Lab 2

Weng Hang Wong

4/27/2020

1. Linear and Polynomial regression

The dataset TempLinkoping.txt contains daily average temperatures (in Celcius degrees) at Malmslätt, Linköping over the course of the year 2018. The response variable is temp and the covariate is

$$time = \frac{the\ number\ of\ days\ since\ beginning\ of\ year}{365}$$

The task is to perform a Bayesian analysis of a quadratic regression

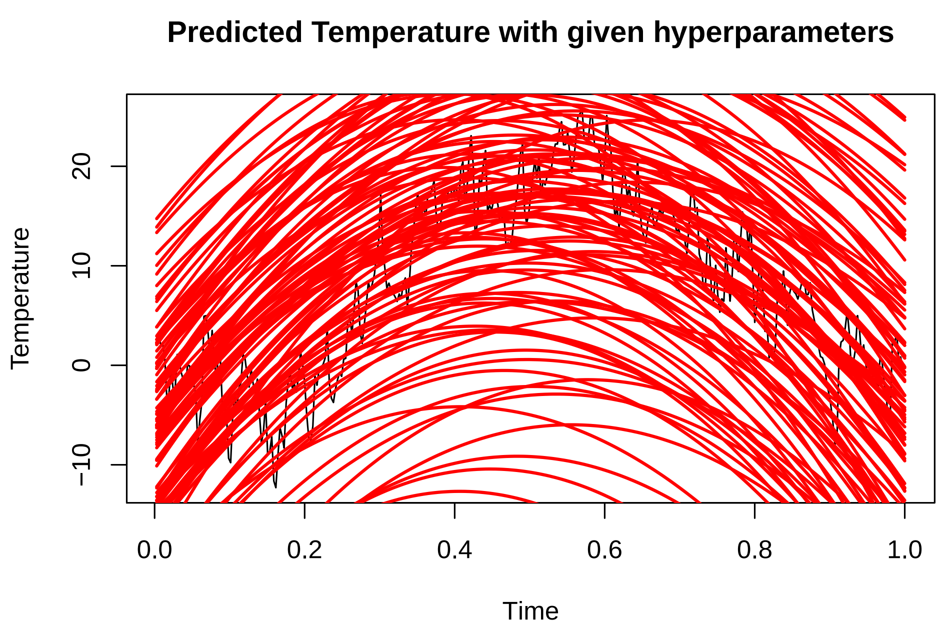
$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \varepsilon, \varepsilon \sim N(0, \sigma^2)$$

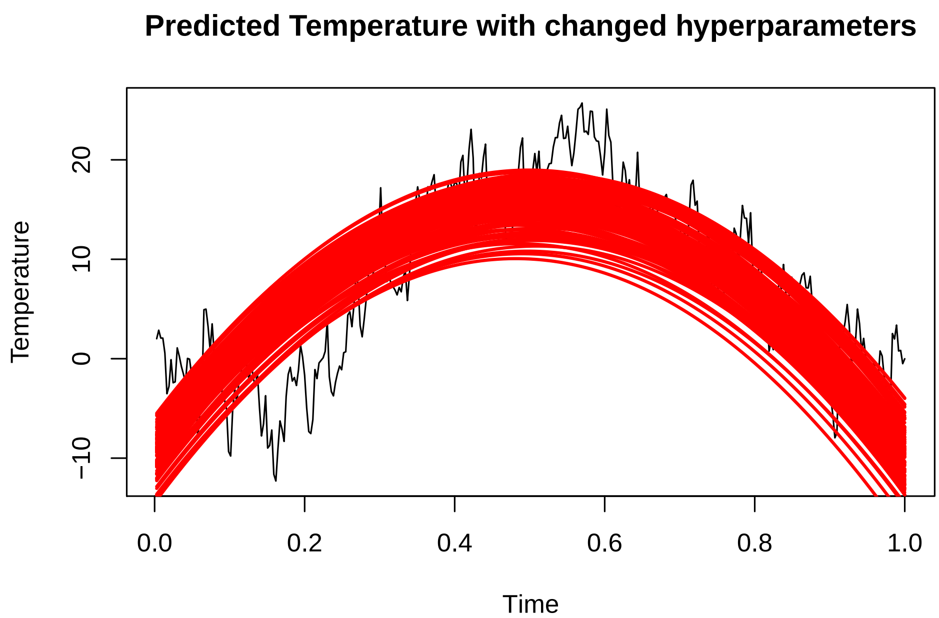
(a) Determining the prior distribution of the model parameters. Use the conjugate prior for the linear regression model. Your task is to set the prior hyperparameters μ_0,Ω_0,v_0 and σ_0 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3$, $v_0 = 4$ and $\sigma_0^2 = 1$. Check if this prior agrees with your prior opinions by simulating draws from the joint prior of all parameters and for every draw compute the regression curve. This gives a collection of regression curves, one for each draw from the prior. Do the collection of curves look reasonable? If not, change the prior hyperparameters until the collection of prior regression curves agrees with your prior beliefs about the regression curve.

