Bernoulli factory based Portkey and Flipped Portkey MCMC Algorithms: Theory and Examples

A. Bhattacharjee* D. Singh* D. Sangani* N. Garg* R. Gupta*

*Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

April 20, 2022

Contents

Motivation

- A search for Markov chain Monte Carlo or "MCMC" articles on Google Scholar yields about 3 lac hits! Why so popular?
- MCMC algorithms are widely used in Bayesian (posterior) inference while estimating expectations under highly complicated (possibly, high-dimensional) intractable distributions.
- Traditional MCMC algorithms, like the Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) use acceptance probabilities that explicitly use the ratio $\pi(y)/\pi(x)$, where $\pi(\cdot)$ is the target distribution known upto a normalizing constant.
- In problems where the functional form of the target is not known upto a normalizing constant, Barker's acceptance probability can be used (Gonçalves et al. (2017a); Gonçalves et al. (2017b)).
- Gonçalves et al. (2017a); Gonçalves et al. (2017b) proposed a Bernoulli factory based 2-coin algorithm, to avoid calculating the acceptance probability explicitly. Computationally burdensome!
- We present Bernoulli factory based algorithms based on a new acceptance probability that is π-invariant and demonstrate its potential via examples. (Vats et al. (2021))

Motivation

- A search for Markov chain Monte Carlo or "MCMC" articles on Google Scholar yields about 3 lac hits! Why so popular?
- MCMC algorithms are widely used in Bayesian (posterior) inference while estimating expectations under highly complicated (possibly, high-dimensional) intractable distributions.
- Traditional MCMC algorithms, like the Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) use acceptance probabilities that explicitly use the ratio $\pi(y)/\pi(x)$, where $\pi(\cdot)$ is the target distribution known upto a normalizing constant.
- In problems where the functional form of the target is not known upto a normalizing constant, Barker's acceptance probability can be used (Gonçalves et al. (2017a); Gonçalves et al. (2017b)).
- Gonçalves et al. (2017a); Gonçalves et al. (2017b) proposed a Bernoulli factory based 2-coin algorithm, to avoid calculating the acceptance probability explicitly. Computationally burdensome!
- We present Bernoulli factory based algorithms based on a new acceptance probability that is π-invariant and demonstrate its potential via examples. (Vats et al. (2021))

Motivation

- A search for Markov chain Monte Carlo or "MCMC" articles on Google Scholar yields about 3 lac hits! Why so popular?
- MCMC algorithms are widely used in Bayesian (posterior) inference while estimating expectations under highly complicated (possibly, high-dimensional) intractable distributions.
- Traditional MCMC algorithms, like the Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) use acceptance probabilities that explicitly use the ratio $\pi(y)/\pi(x)$, where $\pi(\cdot)$ is the target distribution known upto a normalizing constant.
- In problems where the functional form of the target is not known upto a normalizing constant, Barker's acceptance probability can be used (Gonçalves et al. (2017a); Gonçalves et al. (2017b)).
- Gonçalves et al. (2017a); Gonçalves et al. (2017b) proposed a Bernoulli factory based 2-coin algorithm, to avoid calculating the acceptance probability explicitly. Computationally burdensome!
- We present Bernoulli factory based algorithms based on a new acceptance probability that is π-invariant and demonstrate its potential via examples. (Vats et al. (2021))

Motivation

- A search for Markov chain Monte Carlo or "MCMC" articles on Google Scholar yields about 3 lac hits! Why so popular?
- MCMC algorithms are widely used in Bayesian (posterior) inference while estimating expectations under highly complicated (possibly, high-dimensional) intractable distributions.
- Traditional MCMC algorithms, like the Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) use acceptance probabilities that explicitly use the ratio $\pi(y)/\pi(x)$, where $\pi(\cdot)$ is the target distribution known upto a normalizing constant.
- In problems where the functional form of the target is not known upto a normalizing constant, Barker's acceptance probability can be used (Gonçalves et al. (2017a); Gonçalves et al. (2017b)).
- Gonçalves et al. (2017a); Gonçalves et al. (2017b) proposed a Bernoulli factory based 2-coin algorithm, to avoid calculating the acceptance probability explicitly. Computationally burdensome!
- We present Bernoulli factory based algorithms based on a new acceptance probability that is π-invariant and demonstrate its potential via examples. (Vats et al. (2021))

Motivation

- A search for Markov chain Monte Carlo or "MCMC" articles on Google Scholar yields about 3 lac hits! Why so popular?
- MCMC algorithms are widely used in Bayesian (posterior) inference while estimating expectations under highly complicated (possibly, high-dimensional) intractable distributions.
- Traditional MCMC algorithms, like the Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) use acceptance probabilities that explicitly use the ratio $\pi(y)/\pi(x)$, where $\pi(\cdot)$ is the target distribution known upto a normalizing constant.
- In problems where the functional form of the target is not known upto a normalizing constant, Barker's acceptance probability can be used (Gonçalves et al. (2017a); Gonçalves et al. (2017b)).
- Gonçalves et al. (2017a); Gonçalves et al. (2017b) proposed a Bernoulli factory based 2-coin algorithm, to avoid calculating the acceptance probability explicitly. Computationally burdensome!
- We present Bernoulli factory based algorithms based on a new acceptance probability that is π-invariant and demonstrate its potential via examples. (Vats et al. (2021))

Motivation

- A search for Markov chain Monte Carlo or "MCMC" articles on Google Scholar yields about 3 lac hits! Why so popular?
- MCMC algorithms are widely used in Bayesian (posterior) inference while estimating expectations under highly complicated (possibly, high-dimensional) intractable distributions.
- Traditional MCMC algorithms, like the Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) use acceptance probabilities that explicitly use the ratio $\pi(y)/\pi(x)$, where $\pi(\cdot)$ is the target distribution known upto a normalizing constant.
- In problems where the functional form of the target is not known upto a normalizing constant, Barker's acceptance probability can be used (Gonçalves et al. (2017a); Gonçalves et al. (2017b)).
- Gonçalves et al. (2017a); Gonçalves et al. (2017b) proposed a Bernoulli factory based 2-coin algorithm, to avoid calculating the acceptance probability explicitly. Computationally burdensome!
- We present Bernoulli factory based algorithms based on a new acceptance probability that is π-invariant and demonstrate its potential via examples. (Vats et al. (2021))

Basic Definitions

Definition (π-stationary)

Let P be a Markov transition kernel and X_0 be the starting value of the associated Markov chain. Let π be a distribution such that $X_0 \sim \pi$. P is said to be π-invariant if,

$$\pi P(A) = \int_{\mathscr{X}} P(x, A)\pi(dx) = \pi(A).$$

Definition (Bernoulli factory, Latuszynski (2010))

Let $p \in \mathscr{P} \subseteq [0, 1]$ be unknown and let $f : \mathscr{P} \to [0, 1]$. Then the problem known as the Bernoulli Factory is to generate Y, a single coin toss of an $s = f(p)-$coin, given a sequence $X_1, X_2, \ldots$ of independent tosses of a $p-$coin.

