Bayesian Learning

Lab 2

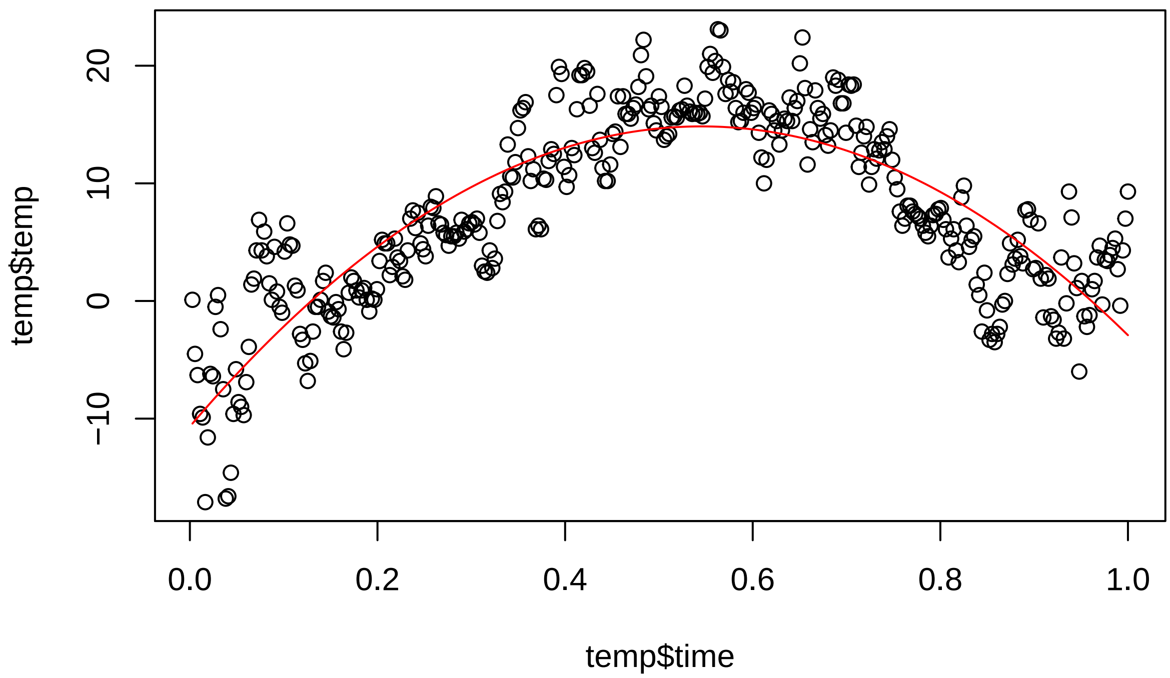
Emil K Svensson and Rasmus Holm

2017-05-02

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

$$
\begin{aligned}
\sigma^2 &\sim \text{Inv} - \chi^2(\nu_0, \sigma_0^2) \\
\beta|\sigma^2 &\sim \text{N}(\mu_0, \sigma^2\Omega_0^{-1})
\end{aligned}
$$

Likelihood

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

Posterior

$$
\begin{aligned}
\sigma^2|\mathbf{y} &\sim \mathrm{Inv} - \chi^2(\nu_n, \sigma_n^2) \\
\beta|\sigma^2, \mathbf{y} &\sim \mathrm{N}(\mu_n, \sigma^2\Omega_n^{-1})
\end{aligned}
$$

where

$$
\begin{aligned}
\mu_n &= (\mathbf{X}^\intercal\mathbf{X} + \Omega_0)^{-1}(\mathbf{X}^\intercal\mathbf{X}\hat{\beta} + \Omega_0\mu_0) \\
\Omega_n &= \mathbf{X}^\intercal\mathbf{X} + \Omega_0 \\
\nu_n &= \nu_0 + n \\
\nu_n\sigma_n^2 &= \nu_0\sigma_0^2 + (\mathbf{y}^\intercal\mathbf{y} + \mu_0^\intercal\Omega_0\mu_0 - \mu_n^\intercal\Omega_n\mu_n)
\end{aligned}
$$

