Non-reversible jump algorithms for Bayesian nested model selection

Philippe Gagnon and Arnaud Doucet

University of Oxford

January 8, 2020

1. Running example

Example: multiple change-point problems

- Context: coal mining disasters.

Example: multiple change-point problems

- Context: coal mining disasters.
- Data: times of occurrence of disasters.

Example: multiple change-point problems

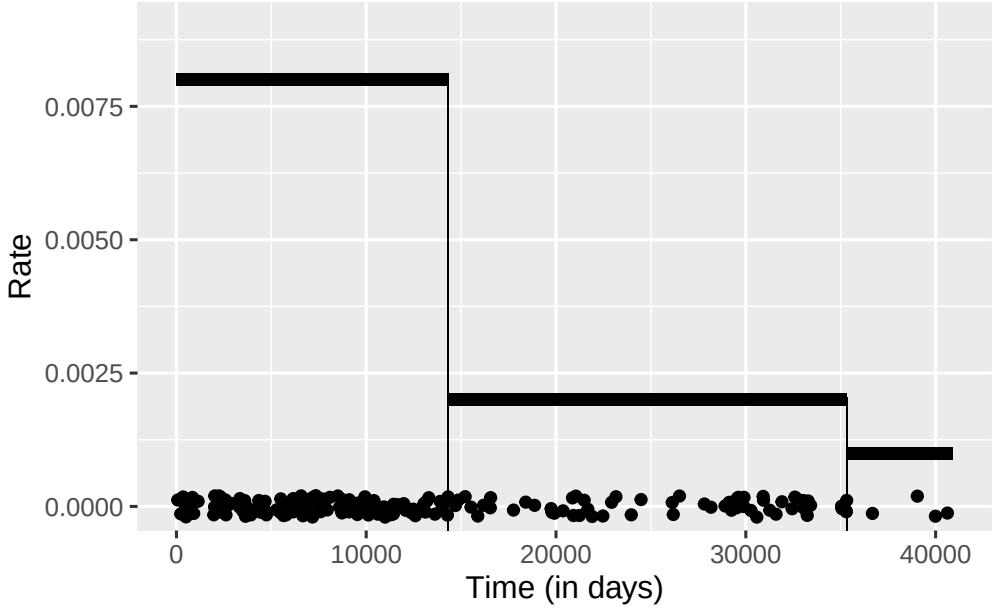
- Context: coal mining disasters.
- Data: times of occurrence of disasters.
- Model: non-homogeneous Poisson process.

Example: multiple change-point problems

- Context: coal mining disasters.
- Data: times of occurrence of disasters.
- Model: non-homogeneous Poisson process.
  - Assumption: the intensity is a step function.

Example: multiple change-point problems

- Context: coal mining disasters.
- Data: times of occurrence of disasters.
- Model: non-homogeneous Poisson process.
  - Assumption: the intensity is a step function.



