

Bayesian Learning - Lab 01

Lakshidaa Saigiridharan (laksa656) and Maximilian Pfundstein (maxpf364)

2019-04-09

Contents

1	Linear and Polynomial Regression	1
1.1	Determining the prior Distribution of the Model Parameters	2
1.2	Simulating from the Joint Posterior Distribution	4
2	Posterior Approximation for Classification with Logistic Regression	4
3	Source Code	5

1 Linear and Polynomial Regression

Exercise: The dataset `TempLinkoping.txt` contains daily temperatures (in Celcius degrees) at Malslätt, Linköping over the course of the year 2016 (366 days since 2016 was a leap year). The response variable is `temp` and the covariate is

$$time = \frac{\text{the number of days since beginning of year}}{366}$$

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

$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

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, ν_0 and σ_0^2 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3$, $\nu_0 = 4$ 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 do agree with your prior beliefs about the regression curve. [Hint: the R package `mvtnorm` will be handy. And use your `Inv - χ^2` simulator from Lab 1.]

b) Write a program that simulates from the joint posterior distribution of $\beta_0, \beta_1, \beta_2$ 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?

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 \tilde{x} . Use the simulations in b) to simulate from the *posterior distribution of \tilde{x}* . [Hint: the regression curve is a quadratic. You can find a simple formula for \tilde{x} given β_0, β_1 and β_2 .]

d) Say now that you want to estimate a polynomial model of order 7, but you suspect that higher order terms may not be needed, and you worry about overfitting. Suggest a suitable prior that mitigates this

potential problem. You do not need to compute the posterior, just write down your prior. [Hint: the task is to specify μ_0 and Ω_0 in a smart way.]

1.1 Determining the prior Distribution of the Model Parameters

Let's first take a look at the data to get a feeling for it.

```
##           time  temp
## 1 0.002732240   0.1
## 2 0.005464481  -4.5
## 3 0.008196721  -6.3
## 4 0.010928962  -9.6
## 5 0.013661202  -9.9
## 6 0.016393443 -17.1
```

We save the initial parameters to variables and provide a function which calculates σ^2 and β from the given parameters, which we use as the parameters for the prior. The calculation is done as follows:

