Lab2 Report

Ludvig Noring, Michael Sörsäter

April 26, 2017

1. Linear and polynomial regression

a & b

We tried to reason about the hyperparameters. After simulating draws from the model we updated the hyperparameters and ended up with the following values. The resulting regression curves can be seen in Figure 1.

$$\mu_0 = (-10, 100, -100)$$

$$\Omega_0 = I$$

$$v_0 = 366 - 3 = 363$$

$$\sigma_0^2 = 5$$

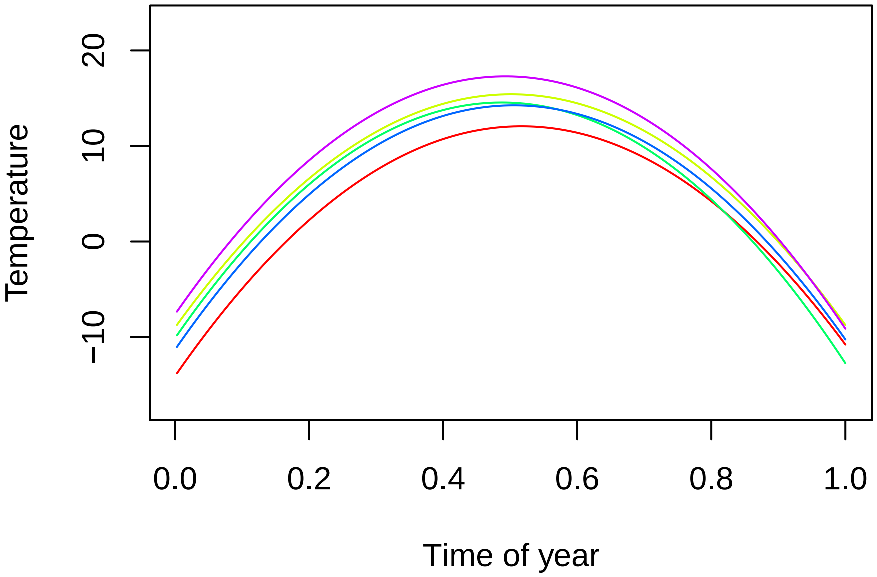

Figure 1: Regression curves

The temperature starts at around $-10°C$ and reaches its peak in the middle of the year at $20°C$ and finally ends at the same starting value $-10°C$.

c

We simulated draws from the joint posterior and calculated the posterior mean of the β values. The plot of the posterior mean together with the credible intervals can be seen in Figure 2.

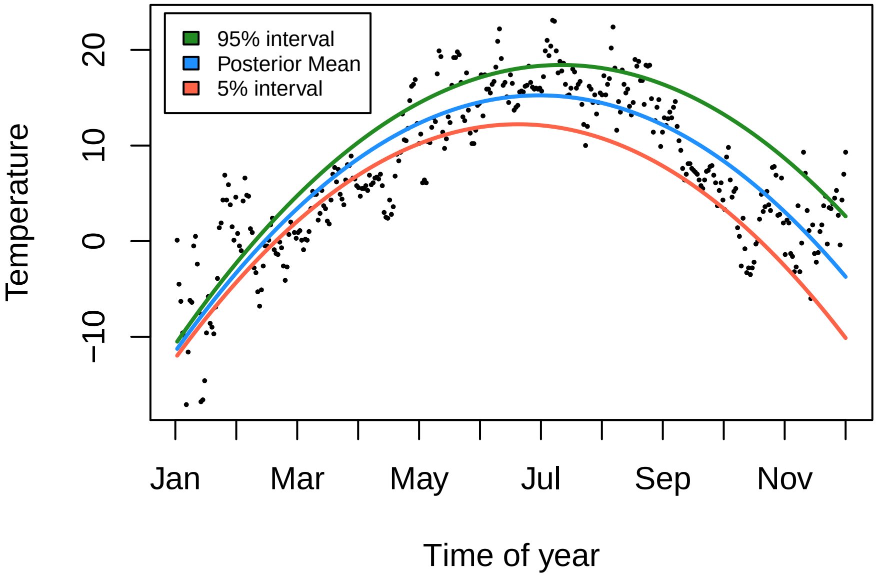

Figure 2: Scatter plot with credible intervals

d

We calculated the maximum temperature with the following formula. This resulted in the numeric value 0.5429473 which corresponds to the date 17th of July.

$$\tilde{x} = -\frac{\beta_1}{2\beta_2}$$

e

When estimating with a polynomial with degree 7 we supress polynomial by setting the coefficients of the higher orders to zero. Below is μ_0 and Ω_0.