Basic Definitions

Definition (π-**stationary**)

Let P be a Markov transition kernel and X_0 be the starting value of the associated Markov chain. Let π be a distribution such that $X_0 \sim \pi$. P is said to be π-invariant if,

$$\pi P(A) = \int_{\mathscr{X}} P(x, A)\pi(dx) = \pi(A).$$

Definition (**Bernoulli factory**, Latuszynski (2010))

Let $p \in \mathscr{P} \subseteq [0,1]$ be unknown and let $f : \mathscr{P} \to [0,1]$. Then the problem known as the Bernoulli Factory is to generate Y, a single coin toss of an $s = f(p)-$coin, given a sequence $X_1, X_2, \ldots$ of independent tosses of a $p-$coin.

Barker's Acceptance Probability

- The Barker's (A. (1965)) acceptance probability, which is π-invariant is given by:

$$\alpha_B(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)},$$

where $q(y,x)$ is the proposal density at the point x, y being the currrent state ofthe Markov chain .

- It can be easily shown that:

$$\frac{\alpha_{MH}(x,y)}{2} \leq \alpha_B(x,y) \leq \alpha_{MH}(x,y),$$

where $\alpha_{MH} = \min\left(1, \frac{\pi(y)}{\pi(x)}\frac{q(y,x)}{q(x,y)}\right)$.

Theorem (Łatuszyński and Roberts (2013), Theorem 4(ii))

Let $f \in L^2(\pi)$ and denote the i.i.d. Monte Carlo variance by $\sigma_\pi^2 := Var_\pi(f)$. If a square root central limit theorem holds for f and the Metropolis–Hastings chain with the CLT asymptotic variance σ_{MH}^2 i.e.

$$\frac{\sum_{i=1}^N f(\theta_i) - \pi(f)}{\sqrt{N}} \Longrightarrow N(0, \sigma_{MH}^2),$$

then a corresponding CLT holds for f and the Barker chain with CLT asymptotic variance σ_B^2 satisfying

$$\sigma_{MH}^2 \leq \sigma_B^2 \leq 2\sigma_{MH}^2 + \sigma_\pi^2.$$

Barker's Acceptance Probability

- The Barker's (A. (1965)) acceptance probability, which is π-invariant is given by:

$$\alpha_B(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)},$$

where $q(y,x)$ is the proposal density at the point x, y being the currrent state ofthe Markov chain .

- It can be easily shown that:

$$\frac{\alpha_{MH}(x,y)}{2} \leq \alpha_B(x,y) \leq \alpha_{MH}(x,y),$$

where $\alpha_{MH} = \min\left(1, \frac{\pi(y)}{\pi(x)}\frac{q(y,x)}{q(x,y)}\right)$.

Theorem (Łatuszyński and Roberts (2013), Theorem 4(ii))

Let $f \in L^2(\pi)$ and denote the i.i.d. Monte Carlo variance by $\sigma_\pi^2 := Var_\pi(f)$. If a square root central limit theorem holds for f and the Metropolis–Hastings chain with the CLT asymptotic variance σ_{MH}^2 i.e.

$$\frac{\sum_{i=1}^N f(\theta_i) - \pi(f)}{\sqrt{N}} \implies N(0,\sigma_{MH}^2),$$

then a corresponding CLT holds for f and the Barker chain with CLT asymptotic variance σ_B^2 satisfying

$$\sigma_{MH}^2 \leq \sigma_B^2 \leq 2\sigma_{MH}^2 + \sigma_\pi^2.$$

Barker's Acceptance Probability

- The Barker's (A. (1965)) acceptance probability, which is π-invariant is given by:

$$\alpha_B(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)},$$

where $q(y,x)$ is the proposal density at the point x, y being the currrent state ofthe Markov chain .

- It can be easily shown that:

$$\frac{\alpha_{MH}(x,y)}{2} \leq \alpha_B(x,y) \leq \alpha_{MH}(x,y),$$

where $\alpha_{MH} = \min\left(1, \frac{\pi(y)}{\pi(x)}\frac{q(y,x)}{q(x,y)}\right)$.

Theorem (Łatuszyński and Roberts (2013), Theorem 4(ii))

Let $f \in L^2(\pi)$ and denote the i.i.d. Monte Carlo variance by $\sigma_\pi^2 := Var_\pi(f)$. If a square root central limit theorem holds for f and the Metropolis–Hastings chain with the CLT asymptotic variance σ_{MH}^2 i.e.

$$\frac{\sum_{i=1}^N f(\theta_i) - \pi(f)}{\sqrt{N}} \implies N(0,\sigma_{MH}^2),$$

then a corresponding CLT holds for f and the Barker chain with CLT asymptotic variance σ_B^2 satisfying

$$\sigma_{MH}^2 \leq \sigma_B^2 \leq 2\sigma_{MH}^2 + \sigma_\pi^2.$$

Barker's Method and the two-coin algorithm

Algorithm 1 Barker's MCMC for x_{m+1}

1: Draw $y \sim \mathrm{q}(x_m, y)$
2: Draw $A \sim \mathrm{Bern}(\alpha_B(x_m, y))$
3: **if** $A = 0$ **then**
4: $\quad x_{m+1} = x_m$
5: **if** $A = 1$ **then**
6: $\quad x_{m+1} = y$

Bernoulli factory to simulate events with probability $\alpha_B(x, y)$:

- Write $\pi(x)q(x, y) = c_x p_x$. Write similarly for y.
- We can get c_x and p_x as :

$$\pi(x)q(x,y) \leq c_x \text{ and then set } p_x = \pi(x)q(x,y)c_x^{-1}. \tag{1}$$

- Similarly, we can get c_y and p_y.
- The 2-coin algorithm (1) Gonçalves et al. (2017a) Gonçalves et al. (2017b) returns events with probability $\alpha_B(x, y)$ as given below.

$$h(p_x, p_y) = \frac{c_y p_y}{c_y p_y + c_x p_x} = \alpha_B(x, y).$$

Barker's Method and the two-coin algorithm

Algorithm 1 Barker's MCMC for x_{m+1}

Draw $y \sim \mathrm{q}(x_m, y)$
Draw $A \sim \mathrm{Bern}(\alpha_B(x_m, y))$
if $A = 0$ **then**
    $x_{m+1} = x_m$
if $A = 1$ **then**
    $x_{m+1} = y$

