Introduction to the Randomness Recycler

Mark Huber
Dept. of Mathematics and Institute of Statistics and Decision Sciences
Duke University
mhuber@math.duke.edu
www.math.duke.edu/~mhuber

Joint work with:

James Allen Fill
Dept. of Mathematical Sciences
The Johns Hopkins University
jimfill@jhu.edu
www.mts.jhu.edu/~fill
When don't we use Markov chains?

Shuffling cards
(Generating permutations uniformly at random)

- Given a deck do use Markov chains
- Computers use a different method

1 2 3 (swap 1) 1 2 3
1 2 3 (swap 2) 2 1 3
2 1 3 (swap 3) 3 1 2
**Building up a permutation**

Given:

\[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\]

after 9 steps:

\[7\ 1\ 9\ 6\ 3\ 2\ 5\ 8\ 4\]

Properties:
- Linear Time
- Perfect Sampling Algorithm
Can this idea be used elsewhere?

Bivariate process:

$$(X_0^*, X_0), (X_1^*, X_1), \ldots, (X_n^*, X_n)$$

$X_i$ determines state after $i$ steps

$X_i^*$ determines distribution after $i$ steps

Example

$$X_6 = 5 \ 6 \ 1 \ 4 \ 2 \ 3 \ 7 \ 8 \ 9$$

$$X_6^* = - \ - \ - \ - \ - \ - \ - \ - \ 7 \ 8 \ 9$$
This talk...

- The Randomness Recycler
  A semigeneral technique for creating these bivariate chains
- Illustration with a simple example
- Comparison to other perfect sampling methods
The Problem

Given a finite measure $\mu$, simulate from:

$$\pi(\cdot) = \frac{\mu(\cdot)}{\mu(\Omega)}$$

$\Omega$ state space of the problem

Often $NP$ or $\#P$ hard to compute
A Perfect Simulation algorithm delivers samples exactly drawn from $\pi$ without needing to find $\mu(\Omega)$

Approximate simulation algorithms delivers samples from some distribution close to $\mu$, without the need to know $\mu(\Omega)$

Direct simulation algorithms compute $\mu(\Omega)$ exactly
**Classes of Perfect Samplers**

- **Complete Couplers**
  Coupling from the past (Propp, Wilson '96)
  (3 problems: noninterruptible, read twice, not linear time because of underlying chain)
  FMMR (Fill, Machida, Murdoch, Rosenthal '00)
  Read once CFTP (Wilson '00)
  Linear time CFTP (Haggstrom, Steif '96)

- **Finding the Distribution**
  Strong stationary stopping times
  (Aldous, Diaconis '87), (Diaconis, Fill '90)
  Randomness recycler (Fill, H. '99)
Ingredients:

\[ X_t \in \Omega \]

\[ X^*_t \in \Omega^* \]

Family \( \Lambda \) of distributions indexed by \( \Omega^* \)

Starting distribution \( X^*_0 \in \Omega^* \)

Starting state \( X_0 \sim \Lambda X^*_0 \)

Special state \( x^*_\pi \) where \( \Lambda x^*_\pi = \pi \)
The Algorithm

Want Markov-like property:

\[ P \left( X_t = x_t \mid X_t^* = x_t^*, \ldots, X_0^* = x_0^* \right) \]

\[ = P \left( X_t = x_t \mid X_t^* = x_t^* \right) \]

\[ = \Lambda_{x_t^*}(x_t) \]

The Algorithm:

1) Start at \( (X_0^*, X_0) \)
2) Take steps from \( (X_t^*, X_t) \to (X_{t+1}^*, X_{t+1}) \)
3) Until \( X_t^* = x_{\pi}^* \)
4) Output \( X_t \) as sample from \( \pi \)
The Randomness Recycler

Given $X_t^*$ find $X_a^*$ that is "closer" to $x_\pi^*$

Generate proposal state $X_a$ from $X_t$

Accept $(X_a^*, X_a)$ as new state with probability calculated so that $X_a \sim \Lambda X_a^*$

If reject move to state $(X_r^*, X_t)$

$(X_t^*, X_t)$ \rightarrow $(X_a^*, X_a)$ \rightarrow $(X_r^*, X_t)$
Metropolis Hastings

Generate proposal state $X_a$ from $X_t$

Accept $X_a$ as new state with probability calculated so that $X_t \sim \pi \rightarrow X_a \sim \pi$

If reject stay at state $X_t$
 Compare and Contrast

- Both RR and MH have process $X_0, X_1, X_2, \ldots$
- For both, finding acceptance probability easy
- RR also keeps track of distributions with $X^*_0, X^*_1, X^*_2, \ldots$
- RR needs to find $X^*_r$, can be very tricky
Metropolis-Hastings

Given a kernel \( K' \) satisfying:

\[
\pi(\cdot) > 0 \Rightarrow \int_\Omega \pi(dx) K'(x, \cdot) > 0
\]

For \( X_t = x_t \) propose state \( x_a \) using \( K' \)

Make \( X_{t+1} = x_a \) with probability

\[
C \frac{\pi(dx_a)}{\int_\Omega \pi(dx) K'(x, dx_a)}
\]