	1	2	3	4	5	6	7	8
1	-10.00	100.00	-100.00	0.00	0.00	0.00	0.00	0.00

Table 1: Mu

	1	2	3	4	5	6	7	8
1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 2: Omega

2. Posterior approximation for classification with logistic regression

a

Using the glm function to fit the logistic regression results in the coefficients seen in table 3.

	Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
1	0.64	-0.02	0.18	0.17	-0.14	-0.08	-1.36	-0.03

Table 3: Coefficents using glm

b

We used the function optim to find optimal values for $\tilde{\beta}$ and $J_y^{-1}(\tilde{\beta})$. The result can be show in the tables below. For the feature NSmallChild the 95% credible interval is shown in Figure 3.

	1	2	3	4	5	6	7	8
1	0.63	-0.02	0.18	0.17	-0.14	-0.08	-1.36	-0.02

Table 4: Beta tilde

c

From the optimal β values we simulated 1000 draws and for each of them calculated the probability to be used in the Bernoulli draws. From the Bernoulli draws we plotted the probability that the woman is working. The plot can be seen in Figure 4.

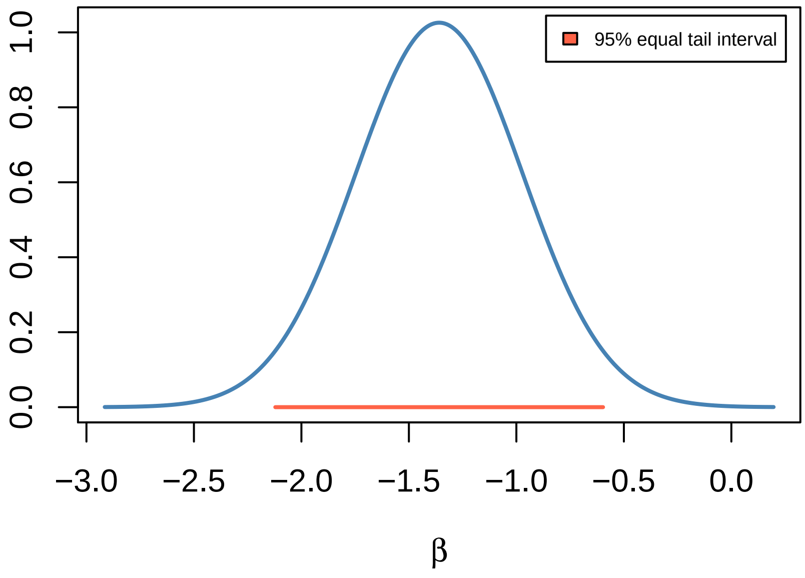

Figure 3: Equal Tail intervals for NSmallChild

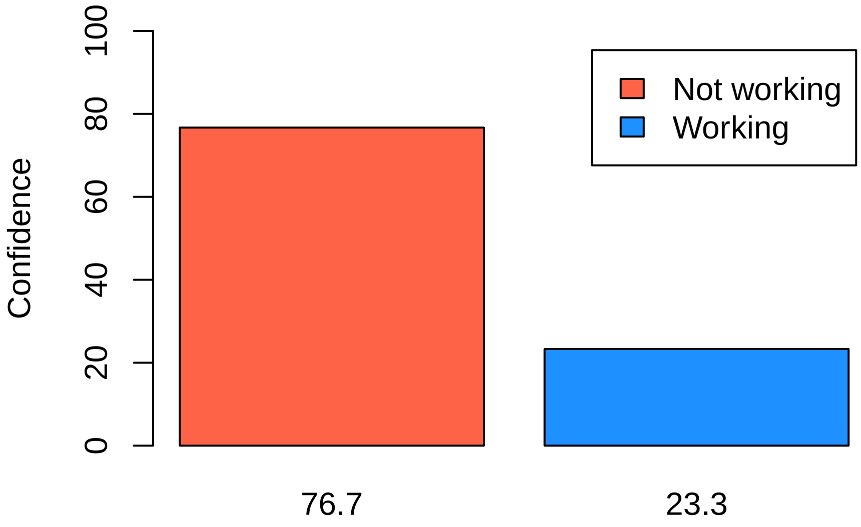

Figure 4: Working confidence

	1	2	3	4	5	6	7	8
1	2.27	0.00	-0.07	-0.01	0.05	-0.03	-0.19	-0.10
2	0.00	0.00	-0.00	-0.00	0.00	-0.00	0.00	-0.00
3	-0.07	-0.00	0.01	-0.00	0.00	-0.00	-0.01	0.00
4	-0.01	-0.00	-0.00	0.00	-0.01	-0.00	-0.00	0.00
5	0.05	0.00	0.00	-0.01	0.06	-0.00	0.00	0.00
6	-0.03	-0.00	-0.00	-0.00	-0.00	0.00	0.01	0.00
7	-0.19	0.00	-0.01	-0.00	0.00	0.01	0.15	0.01
8	-0.10	-0.00	0.00	0.00	0.00	0.00	0.01	0.02

Table 5: Observed hessian evaluated at the posterior mode

Appendix A - Code for assignment 1

```r
require('polynom')
require('MASS')
require('geoR')
set.seed(1234567890)
imgw = 5
imgh = 4

data = read.table('TempLinkoping2016.txt', header=TRUE)
n = nrow(data)

# a
quad = lm(temp ~ time + I(time^2), data=data)
coefs = polynomial(c(quad$coefficients))

#plot(data$time, data$temp, pch=20)
#lines(data$time, predict(coefs, data$time))

# Hyper parameters
mu_0 = c(-10, 100, -100)
omega_0 = diag(3)
v_0 = n - 3
sigma2_0 = 5

# b
cols = rainbow(5)