Bernoulli factory to simulate events with probability $\alpha_B(x, y)$:

- Write $\pi(x)q(x, y) = c_x p_x$. Write similarly for y.
- We can get c_x and p_x as :

$$\pi(x)q(x,y) \leq c_x \text{ and then set } p_x = \pi(x)q(x,y)c_x^{-1}. \tag{1}$$

- Similarly, we can get c_y and p_y.
- The 2-coin algorithm (1) Gonçalves et al. (2017a) Gonçalves et al. (2017b) returns events with probability $\alpha_B(x, y)$ as given below.

$$h(p_x, p_y) = \frac{c_y p_y}{c_y p_y + c_x p_x} = \alpha_B(x, y).$$

Barker's Method and the two-coin algorithm

Algorithm 1 Barker's MCMC for x_{m+1}

1: Draw $y \sim \mathrm{q}(x_m, y)$
2: Draw $A \sim \mathrm{Bern}(\alpha_B(x_m, y))$
3: **if** $A = 0$ **then**
4: $\quad x_{m+1} = x_m$
5: **if** $A = 1$ **then**
6: $\quad x_{m+1} = y$

Bernoulli factory to simulate events with probability $\alpha_B(x, y)$:

- Write $\pi(x)q(x, y) = c_x p_x$. Write similarly for y.
- We can get c_x and p_x as :

$$\pi(x)q(x,y) \leq c_x \text{ and then set } p_x = \pi(x)q(x,y)c_x^{-1}. \tag{1}$$

- Similarly, we can get c_y and p_y.
- The 2-coin algorithm (1) Gonçalves et al. (2017a) Gonçalves et al. (2017b) returns events with probability $\alpha_B(x, y)$ as given below.

$$h(p_x, p_y) = \frac{c_y p_y}{c_y p_y + c_x p_x} = \alpha_B(x, y).$$

Barker's Method and the two-coin algorithm

Algorithm 1 Barker's MCMC for x_{m+1}

1: Draw $y \sim \mathrm{q}(x_m, y)$
2: Draw $A \sim \mathrm{Bern}(\alpha_B(x_m, y))$
3: **if** $A = 0$ **then**
4: $\quad x_{m+1} = x_m$
5: **if** $A = 1$ **then**
6: $\quad x_{m+1} = y$

Bernoulli factory to simulate events with probability $\alpha_B(x, y)$:

- Write $\pi(x)q(x, y) = c_x p_x$. Write similarly for y.
- We can get c_x and p_x as :

$$\pi(x)q(x, y) \leq c_x \text{ and then set } p_x = \pi(x)q(x, y)c_x^{-1}. \tag{1}$$

- Similarly, we can get c_y and p_y.
- The 2-coin algorithm (1) Gonçalves et al. (2017a) Gonçalves et al. (2017b) returns events with probability $\alpha_B(x, y)$ as given below.

$$h(p_x, p_y) = \frac{c_y p_y}{c_y p_y + c_x p_x} = \alpha_B(x, y).$$

Barker's Method and the two-coin algorithm

Algorithm 1 Barker's MCMC for x_{m+1}

1: Draw $y \sim \mathrm{q}(x_m, y)$
2: Draw $A \sim \mathrm{Bern}(\alpha_B(x_m, y))$
3: **if** $A = 0$ **then**
4: $\quad x_{m+1} = x_m$
5: **if** $A = 1$ **then**
6: $\quad x_{m+1} = y$

Bernoulli factory to simulate events with probability $\alpha_B(x, y)$:

- Write $\pi(x)q(x, y) = c_x p_x$. Write similarly for y.
- We can get c_x and p_x as :

$$\pi(x)q(x, y) \leq c_x \text{ and then set } p_x = \pi(x)q(x, y)c_x^{-1}. \tag{1}$$

- Similarly, we can get c_y and p_y.
- The 2-coin algorithm (1) Gonçalves et al. (2017a) Gonçalves et al. (2017b) returns events with probability $\alpha_B(x, y)$ as given below.

$$h(p_x, p_y) = \frac{c_y p_y}{c_y p_y + c_x p_x} = \alpha_B(x, y).$$

Barker's Method and the two-coin algorithm

Algorithm 1 The 2-coin algorithm for $\alpha_B(x, y)$

1: Draw $C_1 \sim Ber\left(\frac{c_y}{c_x + c_y}\right)$.
2: **if** $C_1 = 1$ **then**
3:     Draw $C_2 \sim Ber(p_y)$.
4:     **if** $C_2 = 1$ **then**
5:         output 1.
6:     **else**
7:         go to Step 1.
8: **if** $C_1 = 0$ **then**
9:     Draw $C_2 \sim Ber(p_x)$.
10:     **if** $C_2 = 1$ **then**
11:         output 0.
12:     **else**
13:         go to Step 1.

- The number of loops of Algorithm (1) $\sim Geom((c_y p_y + c_x p_x)/(c_y + c_x))$.
- Mean execution time is :

$$\frac{c_x + c_y}{c_x p_x + c_y p_y} = \frac{c_x + c_y}{\pi(x)q(x,y) + \pi(y)q(x,y)}.$$

Barker's Method and the two-coin algorithm

Algorithm 1 The 2-coin algorithm for $\alpha_B(x, y)$

1: Draw $C_1 \sim Ber\left(\frac{c_y}{c_x + c_y}\right)$.
2: **if** $C_1 = 1$ **then**
3:     Draw $C_2 \sim Ber(p_y)$.
4:     **if** $C_2 = 1$ **then**
5:         output 1.
6:     **else**
7:         go to Step 1.
8: **if** $C_1 = 0$ **then**
9:     Draw $C_2 \sim Ber(p_x)$.
10:     **if** $C_2 = 1$ **then**
11:         output 0.
12:     **else**
13:         go to Step 1.

- The number of loops of Algorithm (1) $\sim Geom((c_y p_y + c_x p_x)/(c_y + c_x))$.
- Mean execution time is :

$$\frac{c_x + c_y}{c_x p_x + c_y p_y} = \frac{c_x + c_y}{\pi(x)q(x,y) + \pi(y)q(x,y)}.$$

Barker's Method and the two-coin algorithm

Algorithm 1 The 2-coin algorithm for $\alpha_B(x, y)$

1: Draw $C_1 \sim Ber\left(\frac{c_y}{c_x + c_y}\right)$.
2: **if** $C_1 = 1$ **then**
3:     Draw $C_2 \sim Ber(p_y)$.
4:     **if** $C_2 = 1$ **then**
5:         output 1.
6:     **else**
7:         go to Step 1.
8: **if** $C_1 = 0$ **then**
9:     Draw $C_2 \sim Ber(p_x)$.
10:     **if** $C_2 = 1$ **then**
11:         output 0.
12:     **else**
13:         go to Step 1.

