Sequential Implementation of Monte Carlo Tests with Uniformly Bounded Resampling Risk

Axel Gandy

Department of Mathematics
Imperial College London
a.gandy@imperial.ac.uk

useR! 2009, Rennes
July 8-10, 2009
Introduction

- Test statistic $T$, reject for large values.
- Observation: $t$.
- $p$-value:
  \[ p = P(T \geq t) \]

Often not available in closed form.

- Monte Carlo Test:
  \[ \hat{p}_{\text{naive}} = \frac{1}{n} \sum_{i=1}^{n} I(T_i \geq t), \]
  where $T, T_1, \ldots, T_n$ i.i.d.

- Examples:
  - Bootstrap,
  - Permutation tests.

- Goal: Estimate $p$ using few $X_i$

Mainly interested in deciding if $p \leq \alpha$ for some $\alpha$. 
Introduction

- Test statistic $T$, reject for large values.
- Observation: $t$.
- $p$-value:
  \[ p = P(T \geq t) \]

Often not available in closed form.
- Monte Carlo Test:
  \[ \hat{p}_{\text{naive}} = \frac{1}{n} \sum_{i=1}^{n} I(T_i \geq t), \]
  where $T, T_1, \ldots, T_n$ i.i.d.
- Examples:
  - Bootstrap,
  - Permutation tests.
- Goal: Estimate $p$ using few $X_i$

Mainly interested in deciding if $p \leq \alpha$ for some $\alpha$. 
Introduction

- Test statistic $T$, reject for large values.
- Observation: $t$.
- $p$-value:
  \[ p = P(T \geq t) \]

Often not available in closed form.
- Monte Carlo Test:
  \[ \hat{p}_{\text{naive}} = \frac{1}{n} \sum_{i=1}^{n} I(T_i \geq t), \]
  where $T, T_1, \ldots, T_n$ i.i.d.
- Examples:
  - Bootstrap,
  - Permutation tests.
- Goal: Estimate $p$ using few $X_i$

Mainly interested in deciding if $p \leq \alpha$ for some $\alpha$. 
Introduction

- Test statistic $T$, reject for large values.
- Observation: $t$.
- $p$-value:
  
  \[ p = P(T \geq t) \]

  Often not available in closed form.

- Monte Carlo Test:

  \[ \hat{p}_{\text{naive}} = \frac{1}{n} \sum_{i=1}^{n} I(T_i \geq t), \]

  where $T, T_1, \ldots, T_n$ i.i.d.

- Examples:
  - Bootstrap,
  - Permutation tests.

- Goal: Estimate $p$ using few $X_i$

  Mainly interested in deciding if $p \leq \alpha$ for some $\alpha$. 
Sequential approaches based on $S_n = \sum_{i=1}^{n} X_i$

- Stop once $S_n \geq U_n$ or $S_n \leq L_n$
- $\tau$: hitting time
- Compute $\hat{p}$ based on $S_{\tau}$ and $\tau$.
- Hit $B_U$: decide $p > \alpha$,
- Hit $B_L$: decide $p \leq \alpha$,
Sequential approaches based on $S_n = \sum_{i=1}^{n} X_i$

- Stop once $S_n \geq U_n$ or $S_n \leq L_n$
- $\tau$: hitting time
- Compute $\hat{p}$ based on $S_\tau$ and $\tau$
- Hit $B_U$: decide $p > \alpha$
- Hit $B_L$: decide $p \leq \alpha$
Sequential approaches based on $S_n = \sum_{i=1}^{n} X_i$

- Stop once $S_n \geq U_n$ or $S_n \leq L_n$
- $\tau$: hitting time
- Compute $\hat{p}$ based on $S_{\tau}$ and $\tau$.
- Hit $B_U$: decide $p > \alpha$,
- Hit $B_L$: decide $p \leq \alpha$, 

\[ S_n = \sum_{i=1}^{n} X_i \]
Previous Approaches

- Besag & Clifford (1991):

- (Truncated) Sequential Probability Ratio Test, Fay et al. (2007)

- R-package MChtest.
What do we really want?

Is \( p \leq \alpha \)?

