Non-Parametric Bayesian Inference for Controlled Branching Processes Through MCMC Methods

M. González, R. Martínez, I. del Puerto, A. Ramos

Department of Mathematics. University of Extremadura
Spanish Branching Processes Group

18th International Conference in Computational Statistics
Porto, August 2008
Contents

1 Controlled Branching Processes

2 Bayesian Inference for Controlled Branching Processes

3 A Simulation-Based Method using Gibbs Sampler
   • Introducing the Method
   • Developing the Method
   • Simulated Example

4 Concluding Remarks and References
   • Concluding Remarks
   • References
Contents

1 Controlled Branching Processes

2 Bayesian Inference for Controlled Branching Processes

3 A Simulation-Based Method using Gibbs Sampler
   • Introducing the Method
   • Developing the Method
   • Simulated Example

4 Concluding Remarks and References
   • Concluding Remarks
   • References
Contents

1 Controlled Branching Processes

2 Bayesian Inference for Controlled Branching Processes

3 A Simulation-Based Method using Gibbs Sampler
   - Introducing the Method
   - Developing the Method
   - Simulated Example

4 Concluding Remarks and References
   - Concluding Remarks
   - References
Contents

1. Controlled Branching Processes
2. Bayesian Inference for Controlled Branching Processes
3. A Simulation-Based Method using Gibbs Sampler
   - Introducing the Method
   - Developing the Method
   - Simulated Example
4. Concluding Remarks and References
   - Concluding Remarks
   - References
Inside the general context concerning Stochastic Models, Branching Processes Theory provides appropriate mathematical models for description of the probabilistic evolution of systems whose components (cell, particles, individuals in general), after certain life period, reproduce and die. Therefore, it can be applied in several fields (biology, demography, ecology, epidemiology, genetics, medicine,...).
Branching Processes

Example

\[ Z_0 = 1 \]

\[ Z_{n+1} = \sum_{j=1}^{z_n} X_{nj} \]
Branching Processes

**Example**

\[ Z_0 = 1 \]

\[ Z_{n+1} = \sum_{j=1}^{z_n} X_{nj} \]
Branching Processes

**Example**

\[ Z_0 = 1 \]

\[ Z_1 = 2 \]

\[ \vdots \]

\[ Z_{n+1} = \sum_{j=1}^{z_n} X_{nj} \]
### Branching Processes

**Example**

\[
Z_0 = 1 \\
Z_1 = 2 \\
\vdots \\
Z_{n+1} = \sum_{j=1}^{z_n} X_{nj}
\]
Branching Processes

Example

\[
Z_0 = 1 \\
Z_1 = 2 \\
Z_2 = 7
\]

\[
Z_{n+1} = \sum_{j=1}^{Z_n} X_{nj}
\]
Branching Processes

Example

\[ Z_0 = 1 \]
\[ Z_1 = 2 \]
\[ Z_2 = 7 \]

\[ Z_{n+1} = \sum_{j=1}^{z_n} X_{nj} \]
Branching Processes

Example

\[ Z_0 = 1 \]
\[ Z_1 = 2 \]
\[ Z_2 = 7 \]
\[ Z_3 = 10 \]
\[ \vdots \]

\[ Z_{n+1} = \sum_{j=1}^{z_n} X_{nj} \]
Branching Processes

Example

\[ Z_0 = 1 \]
\[ Z_1 = 2 \]
\[ Z_2 = 7 \]
\[ Z_3 = 10 \]
\[ \vdots \]

\[ Z_{n+1} = \sum_{j=1}^{z_n} X_{nj} \]
Branching Processes

Example

\[ Z_0 = 1 \]
\[ Z_1 = 2 \]
\[ Z_2 = 7 \]
\[ Z_3 = 10 \]
\[ \vdots \]

\[ Z_{n+1} = \sum_{j=1}^{Z_n} X_{nj} \]
Branching Processes

Main Results for Galton–Watson Branching Processes

Let \( m = E[X_{01}] \) and \( \sigma^2 = \text{Var}[X_{01}] \)

- **Extinction Problem**
  - If \( m \leq 1 \) ⇒ the process dies out with probability 1
  - If \( m > 1 \) ⇒ there exists a positive probability of non-extinction