We have the joint prior β and σ^2

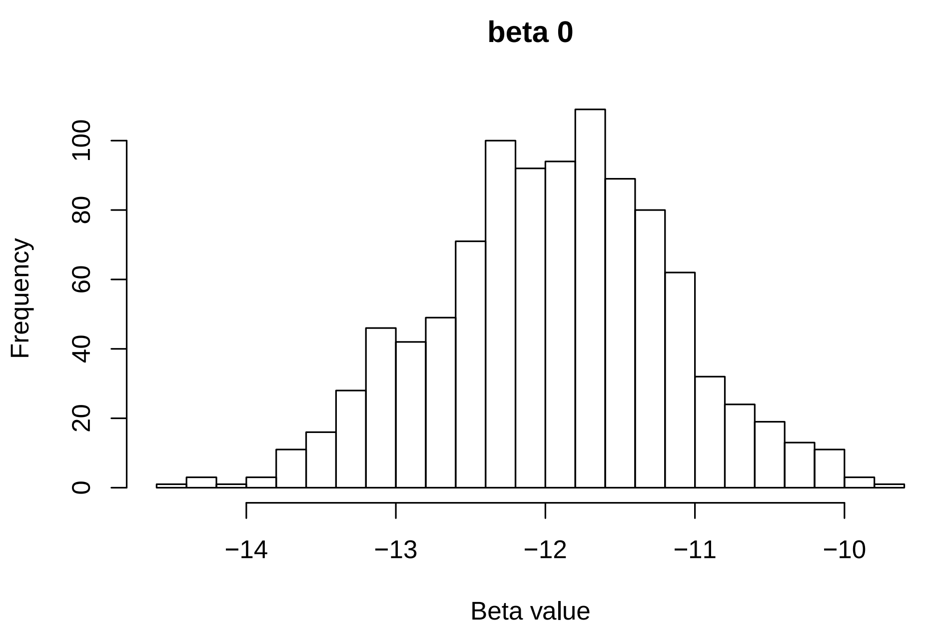
$$\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$$
$$\sigma^2 \sim Inv - \chi^2(v_0, \sigma_0^2)$$