- The number of loops of Algorithm (1) $\sim Geom((c_y p_y + c_x p_x)/(c_y + c_x))$.
- Mean execution time is :

$$\frac{c_x + c_y}{c_x p_x + c_y p_y} = \frac{c_x + c_y}{\pi(x)q(x,y) + \pi(y)q(x,y)}.$$

Portkey Barker's Method

- Consider the following π-invariant acceptance probability:

$$\alpha_{(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)+\frac{(1-\beta)}{\beta}(c_x+c_y)},$$

with c_x and c_y as defined previously.

- As $\alpha_{(\beta)}(x,y) \leq \alpha_B(x,y)$, Barker's method is more efficient by Peskun's ordering (Peskun (1973)). Hence, we choose $\beta \approx 1$ and events with probability $\alpha_{(\beta)}$ can be generated using Algorithm (1).
- The number of loops until the algorithm stops $\sim$Geom(s_β), where

$$s_\beta = (1-\beta)+\beta \cdot \frac{c_y p_y + c_x p_x}{c_x+c_y}.$$

- Ratio of mean execution time of the two-coin algorithm to the portkey two-coin algorithm is:

$$\frac{1/s_1}{1/s_\beta} = \frac{s_\beta}{s_1} = (1-\beta)\cdot\left(\frac{c_y p_y + c_x p_x}{c_x+c_y}\right)^{-1} + \beta \geq 1.$$

- Computational efficiency gained here is at the cost of statistical efficiency.

Portkey Barker's Method

- Consider the following π-invariant acceptance probability:

$$\alpha_{(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)+\dfrac{(1-\beta)}{\beta}(c_x+c_y)},$$

  with c_x and c_y as defined previously.
- As $\alpha_{(\beta)}(x,y) \leq \alpha_B(x,y)$, Barker's method is more efficient by Peskun's ordering (Peskun (1973)). Hence, we choose $\beta \approx 1$ and events with probability $\alpha_{(\beta)}$ can be generated using Algorithm (1).
- The number of loops until the algorithm stops $\sim$Geom(s_β), where

$$s_\beta = (1-\beta)+\beta \cdot \frac{c_y p_y + c_x p_x}{c_x+c_y}.$$

- Ratio of mean execution time of the two-coin algorithm to the portkey two-coin algorithm is:

$$\frac{1/s_1}{1/s_\beta} = \frac{s_\beta}{s_1} = (1-\beta)\cdot\left(\frac{c_y p_y + c_x p_x}{c_x+c_y}\right)^{-1} + \beta \geq 1.$$

- Computational efficiency gained here is at the cost of statistical efficiency.

Portkey Barker's Method

- Consider the following π-invariant acceptance probability:

$$\alpha_{(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)+\frac{(1-\beta)}{\beta}(c_x+c_y)},$$

  with c_x and c_y as defined previously.
- As $\alpha_{(\beta)}(x,y) \leq \alpha_B(x,y)$, Barker's method is more efficient by Peskun's ordering (Peskun (1973)). Hence, we choose $\beta \approx 1$ and events with probability $\alpha_{(\beta)}$ can be generated using Algorithm (1).
- The number of loops until the algorithm stops $\sim$Geom(s_β), where

$$s_\beta = (1-\beta)+\beta \cdot \frac{c_y p_y + c_x p_x}{c_x + c_y}.$$

- Ratio of mean execution time of the two-coin algorithm to the portkey two-coin algorithm is:

$$\frac{1/s_1}{1/s_\beta} = \frac{s_\beta}{s_1} = (1-\beta)\cdot\left(\frac{c_y p_y + c_x p_x}{c_x+c_y}\right)^{-1} + \beta \geq 1.$$

- Computational efficiency gained here is at the cost of statistical efficiency.

Portkey Barker's Method

- Consider the following π-invariant acceptance probability:

$$\alpha_{(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)+\frac{(1-\beta)}{\beta}(c_x+c_y)},$$

  with c_x and c_y as defined previously.
- As $\alpha_{(\beta)}(x,y) \leq \alpha_B(x,y)$, Barker's method is more efficient by Peskun's ordering (Peskun (1973)). Hence, we choose $\beta \approx 1$ and events with probability $\alpha_{(\beta)}$ can be generated using Algorithm (1).
- The number of loops until the algorithm stops $\sim$Geom(s_β), where

$$s_\beta = (1-\beta)+\beta \cdot \frac{c_y p_y + c_x p_x}{c_x + c_y}.$$

- Ratio of mean execution time of the two-coin algorithm to the portkey two-coin algorithm is:

$$\frac{1/s_1}{1/s_\beta} = \frac{s_\beta}{s_1} = (1-\beta)\cdot\left(\frac{c_y p_y + c_x p_x}{c_x + c_y}\right)^{-1} + \beta \geq 1.$$

- Computational efficiency gained here is at the cost of statistical efficiency.

Portkey Barker's Method

- Consider the following π-invariant acceptance probability:

$$\alpha_{(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x)+\pi(x)q(x,y)+\frac{(1-\beta)}{\beta}(c_x+c_y)},$$

  with c_x and c_y as defined previously.
- As $\alpha_{(\beta)}(x,y) \leq \alpha_B(x,y)$, Barker's method is more efficient by Peskun's ordering (Peskun (1973)). Hence, we choose $\beta \approx 1$ and events with probability $\alpha_{(\beta)}$ can be generated using Algorithm (1).
- The number of loops until the algorithm stops $\sim$Geom(s_β), where

$$s_\beta = (1-\beta)+\beta\cdot\frac{c_y p_y + c_x p_x}{c_x+c_y}.$$

- Ratio of mean execution time of the two-coin algorithm to the portkey two-coin algorithm is:

$$\frac{1/s_1}{1/s_\beta} = \frac{s_\beta}{s_1} = (1-\beta)\cdot\left(\frac{c_y p_y + c_x p_x}{c_x+c_y}\right)^{-1} + \beta \geq 1.$$

- Computational efficiency gained here is at the cost of statistical efficiency.

