CS4070: ASSIGNMENT 2: POISSON REGRESSION

Hand in before 22 December, 23:59. Include code as an appendix.

Consider the Poisson regression model, which is a basic model for count-data. So we assume data $\{(x_i, y_i)\}_{i=1}^n$, where $y_i \in \{0, 1, 2, \ldots\}$ and $x_i \in \mathbb{R}^p$. The model is given by

$$y_i \sim Pois\,(\mu_i)\,, \qquad \text{where} \quad \log \mu_i = x_i^T \theta$$

for an unknown parameter vector $\theta \in \mathbb{R}^p$. Hence, for $k \in \{0, 1, 2, \ldots\}$ we have

$$\mathbb{P}(Y_i = k) = e^{-\mu_i} \mu_i^k / (k!).$$

(1) Give the loglikelihood, assuming all y_i are independent.
(2) Derive an expression for the gradient and Hessian of the loglikelihood.
(3) In the following we consider the dataset `dataexercise2.csv`. We take a Bayesian point of view, where we assume

$$\begin{aligned} y_i \mid \theta &\stackrel{\text{ind}}{\sim} Pois\,(\mu_i)\,, \qquad \text{where} \quad \log \mu_i = x_i^T \theta \\ \theta &\sim N\left(0, \tilde{\sigma}^2 I_p\right). \end{aligned}$$

Assume the prior standard deviation is given by $\tilde{\sigma} = 4$. Implement a Newton algorithm for computing the Laplace approximation to the posterior distribution. Report mean and covariance matrix of the approximation.
(4) Implement a random walk Metropolis–Hastings algorithm to sample from the posterior. Take proposals of the form $\theta^\circ := \theta + \sigma_{\text{proposal}} N\,(0, I_p)$. Tune σ_{proposal} to achieve an acceptance rate of about $25\% - 50\%$. Make a plot of the iterates where you plot θ_2 versus θ_1, with colour indicating the iteration number. Report the Monte-Carlo estimate of the posterior mean (where you "throw away" burn-in samples, i.e. initial samples where the chain has not reached its stationary region).
(5) The results may be sensitive to the choice of $\tilde{\sigma}$. For that reason we add an extra layer to the hierarchical model in the following way:

$$\begin{aligned} y_i \mid \theta &\stackrel{\text{ind}}{\sim} Pois\,(\mu_i)\,, \qquad \text{where} \quad \log \mu_i = x_i^T \theta, \\ \theta \mid \tilde{\sigma} &\sim N\left(0, \tilde{\sigma}^2 I_p\right), \\ \tilde{\sigma}^2 &\sim IG\,(\alpha, \beta)\,. \end{aligned}$$

Here $IG\,(\alpha, \beta)$ denotes the inverse Gamma distribution with parameters α and β (its density function is given in Exercise 3.12 in RG). Take $\alpha = \beta = 0.2$. Implement a Gibbs sampler that iteratively samples from the full conditionals of θ and $\tilde{\sigma}$.

Include a derivation for the update step for $\tilde{\sigma}^2$ in your report. Also include a traceplot of the posterior samples of $\tilde{\sigma}^2$ (a traceplot is a plot of iterate value versus iterate number).

Date: December 7, 2023.

Solutions

(1) We have

$$\log p(y_i \mid \theta) = -\mu_i + y_i \log \mu_i - \log y_i!,$$

where $\mu_i = \exp(\theta^T x_i)$. The loglikelihood is just $\ell(\theta) := \sum_{i=1}^n \log p(y_i \mid \theta)$.

(2) We have, for $j \in \{1, \ldots, p\}$,

$$\frac{\partial \ell(\theta)}{\partial \theta_j} = \sum_{i=1}^n \frac{\partial}{\partial \theta_j} \log p(y_i \mid \theta) = \sum_{i=1}^n x_{ij}(y_i - \mu_i).$$

By stacking elements, we get $G(\theta) := \nabla \ell(\theta) = X^T(y - \mu)$.

For the Hessian, note that

$$\frac{\partial^2 \ell(\theta)}{\partial \theta_j \partial \theta_k} = -\sum_{i=1}^n x_{ij} \frac{\partial \mu_i}{\partial \theta_k} = -\sum_{i=1}^n x_{ij} x_{ik} \mu_i.$$

Hence

$$H(\theta) = -X^T \operatorname{diag}(\mu_1, \ldots, \mu_p) X.$$

(3) See files `poisson.jl` (main file) and `samplers.jl`. Newton-Raphson should be coded along the following lines:

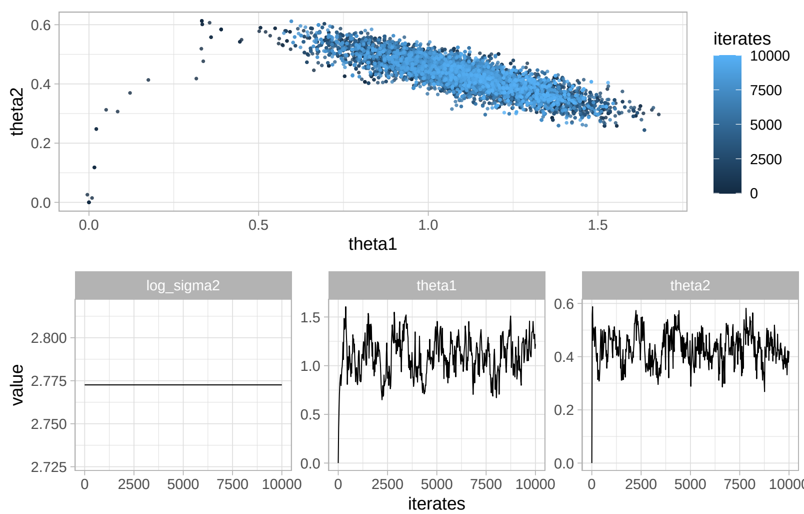
```julia
function NewtonRaphson_step!(θiters,X,y,σprior, πprior)
    θ = θiters[end]
    H = Hessianlogtarget(θ,X,σprior)
    ∇ = ∇logtarget(θ,X,y,σprior)
    θᵒ = θ - inv(H)*∇
    push!(θiters, θᵒ)
    logtarget(θᵒ,X,y,πprior)
end

function NewtonRaphson!(θiters,X,y,σprior, πprior; acc= 10^(-8),verbose=false)
    done = false
    logtargetval = logtarget(θiters[end],X,y,πprior)
    while !done
        logtargetvalnew = NewtonRaphson_step!(θiters,X,y,σprior, πprior)
        done = abs(logtargetval-logtargetvalnew)<acc
        if !verbose
            println(done, round(logtargetvalnew;digits=2), round.(θiters[end];digits=2))
        end
        logtargetval = logtargetvalnew
    end
    H = Hessianlogtarget(θiters[end],X,σprior)
    println("Stationary point is a (local) maximum: ",isposdef(-H))
    H
end
```

I obtain mean and covariance matrix of the Laplace approximation to be equal to

$$\mu_{\text{Lap}} = \begin{bmatrix} 1.13 \\ 0.43 \\ 0.02 \\ -0.05 \end{bmatrix} \qquad \Sigma_{\text{Lap}} = \begin{bmatrix} 0.0314 & -0.0082 & 0.0012 & -0.0014 \\ -0.0082 & 0.0031 & -0.0003 & 0.0007 \\ 0.0012 & -0.0003 & 0.0148 & -0.0015 \\ -0.0014 & 0.0007 & -0.0015 & 0.0117 \end{bmatrix}$$

(4) I took 10000 iterations, discarding the first half as burn-in. I got the posterior mean vector $\begin{bmatrix} 1.116 & 0.428 & 0.037 & -0.062 \end{bmatrix}$. With $\sigma_{\text{proposal}} = 0.05$, I got an acceptance percentage of 47%.

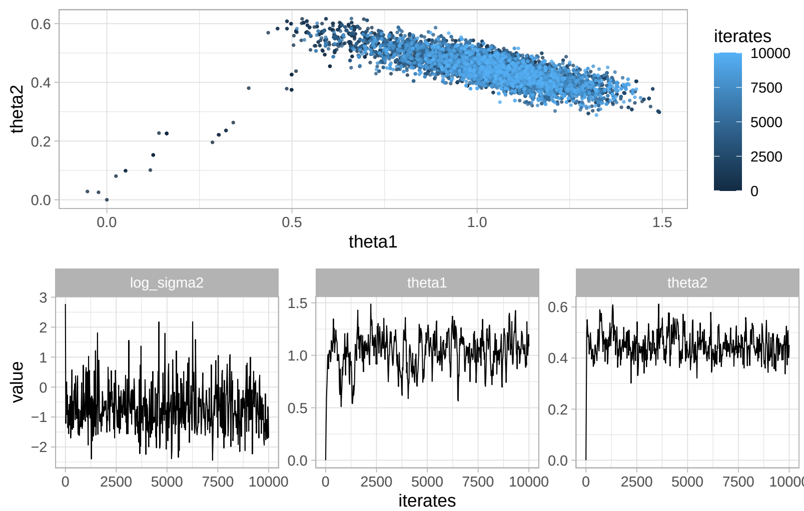


Plots of students may look slightly different of course, they may have initialised at the posterior mode to reduce burn-in.

(5) The update for $\tilde{\sigma}$ follows from

$$p(\tilde{\sigma}^2 \mid \theta, y) \propto p(\theta \mid \tilde{\sigma}^2) p(\tilde{\sigma}^2) \propto (\tilde{\sigma}^2)^{-p/2} e^{-\frac{\|\theta\|^2}{2\tilde{\sigma}^2}} (\tilde{\sigma}^2)^{-\alpha-1} e^{-\frac{\beta}{\tilde{\sigma}^2}}.$$

So the update step for $\tilde{\sigma}^2$ is a draw from the InvGamma$(\alpha + p/2, \beta + \frac{1}{2}\|\theta\|^2)$-distribution.


theta2
theta1
iterates
10000
7500
5000
2500
0
log_sigma2
theta1
theta2
value
iterates