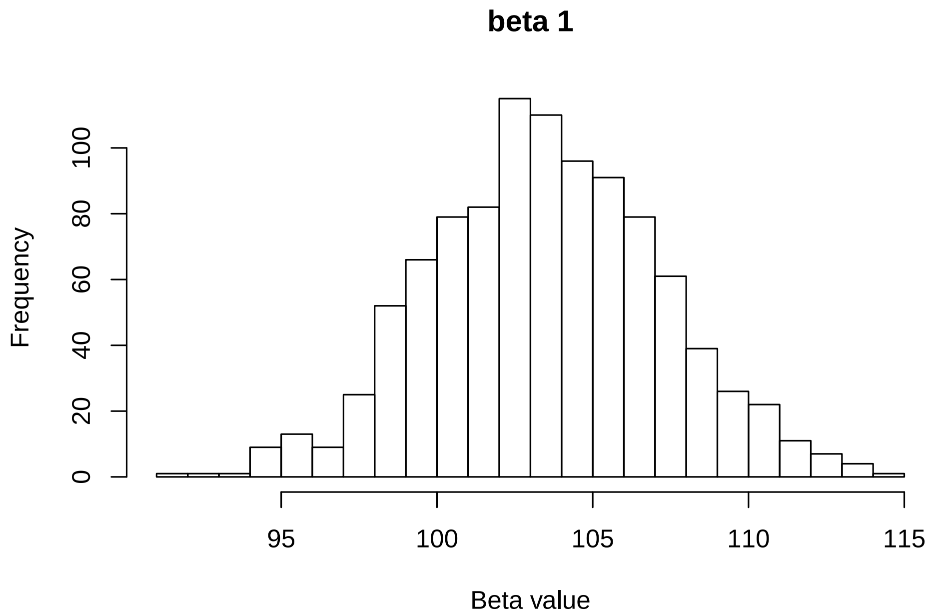
We used 10 draws for the fitting lines, and changed the parameters $sigma^2$to 0.03