Portkey 2-coin algorithm

Algorithm 1 Portkey two-coin algorithm

```
1:  Draw S ~ Bern(β)
2:  if S = 0 then
3:      output 0
4:  if S = 1 then
5:      Draw C_1 ~ Bern( c_y / (c_x + c_y) )
6:      if C_1 = 1 then
7:          Draw C_2 ~ Bern(p_y)
8:          if C_2 = 1 then
9:              output 1
10:         if C_2 = 0 then
11:             go to Step 1
12:     if C_1 = 0 then
13:         Draw C_2 ~ Bern(p_x)
14:         if C_2 = 1 then
15:             output 0
16:         if C_2 = 0 then
17:             go to Step 1
```

Flipped Portkey two-coin algorithm

- For target densities restricted to constrained spaces, it is easier to lower bound $\pi(x)q(x,y)$, i.e., upper bound $\pi(x)^{-1}q(x,y)^{-1}$.
- Suppose, $\pi(x)^{-1}q(x,y)^{-1} = \tilde{c}_x\tilde{p}_x$, for $\tilde{c}_x > 0$, and $0 < \tilde{p}_x < 1$. Here, we consider an acceptance probability $\alpha_{f,(\beta)}(x,y)$ such that

$$\alpha_{f,(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x) + \pi(x)q(x,y) + \frac{(1-\beta)}{\beta}\frac{(\tilde{c}_x + \tilde{c}_y)}{\tilde{c}_x\tilde{p}_x\tilde{c}_y\tilde{p}_y}}$$

  and Algorithm (2) generates events with probability $\alpha_{f,(\beta)}(x,y)$.
- Similar to the Portkey Barker algorithm, the flipped version leads to a loss in statistical efficiency when compared to the two-coin algorithm.
- Can be used if generating $\tilde{p}_x$ coins is easier than generating p_x coins.

Flipped Portkey two-coin algorithm

- For target densities restricted to constrained spaces, it is easier to lower bound $\pi(x)q(x,y)$, i.e., upper bound $\pi(x)^{-1}q(x,y)^{-1}$.
- Suppose, $\pi(x)^{-1}q(x,y)^{-1} = \tilde{c}_x\tilde{p}_x$, for $\tilde{c}_x > 0$, and $0 < \tilde{p}_x < 1$. Here, we consider an acceptance probability $\alpha_{f,(\beta)}(x,y)$ such that

$$\alpha_{f,(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x) + \pi(x)q(x,y) + \frac{(1-\beta)}{\beta}\frac{(\tilde{c}_x + \tilde{c}_y)}{\tilde{c}_x\tilde{p}_x\tilde{c}_y\tilde{p}_y}}$$

  and Algorithm (2) generates events with probability $\alpha_{f,(\beta)}(x,y)$.
- Similar to the Portkey Barker algorithm, the flipped version leads to a loss in statistical efficiency when compared to the two-coin algorithm.
- Can be used if generating $\tilde{p}_x$ coins is easier than generating p_x coins.

Flipped Portkey two-coin algorithm

- For target densities restricted to constrained spaces, it is easier to lower bound $\pi(x)q(x,y)$, i.e., upper bound $\pi(x)^{-1}q(x,y)^{-1}$.
- Suppose, $\pi(x)^{-1}q(x,y)^{-1} = \tilde{c}_x\tilde{p}_x$, for $\tilde{c}_x > 0$, and $0 < \tilde{p}_x < 1$. Here, we consider an acceptance probability $\alpha_{f,(\beta)}(x,y)$ such that

$$\alpha_{f,(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x) + \pi(x)q(x,y) + \frac{(1-\beta)}{\beta}\frac{(\tilde{c}_x + \tilde{c}_y)}{\tilde{c}_x\tilde{p}_x\tilde{c}_y\tilde{p}_y}}$$

  and Algorithm (2) generates events with probability $\alpha_{f,(\beta)}(x,y)$.
- Similar to the Portkey Barker algorithm, the flipped version leads to a loss in statistical efficiency when compared to the two-coin algorithm.
- Can be used if generating $\tilde{p}_x$ coins is easier than generating p_x coins.

Flipped Portkey two-coin algorithm

- For target densities restricted to constrained spaces, it is easier to lower bound $\pi(x)q(x,y)$, i.e., upper bound $\pi(x)^{-1}q(x,y)^{-1}$.
- Suppose, $\pi(x)^{-1}q(x,y)^{-1} = \tilde{c}_x\tilde{p}_x$, for $\tilde{c}_x > 0$, and $0 < \tilde{p}_x < 1$. Here, we consider an acceptance probability $\alpha_{f,(\beta)}(x,y)$ such that

$$\alpha_{f,(\beta)}(x,y) := \frac{\pi(y)q(y,x)}{\pi(y)q(y,x) + \pi(x)q(x,y) + \frac{(1-\beta)}{\beta}\frac{(\tilde{c}_x + \tilde{c}_y)}{\tilde{c}_x\tilde{p}_x\tilde{c}_y\tilde{p}_y}}$$

  and Algorithm (2) generates events with probability $\alpha_{f,(\beta)}(x,y)$.
- Similar to the Portkey Barker algorithm, the flipped version leads to a loss in statistical efficiency when compared to the two-coin algorithm.
- Can be used if generating $\tilde{p}_x$ coins is easier than generating p_x coins.

Flipped Portkey two-coin algorithm

Algorithm 2 Flipped Portkey two-coin algorithm

1: Draw $S \sim \text{Bern}(\beta)$
2: **if** $S = 0$ **then**
3:     output 0
4: **if** $S = 1$ **then**
5:     Draw $C_1 \sim \text{Bern}\left(\frac{\tilde{c}_y}{\tilde{c}_x + \tilde{c}_y}\right)$
6:     **if** $C_1 = 1$ **then**
7:         Draw $C_2 \sim \text{Bern}(\tilde{p}_x)$
8:         **if** $C_2 = 1$ **then**
9:             output 1
10:         **if** $C_2 = 0$ **then**
11:             go to Step 1
12:     **if** $C_1 = 0$ **then**
13:         Draw $C_2 \sim \text{Bern}(\tilde{p}_y)$
14:         **if** $C_2 = 1$ **then**
15:             output 0
16:         **if** $C_2 = 0$ **then**
17:             go to Step 1

Gamma Mixture of Weibulls

- Target Distribution $\pi(\theta) = \int \pi(\theta|\lambda)\nu(d\lambda)$
- $\theta|\lambda \sim$ Weibull(λ, k)
- Proposal Distribution as the Normal Distribution with mean at the current step and variance 0.001.
- $c_\theta := \pi(\theta|\lambda) \leq k/(e\theta)$ and $p_\theta = \pi(\theta)/c_\theta$
- We would draw $\lambda \sim \nu$ and $U \sim \mathscr{U}[0,1]$ independently. Then $Pr\{U \leq \pi(\theta|\lambda)/c_\theta\} = p_\theta$. We set ν = Gamma(10, 100) and $k = 10$.
- We repeat the simulation 1000 times to compare the performances. We observe that ESS decreases as β decreases and ESS/s is higher for portkey barker's than barker's.