- **Asymptotic behaviour**

- **Statistical Inference**
Branching Processes

Main Results for Galton–Watson Branching Processes

Let \( m = E[X_{01}] \) and \( \sigma^2 = \text{Var}[X_{01}] \)

- **Extinction Problem**
  - If \( m \leq 1 \) ⇒ the process **dies out** with probability 1
  - If \( m > 1 \) ⇒ there exists a **positive probability of non-extinction**

- Asymptotic behaviour
- Statistical Inference
Main Results for Galton–Watson Branching Processes

Let $m = E[X_{01}]$ and $\sigma^2 = Var[X_{01}]$

- **Extinction Problem**
  - If $m \leq 1 \Rightarrow$ the process dies out with probability 1
  - If $m > 1 \Rightarrow$ there exists a positive probability of non-extinction

- **Asymptotic behaviour**

- Statistical Inference
Main Results for Galton–Watson Branching Processes

Let $m = E[X_{01}]$ and $\sigma^2 = Var[X_{01}]$

- **Extinction Problem**
  - If $m \leq 1 \Rightarrow$ the process *dies out* with probability 1
  - If $m > 1 \Rightarrow$ there exists a *positive probability of non-extinction*

- **Asymptotic behaviour**

- **Statistical Inference**
Many **monographs** about the **theory and applications** about the branching processes have been published:

A Controlled Branching Process is a discrete-time stochastic growth population model in which the individuals with reproductive capacity in each generation are controlled by some function $\phi$. This branching model is well-suited for describing the probabilistic evolution of populations in which, for various reasons of an environmental, social or other nature, there is a mechanism that establishes the number of progenitors who take part in each generation.
Mathematically: Controlled Branching Process

\[ \{Z_n\}_{n \geq 0} \]

\[ Z_0 = N, \quad Z_{n+1} = \sum_{i=1}^{\phi(Z_n)} X_{ni}, \quad n = 0, 1, \ldots \]

- \{X_{ni}: i = 1, 2, \ldots, n = 0, 1, \ldots\} are i.i.d. random variables.
- \(\{p_k: k \in S\}\) **Offspring Distribution**  \(m = E[X_{01}], \sigma^2 = Var[X_{01}]\)

- \(\phi: \mathbb{R}^+ \rightarrow \mathbb{R}^+\) is assumed to be integer-valued for integer-valued arguments **Control Function**
Controlled Branching Processes

Properties

- \( \{Z_n\}_{n \geq 0} \) is a Homogeneous Markov Chain
- **Duality Extinction-Explosion:** \( P(Z_n \to 0) + P(Z_n \to \infty) = 1 \)

Main Topics Investigated

- **Extinction Problem**
  - Sevast’yanov and Zubkov (1974)
  - Zubkov (1974)
  - Molina, González and Mota (1998)

- **Asymptotic Behaviour: Growth rates**
  - Bagley (1986)
  - Molina, González and Mota (1998)
Controlled Branching Processes

Properties
- \( \{Z_n\}_{n \geq 0} \) is a Homogeneous Markov Chain
- Duality Extinction-Explosion: \( P(Z_n \to 0) + P(Z_n \to \infty) = 1 \)

Main Topics Investigated
- Extinction Problem
  - Sevast’yanov and Zubkov (1974)
  - Zubkov (1974)
  - Molina, González and Mota (1998)
- Asymptotic Behaviour: Growth rates
  - Bagley (1986)
  - Molina, González and Mota (1998)
Controlled Branching Processes

Main Topics Investigated

- **Statistical Inference**
Non-Parametric Framework

Offspring Distribution: \( p = \{p_k : k \in S\} \) \( S \) finite.

Sample: The entire family tree up to the current generation

\[ \{X_{ki} : i = 1, \ldots, \phi(Z_k), k = 0, 1, \ldots, n\} \]

or at least

\[ \mathcal{Z}_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\} \]

where \( Z_j(k) = \sum_{i=1}^{\phi(Z_j)} I_{\{X_{ji} = k\}} = \) number of parents in the jth-generation which generate exactly \( k \) offspring

Objective: Make inference on \( p \)
Non-Parametric Framework

Offspring Distribution: \( p = \{p_k : k \in S\} \) \( S \) finite.

