Homework 4

Senan Hogan-H.

12 April 2018

Consider observing $x_t \sim \text{Poisson}(\lambda)$ from times $t = 1, 2, \dots, N$, assuming that each x_t is independent. At some point in this time span, say $t = n$, the value of λ switches from some value λ_1 to λ_2. Our goal is to estimate both values of λ_i as well as the time point n at which the switch occurs.

First, assume independent gamma priors on λ_1 and λ_2, as well as a discrete uniform prior on n. \\[5pt]

So that the priors are as follows:

$$n \sim \mathcal{U}(1, \dots, n)$$

$$\lambda_1 \sim \Gamma(\alpha_1, \beta_1), \lambda_2 \sim \Gamma(\alpha_2, \beta_2)$$

$$x_i \sim Pois(\lambda_1), i \leqslant n; x_i \sim Pois(\lambda_2) i > n$$

Next, we are looking for the posterior distribution, $p(\lambda_1 \lambda_2, n|x)$. By the Bayes rule:

$$p(\lambda_1, \lambda_2, n|x) \propto p(x|\lambda_1, \lambda_2, n) p(\lambda_1) p(\lambda_2) p(n)$$

The likelihood here spilts to the cases before and after λ_i has changed:

$$p(x|\lambda_1, \lambda_2, n) = p(x_1, \dots, x_n|\lambda_1) p(x_{n+1}, \dots, x_N|\lambda_2)$$

Note that the draws are independent, so the likelihoods take the form:

$$p(x_1, \dots, x_n|\lambda_1) = \prod_{i=1}^{n} p(x_i|\lambda_1)$$
$$p(x_{n+1}, \dots, x_N|\lambda_2) = \prod_{i=n+1}^{N} p(x_i|\lambda_2)$$

And so the posterior takes the form:

$$p(\lambda_1, \lambda_2, n|x) \propto p(\lambda_1) p(\lambda_2) p(n) \prod_{i=1}^{n} p(x_i|\lambda_1) \prod_{i=n+1}^{N} p(x_i|\lambda_2)$$

Giving definitions of the appropriate distributions:

$$= \frac{\beta_1^{\alpha_1} \lambda_1^{\alpha_1 - 1} e^{-\beta_1 \lambda_1}}{\Gamma(\alpha_1)} \frac{\beta_2^{\alpha_2} \lambda_2^{\alpha_2 - 1} e^{-\beta_2 \lambda_2}}{\Gamma(\alpha_2)} \frac{1}{N} \prod_{i=1}^{n} \frac{\lambda_1^{x_i} e^{-\lambda_1}}{x_i!} \prod_{i=n+1}^{N} \frac{\lambda_2^{x_i} e^{-\lambda_2}}{x_i!}$$

The next step is to isolate the relevant posteriors on λ_1 and λ_2.

$$p(\lambda_1|\lambda_2, n, x) \propto \frac{\beta_1^{\alpha_1} \lambda_1^{\alpha_1 - 1} e^{-\beta_1 \lambda_1}}{\Gamma(\alpha_1)} \prod_{i=1}^{n} \frac{\lambda_1^{x_i} e^{-\lambda_1}}{x_i!}$$

$$= \frac{\beta_1^{\alpha_1}\lambda_1^{\alpha_1-1}\lambda_1^{\sum_{i=1}^n x_i}e^{-\beta_1\lambda_1}e^{-n\lambda_1}}{\Gamma(\alpha_1)\prod_{i=1}^n x_i!} \propto \lambda_1^{(\alpha_1+\sum_{i=1}^n x_i)-1}e^{-(\beta_1+n)\lambda_1}$$

Which is clearly the pdf of a Gamma($\alpha_1 + \sum_{i=1}^n x_i, \beta_1 + n$).

So that: $\lambda_1|\lambda_2, n, x \sim Gamma(\alpha_1 + \sum_{i=1}^n x_i, \beta_1 + n)$.

Similarly: $\lambda_2|\lambda_1, n, x \sim Gamma(\alpha_2 + \sum_{i=1}^n x_i, \beta_2 + (N - n))$.