Gamma Mixture of Weibulls

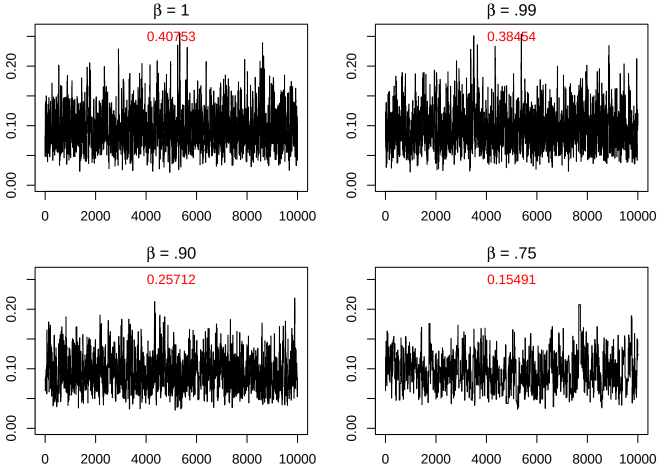

Figure: Trace plots of the chains for four values of β; the acceptance probabilities are 0.40, 0.38, 0.26 and 0.16 for $\beta = 1$, $\beta = 0.99$, $\beta = 0.9$ and $\beta = 0.75$ respectively.

Gamma Mixture of Weibulls

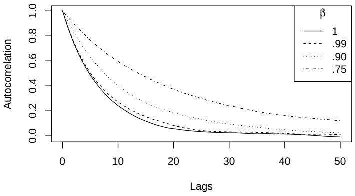

Figure: Autocorrelation plots for a run-length of 10^5 for four β values: 1 (solid), 0.99 (dashed), 0.90 (dotted) and 0.75 (dot-dashed).

Table: Averaged results from 1000 replications. Standard errors are in brackets.

β	1	0.99	0.90	0.75
ESS	7484(7.74)	6939(12.18)	4320(13.86)	2501(9.19)
ESS/s	251.40(2.68)	616.60(3.5)	717.79(4.59)	658.05(4.27)
Mean loops μ	32.00(3.7)	7.63(0)	3.97(0)	2.55(0)
Max loops	1315683	604(3.44)	78(0.37)	32(0.12)

MCMC on constrained spaces

- Let $f(y|\theta)$ be a likelihood and $\pi(\theta|\eta)$ be the prior on θ constrained in a set $\mathscr{A}$ such that

$$\pi(\theta|\eta) = \frac{g(\theta|\eta)I(\theta \in \mathscr{A})}{\int_{\mathscr{A}} g(\theta|\eta)d\theta},$$

where $\int_{\mathscr{A}} g(\theta|\eta)d\theta$ is intractable and becomes an issue when we assign a hyperprior for η.

- **Model:** Suppose $y_1, \ldots, y_n | R \overset{iid}{\sim} N(0, R)$ where R is a $p \times p$ correlation matrix. Assuming that the unique elements in R, r_{ij} are normally distributed and restricting R being positive-definite, we get a prior

$$f(R \mid \mu, \sigma^2) = L(\mu, \sigma^2) \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \mathbb{I}\{R \in S_p^+\},$$

where S_p^+ is the set of $p \times p$ positive-definite matrices and $L(\mu, \sigma^2)$ is the normalising constant.

MCMC on constrained spaces

- Let $f(y|\theta)$ be a likelihood and $\pi(\theta|\eta)$ be the prior on θ constrained in a set $\mathscr{A}$ such that

$$\pi(\theta|\eta) = \frac{g(\theta|\eta)I(\theta \in \mathscr{A})}{\int_{\mathscr{A}} g(\theta|\eta)d\theta},$$

  where $\int_{\mathscr{A}} g(\theta|\eta)d\theta$ is intractable and becomes an issue when we assign a hyperprior for η.

- **Model:** Suppose $y_1, \dots, y_n|R \overset{iid}{\sim} N(0, R)$ where R is a $p \times p$ correlation matrix. Assuming that the unique elements in R, r_{ij} are normally distributed and restricting R being positive-definite, we get a prior

$$f(R \mid \mu, \sigma^2) = L(\mu, \sigma^2) \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(r_{ij}-\mu)^2}{2\sigma^2} \right\} \mathbb{I}\{R \in S_p^+\},$$

  where S_p^+ is the set of $p \times p$ positive-definite matrices and $L(\mu, \sigma^2)$ is the normalising constant.

Full conditional densities

- For known τ^2, a_0, b_0, hyperpriors for $\mu \sim N(0,\tau^2)$ and $\sigma^2 \sim IG(a_0, b_0)$ are chosen.
- The full conditional densities are

$$f(r_{ij} \mid r_{-ij}, \mu, \sigma^2) \propto |R|^{-n/2} \exp\left\{-\frac{\mathrm{tr}(R^{-1}Y^TY)}{2}\right\} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \mathbb{I}\{l_{ij} \le r_{ij} \le u_{ij}\},$$

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\} := L(\mu,\sigma^2) g(\mu, R, \sigma^2),$$

$$f(\sigma^2 \mid R, \mu) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \left(\frac{1}{\sigma^2}\right)^{a_0+I/2+1} \exp\left\{-\frac{b_0}{\sigma^2}\right\}.$$

- Updating μ and σ^2 requires the knowledge of $L(\mu, \sigma^2)$ which is unavailable.

Full conditional densities

- For known τ^2, a_0, b_0, hyperpriors for $\mu \sim N(0, \tau^2)$ and $\sigma^2 \sim IG(a_0, b_0)$ are chosen.
- The full conditional densities are

$$f(r_{ij} \mid r_{-ij}, \mu, \sigma^2) \propto |R|^{-n/2} \exp\left\{-\frac{\mathrm{tr}(R^{-1}Y^TY)}{2}\right\} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \mathbb{I}\{l_{ij} \le r_{ij} \le u_{ij}\},$$

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\} := L(\mu, \sigma^2) g(\mu, R, \sigma^2),$$

$$f(\sigma^2 \mid R, \mu) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \left(\frac{1}{\sigma^2}\right)^{a_0+l/2+1} \exp\left\{-\frac{b_0}{\sigma^2}\right\}.$$

- Updating μ and σ^2 requires the knowledge of $L(\mu, \sigma^2)$ which is unavailable.