Sample: The entire family tree up to the current generation

\[ \{X_{ki} : i = 1, \ldots, \phi(Z_k), k = 0, 1, \ldots, n\} \]

or at least

\[ \mathcal{Z}_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\} \]

where \( Z_j(k) = \sum_{i=1}^{\phi(Z_j)} I_{\{X_{ji}=k\}} \) = number of parents in the jth-generation which generate exactly k offspring

Objective: Make inference on \( p \)
Bayesian Inference for Controlled Branching Processes

Likelihood Function

\[ L(p|Z_n) \propto \prod_{k \in S} p_k^{\sum_{j=0}^{n} Z_j(k)} \]

Conjugate Class of Distributions: Dirichlet Family

- Prior Distribution: \( p \sim D(\alpha_k : k \in S) \)
- Posterior Distribution:

\[ p|Z_n \sim D(\alpha_k + \sum_{j=0}^{n} Z_j(k) : k \in S) \]
Bayesian Inference for Controlled Branching Processes

Likelihood Function

\[ L(p | Z_n) \propto \prod_{k \in S} p_k^{\sum_{j=0}^{n} Z_j(k)} \]

Conjugate Class of Distributions: Dirichlet Family

- Prior Distribution: \( p \sim D(\alpha_k : k \in S) \)
- Posterior Distribution:

\[ p | Z_n \sim D(\alpha_k + \sum_{j=0}^{n} Z_j(k) : k \in S) \]
Setting out the Problem

In real problems it is difficult to observe the entire family tree \( \{X_{ki} : i = 1, 2, \ldots, k = 0, 1, \ldots, n\} \) or even the random variables \( \mathcal{Z}_n = \{Z_j(k) : k \in \mathcal{S}, j = 0, \ldots, n\} \)

Usual Sample Information

\[ \mathcal{Z}_n^* = \{Z_j : j = 0, \ldots, n\} \]

A Solution

We introduce an algorithm to approximate the distribution

\[ p | \mathcal{Z}_n^* \]

using Markov Chain Monte Carlo Methods
Setting out the Problem

In real problems it is difficult to observe the entire family tree \(\{X_{ki} : i = 1, 2, \ldots, k = 0, 1, \ldots, n\}\) or even the random variables \(Z_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\}\).

Usual Sample Information

\[ Z_n^* = \{Z_j : j = 0, \ldots, n\} \]

A Solution

We introduce an algorithm to approximate the distribution \( p|Z_n^* \) using Markov Chain Monte Carlo Methods.
Setting out the Problem

In real problems it is difficult to observe the entire family tree \( \{X_{ki} : i = 1, 2, \ldots, k = 0, 1, \ldots, n\} \) or even the random variables \( \mathcal{Z}_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\} \)

Usual Sample Information

\[ \mathcal{Z}_n^* = \{Z_j : j = 0, \ldots, n\} \]

A Solution

We introduce an algorithm to approximate the distribution

\[ p|\mathcal{Z}_n^* \]

using Markov Chain Monte Carlo Methods
Gibbs Sampler: Introducing the Method

- **Sample:** \( \mathcal{Z}_n^* = \{Z_j : j = 0, \ldots, n\} \)

The Problem

- **Latent Variables:**
  \[ \mathcal{Z}_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\} \]

- **Gibbs Sampler:**
  \[ p | \mathcal{Z}_n, \mathcal{Z}_n^* \quad \mathcal{Z}_n | \mathcal{Z}_n^*, p \]
Gibbs Sampler: Introducing the Method

- **Sample:** \( \mathcal{Z}_n^* = \{Z_j : j = 0, \ldots, n\} \)

The Problem

- **Latent Variables:**
  \[ \mathcal{Z}_n = \{Z_j(k) : k \in \mathcal{S}, j = 0, \ldots, n\} \]

- **Gibbs Sampler:**
  \[ p|\mathcal{Z}_n, \mathcal{Z}_n^* \quad \mathcal{Z}_n|\mathcal{Z}_n^*, p \]
Gibbs Sampler: Introducing the Method

Sample: \( Z^*_n = \{Z_j : j = 0, \ldots, n\} \)