MH/fixed-dimensional MCMC sampler

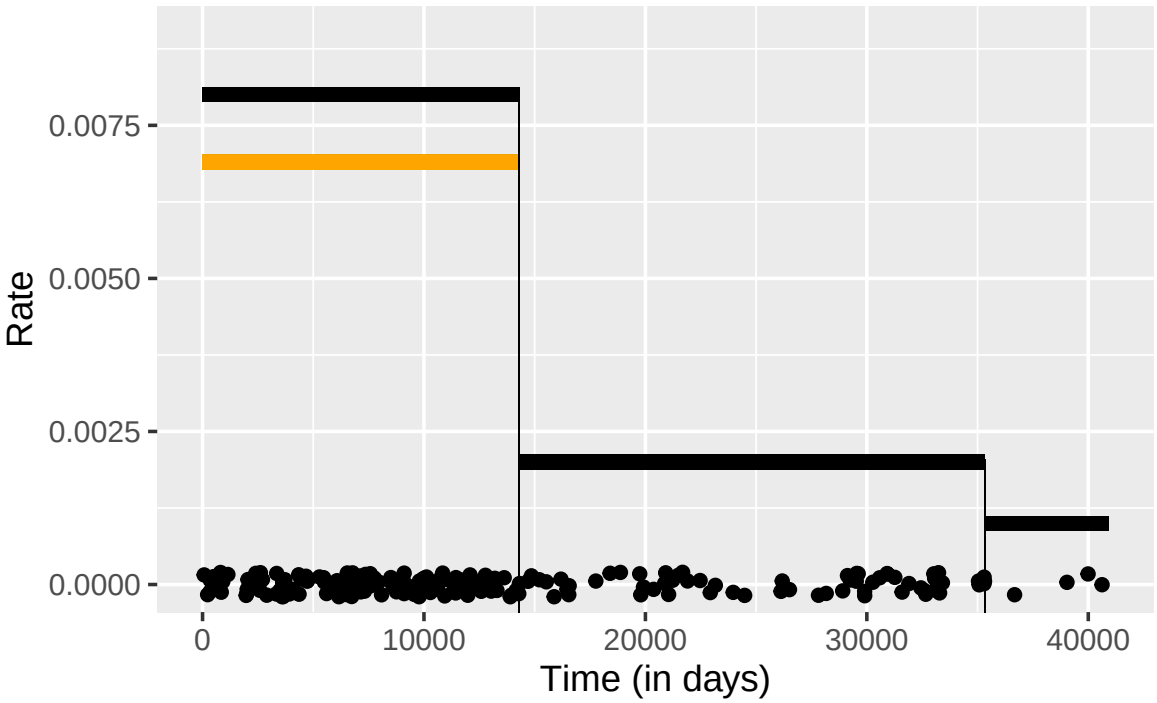


Figure 2: Proposal $\mathbf{y}$ with $\alpha(\mathbf{x}, \mathbf{y}) = 0.08$

Sampling context

- **Goal**: sample from a target distribution $\pi(k, \mathbf{x}_k)$,

Sampling context

- **Goal**: sample from a target distribution $\pi(k, \mathbf{x}_k)$,
  - $k \in \mathcal{K}$: model indicator,

Sampling context

- **Goal**: sample from a target distribution $\pi(k, \mathbf{x}_k)$,
  - $k \in \mathcal{K}$: model indicator,
  - $\mathbf{x}_k \in \mathbb{R}^{d_k}$: parameters of Model k,

Sampling context

- **Goal**: sample from a target distribution $\pi(k, \mathbf{x}_k)$,
  - $k \in \mathcal{K}$: model indicator,
  - $\mathbf{x}_k \in \mathbb{R}^{d_k}$: parameters of Model k,
  - d_k: number of parameters in Model k.

Reversible jump (RJ) algorithms

Reversible jump (RJ) algorithms

1. Generate $k' \sim g(k, \cdot)$.

Reversible jump (RJ) algorithms

1. Generate $k' \sim g(k, \cdot)$.
2. Generate $\mathbf{u}_{k \mapsto k'} \sim q_{k \mapsto k'}$.

Reversible jump (RJ) algorithms

1. Generate $k' \sim g(k, \cdot)$.
2. Generate $\mathbf{u}_{k \mapsto k'} \sim q_{k \mapsto k'}$.
3. Transform $(\mathbf{x}_k, \mathbf{u}_{k \mapsto k'})$: $T_{k \mapsto k'}(\mathbf{x}_k, \mathbf{u}_{k \mapsto k'}) := (\mathbf{y}_{k'}, \mathbf{u}_{k' \mapsto k})$. If

$$u_a \sim \mathcal{U}(0,1) \leq \alpha_{\mathrm{RJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'}))$$
$$:= 1 \wedge \frac{\pi(k', \mathbf{y}_{k'})\, g(k', k)\, q_{k' \mapsto k}(\mathbf{u}_{k' \mapsto k})}{\pi(k, \mathbf{x}_k)\, g(k, k')\, q_{k \mapsto k'}(\mathbf{u}_{k \mapsto k'})\, |J_{T_{k \mapsto k'}}(\mathbf{x}_k, \mathbf{u}_{k \mapsto k'})|^{-1}},$$

   set the next state of the chain to $(k', \mathbf{y}_{k'})$. Otherwise, set it to $(k, \mathbf{x}_k)$.
4. Go to Step 1.

