SWITCH POINT PROCESS MCMC

Inferring the number of switch points by inferring the switch point process without having to use a reversible jump chain algorithm

MORE BAYESIAN APPROACH

- Motivated by Lavielle (2001) paper
- Inference is done on the switch point process, r

$$r_i = \begin{cases} 1 & \text{if } \exists \text{ j st } t_i = \tau_j \\ 0 & \text{ow} \end{cases}.$$

- *Step 1*: Simulate just switch point process to obtain the posterior distribution of k.
    - Prior on switch point process:

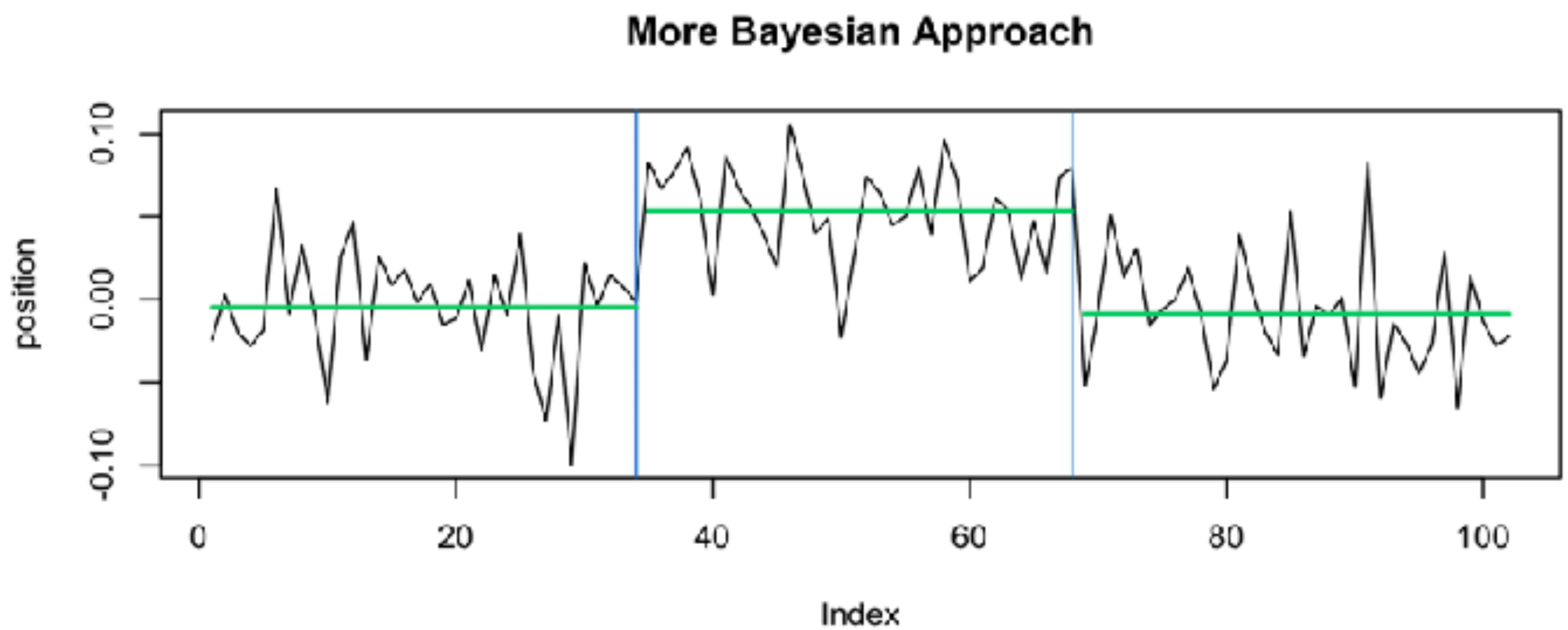
    $$r_i \sim Bernoulli(q) \quad \text{for } i = 2, \dots N - 1$$