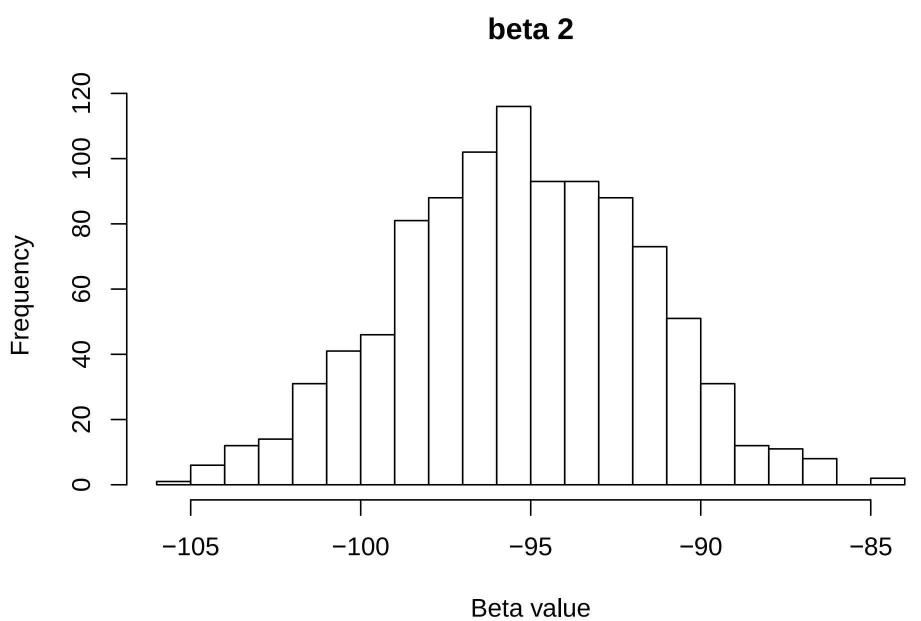


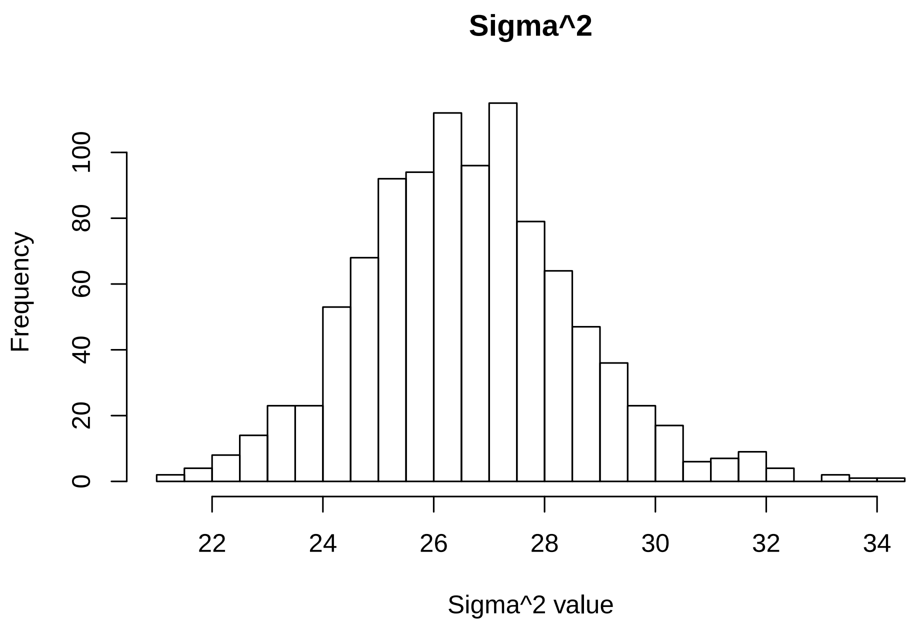


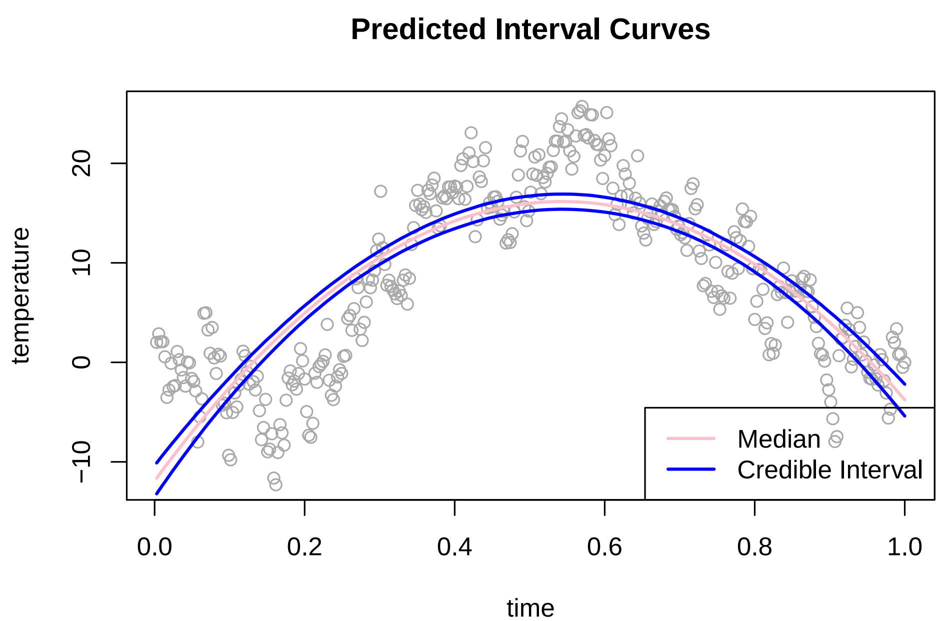
(b)Write a program that simulates from the joint posterior distribution of β_0,β_1,β_2,and σ^2. Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function $f(time) = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$, computed for every value of time. Also overlay curves for the lower 2.5% and upper 97.5% posterior credible interval for f (time). That is, compute the 95% equal tail posterior probability intervals for every value of time and then connect the lower and upper limits of the interval by curves. Does the interval bands contain most of the data points? Should they?




beta 1
Frequency
0
20
40
60
80
100
95
100
105
110
115
Beta value


beta 2
Frequency
0
20
40
60
80
100
120
−105
−100
−95
−90
−85
Beta value


Sigma^2
Frequency
0
20
40
60
80
100
22
24
26
28
30
32
34
Sigma^2 value


Predicted Interval Curves
temperature
−10
0
10
20
0.0
0.2
0.4
0.6
0.8
1.0
time
Median
Credible Interval

(c) It is of interest to locate the time with the highest expected temperature (that is, the time where f (time) is maximal). Let's call this value $\widetilde{x}$ Use the simulations in b) to simulate from the posterior distribution of $\widetilde{x}$