The Problem

Latent Variables:

\( Z_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\} \)

Gibbs Sampler:

\[ p|Z_n, Z^*_n \]

\[ Z_n|Z^*_n, p \]
Gibbs Sampler: Introducing the Method

- **Sample:** $\mathcal{Z}_n^* = \{Z_j : j = 0, \ldots, n\}$

**The Problem**

$$p | \mathcal{Z}_n^*$$

- **Latent Variables:**

$$\mathcal{Z}_n = \{Z_j(k) : k \in S, j = 0, \ldots, n\}$$

- **Gibbs Sampler:**

$$p | \mathcal{Z}_n, \mathcal{Z}_n^* \quad \mathcal{Z}_n | \mathcal{Z}_n^*, p$$
First Conditional Distribution: $p|Z_n, Z^*_n$

$$p|Z_n, Z^*_n \equiv p|Z_n \sim D(\alpha_k + \sum_{j=0}^{n} Z_j(k) : k \in S)$$

For $j = 0, \ldots, n$

$$\phi(Z_j) = \sum_{k \in S} Z_j(k)$$

$$Z_{j+1} = \sum_{k \in S} kZ_j(k)$$
Gibbs Sampler: Introducing the Method

First Conditional Distribution: \( p|Z_n, Z^*_n \)

\[
p|Z_n, Z^*_n \equiv p|Z_n \sim D(\alpha_k + \sum_{j=0}^{n} Z_j(k): k \in S)
\]

For \( j = 0, \ldots, n \)

\[
\phi(Z_j) = \sum_{k \in S} Z_j(k)
\]

\[
Z_{j+1} = \sum_{k \in S} kZ_j(k)
\]
Gibbs Sampler: Introducing the Method

First Conditional Distribution: $p|\mathcal{Z}_n, \mathcal{Z}_n^*$

$$p|\mathcal{Z}_n, \mathcal{Z}_n^* \equiv p|\mathcal{Z}_n \sim D(\alpha_k + \sum_{j=0}^{n} Z_j(k) : k \in S)$$

- For $j = 0, \ldots, n$

$$\phi(Z_j) = \sum_{k \in S} Z_j(k)$$

$$Z_{j+1} = \sum_{k \in S} kZ_j(k)$$
Gibbs Sampler: Introducing the Method

Second Conditional Distribution: \( Z_n | Z^*_n, p \)

\[
P(Z_n | Z^*_n, p) = \prod_{j=0}^{n} P(Z_j(k) : k \in S | Z_j, Z_{j+1}, p)
\]

\( (Z_j(k) : k \in S) | Z_j, Z_{j+1}, p \)

is obtained from a

\( \text{Multinomial}(\phi(Z_j), p) \)

normalized by considering the constraint

\[
Z_{j+1} = \sum_{k \in S} k Z_j(k)
\]
Second Conditional Distribution: $\mathcal{Z}_n | \mathcal{Z}_n^*, p$

$$P(\mathcal{Z}_n | \mathcal{Z}_n^*, p) = \prod_{j=0}^{n} P(Z_j(k) : k \in S | Z_j, Z_{j+1}, p)$$

$$(Z_j(k) : k \in S) | Z_j, Z_{j+1}, p$$

is obtained from a

$$Multinomial(\phi(Z_j), p)$$

normalized by considering the constraint

$$Z_{j+1} = \sum_{k \in S} kZ_j(k)$$
Introducing the Method

Gibbs Sampler: Introducing the Method

Second Conditional Distribution: \( \mathcal{Z}_n | \mathcal{Z}^*_n, p \)

\[
\begin{align*}
p & \quad \phi(Z_0) \\
Z_0(k), \ k \in S & \quad \phi(Z_1) \\
Z_1(k), \ k \in S & \quad \phi(Z_2) \\
\vdots & \quad \vdots & \quad \vdots \\
Z_n(k), \ k \in S & \quad \phi(Z_n) \\
Z_{n+1} & \quad \phi(Z_j) = \sum_{k \in S} Z_j(k), \quad Z_{j+1} = \sum_{k \in S} kZ_j(k)
\end{align*}
\]
Gibbs Sampler: Introducing the Method