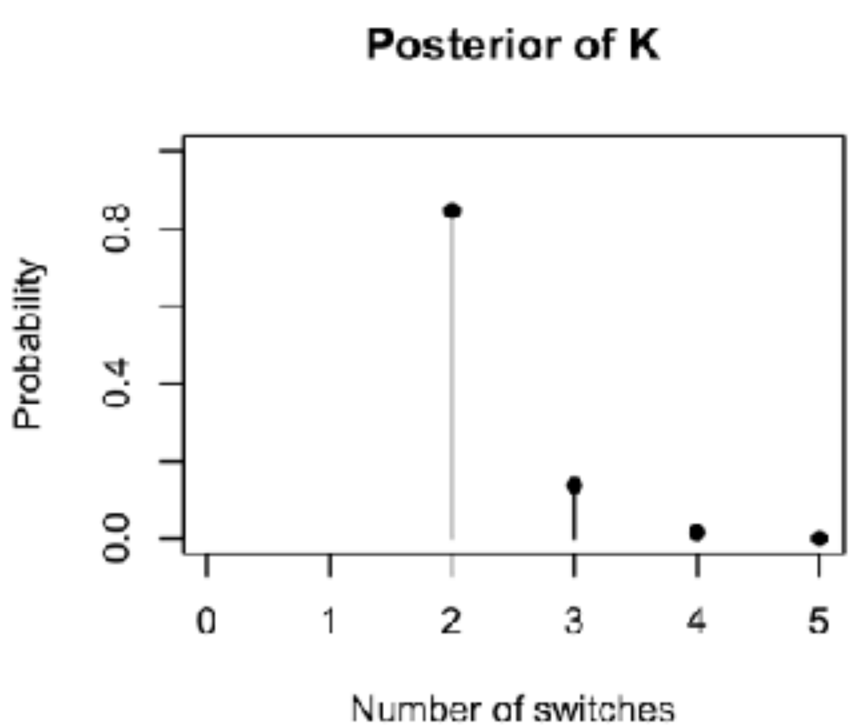
    - Non-informative priors for $\mu_1,\dots,\mu_{k+1}$, $1/\eta$
    - Three proposal for r: (1) Independent switch point process (2) Birth/death of a switch point (3)Position switch of current switch
- *Step* 2: Run our method for the MAP estimator of k.

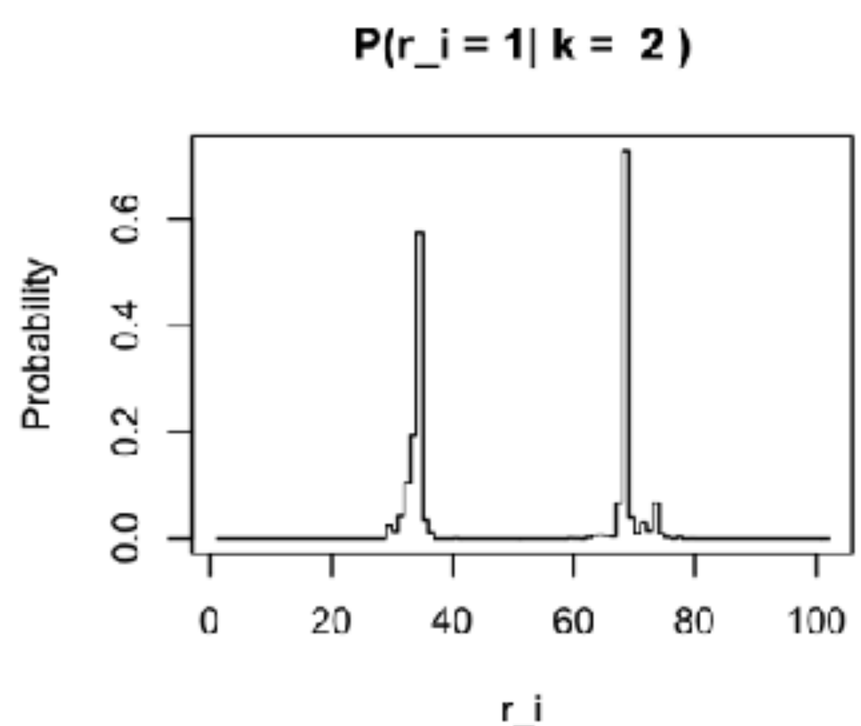
EXAMPLE OF STEP 1 RESULTS

$k = 2, N = 102$
$\mu_0 = 0, \mu_1 = 0.05, \mu_3 = 0$
$1/\eta = 0.001$

Hyper-parameter: $q = 3/102$, *for* $q < 12/102$ *yields the same MAP estimator for* τ