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)
```

Since we both are novice in weather-prediction we have no real prior knowledge we set all mu0 to 0. Our omega0 is set to a diagonal matrix with 0.05 in the trace to express that we are not certain at all in these mus.

The same goes for our priors for sigma, with nu0 (τ_0) we express that we are very uncertain of our set prior for sigmasq.

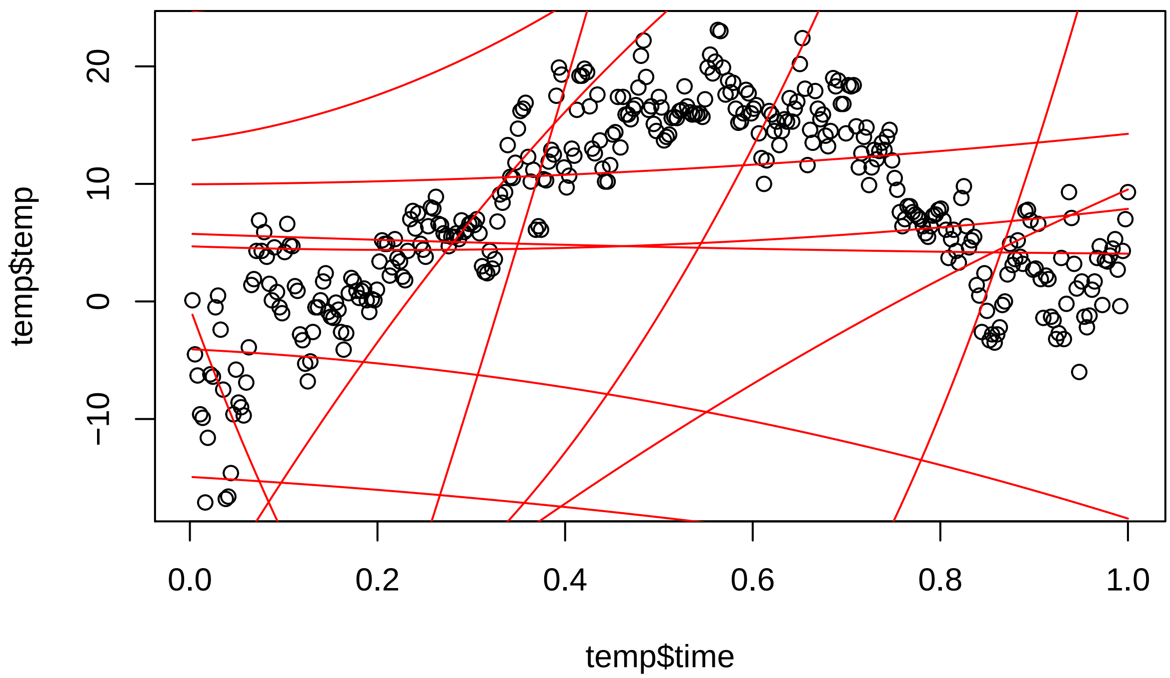
b)

```
library(geoR)
```

```
## ---------------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.2 (built on 2016-05-02) is now loaded
## ---------------------------------------------------------------
```

