STAT 447 Assignment 9

MCMC Hacking

Caden Hewlett

2024-04-04

Data Import

We import the primary data set inspired by Davidson-Pilon, (2013) below:

```
# main data vector
sms_data = c(
    13,24,8,24,7,35,14,11,15,11,22,22,11,57,11,19,29,6,19,
    12,22,12,18,72,32,9,7,13,19,23,27,20,6,17,13,10,14,6,
    16,15,7,2,15,15,19,70,49,7,53,22,21,31,19,11,18,20,12,
    35,17,23,17,4,2,31,30,13,27,0,39,37,5,14,13,22)
# data frame
df = data.frame(
  num_texts = sms_data,
  day = 1:length(sms_data)
)
```

Bayesian Model

We denote C to be the change point, selected uniformly from days $d \in \{1, 2, \dots N\}$ where N is the number of observations. Then, there is a likelihood for days less than change point C and a likelihood for days above the change point.

We can denote the model as follows:

$$\begin{aligned} \lambda_1 &\sim \exp(1/100) \\ \lambda_2 &\sim \exp(1/100) \\ C &\sim \text{unif}\big(\{1, 2, \dots, N\}\big) \\ Y_d \mid \{C, \lambda_1, \lambda_2\} &\sim \text{pois}\big(\mathbb{1}[d < C]\lambda_1 + \mathbb{1}[d \geq C]\lambda_2\big) \end{aligned}$$

We provide an implementation of the joint distribution of this model below.

```
# inputs are lambdas, C and y
log_joint = function(rates, change_point, y) {

  # Return log(0.0) if parameters are outside of the support
  if (rates[[1]] < 0 | rates[[2]] < 0 | change_point < 1 | change_point > length(y))
    return(-Inf)

  log_prior =
    dexp(rates[[1]], 1/100, log = TRUE) +
    dexp(rates[[2]], 1/100, log = TRUE)
```

```
  log_likelihood = 0.0
  for (i in 1:length(y)) {
    rate = if (i < change_point) rates[[1]] else rates[[2]]
    log_likelihood = log_likelihood + dpois(y[[i]], rate, log = TRUE)
  }

  return(log_prior + log_likelihood)
}
```

Question 1: A Custom MCMC Sampler

Part 1: Algorithm

Define mathematically an irreducible and invariant MCMC algorithm to sample from the change point model's posterior.

Part 2: π-Invariance

Prove that the MCMC algorithm you defined in part 1 is π-invariant.

Part 3: Implementation

Implement the MCMC algorithm you describe mathematically in `R`.

```
mcmc = function(rates, change_point, y, n_iterations) {
  change_point_trace = rep(-1, n_iterations)

  for (i in 1:n_iterations) {
    # TODO: implement a MCMC sampler
  }

  # Return:
  # - the trace of the change points (for question 1)
  # - the rates at the last iteration (for question 2)
  return(
    list(
      change_point_trace = change_point_trace,
      last_iteration_rates = rates
    )
  )
}
```