Second Conditional Distribution: $Z_n | Z_n^*, p$

\[
p \quad \phi(Z_0) \\
Z_0(k), k \in S \\
Z_1 \quad \phi(Z_1) \\
Z_1(k), k \in S \\
Z_2 \quad \phi(Z_2) \\
\vdots \quad \vdots \\
Z_n \quad \phi(Z_n) \\
Z_n(k), k \in S \\
Z_{n+1}
\]

\[
\phi(Z_j) = \sum_{k \in S} Z_j(k), \quad Z_{j+1} = \sum_{k \in S} kZ_j(k)
\]
Second Conditional Distribution: $\mathcal{Z}_n \mid \mathcal{Z}_n^*, p$

\[
p \quad \phi(Z_0)
\]
\[
Z_0(k), k \in S
\]
\[
Z_1 \quad \phi(Z_1)
\]
\[
Z_1(k), k \in S
\]
\[
Z_2 \quad \phi(Z_2)
\]
\[
\vdots \quad \vdots \quad \vdots
\]
\[
Z_n \quad \phi(Z_n)
\]
\[
Z_n(k), k \in S
\]
\[
Z_{n+1}
\]

$$\phi(Z_j) = \sum_{k \in S} Z_j(k), \quad Z_{j+1} = \sum_{k \in S} kZ_j(k)$$
Gibbs Sampler: Introducing the Method

Second Conditional Distribution: $\mathcal{Z}_n | \mathcal{Z}_n^*, p$

- $p$
- $\phi(Z_0)$
- $Z_0(k), k \in S$
- $\phi(Z_1)$
- $Z_1(k), k \in S$
- $\phi(Z_2)$
- $Z_2(k), k \in S$
- $\vdots$
- $\vdots$
- $\vdots$
- $\phi(Z_n)$
- $Z_n(k), k \in S$
- $\phi(Z_{n+1})$
- $Z_{n+1}$

$\phi(Z_j) = \sum_{k \in S} Z_j(k)$, $Z_{j+1} = \sum_{k \in S} kZ_j(k)$
Gibbs Sampler: Developing the Method

**Algorithm**

- **Fixed** $p^{(0)}$
- **Do** $l = 1$
  - Generate $Z^{(l)}_n \sim Z_n|Z^*_n, p^{(l-1)}$
  - Generate $p^{(l)} \sim p|Z^{(l)}_n$
- **Do** $l = l + 1$

- For a run of the sequence $\{p^{(l)}\}_{l \geq 0}$, we choose $Q + 1$ vectors in the way $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$, where $G$ is a batch size.
- The vectors $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$ are considered independent samples from $p|Z^*_n$ if $G$ and $N$ are large enough (Tierney (1994)).
- Since these vectors could be affected by the initial state $p^{(0)}$, we apply the algorithm $T$ times, obtaining a final sample of length $T(Q + 1)$.
Algorithm

Fixed $p^{(0)}$

Do $l = 1$

Generate $Z^{(l)}_n \sim Z_n | Z^*_n, p^{(l-1)}$

Generate $p^{(l)} \sim p | Z^{(l)}_n$

Do $l = l + 1$

For a run of the sequence $\{p^{(l)}\}_{l \geq 0}$, we choose $Q + 1$ vectors in the way $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$, where $G$ is a batch size.

The vectors $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$ are considered independent samples from $p | Z^*_n$ if $G$ and $N$ are large enough (Tierney (1994)).

Since these vectors could be affected by the initial state $p^{(0)}$, we apply the algorithm $T$ times, obtaining a final sample of length $T(Q + 1)$. 
Gibbs Sampler: Developing the Method

Algorithm

Fixed $p^{(0)}$
Do $l = 1$
  Generate $Z_n^{(l)} \sim Z_n|Z_n^*, p^{(l-1)}$
  Generate $p^{(l)} \sim p|Z_n^{(l)}$
Do $l = l + 1$

For a run of the sequence $\{p^{(l)}\}_{l \geq 0}$, we choose $Q + 1$ vectors in the way $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$, where $G$ is a batch size.

- The vectors $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$ are considered independent samples from $p|Z_n^*$ if $G$ and $N$ are large enough (Tierney (1994)).