Two individuals using the same statistical method on the same data should arrive at the same conclusion. 

*First law of applied statistics, Gleser (1996)*

Consider the resampling risk

\[
RR_p(\hat{p}) \equiv \begin{cases} 
P_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\
P_p(\hat{p} \leq \alpha) & \text{if } p > \alpha. \
\end{cases}
\]

Want:

\[
\sup_{p \in [0,1]} RR_p(\hat{p}) \leq \epsilon
\]

for some (small) \( \epsilon > 0 \).

For Besag & Clifford (1991), SPRT: \( \sup_p RR_p \geq 0.5 \)
What do we really want?

Is \( p \leq \alpha \)?

Two individuals using the same statistical method on the same data should arrive at the same conclusion.

*First law of applied statistics, Gleser (1996)*

Consider the resampling risk

\[
RR_p(\hat{p}) \equiv \begin{cases} 
  P_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\
  P_p(\hat{p} \leq \alpha) & \text{if } p > \alpha.
\end{cases}
\]

Want:

\[
\sup_{p \in [0,1]} RR_p(\hat{p}) \leq \epsilon
\]

for some (small) \( \epsilon > 0 \).

For Besag & Clifford (1991), SPRT: \( \sup_p RR_p \geq 0.5 \)
What do we really want?

Is \( p \leq \alpha \)?

Two individuals using the same statistical method on the same data should arrive at the same conclusion.

*First law of applied statistics, Gleser (1996)*

Consider the resampling risk

\[
RR_p(\hat{p}) \equiv \begin{cases} 
P_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\
P_p(\hat{p} \leq \alpha) & \text{if } p > \alpha.
\end{cases}
\]

Want:

\[
\sup_{\hat{p}} RR_p(\hat{p}) \leq \epsilon
\]

for some (small) \( \epsilon > 0 \).

For Besag & Clifford (1991), SPRT: \( \sup_p RR_p \geq 0.5 \).
What do we really want?

Is $p \leq \alpha$?

Two individuals using the same statistical method on the same data should arrive at the same conclusion.

*First law of applied statistics, Gleser (1996)*

Consider the resampling risk

$$RR_p(\hat{p}) \equiv \begin{cases} P_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\ P_p(\hat{p} \leq \alpha) & \text{if } p > \alpha. \end{cases}$$

Want:

$$\sup_{p \in [0,1]} RR_p(\hat{p}) \leq \epsilon$$

for some (small) $\epsilon > 0$.

For Besag & Clifford (1991), SPRT: $\sup_p RR_P \geq 0.5$
What do we really want?

Is $p \leq \alpha$?

Two individuals using the same statistical method on the same data should arrive at the same conclusion.

First law of applied statistics, Gleser (1996)

Consider the resampling risk

$$RR_p(\hat{p}) \equiv \begin{cases} 
P_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\
P_p(\hat{p} \leq \alpha) & \text{if } p > \alpha.
\end{cases}$$

Want:

$$\sup_{p \in [0,1]} RR_p(\hat{p}) \leq \epsilon$$

for some (small) $\epsilon > 0$.

For Besag & Clifford (1991), SPRT: $\sup_p RR_p \geq 0.5$
What do we really want?

Is $p \leq \alpha$?

Two individuals using the same statistical method on the same data should arrive at the same conclusion.

*First law of applied statistics, Gleser (1996)*

Consider the resampling risk

$$RR_p(\hat{p}) \equiv \begin{cases} 
P_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\
P_p(\hat{p} \leq \alpha) & \text{if } p > \alpha.
\end{cases}$$

Want:

$$\sup_{p \in [0,1]} RR_p(\hat{p}) \leq \epsilon$$

for some (small) $\epsilon > 0$.

