Stat 535 C - Statistical Computing & Monte Carlo Methods

Lecture 24 - 6th April 2006

Arnaud Doucet

Email: `arnaud@cs.ubc.ca`

1.1– Outline

- Sequential Monte Carlo for Static Problems.
- Algorithms Settings.
- Applications.

2.1– Objectives

- Let $\{\pi_n\}_{n\geq 1}$ be a *sequence of probability distributions* defined on E such that $\pi_n(dx) = \pi_n(x)\,dx$ and each $\pi_n(x)$ is known *up to a normalizing constant*, i.e.

$$\pi_n(x) = \underbrace{Z_n^{-1}}_{\text{unknown}} . \underbrace{\gamma_n(x)}_{\text{known}} .$$

- Estimate expectations $\int \varphi(x)\,\pi_n(dx)$ and/or normalizing constants Z_n *sequentially*; i.e. first π_1 then π_2 and so on.
- *Objectives*: Obtain SMC (sampling/resampling population-based) algorithms to solve this problem.
- Standard SMC methods only apply to $\pi_n(x_{1:n}) = Z_n^{-1}\gamma_n(x_{1:n})$.

2.2– Examples

- *Sequential Bayesian Inference:* $\pi_n(x) = p(x|\,y_{1:n})$.
- *Global optimization:* $\pi_n(x) \propto [\pi(x)]^{\eta_n}$ with $\{\eta_n\}$ increasing sequence such that $\eta_n \to \infty$.
- *Sampling from a fixed target* π : $\pi_n(x) \propto [\mu_1(x)]^{\eta_n}\,[\pi(x)]^{1-\eta_n}$ where μ_1 easy to sample and $\eta_1 = 1$, $\eta_n < \eta_{n-1}$ and $\eta_P = 0$.

$\Rightarrow$ In all cases, we select π_1 easy to sample and $\pi_{n-1} \simeq \pi_n$.

2.5– How to use local moves

- Similarly at time n,
$$Z_n^{-1} w_n(x_n) = \frac{\pi_n(x_n)}{\mu_n(x_n)} \text{ IMPOSSIBLE so USE } Z_n^{-1} w_n(x_{1:n}) = \frac{\widetilde{\pi}_n(x_{1:n})}{\mu_n(x_{1:n})}$$
where $\{\widetilde{\pi}_n\}$ is defined using an *sequence of arbitrary backwards* Markov kernels $\{L_n\}$

$$\text{Artificial joint target} \quad : \quad \widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k),$$

$$\text{Joint importance distribution} \quad : \quad \mu_n(x_{1:n}) = \mu_1(x_1) \prod_{k=2}^{n} K_k(x_{k-1}, x_k).$$

- “Proof” of validity
$$\int \widetilde{\pi}_n(x_{1:n})\, dx_{1:n-1} = \pi_n(x_n) \underbrace{\int \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)\, dx_{1:n-1}}_{=1! \text{ whatever being } \{L_k\}} = \pi_n(x_n).$$

2.5– How to use local moves

- *Problem summary*: It is impossible to compute pointwise $\mu_n(x_n)$ hence $\gamma_n(x_n)/\mu_n(x_n)$ except when $n = 1$.

- *Solution*: Perform importance sampling on extended space.
  - At time 2,
$$\frac{\pi_2(x_2)}{\mu_2(x_2)} = \frac{\pi_2(x_2)}{\int \mu_1(dx_1) K_2(x_1, x_2)} \text{ cannot be evaluated}$$
but alternative weights can be defined
$$\frac{\text{new joint target distribution}}{\text{joint importance distribution}} = \frac{\pi_2(x_2) L_1(x_2, x_1)}{\mu_1(x_1) K_2(x_1, x_2)}$$
where $L_1(x_2, x_1)$ is an *arbitrary* (backward) Markov kernel.

- “Proof” of validity:
$$\int \pi_2(x_2) L_1(x_2, x_1)\, dx_1 = \pi_2(x_2) \underbrace{\int L_1(x_2, x_1)\, dx_1}_{=1! \text{ whatever being } L_1} = \pi_2(x_2)$$

2.4– What we propose to do

- At time n, we use μ_{n-1} to build μ_n using $X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right)$, i.e.

$$\begin{aligned} \mu_n(x_n) &= \int \mu_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)\, dx_{n-1} \\ &= \int \mu_1(x_1) \prod_{k=2}^{n} K_k(x_{k-1}, x_k)\, dx_{1:n-1} \end{aligned}$$

- A sensible approach consists of selecting K_n an MCMC kernel of invariant

distribution π_n or
approximate Gibbs move.