pdf('plots/hyper.pdf', width=imgw, height=imgh)
  plot(data, pch=20, cex=0.3, type='n', xlab='Time of year', ylab='Temperature')
  for (i in 1:5){
    sigma2 = rinvchisq(1, df=v_0, scale=sigma2_0)
    betas = mvrnorm(n = 1, mu = mu_0, Sigma=sigma2 * solve(omega_0))
    coefs = polynomial(betas)
    lines(data$time, predict(coefs, data$time), col=cols[i])
  }
dev.off()

# c
y = data$temp
X = as.matrix(data.frame(beta0=rep(1, n), beta1=data$time, beta2=data$time^2))

beta_hat = solve(t(X) %*% X) %*% t(X) %*% y
mu_n = solve(t(X) %*% X + omega_0) %*% (t(X) %*% X %*% beta_hat + omega_0 %*% mu_0)
omega_n = t(X) %*% X + omega_0
v_n = v_0 + n
v_n.sigma2_n = v_0 * sigma2_0 +
                (t(y) %*% y +
                t(mu_0) %*% omega_0 %*% mu_0 -
                t(mu_n) %*% omega_n %*% mu_n)

sigma2 = rinvchisq(1000, df=v_n, scale=v_n.sigma2_n/v_n)
betas = sapply(sigma2, function(sigma2){mvrnorm(n=1, mu=mu_n, Sigma=sigma2*solve(omega_n))})
```

```
betas_mean = apply(betas, 1, mean)
print(betas_mean)

ci = apply(betas, 1, quantile, probs=c(0.05, 0.95))

cols = c('forestgreen', 'dodgerblue', 'tomato')
pdf('plots/posterior.pdf', width=imgw, height=imgh)
  plot(data, pch=20, cex=0.3, xlab='Time of year', ylab='Temperature', xaxt='n')
  lines(data$time, predict(polynomial(ci[2,]), data$time), col=cols[1], lwd=2)
  lines(data$time, predict(polynomial(betas_mean), data$time), col=cols[2], lwd=2)
  lines(data$time, predict(polynomial(ci[1,]), data$time), col=cols[3], lwd=2)
  legend('topleft', inset=0.02,
         legend = c('95% interval', 'Posterior Mean','5% interval'),
         fill = cols, cex=0.65)
  axis(1, at=c(seq(0,1,length=12)),
       labels=c('Jan', 'Feb','Mar', 'Apr','May', 'Jun','Jul', 'Aug', 'Sep', 'Oct','Nov', 'Dec'))
dev.off()
# d
hot_day = data$time[which.max(predict(polynomial(betas_mean), data$time))]
hot_day
hot_day = as.numeric(betas_mean[2] / (-betas_mean[3]*2))
hot_day
hot_date = as.Date(hot_day*n, origin="2016-01-01")
hot_date

xGrid = seq(0,1,0.001)
a = dnorm(xGrid, mean=hot_day, sd=0.1)
tjo = max(a)
plot(as.Date(xGrid*n, origin="2016-01-01"),a, type='l')
abline(h=tjo/2)
```