Appendix

```
#~1.a
library(mvtnorm)
data = read.table("~/Desktop/Bayesian/732A91_Lab_1/lab2/TempLinkoping.txt",header=T)

#hyperparameters
mu0=matrix(c(-10,100,-100))
omega0=diag(x=0.01, nrow=3, ncol=3)
v0=4
sigma20=1


#prior

PriorReg = function(mu0,omega0,v0,sigma20){
  set.seed(12345)
  for(i in 1:100){
    #using chi_sq to sample sigma^2
    chi_sample = rchisq(n=1, df=v0)
    sigma2 = v0*sigma20/chi_sample

    #using mvtnorm sample beta
    beta = rmvnorm(n=1, mean=mu0, sigma=sigma20*solve(omega0))

    #quadratic regression
    quad_regre= beta[1]+beta[2]*data$time+beta[3]*(data$time^2)+rnorm(1,mean=0, sd=sqrt(sigma2))
    lines(x=data$time, y=quad_regre,col="red",lwd=2)
  }
}

### Check the given hyperpara
plot(data, main="Predicted Temperature with given hyperparameters", ylab="Temperature", xlab="Time", typ
PriorReg( mu0, omega0, v0, sigma20)


# Change the hyperpara
plot(data, main="Predicted Temperature with changed hyperparameters", ylab="Temperature", xlab="Time", t
PriorReg( mu0, omega0, v0, sigma20=0.03)


#1.b

### find beta hat
n=dim(data)[1]
X = data.frame(intercept=rep(1,n), x1=data$time, x2=data$time^2)
X = as.matrix(X)
y = data$temp
```

```r
betaHat = solve(t(X)%*%X)%*%t(X)%*%y

### calculate mu, omega, nu sigma
mu_n = solve(t(X)%*%X+omega0) %*% (t(X)%*%X%*%betaHat+omega0%*%mu0)
omega_n = t(X)%*%X+omega0
v_n = v0 + n
sigma2_n = (v0*sigma20+(t(y)%*%y+t(mu0)%*%omega0%*%mu0-t(mu_n)%*%omega_n%*%mu_n))/v_n

### Marginal posterior
set.seed(12345)
paras = NULL
final = NULL
for(i in 1:1000){
  #using chi_sq to sample posterior sigma^2
  chi_sample = rchisq(n=1, df=v_n)
  post_sigma2 = v_n*sigma2_n/chi_sample

  #using mvtnorm sample posterior beta
  post_beta = rmvnorm(n=1, mean=mu_n, sigma=post_sigma2[1]*solve(omega_n))

  paras = cbind(post_beta,post_sigma2)
  final = rbind(paras, final)
}

colnames(final) = c("beta0","beta1","beta2","sigma2")

## histogram of each parameters
hist(final[,1], main="beta 0", xlab="Beta value", breaks=30)
hist(final[,2], main="beta 1", xlab="Beta value", breaks=30)
hist(final[,3], main="beta 2", xlab="Beta value", breaks=30)
hist(final[,4], main="Sigma^2",xlab="Sigma^2 value", breaks=30)

### median curve and intervals
post_beta = final[,1:3]

PredictedVal=matrix(0,nrow=n,ncol=nrow(post_beta))
for(i in 1:nrow(post_beta)){
  PredictedVal[,i] = X %*% post_beta[i,]
}


## find median and credible interval
medianInterval=c()
crediInterval = matrix(0,nrow=n,ncol=2)
for(i in 1:n){
  medianInterval[i] = median(PredictedVal[i,])
  crediInterval[i,] = quantile(PredictedVal[i,], c(0.025,0.975))

}

plot(data, main="Predicted Interval Curves", col="darkgrey", ylab="temperature")
lines(data$time,medianInterval, col="pink",lwd=2)
lines(data$time,crediInterval[,1], col="blue",lwd=2)
```

```r
lines(data$time,crediInterval[,2], col="blue",lwd=2)
legend("bottomright",legend=c("Median", "Credible Interval"), col=c("pink","blue"),lwd=2 )
```