1. First draw σ^2 from $Inv - \chi^2(\nu_0, \sigma_0^2)$
2. Then draw β from $\mathcal{N}(\mu_0, \sigma^2 \Omega_0^{-1})$
3. Calculate the prior according to $\hat{y} = X\beta$ where $X = (1, \text{temp}, \text{temp}^2)$. The first 1 is the intercept.

```
# Initial Parameters
mu_0 = c(-10, 100, -100)
omega_0 = 0.01 * diag(3)
nu_0 = 4
sigma_sq_0 = 1
n = 5 # Number of prior regression curves

predict_using_model_params = function(time, mu_0, omega_0, nu_0, sigma_sq_0) {

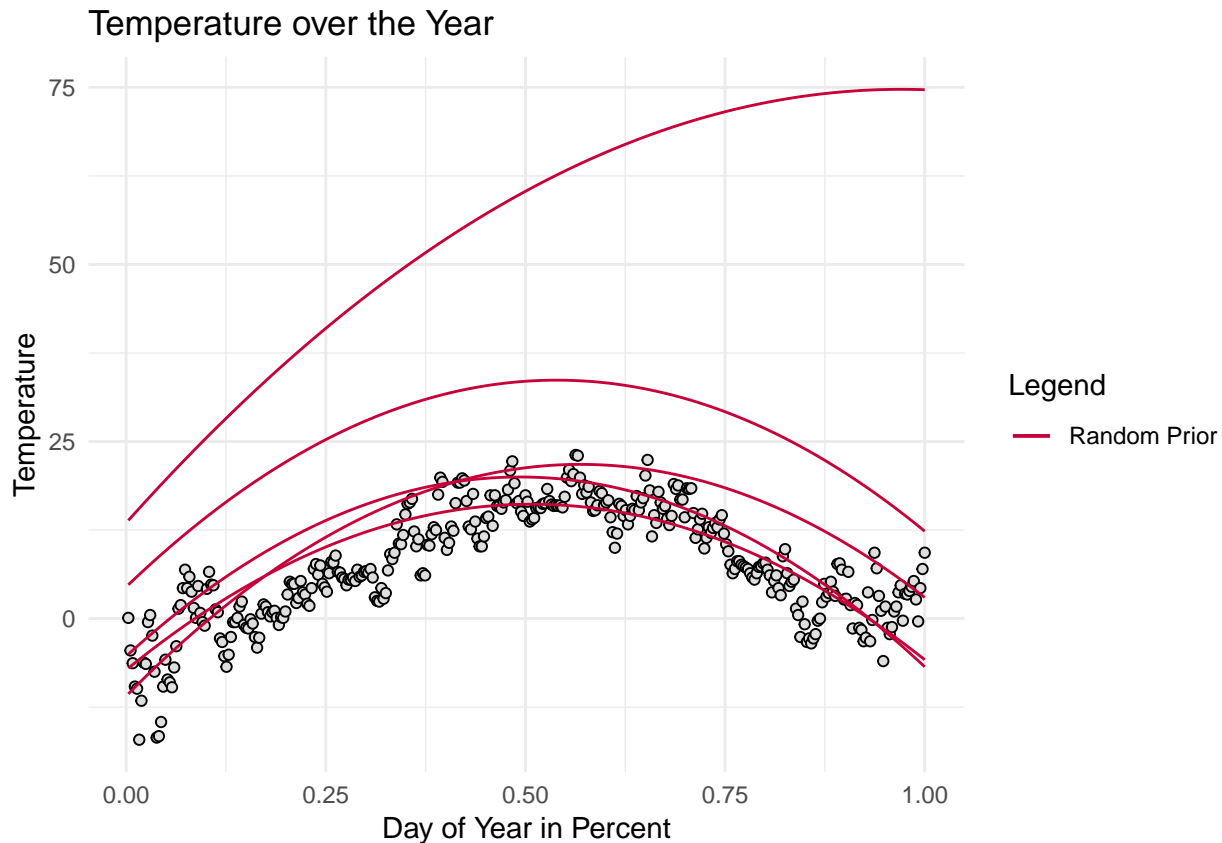
  # Taken from slides (set 5, page 7)
  # First we sample the model parameters
  sigma_sq = rinvchisq(n = 1, df = nu_0, scale = sigma_sq_0)
  beta = mvnrm(n = 1, mu = mu_0, Sigma = sigma_sq * solve(omega_0))

  # Then we perform the regression
  temp =
    matrix(c(rep(1, length(time)), time, time^2), ncol = 3) %*% matrix(beta)

  return(temp)
}

priors = matrix(0, ncol = n + 2, nrow = length(temperatures_linkoeeping$time))
priors[,1] = temperatures_linkoeeping$time
priors[,2] = temperatures_linkoeeping$temp

for (i in 3:(n+2)) {
  priors[, i] = predict_using_model_params(
    temperatures_linkoeeping$time, mu_0, omega_0, nu_0, sigma_sq_0)
}
```



We can see that the chosen parameters for our prior are not optimal and it should be easy to get a better prior. We therefore select the following new parameters:

```
# Adjusted Parameters
mu_0 = c(-8, 90, -82)
omega_0 = 0.2 * diag(3)
nu_0 = 4
sigma_sq_0 = 2
n = 5 # Number of prior regression curves
```