Appendix B - Code for assignment 2

```
require('mvtnorm')
require('LaplacesDemon')
imgw = 5
imgh = 4
data = read.table('WomenWork.dat.txt', header=TRUE)
response = data$Work
features = as.matrix(data[,-data$Work])

# a
fit = glm(Work ~ 0 + ., data = data, family = 'binomial')

# b
nFeats = ncol(features)
mu = rep(0, nFeats)
tau = 10
sigma2 = tau^2 * diag(nFeats)
beta_prior = rnorm(mu, sigma2)

logPostLogistic = function(betas, y, X, mu, sigma2) {
  nPara = length(betas)
  yPred = as.matrix(X) %*% betas

  logLike = sum(yPred*y - log(1 + exp(yPred)))
  if (abs(logLike) == Inf) {
    logLike = -20000
  }
  logPrior = dmvnorm(betas, rep(0, nPara), sigma2, log=TRUE)
  logPost = logPrior + logLike
  return (logPost)
}

initValues = rep(0, nFeats)
optimResults = optim(initValues,
                     logPostLogistic, y=response, X=features, mu=mu, sigma2=sigma2,
                     method=c('BFGS'), control=list(fnscale=-1), hessian=TRUE)

postMode = optimResults$par
postCov = -solve(optimResults$hessian)
approxPostStd = sqrt(diag(postCov))

names(postMode) = colnames(features)
names(approxPostStd) = colnames(features)

childMean = postMode['NSmallChild']
childStd = approxPostStd['NSmallChild']

betaGrid = seq(-abs(childMean - 4*childStd), abs(childMean + 4* childStd), length = 1000)

pdf('plots/smallchildren.pdf', width=imgw, height=imgh)
  plot(betaGrid, dnorm(betaGrid, childMean, childStd),
       type = "l", lwd = 2, ylab = '', xlab = expression(beta), col='steelblue')
```

```r
  lines(qnorm(c(0.025, 0.975), childMean, childStd), c(0,0), lwd=2, col='tomato')
  legend('topright', legend=c('95% equal tail interval'), fill=c('tomato'), inset=0.02, cex=0.6)
dev.off()
# c

inverseLogit = function(betas, features){
  return (exp(features %*% betas) / (1 + exp(features %*% betas)))
}

jane_doe = c(1, 10, 8, 10, 1, 40, 1, 1)
betas = rmvnorm(1000, mean=postMode, sigma=postCov)

p = apply(betas, 1, inverseLogit, features=jane_doe)
y = sapply(p, rbern, n=1)

h = hist(y, breaks=2, plot=FALSE)
h$counts = h$counts / sum(h$counts)
h$counts = h$counts * 100
names(h$counts) = h$counts

pdf('plots/working.pdf', width=imgw, height=imgh)
  barplot(h$counts, ylab='Confidence', ylim=c(0,100), col=c('tomato', 'dodgerblue'),
          main='', legend=c('Not working', 'Working'))
dev.off()
```