Example (cont'd)

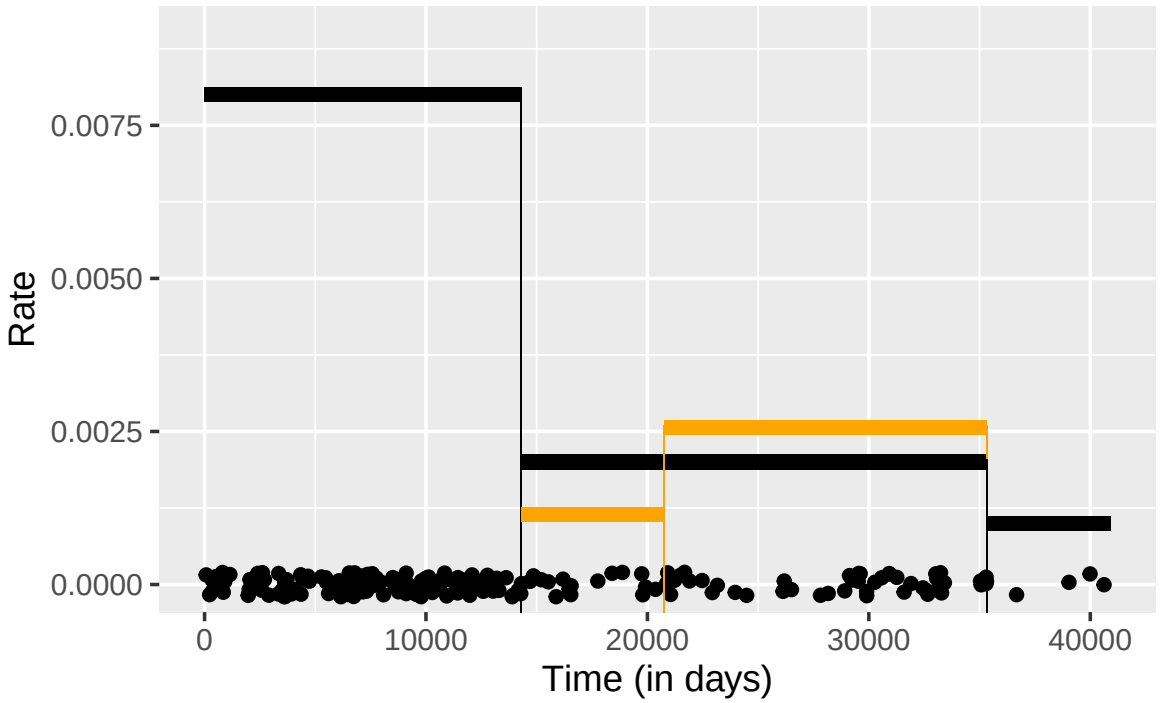


Figure 3: Proposal $(k', \mathbf{y}_{k'})$ with $\alpha_{\mathrm{RJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) = 1$

Nested models

- We focus on nested models.

Nested models

- We focus on nested models.
- K: ordinal discrete random variable reflecting the complexity of the models.

Nested models

- We focus on nested models.
- K: ordinal discrete random variable reflecting the complexity of the models.
- **Examples**:

Nested models

- We focus on nested models.
- K: ordinal discrete random variable reflecting the complexity of the models.
- **Examples**:
  1. number of change-points in multiple change-point problems,

Nested models

- We focus on nested models.
- K: ordinal discrete random variable reflecting the complexity of the models.
- **Examples**:
  1. number of change-points in multiple change-point problems,
  2. order of an autoregressive process,

Nested models

- We focus on nested models.
- K: ordinal discrete random variable reflecting the complexity of the models.
- **Examples**:
  1. number of change-points in multiple change-point problems,
  2. order of an autoregressive process,
  3. number of components in mixture modelling.

RJ Algorithms (Cont'd)

- Usually,

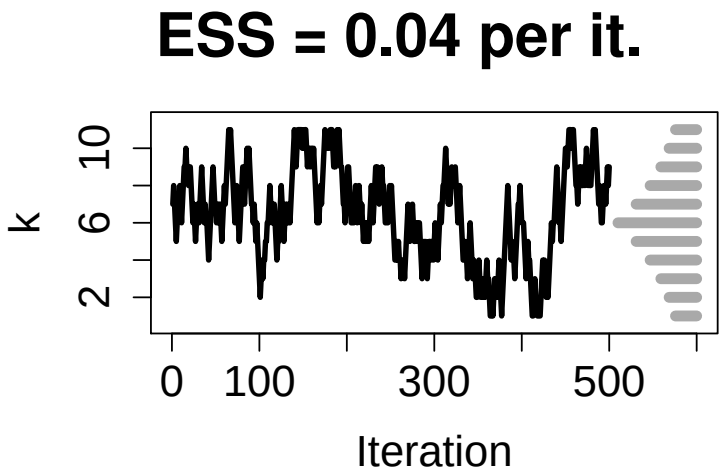
$$g(k,k') = \begin{cases} \tau & \text{if} \quad k' = k, \\ (1-\tau)/2 & \text{if} \quad k' = k-1 \text{ or } k+1. \end{cases}$$

RJ Algorithms (Cont'd)

- Usually,

$$g(k,k') = \begin{cases} \tau & \text{if} \quad k' = k, \\ (1-\tau)/2 & \text{if} \quad k' = k-1 \text{ or } k+1. \end{cases}$$

- **Remark**: the model space $\mathcal{K}$ is explored through a random walk.