```
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')
}
```



Given our prior the curves are very flexible and go all over the plot which is about what we expected when setting such vauge priors.

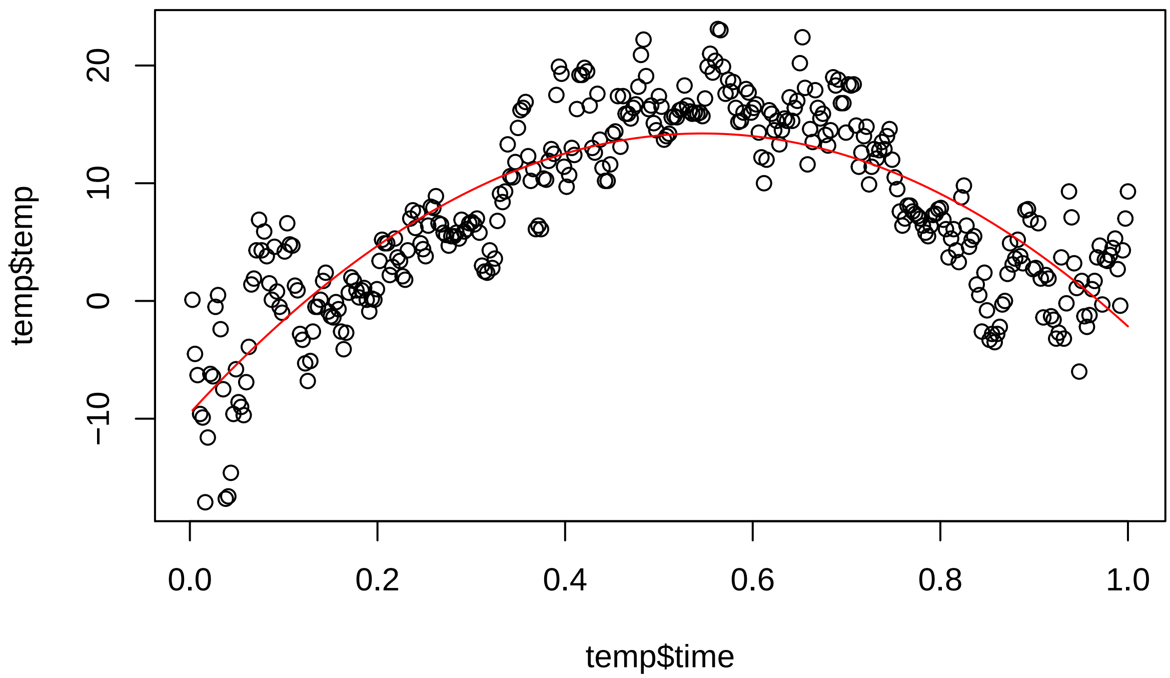
Since we say that we don't really know anything about the weather-forecast we are satitisfied with this.

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 +
```

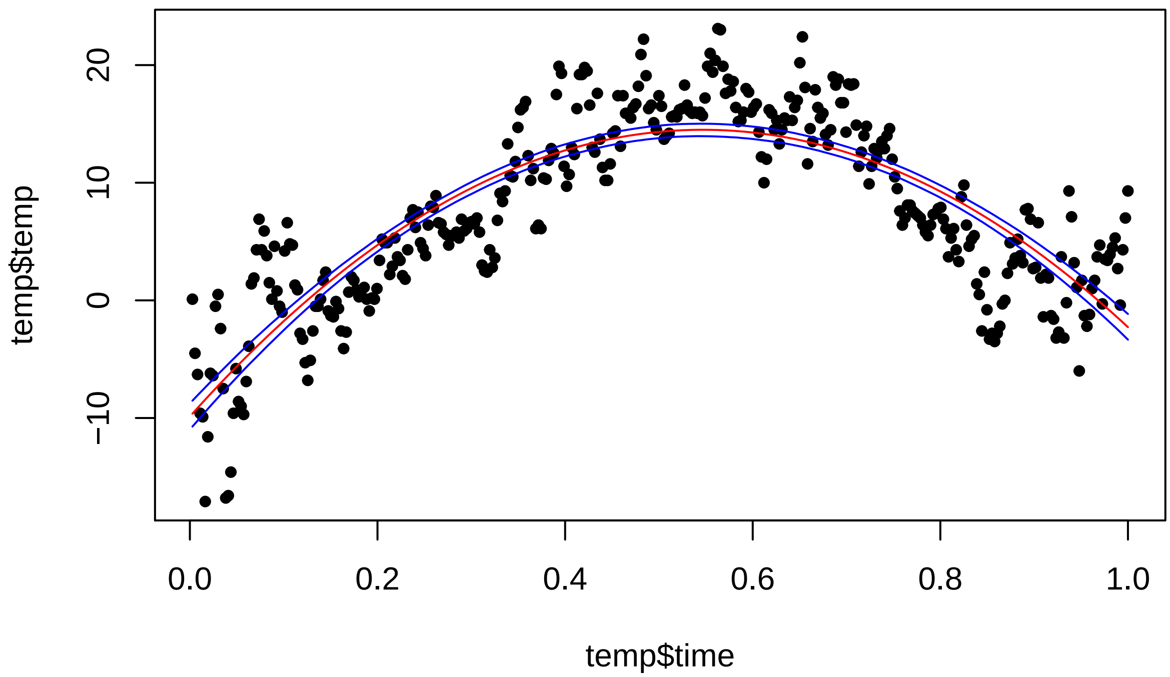
```r
        (t(y) %*% y +
         t(hyperparams$mu) %*% hyperparams$omega %*% hyperparams$mu -
         t(mun) %*% omegan %*% mun )

    sigmasqn <- nunsigmasqn / nun

    sigmasq  <- rinvchisq(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
}
```

```r
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')
```




```r
set.seed(12345)
library(fields)
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))
```

```r
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, pch = 20 )
lines(x, y1, col='red', type='l')
lines(x, y2, col='blue', type='l')
lines(x, y3, col='blue', type='l')
```