- Since these vectors could be affected by the initial state $p^{(0)}$, we apply the algorithm $T$ times, obtaining a final sample of length $T(Q + 1)$. 
Gibbs Sampler: Developing the Method

**Algorithm**

Fixed $p^{(0)}$

Do $l = 1$

Generate $Z_n^{(l)} \sim Z_n | Z_n^*, p^{(l-1)}$

Generate $p^{(l)} \sim p | Z_n^{(l)}$

Do $l = l + 1$

- For a run of the sequence $\{p^{(l)}\}_{l \geq 0}$, we choose $Q + 1$ vectors in the way $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$, where $G$ is a batch size.

- The vectors $\{p^{(N)}, p^{(N+G)}, \ldots, p^{(N+QG)}\}$ are considered independent samples from $p | Z_n^*$ if $G$ and $N$ are large enough (Tierney (1994)).

- Since these vectors could be affected by the initial state $p^{(0)}$, we apply the algorithm $T$ times, obtaining a final sample of length $T(Q + 1)$. 
Gibbs Sampler: Simulated Example

Offspring Distribution:

<table>
<thead>
<tr>
<th>$k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_k$</td>
<td>0.28398</td>
<td>0.42014</td>
<td>0.233090</td>
<td>0.05747</td>
<td>0.00531</td>
</tr>
</tbody>
</table>

Parameters: $m = 1.08$, $\sigma^2 = 0.7884$

Control function: $\phi(x) = 7$ if $x \leq 7$; $x$ if $7 < x \leq 20$; 20 if $x > 20$

Simulated Data
Gibbs Sampler: Simulated Example

Observed Data: $n = 15$

<table>
<thead>
<tr>
<th>$n$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_n$</td>
<td>10</td>
<td>12</td>
<td>17</td>
<td>13</td>
<td>12</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n$</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_n$</td>
<td>15</td>
<td>21</td>
<td>24</td>
<td>20</td>
<td>19</td>
</tr>
</tbody>
</table>

$p \sim D(1/2, \ldots, 1/2)$

Selection of $N, G, Q$ and $T$

$N = 5000, G = 100, Q = 49$ and $T = 100$

- Gelman-Rubin-Brooks diagnostic plots.
- Estimated potential scale reduction factor.
- Autocorrelation values.
Gibbs Sampler: Simulated Example

Gelman-Rubin-Brooks diagnostic plots (CODA package for R)

- $p_0$
- $p_1$
- $p_2$
- $p_3$
- $p_4$
Gibbs Sampler: Simulated Examples

\[ p \sim D(1/2, \ldots, 1/2) \]

**Selection of \( N, G, Q \) and \( T \)**

\( N = 5000, \ G = 100, \ Q = 49 \) and \( T = 100 \)

- Gelman-Rubin-Brooks diagnostic plots.
- Estimated potential scale reduction factor.
- Autocorrelation values.

**Computation Time:** 60.10s for each chain on an Intel(R) Core(TM)2 Duo CPU T7500 running at 2.20GHz with 2038 MB RAM.
Gibbs Sampler: Simulated Example

Sample Information: $Z_n^*$

$N = 5000, G = 100, Q = 49$ and $T = 100$ (Sample Size: 5000)

Algorithm’s Efficiency

<table>
<thead>
<tr>
<th></th>
<th>MEAN</th>
<th>SD</th>
<th>MCSE</th>
<th>TSSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offspring Mean</td>
<td>1.0766931</td>
<td>0.0579821</td>
<td>0.0008200</td>
<td>0.0008179</td>
</tr>
</tbody>
</table>
In a non-parametric Bayesian framework we can make inference on the offspring distribution of Controlled Branching Processes, and consequently on the rest of offspring parameters, without observing the entire family tree, but only considering the total number of individuals in each generation.

We use a MCMC method (Gibbs sampler) and the statistical software and programming environment R, in order to give a "likely" approach to family trees.
In a non-parametric Bayesian framework we can make inference on the offspring distribution of Controlled Branching Processes, and consequently on the rest of offspring parameters, without observing the entire family tree, but only considering the total number of individuals in each generation.

We use a MCMC method (Gibbs sampler) and the statistical software and programming environment R, in order to give a "likely" approach to family trees.


Thank you very much!