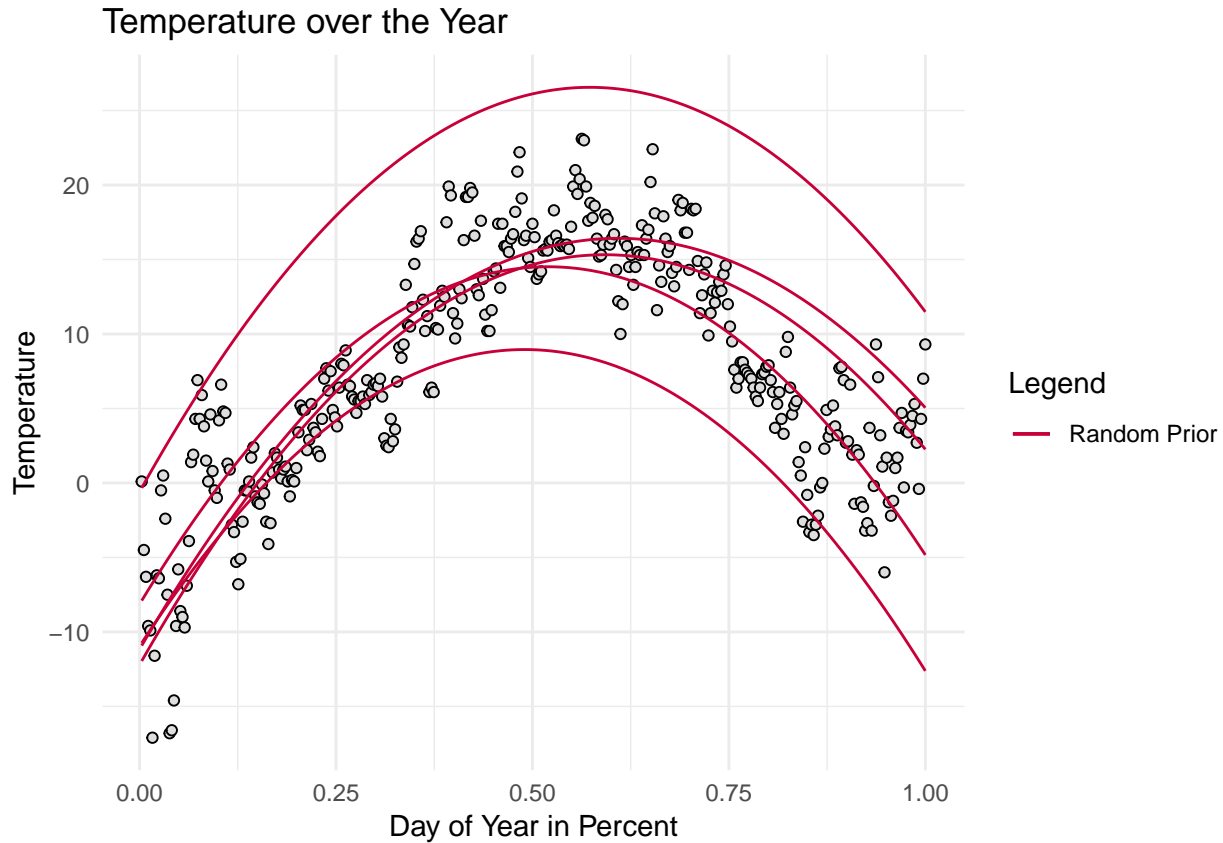
μ_0 : Looking at the data we want to slightly adjust the intercept, so we set the first entry of μ_0 to -8 (degrees). For the second linear regression coefficient we choose to set it to 90 as the data trend seems to be warmer in the second half of the year and to capture the warm spike in the middle around 0.6 of the year. For the last linear regression coefficient we chose a value that pulls the curve down again on the right side so that it fits the data the most..

Ω_0 : The variance seems too high as we don't think the mean temperature will fluctuate that much, we thereby decrease it to 0.2 (decrease as the inverse is taken).

σ^2 : We increase the variance of the error to allow for more fluctuation (and increasing uncertainty).

```
priors = matrix(0, ncol = n + 2, nrow = length(temperatures_linkoepping$time))
priors[,1] = temperatures_linkoepping$time
priors[,2] = temperatures_linkoepping$temp

for (i in 3:(n+2)) {
  priors[, i] = predict_using_model_params(
    temperatures_linkoepping$time, mu_0, omega_0, nu_0, sigma_sq_0)
}
```



We see that the parameters make the prior fit better to the observed (optical inspected) data.