For Besag & Clifford (1991), SPRT: $\sup_p RR_P \geq 0.5$
Recursive Definition of the Boundaries

Want:

\[
\sup_p \text{RR}_p(\hat{p}) \leq \epsilon
\]

Suffices to ensure

\[
P_\alpha(\text{hit } B_U) \leq \epsilon
\]
\[
P_\alpha(\text{hit } B_L) \leq \epsilon
\]

Recursive definition:

Given \( U_1, \ldots, U_{n-1} \) and \( L_1, \ldots, L_{n-1} \), define

\( U_n \) as the minimal value such that

\[
P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n
\]

and \( L_n \) as the maximal value such that

\[
P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n
\]

where \( \epsilon_n \geq 0 \) with \( \epsilon_n \uparrow \epsilon \) (spending sequence).
Recursive Definition of the Boundaries

Want:

\[ \sup_p \text{RR}_p(\hat{p}) \leq \epsilon \]

Suffices to ensure

\[ P_\alpha(\text{hit } B_U) \leq \epsilon \]
\[ P_\alpha(\text{hit } B_L) \leq \epsilon \]

Recursive definition:
Given \( U_1, \ldots, U_{n-1} \) and \( L_1, \ldots, L_{n-1} \), define

\( U_n \) as the minimal value such that

\[ P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n \]

and \( L_n \) as the maximal value such that

\[ P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n \]

where \( \epsilon_n \geq 0 \) with \( \epsilon_n \uparrow \epsilon \) (spending sequence).
Recursive Definition of the Boundaries

Want:

\[ \sup_{p} RR_{p}(\hat{p}) \leq \epsilon \]

Suffices to ensure

\[ P_{\alpha}(\text{hit } B_{U}) \leq \epsilon \]
\[ P_{\alpha}(\text{hit } B_{L}) \leq \epsilon \]

Recursive definition:

Given \( U_{1}, \ldots, U_{n-1} \) and \( L_{1}, \ldots, L_{n-1} \), define

\( U_{n} \) as the minimal value such that

\[ P_{\alpha}(\text{hit } B_{U} \text{ until } n) \leq \epsilon_{n} \]

\( L_{n} \) as the maximal value such that

\[ P_{\alpha}(\text{hit } B_{L} \text{ until } n) \leq \epsilon_{n} \]

where \( \epsilon_{n} \geq 0 \) with \( \epsilon_{n} \uparrow \epsilon \) (spending sequence).
Recursive Definition of the Boundaries

Want:

\[
\sup_p \text{RR}_p(\hat{p}) \leq \epsilon
\]

Suffices to ensure

\[
P_\alpha(\text{hit } B_U) \leq \epsilon
\]
\[
P_\alpha(\text{hit } B_L) \leq \epsilon
\]

Recursive definition:
Given \(U_1, \ldots, U_{n-1}\) and \(L_1, \ldots, L_{n-1}\), define

\(U_n\) as the minimal value such that

\[
P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n
\]

and \(L_n\) as the maximal value such that

\[
P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n
\]

where \(\epsilon_n \geq 0\) with \(\epsilon_n \uparrow \epsilon\) (spending sequence).
Recursive Definition - Example

- $\alpha = 0.2, \epsilon_n = 0.4 \frac{n}{5+n}$.
- $U_n =$ the minimal value such that
  $$P_{\alpha}(\text{hit } B_U \text{ until } n) \leq \epsilon_n$$
- $L_n =$ maximal value such that
  $$P_{\alpha}(\text{hit } B_L \text{ until } n) \leq \epsilon_n$$

<table>
<thead>
<tr>
<th>$P_{\alpha}(S_n = k, \tau \geq n)$</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 3$</td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td></td>
</tr>
<tr>
<td>$k = 1$</td>
<td></td>
</tr>
<tr>
<td>$k = 0$</td>
<td>1</td>
</tr>
<tr>
<td>$\epsilon_n$</td>
<td>0</td>
</tr>
<tr>
<td>$U_n$</td>
<td>1</td>
</tr>
<tr>
<td>$L_n$</td>
<td>-1</td>
</tr>
</tbody>
</table>
### Recursive Definition - Example

- \( \alpha = 0.2, \ \epsilon_n = 0.4 \frac{n}{5+n} \).
- \( U_n = \) the minimal value such that 
  \[
  P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n
  \]
- \( L_n = \) maximal value such that 
  \[
  P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n
  \]

