) %*% y

  mun <- solve(XX + hyperparams$omega) %*%
    (XX %*% betahat + hyperparams$omega %*% hyperparams$mu)

  omegan <- XX + hyperparams$omega

  nun <- hyperparams$nu + nrow(X)

  nunsigmasqn <- hyperparams$nu * hyperparams$sigmasq +
    (t(y) %*% y +
     t(hyperparams$mu) %*% hyperparams$omega %*% hyperparams$mu -
     t(mun) %*% omegan %*% mun )

  sigmasqn <- nunsigmasqn / nun

  sigmasq <- rinvcchisq(n = 1, df=nun, scale=sigmasqn)
  beta <- mvrnorm(n = 1, mu = mun, Sigma = as.numeric(sigmasq) * solve(omegan))
}
```

```

    list(beta = beta, sigmasq = sigmasq)
  }

posterior_estimate <- function(X, y, hyperparams){
  sample <- posterior_param_sample(X, y, hyperparams)
  X %*% sample$beta
}

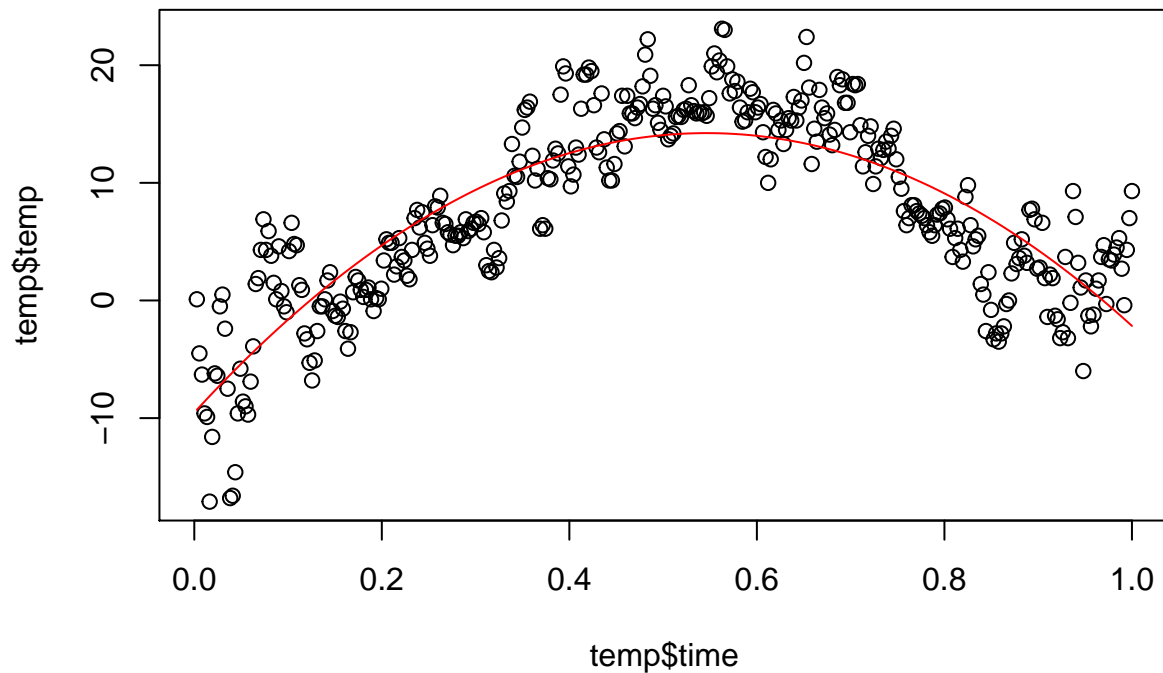
plot(temp$time, temp$temp)

set.seed(12345)

idx <- order(temp$time)
x <- temp$time[idx]
y <- posterior_estimate(mtime, mtemp, hyperparams)[idx]

lines(x, y, col='red', type='l')

```



```

set.seed(12345)

ests <- sapply(1:1000, FUN = function(x) posterior_estimate(mtime, mtemp, hyperparams))
cred_interval <- apply(ests, MARGIN = 1, quantile, probs = c(0.05, 0.95))

idx <- order(temp$time)
x <- temp$time[idx]
y1 <- rowMeans(ests)[idx]