1.2 Simulating from the Joint Posterior Distribution

2 Posterior Approximation for Classification with Logistic Regression

Exercise: The dataset `WomenWork.dat` contains $n = 200$ observations (i.e. women) on the following nine variables:

Variable	Data Type	Meaning	Role
Work	Binary	Whether or not the woman works	Response
Constant	1	Constant to the intercept	Feature
HusbandInc	Numeric	Husband's income	Feature
EducYears	Counts	Years of education	Feature
ExpYears	Counts	Years of experience	Feature
ExpYears2	Numeric	$(\text{Years of experience}/10)^2$	Feature
Age	Counts	Age	Feature
NSmallChildren	Counts	Number of child ≤ 6 years in household	Feature
NBigChildren	Counts	Number of child > 6 years in household	Feature

a) Consider the logistic regression

$$Pr(y = 1|x) = \frac{\exp(x^T \beta)}{1 + \exp(x^T \beta)}$$

where y is the binary variable with $y = 1$ if the woman works and $y = 0$ if she does not. x is a 8-dimensional vector containing the eight features (including a one for the constant term that models the intercept). Fit the logistic regression using maximum likelihood estimation by the command:

```
gmlModel = glm(Work ~ 0 + ., data = WomanWork, family = binomial)
```

Note how I added a zero in the model formula so that R doesn't add an extra intercept (we already have an intercept term from the Constant feature). Note also that a dot (.) in the model formula means to add all other variables in the dataset as features. `family = binomial` tells R that we want to fit a logistic regression.

- b) Now the fun begins. Our goal is to approximate the posterior distribution of the 8-dim parameter vector β with a multivariate normal distribution

$$\beta|y, X \sim \mathcal{N}(\tilde{\beta}, J_y^{(-1)}(\tilde{\beta})),$$

where $\tilde{\beta}$ is the posterior mode and $J(\tilde{\beta}) = -\frac{\partial^2 \ln p(\beta|y)}{\partial \beta \partial \beta^T} |_{\beta=\tilde{\beta}}$ is the observed Hessian evaluated at the posterior mode. Note that $\frac{\partial^2 \ln p(\beta|y)}{\partial \beta \partial \beta^T}$ is an 8×8 matrix with second derivatives on the diagonal and cross-derivatives $\frac{\partial^2 \ln p(\beta|y)}{\partial \beta_i \partial \beta_i^T}$ on the offdiagonal. It is actually not hard to compute this derivative by hand, but don't worry, we will let the computer do it numerically for you. Now, both $\tilde{\beta}$ and $J(\tilde{\beta})$ are computed by the `optim` function in R. See my code

<https://github.com/mattiasvillani/BayesLearnCourse/blob/master/Code/MainOptimizeSpam.zip>

where I have coded everything up for the spam prediction example (it also does probit regression, but that is not needed here). I want you to implement your own version of this. You can use my code as a template, but I want you to write your own file so that you understand every line of your code. Don't just copy my code. Use the prior $\beta \sim \mathcal{N}(0, \tau^2 I)$, with $\tau = 10$. Your report should include your code as well as numerical values for $\tilde{\beta}$ and $J_y^{(-1)}(\tilde{\beta})$ for the `WomanWork` data. Compute an approximate 95% credible interval for the variable `NSmallChild`. Would you say that this feature is an important determinant of the probability that a woman works?

- c) Write a function that simulates from the predictive distribution of the response variable in a logistic regression. Use your normal approximation from 2(b). Use that function to simulate and plot the predictive distribution for the `Work` variable for a 40 year old woman, with two children (3 and 9 years old), 8 years of education, 10 years of experience. and a husband with an income of 10. [Hint: the R package `mvtnorm` will again be handy. And remember my discussion on how Bayesian prediction can be done by simulation.]