- It is typically impossible to compute $\mu_n(x)$ pointwise, hence the importance

weights.

2.3– Brief Review of Standard Importance Sampling

- Let the *target distribution* be $\pi_k(x) = Z_k^{-1} \gamma_k(x)$ and μ_k be a so-called *importance distribution* then
$$\begin{aligned} \pi_k(x) &= \frac{w_k(x)\, \mu_k(x)}{\int w_k(x)\, \mu_k(x)\, dx} \text{ where } w_k(x) = \frac{\gamma_k(x)}{\mu_k(x)}, \\ Z_k &= \int w_k(x)\, \mu_k(x)\, dx \end{aligned}$$

- By sampling N i.i.d. particles $X_k^{(i)} \sim \mu_k$ then $\widehat{\mu}_k(dx) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}}(dx)$ and

$$\begin{aligned} \widehat{\pi}_k(dx) &= \sum_{i=1}^{N} W_k^{(i)} \delta_{X_k^{(i)}}(dx) \text{ where } W_k^{(i)} \propto w_k\left(X_k^{(i)}\right), \ \sum_{i=1}^{N} W_k^{(i)} = 1, \\ \widehat{Z}_k &= \frac{1}{N} \sum_{i=1}^{N} w_k\left(X_k^{(i)}\right). \end{aligned}$$

3.1– How to select the backward Markov kernels

• **No free lunch**: By extending the integration space, the variance of the importance weights can only increase.

• The optimal kernel $\{L_{n-1}\}$ is the one bringing us back to the case where there is no space extension; i.e.

$$L_{n-1}^{\text{opt}}(x_n, x_{n-1}) = \frac{\mu_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}{\mu_n(x_n)}$$

• The result follows straightforwardly from the forward-backward formula for Markov processes

$$\mu_n(x_{1:n}) = \mu_1(x_1) \prod_{k=2}^{n} K_k(x_{k-1}, x_k) = \mu_n(x_n) \prod_{k=2}^{n} L_{k-1}^{\text{opt}}(x_k, x_{k-1})$$

• L_{n-1}^{opt} cannot typically be computed (though there are important exceptions) but can be properly approximated in numerous cases (see later). *Even if an approximation is used, the estimates are still asymptotically consistent.*

2.8– SMC Sampler Estimates

• Monte Carlo approximation

$$\widehat{\pi}_n(dx) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_n^{(i)}}(dx).$$

• Ratio of normalizing constants

$$\begin{aligned}
\frac{Z_n}{Z_{n-1}} &= \frac{\int \gamma_n(x_n)\, dx_n}{\int \gamma_{n-1}(x_{n-1})\, dx_{n-1}} \\
&= \int \frac{\gamma_n(x_n) L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)} \pi_{n-1}(dx_{n-1}) K_n(x_{n-1}, dx_n) \\
\Rightarrow \widehat{\frac{Z_n}{Z_{n-1}}} &= \sum_{i=1}^{N} W_{n-1}^{(i)} \frac{\gamma_n\left(X_n^{(i)}\right) L_{n-1}\left(X_n^{(i)}, X_{n-1}^{(i)}\right)}{\gamma_{n-1}\left(X_{n-1}^{(i)}\right) K_n\left(X_{n-1}^{(i)}, X_n^{(i)}\right)}.
\end{aligned}$$

2.7– SMC Sampler

Initialization; $n = 1$.

For $i = 1, ..., N$, sample $X_1^{(i)} \sim \mu_1(\cdot)$ and set

$$W_1^{(i)} \propto \frac{\pi_1\left(X_1^{(i)}\right)}{\mu_1\left(X_1^{(i)}\right)}.$$

Resample $\left\{W_1^{(i)}, X_1^{(i)}\right\}$ to obtain N new particles $\left\{N^{-1}, X_1^{(i)}\right\}$.

At time n; $n > 1$.