<table>
<thead>
<tr>
<th>( P_\alpha(S_n = k, \tau \geq n) )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = 3 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( k = 2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( k = 1 )</td>
<td></td>
<td>.2</td>
</tr>
<tr>
<td>( k = 0 )</td>
<td>1</td>
<td>.8</td>
</tr>
<tr>
<td>( \epsilon_n )</td>
<td>0</td>
<td>.07</td>
</tr>
<tr>
<td>( U_n )</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>( L_n )</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Recursive Definition - Example

- $\alpha = 0.2$, $\epsilon_n = 0.4 \frac{n}{5+n}$.
- $U_n =$ the minimal value such that
  \[ P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n \]
- $L_n =$ maximal value such that
  \[ P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n \]

<table>
<thead>
<tr>
<th>$P_\alpha(S_n = k, \tau \geq n)$</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 3$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td></td>
<td>.04</td>
<td></td>
</tr>
<tr>
<td>$k = 1$</td>
<td>.2</td>
<td>.32</td>
<td></td>
</tr>
<tr>
<td>$k = 0$</td>
<td>1</td>
<td>.8</td>
<td>.64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\epsilon_n$</th>
<th>0</th>
<th>.07</th>
<th>.11</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_n$</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$L_n$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Recursive Definition - Example

- \( \alpha = 0.2, \ \epsilon_n = 0.4 \frac{n}{5+n}. \)
- \( U_n = \) the minimal value such that
  \[ P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n \]
- \( L_n = \) maximal value such that
  \[ P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n \]

<table>
<thead>
<tr>
<th>( P_\alpha(S_n = k, \tau \geq n) )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = 3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( k = 2 )</td>
<td></td>
<td>.04</td>
<td>.06</td>
<td></td>
</tr>
<tr>
<td>( k = 1 )</td>
<td>.2</td>
<td>.32</td>
<td>.38</td>
<td></td>
</tr>
<tr>
<td>( k = 0 )</td>
<td>1</td>
<td>.8</td>
<td>.64</td>
<td>.51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \epsilon_n )</th>
<th>0</th>
<th>.07</th>
<th>.11</th>
<th>.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U_n )</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( L_n )</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Recursive Definition - Example

- $\alpha = 0.2$, $\epsilon_n = 0.4 \frac{n}{5+n}$.
- $U_n =$ the minimal value such that
  $$P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n$$
- $L_n =$ maximal value such that
  $$P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n$$

<table>
<thead>
<tr>
<th>$P_\alpha(S_n = k, \tau \geq n)$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td></td>
<td>0.04</td>
<td>0.06</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>$k = 1$</td>
<td></td>
<td>0.2</td>
<td>0.32</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>$k = 0$</td>
<td></td>
<td>1</td>
<td>0.8</td>
<td>0.64</td>
<td>0.51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\epsilon_n$</th>
<th>0</th>
<th>0.07</th>
<th>0.11</th>
<th>0.15</th>
<th>0.18</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_n$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$L_n$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Recursive Definition - Example

- $\alpha = 0.2$, $\epsilon_n = 0.4 \frac{n}{5+n}$.
- $U_n =$ the minimal value such that
  \[ P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n \]
- $L_n =$ maximal value such that
  \[ P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n \]

\[
\begin{array}{c|cccccc}
\alpha & S_n = k, \tau \geq n & & & & & \\
\hline
\epsilon_n & 0 & .07 & .11 & .15 & .18 & .20 \\
\hline
U_n & 1 & 2 & 2 & 2 & 3 & 3 \\
L_n & -1 & -1 & -1 & -1 & -1 & -1 \\
\hline
\end{array}
\]
Recursive Definition - Example

- $\alpha = 0.2$, $\epsilon_n = 0.4 \frac{n}{5+n}$.
- $U_n =$ the minimal value such that
  \[ P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n \]
- $L_n =$ maximal value such that
  \[ P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n \]

<table>
<thead>
<tr>
<th>$P_\alpha(S_n = k, \tau \geq n)$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n =$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$\epsilon_n$</td>
<td>0</td>
<td>.07</td>
<td>.11</td>
<td>.15</td>
<td>.18</td>
<td>.20</td>
<td>.22</td>
</tr>
<tr>
<td>$U_n$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$L_n$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Recursive Definition - Example