The last part here to solve for the posterior conditional of n:

$$p(n|\lambda_1, \lambda_2, x) \propto \lambda_1^{(\alpha_1+\sum_{i=1}^n x_i)-1}\lambda_2^{(\alpha_2+\sum_{i=1}^n x_i)-1}e^{-(\beta_1+n)\lambda_1-(\beta_2-(N-n))\lambda_2}$$
$$\Rightarrow p(n|\lambda_1, \lambda_2, x) \propto \lambda_1^{\sum_{i=1}^n x_i}\lambda_2^{\sum_{i=n}^N x_i}e^{n(\lambda_2-\lambda_1)}$$

Gibbs Sampler

Suppose the distribution we are sampling from has $\lambda_1 = 2$, $\lambda_2 = 5$, with data observed where $n = 50$, $N = 100$.

```
lambda_1 <- 2
lambda_2 <- 5
N <- c(1:100)
n <- round(max(N)/2) # change point, halfway

# generate data
x <- c()
for (i in N){
  if (N[i] <= n){
    x_i <- rpois(1, lambda_1)
  }
  if (N[i] > n){
    x_i <- rpois(1, lambda_2)
  }
  x <- c(x, x_i)
}
```

Use a Gibbs sampler to obtain samples from the posterior. Suppose $\alpha_1 = \alpha_2 = 2\beta_1 = 2\beta_2 = 2$.

For convenience, consider the log tranformation of the conditional probability on n:

$$log(p(n|\lambda_1, \lambda_2, x)) =^+ \sum_{i=1}^n x_i log(\lambda_1) + \sum_{i=n}^N x_i log(\lambda_2) + n(\lambda_2 - \lambda_1)$$

```
alpha_1 <- alpha_2 <- 2 # define priors
beta_1 <- beta_2 <- 1 # define priors

lambda_1_chain <- lambda_2_chain <- n_chain <- c() # initialise chains
lambda_1_chain[1] <- lambda_2_chain[1] <-  5 # First values
n_chain[1] <- 10 # First values

p_n <- function(lambda_1_input, lambda_2_input){
  n_input <- c(1:length(x))
```

```r
  x <- c(x,0)
  y <- c()
  for (i in c(1:(length(x)-1))){
    sum1 <- sum(x[1:i])
    sum2 <- sum(x[(i+1):length(x)])
    y <- c(y, sum1*log(lambda_1_input) + sum2*log(lambda_2_input) -
      i*lambda_1_input - (length(x) - i )*lambda_2_input)
  }
  x <- x[-length(x)]
  y <- y - max(y)
  y <- exp(y)
  y <- y / sum(y)
  return(y)
}

for (i in 2:15000) {
  # print(i)
  lambda_1_chain[i] <- rgamma(1, alpha_1 + sum(x[1:n_chain[i-1]]),
                              beta_1 + n_chain[i-1])
  lambda_2_chain[i] <- rgamma(1, alpha_2 + sum(x[(n_chain[i-1]):length(x)]),
                              beta_2 + (length(x)-n_chain[i-1]))

  pr <- p_n(lambda_1_chain[i], lambda_2_chain[i])

  n_chain[i] <- sample(c(1:100), size = 1,
                       prob = pr)
}

##posterior inference
mean(lambda_1_chain)
```

```
## [1] 2.101
```

```r
quantile(lambda_1_chain, c(.025, .975))
```

```
##  2.5% 97.5%
## 1.721 2.507
```

```r
mean(lambda_2_chain)
```

```
## [1] 4.765
```

```r
quantile(lambda_2_chain, c(.025, .975))
```

```
##  2.5% 97.5%
## 4.091 5.461
```

```r
mean(n_chain)
```

```
## [1] 56.75
```

```r
quantile(n_chain, c(.025, .975))
```

```
##  2.5% 97.5%
##    47    60
```

```r
##posterior inference with time for burn-in
lambda_1_chain <- lambda_1_chain[
```

```r
  round(length(lambda_1_chain)/5) : length(lambda_1_chain)]
lambda_2_chain <- lambda_2_chain[
  round(length(lambda_2_chain)/5) : length(lambda_2_chain)]
n_chain <- n_chain[
  round(length(n_chain)/5) : length(n_chain)]

mean(lambda_1_chain)
```

```
## [1] 2.1
```

```r
quantile(lambda_1_chain, c(.025, .975))
```

```
##  2.5% 97.5%
## 1.718 2.506
```

```r
mean(lambda_2_chain)
```

```
## [1] 4.766
```

```r
quantile(lambda_2_chain, c(.025, .975))
```

```
##  2.5% 97.5%
## 4.089 5.461
```

```r
mean(n_chain)
```

```
## [1] 56.72
```

```r
quantile(n_chain, c(.025, .975))
```

```
##  2.5% 97.5%
##    47    60
```