For $i = 1, ..., N$, sample $X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right)$ and set

$$W_n^{(i)} \propto \frac{\widetilde{\pi}_n\left(X_{1:n}^{(i)}\right)}{\mu_n\left(X_{1:n}^{(i)}\right)} \propto W_{n-1}^{(i)} \frac{\pi_n\left(X_n^{(i)}\right) L_{n-1}\left(X_n^{(i)}, X_{n-1}^{(i)}\right)}{\pi_{n-1}\left(X_{n-1}^{(i)}\right) K_n\left(X_{n-1}^{(i)}, X_n^{(i)}\right)}.$$

Resample $\left\{W_n^{(i)}, X_n^{(i)}\right\}$ to obtain N new particles $\left\{N^{-1}, X_n^{(i)}\right\}$.

2.6– Connections to standard SMC methods

• We are back to "standard" SMC methods where one is interested in sampling from a sequence of (artificial) distributions $\{\widetilde{\pi}_n\}$ whose dimension is increasing over time.

• **Key difference**: Given $\{K_n\}$, $\{\widetilde{\pi}_n\}$ has been constructed in a "clever" way such that

$$\int \widetilde{\pi}_n(x_{1:n})\, dx_{1:n-1} = \pi_n(x_n)$$

whereas usually the sequence of targets $\{\widetilde{\pi}_n\}$ is fixed and $\{K_n\}$ is designed accordingly.

4.2– Bayesian Analysis of Finite Mixture of Gaussians

- Model

$$Y_i \overset{\text{i.i.d.}}{\sim} \sum_{i=1}^{4} \omega_i \mathcal{N}(\mu_i, \lambda_i).$$

- Standard conjugate priors on $\theta = (\omega_{1:4}, \mu_{1:4}, \lambda_{1:4})$, no identifiability constraint

$$\mu_i \sim \mathcal{N}(\xi, \kappa^{-1}), \lambda_i \sim \mathcal{G}a(\nu, \chi),\ \omega_{1:4} \sim \mathcal{D}(\rho).$$

- The posterior is a mixture of 4! components

4.1– From MCMC to SMC

- **First step**: Build a sequence of distributions $\{\pi_n\}$ going from π_1 easy to sample/approximate to $\pi_P = \pi$; e.g. $\pi(x) \propto [\mu_1(x)]^{\eta_n} [\pi(x)]^{1-\eta_n}$ where μ_1 easy to sample and $\eta_1 = 1$, $\eta_n < \eta_{n-1}$ with $\eta_P = 0$.
- **Second step**: Introduce a sequence of transition kernels $\{K_n\}$; e.g. K_n MCMC sampler of invariant distribution π_n.
- **Third step**: Introduce a sequence of backward kernels $\{L_n\}$ equal/approximating L_n^{opt}; e.g.

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}{\int \pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1}}$$

$$\Rightarrow \alpha_n(x_{n-1}, x_n) = \frac{\pi_n(x_n)}{\int \pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1}}$$

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1}) K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \Rightarrow \alpha_n(x_{n-1}, x_n) = \frac{\pi_n(x_{n-1})}{\pi_{n-1}(x_{n-1})}$$

3.3– Be careful

- If the supports $S_n = \{x \in E : \pi_n(x) > 0\}$ are nested, i.e. $S_{n-1} \subset S_n$, then

you cannot

use $L_{n-1}(x_n, x_{n-1}) = \pi_n(x_{n-1}) K_n(x_{n-1}, x_n) / \pi_n(x_n)$ as

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1}) K_n(x_{n-1}, x_n)}{\int_{S_{n-1}} \pi_n(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1}}$$

but

$$\int_{S_{n-1}} \pi_n(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1} \neq \pi_n(x_n).$$

3.2– Approximations to Optimal Backward Kernels

- First approximation

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}{\int \pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1}}$$

$$\Rightarrow \frac{\pi_n(x_n) L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)} = \frac{\pi_n(x_n)}{\int \pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1}}$$

- Second approximation: If $K_n(x_{n-1}, x_n)$ is π_n-invariant

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1}) K_n(x_{n-1}, x_n)}{\pi_n(x_n)}$$

$$\Rightarrow \frac{\pi_n(x_n) L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)} = \frac{\pi_n(x_{n-1})}{\pi_{n-1}(x_{n-1})}.$$