3 Source Code

```
knitr::opts_chunk$set(echo = TRUE)
library(kableExtra)
library(geoR)
library(MASS)
library(ggplot2)
set.seed(42)

temperatures_linkoeeping = read.table("data/TempLinkoeeping.txt", header=TRUE)
head(temperatures_linkoeeping)
```

```

# Initial Parameters
mu_0 = c(-10, 100, -100)
omega_0 = 0.01 * diag(3)
nu_0 = 4
sigma_sq_0 = 1
n = 5 # Number of prior regression curves

predict_using_model_params = function(time, mu_0, omega_0, nu_0, sigma_sq_0) {

  # Taken from slides (set 5, page 7)
  # First we sample the model parameters
  sigma_sq = rinvchisq(n = 1, df = nu_0, scale = sigma_sq_0)
  beta = mvrnorm(n = 1, mu = mu_0, Sigma = sigma_sq * solve(omega_0))

  # Then we perform the regression
  temp =
    matrix(c(rep(1, length(time)), time, time^2), ncol = 3) %*% matrix(beta)

  return(temp)
}

priors = matrix(0, ncol = n + 2, nrow = length(temperatures_linkoeeping$time))
priors[,1] = temperatures_linkoeeping$time
priors[,2] = temperatures_linkoeeping$temp

for (i in 3:(n+2)) {
  priors[, i] = predict_using_model_params(
    temperatures_linkoeeping$time, mu_0, omega_0, nu_0, sigma_sq_0)
}

df = data.frame(priors)

plot = ggplot(df) +
  geom_point(aes(x = df[, 1], y = df[,2]), color = "black", fill = "#dedede",
    shape = 21)
  #for (i in 3:(n+2)) {
  # plot = plot + geom_line(aes(x = df[,1], y = df[,i], color = "Random Prior"))
  #}
  plot = plot + geom_line(aes(x = df[,1], y = df[,3], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,4], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,5], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,6], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,7], color = "Random Prior"))
  plot = plot + labs(title = "Temperature over the Year",
    y = "Temperature", x = "Day of Year in Percent", color = "Legend")
  plot = plot + scale_color_manual("Legend", values = rep("#C70039", n))
  plot = plot + theme_minimal()

plot

# Adjusted Parameters

```

```

mu_0 = c(-8, 90, -82)
omega_0 = 0.2 * diag(3)
nu_0 = 4
sigma_sq_0 = 2
n = 5 # Number of prior regression curves

priors = matrix(0, ncol = n + 2, nrow = length(temperatures_linkoeeping$time))
priors[,1] = temperatures_linkoeeping$time
priors[,2] = temperatures_linkoeeping$temp

for (i in 3:(n+2)) {
  priors[, i] = predict_using_model_params(
    temperatures_linkoeeping$time, mu_0, omega_0, nu_0, sigma_sq_0)
}

df = data.frame(priors)

plot = ggplot(df) +
  geom_point(aes(x = df[, 1], y = df[,2]), color = "black", fill = "#dedede",
    shape = 21)
  #for (i in 3:(n+2)) {
  # plot = plot + geom_line(aes(x = df[,1], y = df[,i], color = "Random Prior"))
  #}
  plot = plot + geom_line(aes(x = df[,1], y = df[,3], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,4], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,5], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,6], color = "Random Prior"))
  plot = plot + geom_line(aes(x = df[,1], y = df[,7], color = "Random Prior"))
  plot = plot + labs(title = "Temperature over the Year",
    y = "Temperature", x = "Day of Year in Percent", color = "Legend")
  plot = plot + scale_color_manual("Legend", values = rep("#C70039", n))
  plot = plot + theme_minimal()

plot

variable = c("Work", "Constant", "HusbandInc", "EducYears", "ExpYears",
  "ExpYears2", "Age", "NSmallChildren", "NBigChildren")
dataType = c("Binary", "1", "Numeric", "Counts", "Counts", "Numeric",
  "Counts", "Counts", "Counts")
meaning = c("Whether or not the woman works", "Constant to the intercept",
  "Husband's income", "Years of education", "Years of experience",
  "(Years of experience/10)^2", "Age",
  "Number of child <= 6 years in household",
  "Number of child > 6 years in household")
feature = c("Response", "Feature", "Feature", "Feature", "Feature", "Feature",
  "Feature", "Feature", "Feature")

df = data.frame(variable, dataType, meaning, feature)
colnames(df) = c("Variable", "Data Type", "Meaning", "Role")

kable(df) %>%

```

```
kable_styling(position = "center") %>%  
row_spec(0,bold=TRUE)  
  
gmlModel = glm(Work ~ 0 + ., data = WomanWork, family = binomial)
```