Else set \( X_{t+1} = x_t \)
For \( X_t^* = x_t^* \) choose \( x_a^* \)

Given a kernel \( K' \) satisfying:

\[
\Lambda_{x_a^*}(\cdot) > 0 \Rightarrow \int_{\Omega} \Lambda_{x_t^*}(dx) K'(x, \cdot) > c
\]

If \( X_t = x_t \) propose state \( x_a \) using \( K' \)

Make \( X_{t+1} = x_a \) with probability

\[
C(x_t^*, x_a^*) \frac{\Lambda_{x_a^*}(dx_a)}{\int_{\Omega} \Lambda_{x_t^*}(dx) K'(x, dx_a)}
\]

Else set

\[
(X_{t+1}, X_{t+1}^*) = (x_t, x_r^*)
\]
A Simple Example

Goal: uniformly sample from
\[ \Omega = \{1, 2, \ldots, n\} \]

Family of distributions
\[ \Omega^* = \{1, 2, \ldots, n\} \]
\[ x^* = i, \quad \Lambda_{x^*} = \text{Unif} \{1, \ldots, i\} \]

Note:
\[ \Lambda_n = \text{Unif} \{1, \ldots, n\} \]

So
\[ x^*_\pi = n \]
Proposal Distribution

Randomly take a step to the left or the right

1/2

1/2

1/2

1/2

1/2

1

2
After one step

When $x_t^* = 4$

<table>
<thead>
<tr>
<th>1/4</th>
<th>1/4</th>
<th>1/4</th>
<th>1/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1/8</td>
<td>1/8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Accept/Reject

Accept state $x_a^* = 5$

Accept Purple

Rejection Yellow

If $x_a \in \{4, 5\}$ then always accept

If $x_a \in \{1, 2, 3\}$ then accept with probability $1/2$

If reject, move to $x_r^* = 3$
Given \( x_t^* = i \)

\[
P(x_{t+1}^* = i + 1) = \frac{1}{2} \left( 1 + \frac{1}{i} \right)
\]

\[
P(x_{t+1}^* = i - 1) = \frac{1}{2} \left( 1 - \frac{1}{i} \right)
\]

So

\[
E\left[ x_{t+1}^* \mid x_t^* \right] = x_t^* + \frac{1}{x_t^*}
\]

Expected time until \( x_t^* = n \) is at most \( O(n^2) \)
**RR and mixing times**

- When proposal distribution a Markov chain, running time bounds mixing time

- Linked via Strong Stationary Stopping Times

- On the other hand, proposal distribution does not have to be an MC

- Allows RR to beat Markov chain time bounds

- For our example a better proposal gives $O(n)$ algorithm
What's in a name?

Why call this recycling?

- Difficulty for RR is designing family of distribution, and finding $x^*_r$.

- In example, able to "recycle" much of the sample, with $x^*_r = x^*_t - 1$.

- Good recycling key to a fast algorithm.
<table>
<thead>
<tr>
<th>Other Problems</th>
<th>Approximate</th>
<th>CFTP</th>
<th>RR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proper Colorings</td>
<td>$k &gt; 11 \Delta/6$</td>
<td>$k &gt; \Delta(\Delta+1)$</td>
<td>($\text{Fill, H. '99}$)</td>
</tr>
<tr>
<td></td>
<td>(Vigoda '00)</td>
<td>(H. '97)</td>
<td>$k &gt; 2.3 \Delta$</td>
</tr>
<tr>
<td>Random Cluster</td>
<td>$p &lt; 1/\Delta$</td>
<td>$p &lt; 1/\Delta$</td>
<td>$p &lt; 2/\Delta$</td>
</tr>
<tr>
<td>Autonormal</td>
<td>$O(n \log n)$</td>
<td>$O(n \log n)$</td>
<td>Fast in practice</td>
</tr>
<tr>
<td></td>
<td>(Gibbs '01)</td>
<td>(Wilson '00)</td>
<td></td>
</tr>
<tr>
<td>Hard Core Gas</td>
<td>$\lambda &lt; 2/((\Delta-2)$</td>
<td>$\lambda &lt; 2/((\Delta-2)$</td>
<td>$\lambda &lt; \frac{4}{3\Delta-2}$</td>
</tr>
<tr>
<td></td>
<td>(Dyer Greenhill '00)</td>
<td>(H. '99)</td>
<td></td>
</tr>
<tr>
<td>Move Ahead 1</td>
<td>$r &lt; .2$</td>
<td>$r &lt; .2$</td>
<td>$r &lt; .999$ in practice</td>
</tr>
<tr>
<td></td>
<td>(H. '99)</td>
<td>(H. '99)</td>
<td></td>
</tr>
<tr>
<td>Two sided interval permutations</td>
<td>$O(n^{26})$</td>
<td>Not tested</td>
<td>Fast in practice</td>
</tr>
<tr>
<td></td>
<td>Jerrum, Sinclair,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vigoda '01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- RR technique for difficult to obtain samples
  - Does not use classical Markov chains
  - Different from complete coupling perfect samplers (CFTP and FMMR)
  - Interruptible and read once
- First read once interruptible linear time algorithms for several problems of interest
- Big Challenge:
  - Handling recycling automatically