4.3– SMC Estimates of conditional expectations of mean parameters

Sampler Details	Component			
	1	2	3	4
SMC (100 steps, 1 iteration)	0.68	0.91	2.02	2.14
SMC (100 steps, 10 iterations)	1.34	1.44	1.44	1.54
SMC (200 steps, 1 iteration)	1.11	1.29	1.39	1.98
SMC (200 steps, 10 iterations)	1.34	1.37	1.53	1.53
SMC (500 steps, 1 iteration)	0.98	1.38	1.54	1.87
SMC (500 steps, 10 iterations)	1.40	1.44	1.42	1.50

4.3– SMC Estimates of conditional expectations of mean parameters

4.2– Bayesian Analysis of Finite Mixture of Gaussians

- K_P admits as invariant distribution $\pi_P = \pi$. Very long runs of MCMC get trapped in one of the 4!=24 modes of the distributions.

- We select simply here for $L_{n-1}(x_n, x_{n-1})$ the reversal kernel

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1}) K_n(x_{n-1}, x_n)}{\pi_n(x_n)}.$$

- We ran SMC samplers with MCMC kernels for P =50, 100, 200 and 500 time steps with 1 and 10 MCMC iterations per time step.

4.2– Bayesian Analysis of Finite Mixture of Gaussians

- T =100 data with $M = 4$, with $\mu = (-3, 0, 3, 6)$, $\lambda = (0.55, 0.55, 0.55, 0.55)$; components "far" from each other.

- We build the sequence of P distributions

$$\pi_n(\theta) \propto l(y_{1:T}; \theta)^{\phi_n} f(\theta)$$

where $\phi_1 = 0 < \phi_2 < ... < \phi_P = 1$.

- MCMC sampler to sample from π_n
  - Update $\mu_{1:4}$ via a MH kernel with additive normal random walk.
  - Update $\lambda_{1:4}$ via a MH kernel with multiplicative log-normal random walk.
  - Update $\omega_{1:4}$ via a MH kernel with additive normal random walk on the logit scale.

4.7– Simulation settings

- We simulate T =200 data points with $r = 20$ covariates.
- We ran the MCMC sampler for 100000 iterations, thinning the samples to every 100.
- The CPU time was approximately 421 seconds.

4.6– Gibbs Sampling for Bayesian Probit Regression

- The Gibbs sampler to sample from $\pi\left(\beta, z_{1:T} \,|\, y_{1:T}\right)$ when $\pi\left(\beta\right) = \mathcal{N}\left(\beta, 0, \mathrm{diag}(100)\right)$ proceeds as follows

$$\beta|\cdots \sim \mathcal{N}_r(B, V),\ B = V(v^{-1}b + x'z),\ V = (v^{-1} + x'x)^{-1}$$

$$\pi(z_i|\cdots) \sim \begin{cases} \phi(z_i; x_i'\beta, 1)\mathbb{I}_{\{z_i>0\}}(z_i) & \text{if } y_i = 1 \\ \phi(z_i; x_i'\beta, 1)\mathbb{I}_{\{z_i\leq 0\}}(z_i) & \text{otherwise} \end{cases}$$

4.5– Bayesian Probit Regression

- We observed i.i.d. binary data $Y_1, \ldots, Y_T$, with associated $r-$dimensional covariates $X_1, \ldots, X_u$

$$\Pr\left(Y_i = 1|\beta\right) = \Phi(x_i'\beta))$$

where β is a $r-$dimensional vector and Φ is the standard normal CDF.

- To do Gibbs sampling, we introduce an auxiliary variable Z_i

$$\mathrm{Y}_i|Z_i = \begin{cases} 1 & \text{if } Z_i > 0 \\ 0 & \text{otherwise} \end{cases},\ Z_i = x_i'\beta + \epsilon_i$$