ALGORITHM DETAILS

Assumptions: (1)switch occurs at an observation time, (2) no minimum length of a state

$$\text{Model: } y_i \sim N(m_i, 1/\eta) \text{ for } \tau_{j-1} + 1 \leq i \leq \tau_j$$

$$L(y; r, \mu, \eta) = \left(\frac{\eta}{2\pi}\right)^{N/2} \sum_{j=1}^{K_r} \sum_{i=\tau_{j-1}+1}^{\tau_j} (y_i - \bar{y}_j)^2$$

$$K_r = \sum_{i=2}^{n-1} r_i + 1, \quad \bar{y}_j = \frac{1}{\tau_j - \tau_{j-1}} \sum_{i=\tau_{j-1}+1}^{\tau_j} y_i$$

$$\text{Priors: } \begin{cases} r \sim \text{Bernoulli}(q) & \textit{q is a hyperparameters} \\ p(\mu_i) = 1 & \text{for } i = 1, \ldots, k+1 \\ p(\eta) = 1/\eta \end{cases}$$

ALGORITHM DETAILS CONTINUE

Joint Posterior Distribution:

$$p(r, \mu, \eta; y) \stackrel{c}{=} L(y; r, \mu, \eta)p(r, \mu, \eta; q)$$

Marginal Posterior Distribution:

$$p(r; y) \stackrel{c}{=} \int_{\mathbb{R}^{K_r+1}} \int_{\mathbb{R}} L(y; r, \mu, \eta)p(r, \mu, \eta; q)d\eta d\mu$$

After some algebra:

$$p(r; y) \stackrel{c}{=} \left(\frac{q}{1-q}\right)^{K_r} \left(\prod_{j=1}^{K_r} n_j^{-1}\right) \pi^{K_r} S_r^{-(N-K_r)/2} \Gamma\left(\tfrac{N-K_r}{2}\right)$$

$$K_r = \sum_{i=2}^{N-1} r_i + 1, \quad n_j = \tau_j - \tau_{j-1}, \quad S_r = \sum_{j=1}^{K_r} \sum_{i=\tau_{j-1}+1}^{\tau_i} (y_i - \bar{y}_j)^2$$

PROPOSAL FOR SWITCH POINT PROCESS

Proposal 1: Independent draw $q(r_{\text{prop}}|r_{\text{cur}}) = q(r_{\text{prop}}) \sim \text{Bernoulli}(q)$

$$\alpha(r_{\text{prop}}, r_{\text{cur}}) = \min\left\{1, \frac{p(r_{\text{prop}}; y, q)p(r_{\text{cur}}; q)}{p(r_{\text{cur}}; y, q)p(r_{\text{prop}}; q)}\right\}$$

Proposal 2: Birth/Death of a switch point

$$s \sim Unif(\{2, \ldots, N-1\})$$

$$r_{\text{prop}} = \begin{cases} r_{\text{prop}} & \text{for } i \neq s \\ 1 - r_{\text{prop}} & \text{for } i = s \end{cases}$$

$$\alpha(r_{\text{prop}}, r_{\text{cur}}) = \min\left\{1, \frac{p(r_{\text{prop}}; y, q)}{p(r_{\text{cur}}; y, q)}\right\}$$

PROPOSAL FOR SWITCH POINT PROCESS

Proposal 3: Position Switch

$$s \sim Unif(\{\tau_1, \ldots, \tau_{K_r}\})$$

$$s' \sim Unif(\{2, \ldots, N-1\} \setminus \{\tau_1, \ldots, \tau_{K_r}\})$$

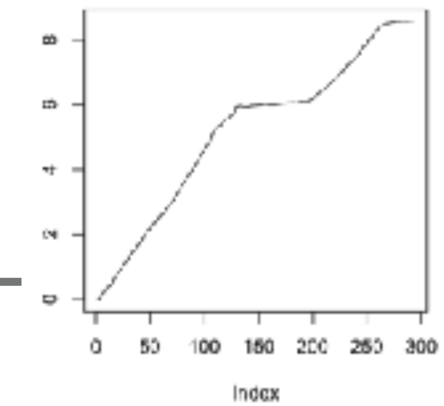
$$r_{\text{prop}} = \begin{cases} r_{\text{prop}} & \text{for } i \neq s, s' \\ 1 - r_{\text{prop}} & \text{for } i = s, s' \end{cases}$$

$$\alpha(r_{\text{prop}}, r_{\text{cur}}) = \min\left\{1, \frac{p(r_{\text{prop}}; y, q)}{p(r_{\text{cur}}; y, q)}\right\}$$