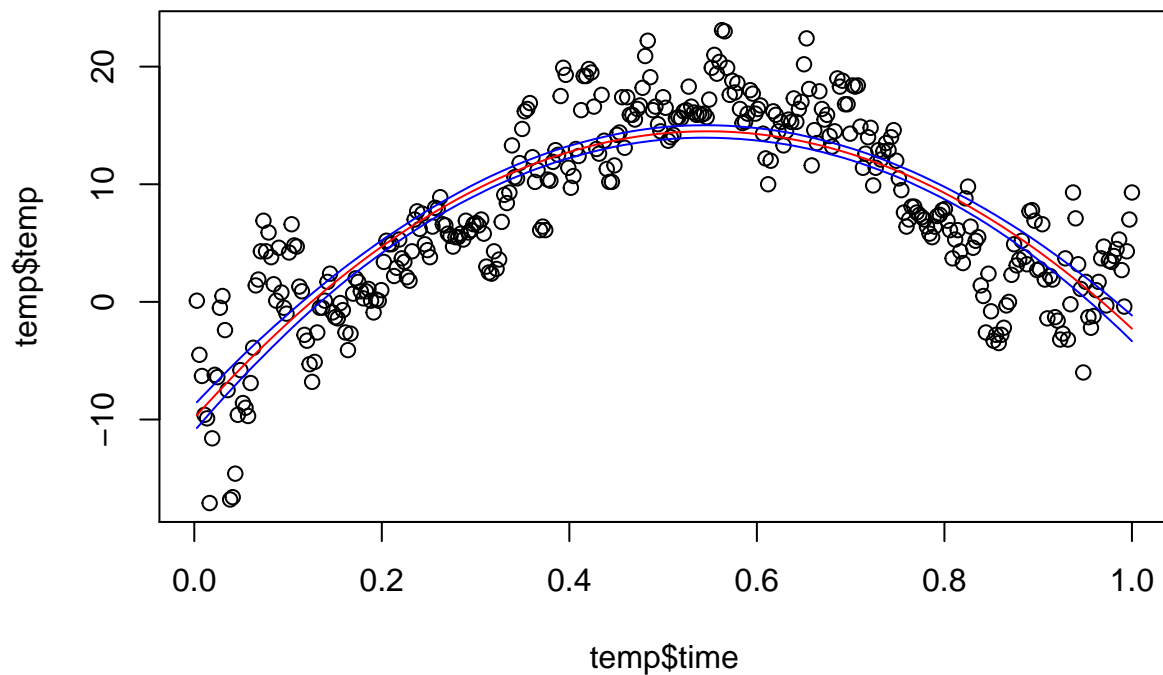
```

```

y2 <- cred_interval[1,][idx]
y3 <- cred_interval[2,][idx]

plot(temp$time, temp$temp)
lines(x, y1, col='red', type='l')
lines(x, y2, col='blue', type='l')
lines(x, y3, col='blue', type='l')

```



d)

```

set.seed(12345)

betas <- sapply(1:1000, FUN = function(x) posterior_param_sample(mtime, mtemp, hyperparams)$beta)
hot <- mean(-betas[2,] / (2 * betas[3,]))
hot * 366 # July 27, 2016 (Wed)

## [1] 200.0608

```

e)

Set μ_0 to zeros and a high Ω_0 that expresses a high degree of certainty in our prior.

Question 2

```
women <- read.table("../data/WomenWorks.txt", header = TRUE)
```

a)

```
glmModel <- glm(Work ~ 0 + ., data = women, family = binomial)
summary(glmModel)

##
## Call:
## glm(formula = Work ~ 0 + ., family = binomial, data = women)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1662  -0.9299   0.4391   0.9494   2.0582
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## Constant         0.64430     1.52307   0.423 0.672274
## HusbandInc     -0.01977     0.01590  -1.243 0.213752
## EducYears       0.17988     0.07914   2.273 0.023024 *
## ExpYears        0.16751     0.06600   2.538 0.011144 *
## ExpYears2     -0.14436     0.23585  -0.612 0.540489
## Age            -0.08234     0.02699  -3.050 0.002285 **
## NSmallChild   -1.36250     0.38996  -3.494 0.000476 ***
## NBigChild     -0.02543     0.14172  -0.179 0.857592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 277.26  on 200  degrees of freedom
## Residual deviance: 222.73  on 192  degrees of freedom
## AIC: 238.73
##
## Number of Fisher Scoring iterations: 4
```

b)

```
library(mvtnorm)

logprior <- function(beta, mean, sigma){
  dmvnorm(beta, mean = mean, sigma = sigma, log = TRUE)
}

loglikelihood <- function(beta, X, Y){
  linear_prediction <- t(X) %*% beta

  probabilities <- (Y * linear_prediction) - log(1 + exp(linear_prediction))
}
```

```

loglike <- sum(probabilities)

## if (abs(loglike) == Inf)
##   loglike = -20000

loglike
}

logposterior <- function(beta, X, Y, mean, sigma){
  loglikelihood(beta, X, Y) + logprior(beta, mean, sigma)
}

tau <- 10
mu <- rep(0,8)
sigma <- tau^2 * diag(8)

womenX <- as.matrix(women[,2:ncol(women)])
womenY <- as.matrix(women[,1])

optpost <- optim(par = matrix(rep(0, 8), ncol = 1),
  fn = logposterior, method = "BFGS", hessian = TRUE,
  X = t(womenX), Y = womenY,
  mean = mu, sigma = sigma,
  control=list(fnscale=-1))

posterior_beta_sample <- function(n, mu, sigma){
  rmvnorm(n, mean = mu, sigma = sigma)
}

mu <- optpost$par
sigma <- -solve(optpost$hessian)

set.seed(12345)
beta_samples <- posterior_beta_sample(n=1000, mu=mu, sigma=sigma)
cred_intervals <- apply(beta_samples, 2, quantile, prob=c(0.025, 0.975))

cred_intervals

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## 2.5% -2.313434 -0.05091541 0.03095675 0.04723847 -0.6143409 -0.13476231
## 97.5% 3.533276 0.00780642 0.32739377 0.29656573 0.2943841 -0.03095853
##           [,7]      [,8]
## 2.5% -2.1296887 -0.3023168
## 97.5% -0.5759195 0.2692414

```

c)

```

posterior_predictive_sample <- function(X, beta){
  linear_prediction <- t(X) %*% beta
  probability <- exp(linear_prediction) / (1 + exp(linear_prediction))
  rbinom(n=1, size=1, prob=probability)
}

```



```
x <- matrix(c(1, 10, 8, 10, (10 / 10)^2, 40, 1, 1), ncol=1)
prediction_samples <- apply(beta_samples, 1, function(beta) posterior_predictive_sample(x, beta))

counts <- table(prediction_samples)
barplot(counts / sum(counts), ylim=c(0, 1))
```