Full conditional densities

- For known τ^2, a_0, b_0, hyperpriors for $\mu \sim N(0,\tau^2)$ and $\sigma^2 \sim IG(a_0, b_0)$ are chosen.
- The full conditional densities are

$$f(r_{ij} \mid r_{-ij}, \mu, \sigma^2) \propto |R|^{-n/2} \exp\left\{-\frac{\mathrm{tr}(R^{-1}Y^TY)}{2}\right\} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \mathbb{I}\{l_{ij} \le r_{ij} \le u_{ij}\},$$

$$f(\mu \mid R, \sigma^2) \propto L(\mu,\sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\} := L(\mu,\sigma^2) g(\mu, R, \sigma^2),$$

$$f(\sigma^2 \mid R, \mu) \propto L(\mu,\sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \left(\frac{1}{\sigma^2}\right)^{a_0+l/2+1} \exp\left\{-\frac{b_0}{\sigma^2}\right\}.$$

- Updating μ and σ^2 requires the knowledge of $L(\mu,\sigma^2)$ which is unavailable.

Methods used

- **Shadow Prior Technique**: Use $r_{ij} \sim N(\delta_{ij}, v^2)$ and $\delta_{ij} \sim N(\mu, \sigma^2)$, with unchanged hyper-priors on μ and σ^2. The full conditionals of μ, σ^2 and δ are available in closed-form if $v^2 \approx 0$.
- **Flipped Portkey**($\beta = 0.9$): Use Gaussian random walk proposals for both components so that we can ignore the proposal density in the Bernoulli factories. As each $r_{ij} \in [-1, 1]$,

$$g(\mu, R, \sigma^2)^{-1} L^{-1}(\mu, \sigma^2) \le g(\mu, R, \sigma^2)^{-1} \left[\Phi\left(\sigma^{-1}(1-\mu)\right) - \Phi\left(\sigma^{-1}(-1-\mu)\right) \right]^{I} := \tilde{c}_\mu .$$

  Set $f(\mu|R, \sigma^2)^{-1} = \tilde{c}_\mu \tilde{c}_\mu^{-1} f(\mu \mid R, \sigma^2)^{-1} := \tilde{c}_\mu \tilde{p}_\mu$. To generate coins of probability $\tilde{p}_\mu$: draw $z_{ij} \sim \text{TN}(-1, 1, \mu, \sigma^2)$ for $i < j$, and construct matrix Z with z_{ij} as the lower-triangular entries. If Z is positive-definite, return 1, else return 0.
- **Barker**: Use $\beta = 1$ instead of 0.9 in flipped portkey.

Methods used

- **Shadow Prior Technique**: Use $r_{ij} \sim N(\delta_{ij}, v^2)$ and $\delta_{ij} \sim N(\mu, \sigma^2)$, with unchanged hyper-priors on μ and σ^2. The full conditionals of μ, σ^2 and δ are available in closed-form if $v^2 \approx 0$.
- **Flipped Portkey**($\beta = 0.9$): Use Gaussian random walk proposals for both components so that we can ignore the proposal density in the Bernoulli factories. As each $r_{ij} \in [-1, 1]$,

$$g(\mu, R, \sigma^2)^{-1} L^{-1}(\mu, \sigma^2) \leq g(\mu, R, \sigma^2)^{-1} \left[\Phi\left(\sigma^{-1}(1-\mu)\right) - \Phi\left(\sigma^{-1}(-1-\mu)\right)\right]^{I} := \tilde{c}_{\mu}.$$

  Set $f(\mu|R, \sigma^2)^{-1} = \tilde{c}_{\mu} \tilde{c}_{\mu}^{-1} f(\mu \mid R, \sigma^2)^{-1} := \tilde{c}_{\mu} \tilde{p}_{\mu}$. To generate coins of probability $\tilde{p}_{\mu}$: draw $z_{ij} \sim \mathrm{TN}(-1, 1, \mu, \sigma^2)$ for $i < j$, and construct matrix Z with z_{ij} as the lower-triangular entries. If Z is positive-definite, return 1, else return 0.
- **Barker**: Use $\beta = 1$ instead of 0.9 in flipped portkey.

Methods used

- **Shadow Prior Technique**: Use $r_{ij} \sim N(\delta_{ij}, v^2)$ and $\delta_{ij} \sim N(\mu, \sigma^2)$, with unchanged hyper-priors on μ and σ^2. The full conditionals of μ, σ^2 and δ are available in closed-form if $v^2 \approx 0$.
- **Flipped Portkey**($\beta = 0.9$): Use Gaussian random walk proposals for both components so that we can ignore the proposal density in the Bernoulli factories. As each $r_{ij} \in [-1, 1]$,

$$g(\mu, R, \sigma^2)^{-1} L^{-1}(\mu, \sigma^2) \leq g(\mu, R, \sigma^2)^{-1} \left[\Phi\left(\sigma^{-1}(1-\mu)\right) - \Phi\left(\sigma^{-1}(-1-\mu)\right)\right]^{I} := \tilde{c}_\mu .$$

  Set $f(\mu|R, \sigma^2)^{-1} = \tilde{c}_\mu \tilde{c}_\mu^{-1} f(\mu \mid R, \sigma^2)^{-1} := \tilde{c}_\mu \tilde{p}_\mu$. To generate coins of probability $\tilde{p}_\mu$: draw $z_{ij} \sim \mathrm{TN}(-1, 1, \mu, \sigma^2)$ for $i < j$, and construct matrix Z with z_{ij} as the lower-triangular entries. If Z is positive-definite, return 1, else return 0.
- **Barker**: Use $\beta = 1$ instead of 0.9 in flipped portkey.

Plots

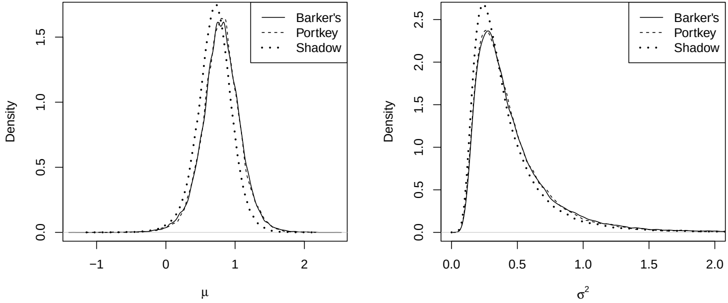

Figure: Estimated posterior density of μ and σ^2 from Markov chain lengths of 105 with v2 = 0.001 as recommended Chen et al. (2010) for Barker's (solid), portkey (dashed) and shadow (dotted).