Construction of non-reversible jump (NRJ) algorithms

1. Extend the state space: add the set $\{-1, +1\}$.

Construction of non-reversible jump (NRJ) algorithms

1. Extend the state space: add the set $\{-1, +1\}$.
   - Associate to it a direction variable $\nu \sim \mathcal{U}\{-1, +1\}$ (new target $= \pi \otimes \mathcal{U}\{-1, +1\}$).

Construction of non-reversible jump (NRJ) algorithms

1. Extend the state space: add the set $\{-1, +1\}$.
   - Associate to it a direction variable $\nu \sim \mathcal{U}\{-1, +1\}$ (new target $= \pi \otimes \mathcal{U}\{-1, +1\}$).
2. Set

$$g(k, k') := \begin{cases} \tau & \text{if} \quad k' = k, \\ 1 - \tau & \text{if} \quad k' = k + \nu. \end{cases}$$

Construction of non-reversible jump (NRJ) algorithms

1. Extend the state space: add the set $\{-1, +1\}$.
   - Associate to it a direction variable $\nu \sim \mathcal{U}\{-1, +1\}$ (new target $= \pi \otimes \mathcal{U}\{-1, +1\}$).
2. Set

$$g(k, k') := \begin{cases} \tau & \text{if} \quad k' = k, \\ 1 - \tau & \text{if} \quad k' = k + \nu. \end{cases}$$

3. Define the acceptance probability as:

$$\alpha_{\text{NRJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) := 1 \wedge \frac{\pi(k', \mathbf{y}_{k'})\, q_{k' \mapsto k}(\mathbf{u}_{k' \mapsto k})}{\pi(k, \mathbf{x}_k)\, q_{k \mapsto k'}(\mathbf{u}_{k \mapsto k'})\, |J_{T_{k \mapsto k'}}(\mathbf{x}_k, \mathbf{u}_{k \mapsto k'})|^{-1}}.$$

Construction of non-reversible jump (NRJ) algorithms

1. Extend the state space: add the set $\{-1, +1\}$.
   - Associate to it a direction variable $\nu \sim \mathcal{U}\{-1, +1\}$ (new target $= \pi \otimes \mathcal{U}\{-1, +1\}$).
2. Set

$$g(k, k') := \begin{cases} \tau & \text{if} \quad k' = k, \\ 1 - \tau & \text{if} \quad k' = k + \nu. \end{cases}$$

3. Define the acceptance probability as:

$$\alpha_{\text{NRJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) := 1 \wedge \frac{\pi(k', \mathbf{y}_{k'})\, q_{k' \mapsto k}(\mathbf{u}_{k' \mapsto k})}{\pi(k, \mathbf{x}_k)\, q_{k \mapsto k'}(\mathbf{u}_{k \mapsto k'})\, |J_{T_{k \mapsto k'}}(\mathbf{x}_k, \mathbf{u}_{k \mapsto k'})|^{-1}}.$$

4. Accepted proposals: set the next state to $(k', \mathbf{y}_{k'}, \nu)$.

Construction of non-reversible jump (NRJ) algorithms

1. Extend the state space: add the set $\{-1, +1\}$.
   - Associate to it a direction variable $\nu \sim \mathcal{U}\{-1, +1\}$ (new target $= \pi \otimes \mathcal{U}\{-1, +1\}$).
2. Set

$$g(k, k') := \begin{cases} \tau & \text{if} \quad k' = k, \\ 1 - \tau & \text{if} \quad k' = k + \nu. \end{cases}$$

3. Define the acceptance probability as:

$$\alpha_{\text{NRJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) := 1 \wedge \frac{\pi(k', \mathbf{y}_{k'})\, q_{k' \mapsto k}(\mathbf{u}_{k' \mapsto k})}{\pi(k, \mathbf{x}_k)\, q_{k \mapsto k'}(\mathbf{u}_{k \mapsto k'})\, |J_{T_{k \mapsto k'}}(\mathbf{x}_k, \mathbf{u}_{k \mapsto k'})|^{-1}}.$$

4. Accepted proposals: set the next state to $(k', \mathbf{y}_{k'}, \nu)$.
5. Rejected proposals: set the next state to $(k, \mathbf{x}_k, -\nu)$.