- $\alpha = 0.2$, $\epsilon_n = 0.4 \frac{n}{5+n}$.
- $U_n$ = the minimal value such that
  \[ P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n \]
- $L_n$ = maximal value such that
  \[ P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n \]

<table>
<thead>
<tr>
<th>$P_\alpha(S_n = k, \tau \geq n)$</th>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>k= 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k= 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k= 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k= 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\epsilon_n$</th>
<th>0</th>
<th>.07</th>
<th>.11</th>
<th>.15</th>
<th>.18</th>
<th>.20</th>
<th>.22</th>
<th>.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_n$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$L_n$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>
Recursive Definition - Example

- $\alpha = 0.2$, $\epsilon_n = 0.4 \frac{n}{5+n}$.

- $U_n=$ the minimal value such that

  $$P_\alpha(\text{hit } B_U \text{ until } n) \leq \epsilon_n$$

- $L_n =$ maximal value such that

  $$P_\alpha(\text{hit } B_L \text{ until } n) \leq \epsilon_n$$

<table>
<thead>
<tr>
<th>$P_\alpha(S_n = k, \tau \geq n)$</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>
</tr>
</thead>
<tbody>
<tr>
<td>$k=3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k=2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k=1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k=0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\epsilon_n$</td>
<td>0</td>
<td>.07</td>
<td>.11</td>
<td>.15</td>
<td>.18</td>
<td>.20</td>
<td>.22</td>
<td>.23</td>
<td>.25</td>
</tr>
<tr>
<td>$U_n$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$L_n$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Sequential Decision Procedure - Example

\[ \alpha = 0.2, \quad \epsilon_n = 0.4 \frac{n}{5+n}. \]
Influence of $\epsilon$ on the stopping rule

$\epsilon = 0.1, 0.001, 10^{-5}, 10^{-7}; \epsilon_n = \epsilon \frac{n}{1000+n}$
Sequential Estimation based on the MLE

$$\hat{p} = \begin{cases} \frac{S_\tau}{\tau}, & \tau < \infty \\ \alpha, & \tau = \infty, \end{cases}$$

- One can show:
  - hitting the upper boundary implies $\hat{p} > \alpha$,
  - hitting the lower boundary implies $\hat{p} < \alpha$.

Hence,

$$\sup_p RR_p(\hat{p}) \leq \epsilon$$

- Furthermore, $\exists$ random interval $I_n$ s.t.
  - $I_n$ only depends on $X_1, \ldots, X_n$,
  - $\hat{p} \in I_n$. 
Example - Two-way sparse contingency table

\[
\begin{array}{ccccccc}
1 & 2 & 2 & 1 & 1 & 0 & 1 \\
2 & 0 & 0 & 2 & 3 & 0 & 0 \\
0 & 1 & 1 & 1 & 2 & 7 & 3 \\
1 & 1 & 2 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 \\
\end{array}
\]

- $H_0$: variables are independent.
- Reject for large values of the likelihood ratio test statistic $T$.
- $T \xrightarrow{d} \chi^2_{(7-1)(5-1)}$ under $H_0$. Based on this: $p = 0.031$.
- Matrix sparse - approximation poor?
- Use parametric bootstrap based on row and column sums.
- Naive test statistic $\hat{p}_{naive}$ with $n = 1,000$ replicates: $p = 0.041 < 0.05$.
- Probability of reporting $p > 0.05$: roughly 0.08.
Example - Two-way sparse contingency table

\[
\begin{array}{cccccccc}
1 & 2 & 2 & 1 & 1 & 0 & 1 \\
2 & 0 & 0 & 2 & 3 & 0 & 0 \\
0 & 1 & 1 & 1 & 2 & 7 & 3 \\
1 & 1 & 2 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 \\
\end{array}
\]