Plots

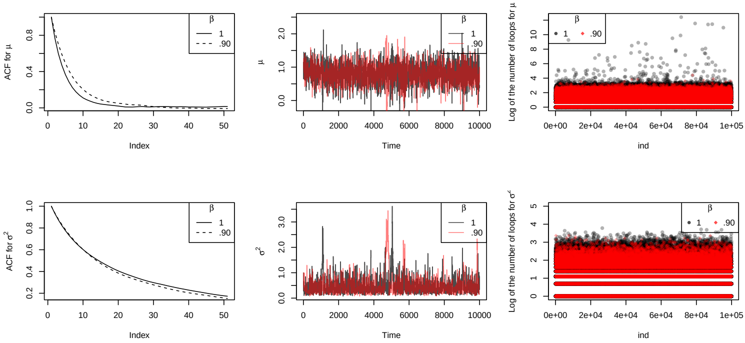

Figure: Autocorrelation, trace and log of Bernoulli factory loops for Barker's and portkey Barker's for μ (right), $\beta = 1$(grey), $\beta = 0.9$ (red), and σ^2 (left), $\beta = 1$(solid), $\beta = 0.90$ (dashed).

Results of simulation

Table: Averaged results from 10 replications.

β	1	.90
ESS	542	496
ESS/s	4.297	6.902
Mean loops μ	218.43	2.99
Mean loops σ^2	3.21	2.49
Max loops μ	2084195	34
Max loops σ^2	38	27

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for 10,000 steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s)-\theta_2 Y_s)+\sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for 10,000 steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for $10,000$ steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for 10,000 steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s,\, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for 10,000 steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1 - Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1 - Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0, 1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for 10,000 steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1 / \theta_1 + \theta_2)$
- The transition density $p(\gamma | y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma | X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for 10,000 steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1 / \theta_1 + \theta_2)$
- The transition density $p(\gamma | y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma | X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for $10,000$ steps.

Bayesian inference for the Wright-Fisher Diffusions

- Discretely observed Wright-Fisher diffusion model is widely used in genetics.
- Two-coin Barker's MCMC algorithm performs exact inference without requiring model restrictions.
- The neutral Wright–Fisher family of diffusions with mutations

$$dY_s = 0.5(\theta_1(1-Y_s) - \theta_2 Y_s) + \sqrt{Y}_s(1-Y_s)dW_s, Y_0 = y_0, \theta_1, \theta_2 > 0,$$

  where Ws is a standard Brownian motion. $Y \in [0,1]$.
- The parameters of interest are (θ_1, θ_2), which are reparametrized as $\gamma = (\gamma_1, \gamma_2) = (\theta_1 + \theta_2, \theta_1/\theta_1 + \theta_2)$
- The transition density $p(\gamma|y_{i-1}, y_i)$, which determines the likelihood, is not available in a closed form and are characterized by poor numerical properties near the 0 and 1 barriers that undermine reliable Bayesian inference based on approximations.
- Apply Lamperti transform to Y, leading to unit diffusion coefficient process X.
- Update γ as $\gamma_1 \sim U(\gamma_1 - 0.3, \gamma_1 - 0.3)$ and $\gamma_2 \sim U(\gamma_1 - 0.01, \gamma_1 - 0.01)$. Update missing paths, X_{mis}, given the transformed observations, X_{obs} using standard Brownian bridge proposals.
- It can be shown that $\pi(\gamma|X_{min}, X_{obs}) = c_\gamma p_\gamma$ (Gonçalves et al. (2017b)).
- We simulate a Wright-Fisher diffusion with $\gamma_1 = 8$ and $\gamma_2 = 0.5$ and run the samplers for $10,000$ steps.

Bayesian inference for the Wright-Fisher Diffusions

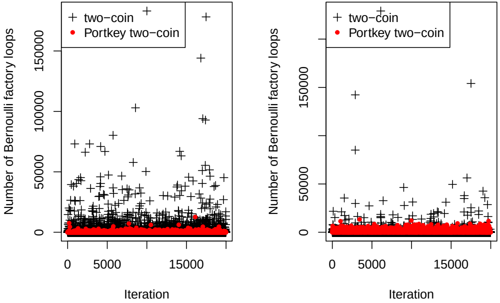

Figure: Comparing the number of loops of the Bernoulli factory for two-coin (cross) and portkey (red dot) two-coin algorithm for the γ_1 update (left) and the γ_2 update (right).

Bayesian inference for the Wright-Fisher Diffusions

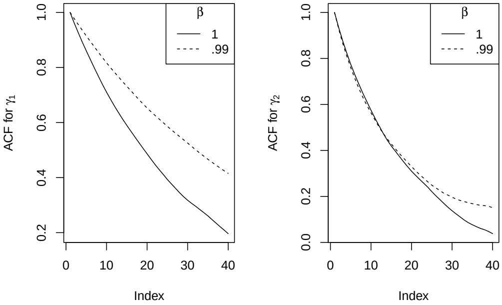

Figure: Autocorrelation plots for γ_1 and γ_2;$\beta = 1$ (solid), $\beta = 0.99$ (dashed).

Bayesian inference for the Wright-Fisher Diffusions

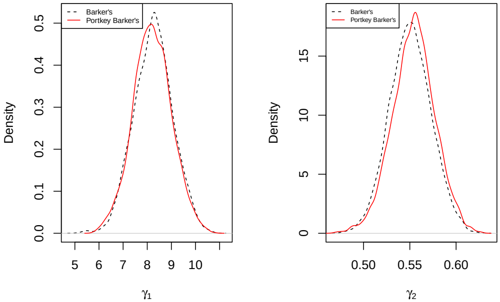

Figure: Density plots of γ_1 (left) and γ_2 (right) using Barker's (dashed) and Portkey Barker's (solid).

A., B. A. (1965). Monte carlo calculations of the radial distribution functions for a proton-electron plasma. *Computing in Science & Engineering*, pages 119–134.

Gonçalves, F. B., Łatuszyński, K., and Roberts, G. O. (2017a). Barker's algorithm for bayesian inference with intractable likelihoods. *Brazilian Journal of Probability and Statistics*, 31(4):732–745.

Gonçalves, F. B., Łatuszyński, K. G., and Roberts, G. O. (2017b). Exact monte carlo likelihood-based inference for jump-diffusion processes. *arXiv preprint arXiv:1707.00332*.

Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Latuszynski, K. (2010). The bernoulli factory, its extensions and applications. *Proceedings of IWAP*, pages 1–5.

Łatuszyński, K. and Roberts, G. O. (2013). Clts and asymptotic variance of time-sampled markov chains. *Methodology and Computing in Applied Probability*, 15(1):237–247.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Peskun, P. H. (1973). Optimum monte-carlo sampling using markov chains. *Biometrika*, 60(3):607–612.

Vats, D., Gonçalves, F., Łatuszyński, K., and Roberts, G. (2021). Efficient bernoulli factory markov chain monte carlo for intractable posteriors. *Biometrika*.