NRJ VS RJ

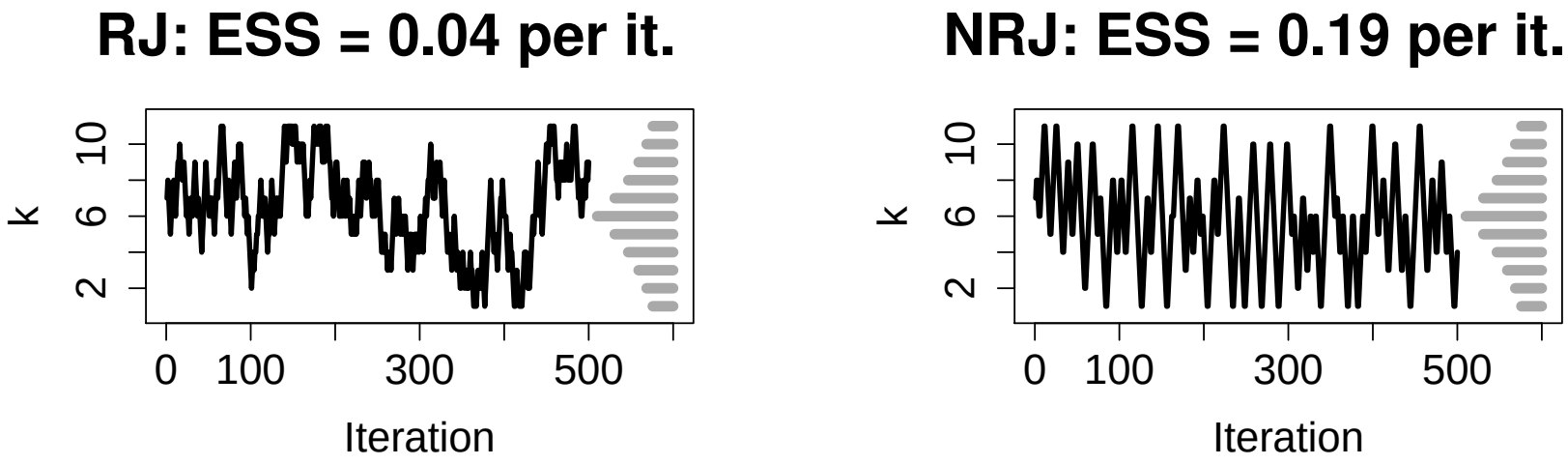


Figure 5: Trace plots for *ideal* RJ selecting uniformly at random which model to switch to and NRJ algorithms, and showing only the iterations in which model switches are proposed

Ideal NRJ

- Ideal situation: $\mathbf{y}_{k'} = \mathbf{u}_{k \mapsto k'} \sim q_{k \mapsto k'} = \pi(\cdot \mid k')$,

Ideal NRJ

- Ideal situation: $\mathbf{y}_{k'} = \mathbf{u}_{k \mapsto k'} \sim q_{k \mapsto k'} = \pi(\cdot \mid k')$,
  - implying that

$$\alpha_{\mathrm{NRJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) = 1 \wedge \frac{\pi(k')}{\pi(k)}.$$

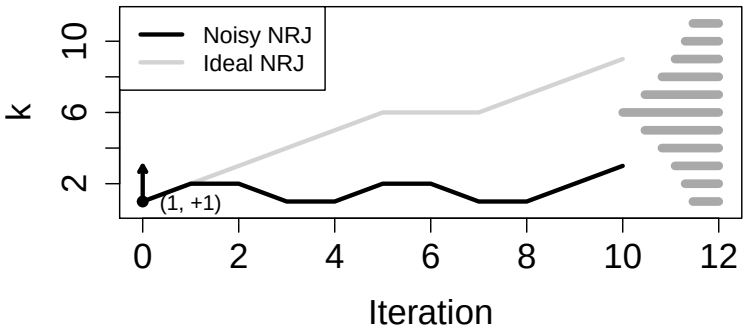
Ideal NRJ

- Ideal situation: $\mathbf{y}_{k'} = \mathbf{u}_{k \mapsto k'} \sim q_{k \mapsto k'} = \pi(\cdot \mid k')$,
  - implying that

$$\alpha_{\mathrm{NRJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) = 1 \wedge \frac{\pi(k')}{\pi(k)}.$$