d)

```r
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 for all of them. For Ω_0 the first three elements in the diagonal are set low and the later ones set high. The prior now express that we are uncertain of what the first three mus are and that we are certain that the other five terms are the mus specified (i.e zero, not needed).

Question 2

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

a)

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

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
```

```r
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))
colnames(cred_intervals) <- colnames(women)[-1]
```

The 95% credibility intervals below show that the intercept, husband income, transformed years in work experience, and the number of children older than 6 years are statistical insignificant, i.e. 0 (no impact) is contained in the intervals. NsmallChilds credibility interval is not crossing 0 and therefor considerd significant and important for the model.

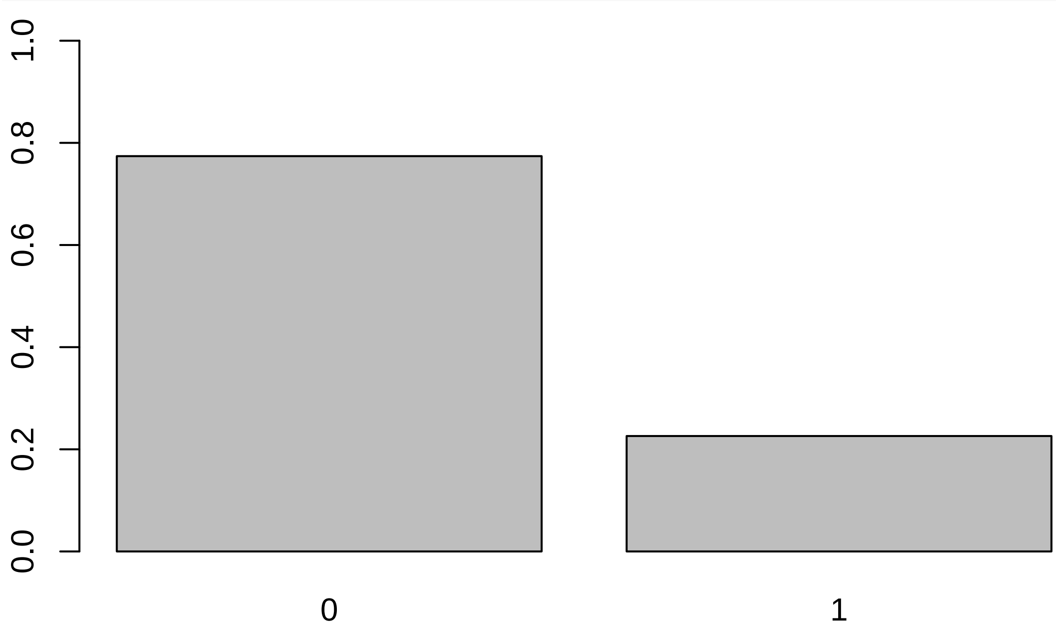
```r
cred_intervals
```

```
##        Constant  HusbandInc  EducYears   ExpYears  ExpYears2         Age
## 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
##       NSmallChild  NBigChild
## 2.5%   -2.1296887 -0.3023168
## 97.5%  -0.5759195  0.2692414
```

c)

```r
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))
```

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



The barplot above indicates that it is much more probable that given the data the woman is not working under our model.