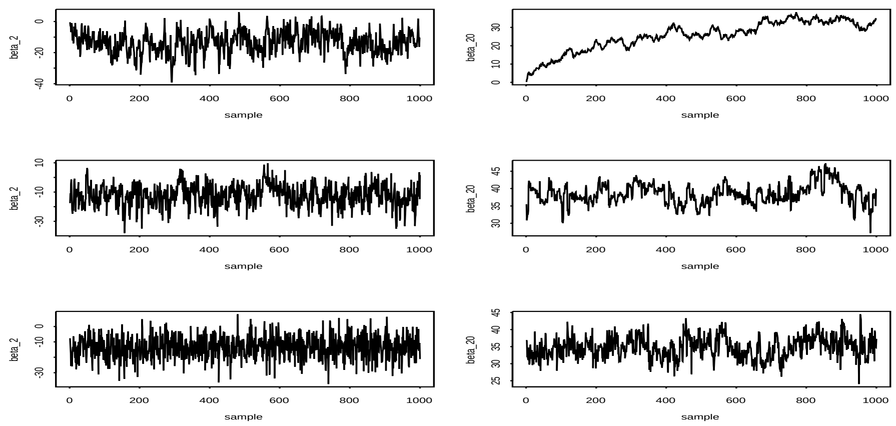
$$\epsilon_i \sim \mathcal{N}(0, 1).$$

4.4– Discussion

- With reasonable number of intermediate distributions and $N = 1000$, SMC manage to provide reasonable estimates of conditional expectations.
- For a fixed computational complexity, it outperforms very significantly MCMC.which gets stuck in a mode.
- Local MCMC kernels can be combined efficiently through SMC to explore more efficiently the space in a simple way.

4.8– Simulation Results: Traces of Regression Coefficients

1st row (Gibbs), 2nd row (SMC Reversal kernel), 3rd row (SMC Gibbs)



4.9– SMC Samplers Design

• We introduce an artificial sequence of targets through

$$\epsilon_i \sim \mathcal{N}(0, \zeta_n)$$

with $1 < \zeta_1 > \cdots > \zeta_P = 1$. This defines the targets $\pi_n\left(\beta, z_{1:T}\right)$ and $\pi_P\left(\beta, z_{1:T}\right) = \pi\left(\beta, z_{1:T}\right)$

• We sample the particles according to K_n which is the Gibbs sampler associated to π_n, i.e. we sample $\pi_n(z_{1:T}|\beta)$ then $\pi_n(\beta|z_{1:T})$.

• Similarly to the mixture model, the MCMC kernels are not mixing very well.

4.9– SMC Samplers Design

• For the auxiliary kernel associated to

$$K_n((z_{1:T}, \beta), (z'_{1:T}, \beta')) = \pi_n(z'_{1:T}|\beta)\pi_n(\beta'|z'_{1:T})$$

we can consider either the reversal kernel

$$L_{n-1}((z'_{1:T}, \beta'), (z_{1:T}, \beta)) = \frac{\pi_n(\beta, z_{1:T})\pi_n(z'_{1:T}|\beta)\pi_n(\beta'|z'_{1:T})}{\pi_n(\beta', z'_{1:T})}$$

or a better approximation of the optimal kernel

$$\begin{aligned} L_{n-1}((z'_{1:T}, \beta'), (z_{1:T}, \beta)) &= \frac{\pi_{n-1}(\beta, z_{1:T})\pi_n(z'_{1:T}|\beta)\pi_n(\beta'|z'_{1:T})}{\int \pi_{n-1}(\beta, z_{1:T})\pi_n(z'_{1:T}|\beta)\pi_n(\beta'|z'_{1:T})d\beta dz_{1:T}} \\ &= \frac{\pi_{n-1}(\beta, z_{1:T})\pi_n(z'_{1:T}|\beta)}{\int \pi_{n-1}(\beta)\pi_n(z'_{1:T}|\beta)d\beta} \end{aligned}$$

4.9– SMC Samplers Design

4.10– Sequential Bayesian Trans-dimensional Estimation

- We consider strictly increasing times $\{t_n\}$ and we have a sequence of distributions on spaces:

$$E_n = \bigcup_{k \in \mathbb{N}_0} \Big(\{k\} \times (\mathbb{R}^+)^{k+1} \times \Theta_{n,k} \Big).$$

- This is a sequence of *nested* trans-dimensional spaces; i.e. $E_{n-1} \subset E_n$.

- We use an "extend" move

$$K_n(x, dx') = \delta_{\tau_{1:k-1}, \lambda_{0:k}, k}(d(\tau'_{1:k-1}, \lambda'_{0:k}, k'))\pi_n(d\tau'_k | \tau_{1:k-1}, \lambda_{0:k}, k)$$

and a "birth" move: τ'_{k+1} from a uniform distribution on $[\tau_k, n)$

$$\pi_n\big((\tau'_{k+1}, \lambda_k), \lambda'_{k+1}\big) \propto (\lambda'_{k+1})^{n_{\tau'_{k+1}:n} + \lambda_k^2/\chi} \exp\big\{ - \lambda'_{k+1}[(n - \tau'_{k+1}) + \lambda_k/\chi]\big\}$$

and RJMCMC moves.