- Typically:

$$\alpha_{\mathrm{NRJ}}((k, \mathbf{x}_k), (k', \mathbf{y}_{k'})) = 1 \wedge \frac{\pi(k')}{\pi(k)} \varepsilon(\mathbf{x}_k, \mathbf{y}_{k'}).$$



Towards ideal NRJ

- Methods allow to get as close as we want to the ideal NRJ:

Towards ideal NRJ

- Methods allow to get as close as we want to the ideal NRJ:
  - [Karagiannis and Andrieu, 2013] and [Andrieu et al., 2018].

Towards ideal NRJ

- Methods allow to get as close as we want to the ideal NRJ:
  - [Karagiannis and Andrieu, 2013] and [Andrieu et al., 2018].
- **How?**

Towards ideal NRJ

- Methods allow to get as close as we want to the ideal NRJ:
  - [Karagiannis and Andrieu, 2013] and [Andrieu et al., 2018].
- **How?**
  - Sophisticated proposal schemes.

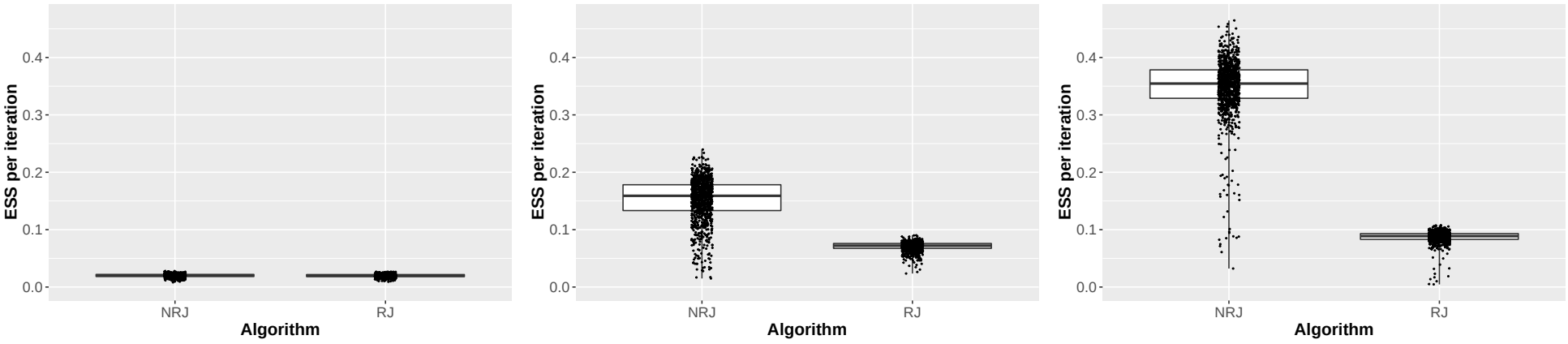
Towards ideal NRJ

- Methods allow to get as close as we want to the ideal NRJ:
  - [Karagiannis and Andrieu, 2013] and [Andrieu et al., 2018].
- **How?**
  - Sophisticated proposal schemes.
  - By-product: $\varepsilon \longrightarrow 1$.

Towards ideal NRJ

- Methods allow to get as close as we want to the ideal NRJ:
  - [Karagiannis and Andrieu, 2013] and [Andrieu et al., 2018].
- **How?**
  - Sophisticated proposal schemes.
  - By-product: $\varepsilon \longrightarrow 1$.
- **Fact:** Markov chains simulated by NRJ converge weakly to those produced by ideal NRJ.

Example (cont'd)



(a) Vanilla samplers **(b) In between** **(c) Ideal samplers**

Figure 8: ESS for NRJ and RJ when the samplers are: (a) vanilla samplers, (b) in between samplers, (c) ideal samplers

Thank you

- Thank you for your attention.

Thank you

- Thank you for your attention.
- Questions?

Andrieu, C., Doucet, A., Yıldırım, S., and Chopin, N. (2018).

On the utility of Metropolis–Hastings with asymmetric acceptance ratio.
arXiv:1803.09527.

Karagiannis, G. and Andrieu, C. (2013).
Annealed importance sampling reversible jump MCMC algorithms.
J. Comp. Graph. Stat., 22(3):623–648.