Bertil Wegmann
Dept of Computer and Information Science
Linköping University

Bayesian Learning

Computer Lab 3

You are recommended to use R for solving the labs.
You work and submit your labs in pairs, but both of you should contribute equally and understand all parts of your solutions.
`It is not allowed to share exact solutions` with other student pairs.
The submitted lab reports will be verified through URKUND and indications of plagiarism will be investigated by the Disciplinary Board.
Submit your solutions via LISAM, no later than May 19 at 23:30.

---

1. **Gibbs sampler for a normal model**
   The dataset `rainfall.dat` consists of daily records, from the beginning of 1948 to the end of 1983, of precipitation (rain or snow in units of $\frac{1}{100}$ inch, and records of zero precipitation are excluded) at Snoqualmie Falls, Washington. Assume the natural log of the daily precipitation $\{y_1, ..., y_n\}$ are independent normally distributed, $\ln y_1, ..., \ln y_n | \mu, \sigma^2 \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$, where both μ and σ^2 are unknown. Let $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$ independently of $\sigma^2 \sim Inv\text{-}\chi^2(\nu_0, \sigma_0^2)$.

   (a) Implement (code!) a Gibbs sampler that simulates from the joint posterior $p(\mu, \sigma^2 | \ln y_1, ..., \ln y_n)$. The full conditional posteriors are given on the slides from Lecture 7. Evaluate the convergence of the Gibbs sampler by calculating the Inefficiency Factors (IFs) and plotting the trajectories of the sampled Markov chains.

   (b) Plot the following in one figure: 1) a histogram or kernel density estimate of the daily precipitation $\{y_1, ..., y_n\}$. 2) The resulting posterior predictive density $p(\tilde{y}|y_1, \ldots, y_n)$ using the simulated posterior draws from (a). How well does the posterior predictive density agree with this data?

2. **Metropolis Random Walk for Poisson regression**
   Consider the following Poisson regression model

$$y_i | \beta \sim \text{Poisson}\left[\exp\left(\mathbf{x}_i^T \beta\right)\right], \; i = 1, ..., n,$$

   where y_i is the count for the ith observation in the sample and x_i is the p-dimensional vector with covariate observations for the ith observation. Use the data set `eBayNumberOfBidderData.dat`. This dataset contains observations from 1000 eBay auctions of coins. The response variable is **nBids** and records the number of bids in each auction. The remaining variables are features/covariates ($\mathbf{x}$):

   - **Const** (for the intercept)

- **PowerSeller** (equal to 1 if the seller is selling large volumes on eBay)
- **VerifyID** (equal to 1 if the seller is a verified seller by eBay)
- **Sealed** (equal to 1 if the coin was sold in an unopened envelope)
- **MinBlem** (equal to 1 if the coin has a minor defect)
- **MajBlem** (equal to 1 if the coin has a major defect)
- **LargNeg** (equal to 1 if the seller received a lot of negative feedback from customers)
- **LogBook** (logarithm of the book value of the auctioned coin according to expert sellers. Standardized)
- **MinBidShare** (ratio of the minimum selling price (starting price) to the book value. Standardized).

(a) Obtain the maximum likelihood estimator of β in the Poisson regression model for the eBay data [Hint: `glm.R`, don't forget that `glm()` adds its own intercept so don't input the covariate Const]. Which covariates are significant?

(b) Let's do a Bayesian analysis of the Poisson regression. Let the prior be $\beta \sim \mathcal{N}\left[\mathbf{0}, 100 \cdot (\mathbf{X}^T\mathbf{X})^{-1}\right]$, where $\mathbf{X}$ is the $n \times p$ covariate matrix. This is a commonly used prior, which is called Zellner's g-prior. Assume first that the posterior density is approximately multivariate normal:

$$\beta|y \;\sim\; \mathcal{N}\left(\tilde{\beta}, J_{\mathbf{y}}^{-1}(\tilde{\beta})\right),$$

where $\tilde{\beta}$ is the posterior mode and $J_{\mathbf{y}}(\tilde{\beta})$ is the negative Hessian at the posterior mode. $\tilde{\beta}$ and $J_{\mathbf{y}}(\tilde{\beta})$ can be obtained by numerical optimization (`optim.R`) exactly like you already did for the logistic regression in Lab 2 (but with the log posterior function replaced by the corresponding one for the Poisson model, which you have to code up.).

(c) Let's simulate from the actual posterior of β using the Metropolis algorithm and compare the results with the approximate results in b). Program a general function that uses the Metropolis algorithm to generate random draws from an *arbitrary* posterior density. In order to show that it is a general function for any model, we denote the vector of model parameters by θ. Let the proposal density be the multivariate normal density mentioned in Lecture 8 (random walk Metropolis):

$$\theta_p|\theta^{(i-1)} \sim N\left(\theta^{(i-1)}, c \cdot \Sigma\right),$$

where $\Sigma = J_{\mathbf{y}}^{-1}(\tilde{\beta})$ was obtained in b). The value c is a tuning parameter and should be an input to your Metropolis function. The user of your Metropolis function should be able to supply her own posterior density function, not necessarily for the Poisson regression, and still be able to use your Metropolis function. This is not so straightforward, unless you have come across *function objects* in `R`. The note **HowToCodeRWM.pdf** in Lisam describes how you can do this in R.

Now, use your new Metropolis function to sample from the posterior of β in the Poisson regression for the eBay dataset. Assess MCMC convergence by graphical methods.

(d) Use the MCMC draws from c) to simulate from the predictive distribution of the number of bidders in a new auction with the characteristics below. Plot the predictive distribution. What is the probability of no bidders in this new auction?

- **PowerSeller** = 1
- **VerifyID** = 1
- **Sealed** = 1
- **MinBlem** = 0
- **MajBlem** = 1
- **LargNeg** = 0
- **LogBook** = 1
- **MinBidShare** = 0.7

3. **Time series models in Stan**

(a) Write a function in R that simulates data from the AR(1)-process

$$x_t = \mu + \phi\left(x_{t-1} - \mu\right) + \varepsilon_t, \;\; \varepsilon_t \overset{iid}{\sim} N\left(0, \sigma^2\right),$$

for given values of μ, ϕ and σ^2. Start the process at $x_1 = \mu$ and then simulate values for x_t for $t = 2, 3 \ldots, T$ and return the vector $x_{1:T}$ containing all time points. Use $\mu = 20, \sigma^2 = 4$ and $T = 200$ and look at some different realizations (simulations) of $x_{1:T}$ for values of ϕ between -1 and 1 (this is the interval of ϕ where the AR(1)-process is stationary). Include a plot of at least one realization in the report. What effect does the value of ϕ have on $x_{1:T}$?

(b) Use your function from a) to simulate two AR(1)-processes, $x_{1:T}$ with $\phi = 0.3$ and $y_{1:T}$ with $\phi = 0.9$. Now, treat your simulated vectors as synthetic data, and treat the values of μ, ϕ and σ^2 as unknown parameters. Implement Stan-code that samples from the posterior of the three parameters, using suitable non-informative priors of your choice. [Hint: Look at the time-series models examples in the Stan user's guide/reference manual, and note the different parameterization used here.]

i. Report the posterior mean, 95% credible intervals and the number of effective posterior samples for the three inferred parameters for each of the simulated AR(1)-process. Are you able to estimate the true values?

ii. For each of the two data sets, evaluate the convergence of the samplers and plot the joint posterior of μ and ϕ. Comments?

GOOD LUCK!
BEST, BERTIL