- $H_0$: variables are independent.
- Reject for large values of the likelihood ratio test statistic $T$
- $T \overset{d}{\to} \chi^2_{(7-1)(5-1)}$ under $H_0$. Based on this: $p = 0.031$.
- Matrix sparse - approximation poor?
- Use parametric bootstrap based on row and column sums.
- Naive test statistic $\hat{p}_{naive}$ with $n = 1,000$ replicates: $p = 0.041 < 0.05$.
  Probability of reporting $p > 0.05$: roughly 0.08.
Example - Two-way sparse contingency table

```
1  2  2  1  1  0  1
2  0  0  2  3  0  0
0  1  1  1  2  7  3
1  1  2  0  0  0  1
0  1  1  1  1  0  0
```

- **$H_0$:** variables are independent.
- Reject for large values of the likelihood ratio test statistic $T$
- $T \overset{d}{\rightarrow} \chi^2_{(7-1)(5-1)}$ under $H_0$. Based on this: $p = 0.031$.
- Matrix sparse - approximation poor?
- Use parametric bootstrap based on row and column sums.
- Naive test statistic $\hat{p}_{naive}$ with $n = 1,000$ replicates: $p = 0.041 < 0.05$.
  Probability of reporting $p > 0.05$: roughly 0.08.
Example - Bootstrap and Sequential Algorithm

```r
> dat <- matrix(c(1,2,2,1,1,0,1, 2,0,0,2,3,0,0, 0,1,1,1,2,7,3, 1,1,2,0,0,0,1, 0,1,1,1,1,0,0), nrow=5,ncol=7,byrow=TRUE)
> loglikrat <- function(data){
+ cs <- colSums(data); rs <- rowSums(data); mu <- outer(rs,cs)/sum(rs)
+ 2*sum(ifelse(data<=0.5, 0, data*log(data/mu)))
+ }
> resample <- function(data){
+ cs <- colSums(data); rs <- rowSums(data); n <- sum(rs)
+ mu <- outer(rs,cs)/n/n
+ matrix(rmultinom(1,n,c(mu)),nrow=dim(data)[1],ncol=dim(data)[2])
+ }
> t <- loglikrat(dat);
> library(simctest)
> res <- simctest(function(){loglikrat(resample(dat))>=t},maxsteps=1000)
> res
No decision reached.
Final estimate will be in [ 0.02859135 , 0.07965451 ]
Current estimate of the p.value: 0.041
Number of samples: 1000
> cont(res, steps=10000)
> p.value: 0.04035456
Number of samples: 8574
```
Further Uses of the Algorithm

- Simulation study to evaluate whether a test is liberal/conservative.
- Determining the sample size to achieve a certain power.
- Iterated Use:
  - Determining the power of a bootstrap test.
  - Simulation study to evaluate whether a bootstrap test is liberal/conservative.
  - Double bootstrap test.
Expected Hitting Time

Result: $E_p(\tau) < \infty \ \forall p \neq \alpha$

Example with $\alpha = 0.05, \epsilon_n = \epsilon \frac{n}{1000+n}$:

$\mu_p = \text{theoretical lower bound on } E_p(\tau)$.

▶ Note: $\int_0^1 \mu_p \, dp = \infty$;
▶ for iterated use: Need to limit the number of steps.
Expected Hitting Time

Result: $E_p(\tau) < \infty \ \forall p \neq \alpha$

Example with $\alpha = 0.05$, $\epsilon_n = \epsilon \frac{n}{1000+n}$:

$\mu_p =$ theoretical lower bound on $E_p(\tau)$.

- Note: $\int_0^1 \mu_p \, dp = \infty$;
- for iterated use: Need to limit the number of steps.
Summary

▶ Sequential implementation of Monte Carlo Tests and computation of $p$-values.
▶ Useful when implementing tests in packages.
▶ After a finite number of steps:
  ▶ $\hat{p}$ or
  ▶ interval $[\hat{p}_n^L, \hat{p}_n^U]$ in which $\hat{p}$ will lie.
▶ Guarantee (up to a very small error probability):
  $\hat{p}$ is on the “correct side” of $\alpha$.
▶ R-package `simctest` available on CRAN.
  (efficient implementation with C-code)
▶ For details see Gandy (2009).
References


Gandy, A. (2009). Sequential implementation of Monte Carlo tests with uniformly bounded resampling risk. Accepted for publication in JASA.