POINTS TO CONSIDER

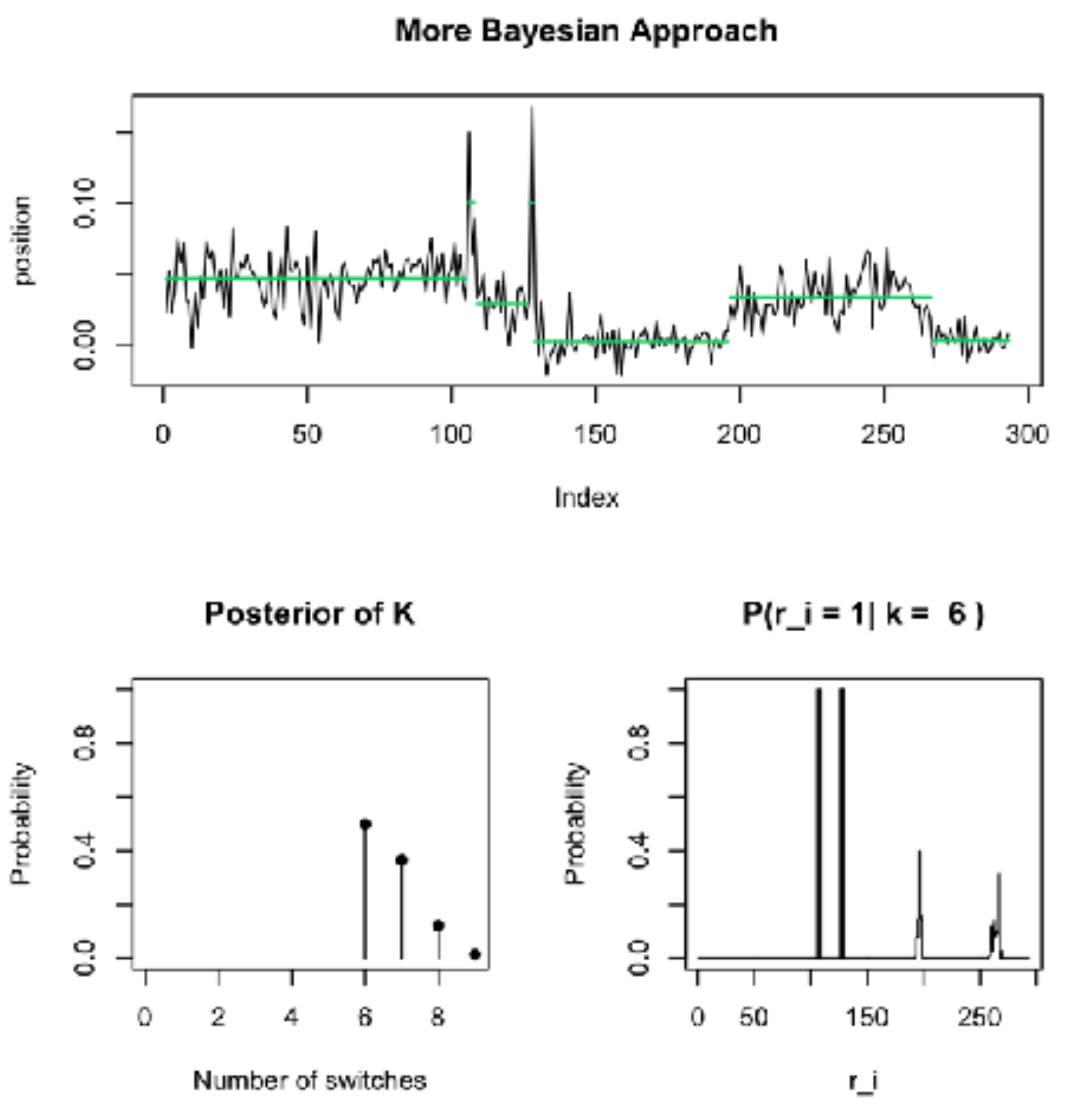
- Algorithm is sensitive to outliers (more common in experimental data)
    - *Solution 1*: impose a minimum duration for each state,
        - the switch point process can no longer be modeled as iid Bernoulli random variables
        - duration can affect the posterior distributions
    - *Solution 2:* Remove or smooth out the outliers
- Change points are discrete random variables
- Posterior distribution of K is not robust to changes in the hyper parameter q
    - Enforcing minimum state durations might help but also introduce more bias

SOLUTION 1: MINIMUM STATE DURATION



kin1-DDB particle 57

Minimum state duration:2



Minimum state duration: 10