4.10– Sequential Bayesian Trans-dimensional Estimation

- At time t_n we are estimating $\lambda(u)$ over $[0, t_n]$. Over this interval the prior on the number k of changepoints follows a Poisson distribution of parameter νt_n

$$f_n(k) = e^{-\nu t_n} \frac{(\nu t_n)^k}{k!}$$

and, conditional on k, we have

$$f_n(\tau_{1:k}) = \frac{k!}{(t_n)^k} \mathbb{I}_{\Theta_{n,k}}(\tau_1, \ldots, \tau_k)$$

where $\Theta_{n,k} = \{\tau_{1:k} : 0 < \tau_1 < \cdots < \tau_k < t_n\}$. Thus at time t_n we have the density

$$\pi_n(\lambda_{0:k}, \tau_{1:k}, k) \propto l_n(y_{1:c_n} | \{\lambda(u)\}_{u \le t_n}) f(\lambda_0) \left[\prod_{j=1}^{k} f(\lambda_j | \lambda_{j-1}) \right] f_n(\tau_{1:k}) f_n(k).$$

4.10– Sequential Bayesian Trans-dimensional Estimation

- We record data $y_1, \ldots, y_{c_n}$ up to some time t_n with associated likelihood:

$$l_n(y_{1:c_n} | \{\lambda(u)\}_{u \le t_n}) \propto \left[\prod_{j=1}^{c_n} \lambda(y_j) \right] \exp\left\{ - \int_0^{t_n} \lambda(u) du \right\}.$$

- We adopt a piecewise constant function, defined for $u \le t_n$:

$$\lambda(u) = \sum_{j=0}^{k} \lambda_j \mathbb{I}_{[\tau_j, \tau_{j+1})}(u)$$

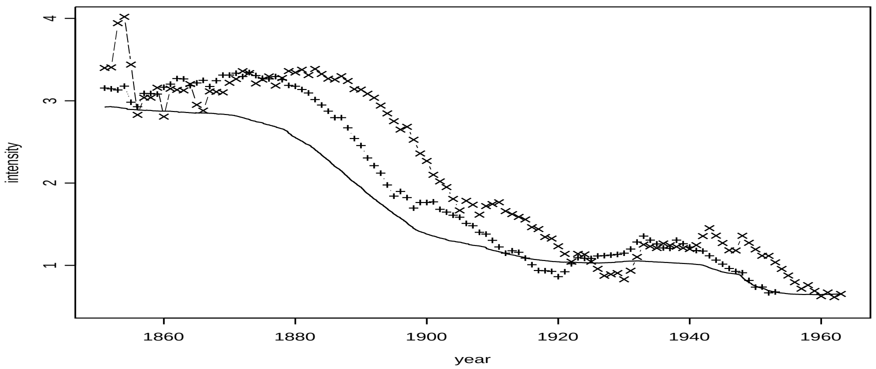
where $\tau_0 = 0$, $\tau_{k+1} = t_n$ and the changepoints (or knots) $\tau_{1:k}$ of the regression function follow a Poisson process of intensity ν whereas for any $k > 0$ $\lambda_0 \sim \mathcal{G}a(\mu, \upsilon)$ and $\lambda_j | \lambda_{j-1} \sim \mathcal{G}a(\lambda_{j-1}^2/\chi, \lambda_{j-1}/\chi)$.

4.9– SMC Samplers Design

Time points	**50**	**100**	**200**
CPU Time	115.33	251.70	681.33
CPU Time	118.93	263.61	677.65
# Times Resampled	29	29	28
# Times Resampled	7	6	8

Table 1: The first entry is for the reversal (i.e. the first column row entry is the reversal kernel for 50 time points). The CPU time is in seconds.

Estimate of $E\left[\lambda(t)|\, y_t\right]$, $E\left[\lambda(t)|\, y_{t+10}\right]$ and $E\left[\lambda(t)|\, y_T\right]$.



ESS over time