Perfect Simulation of Processes with Long Memory [arXiv:1106.5971]

Aurélien Garivier

CNRS & Telecom ParisTech

Groupe de travail “Modélisation” de Paris VII, le 31 Mai 2012
Stationary Markov Chains

Markov Chain \((X_t)_{t \in \mathbb{Z}}\) on the finite set \(G = \{1, \ldots, K\}\)

Dynamical System \(X_{t+1} = \phi(U_t, X_t)\)

Kernel \(P(i, \cdot) \in M_1(G)\), such that

\[
\forall i, j \in G, \quad \mathbb{P}(X_{t+1} = j | X_t = i) = P(i, j)
\]

Stationary distribution \(\pi\) such that \(\pi P = \pi\)
Simulating the chain

Problem  given a kernel $P$, simulate a sample path $X_0, X_1, \ldots, X_n$ from the stationary Markov Chain with kernel $P$

Update rule  $\phi : [0, 1[ \times \{1, \ldots, K\} \to \{1, \ldots, K\} \text{ such that }$

$$\forall i, j \in G : \quad \lambda(\{u : \phi(u, i) = j\}) = P(i, j)$$

Recursion  Given $X_t$, taking $X_{t+1} = \phi(U_t, X_t)$ works

$\implies$  it is sufficient to sample $X_0$ from $\pi$. 
Idea: given the sequence $\left( U_t \right)_{t \leq 0}$, I may know $X_0$ even if I do not know the value of $X_{-8}$!
Coupling from the Past: more formally

Local transition  for each \( t < 0 \) let \( f_t : G \to G \) be defined by

\[
f_t(g) = \phi(U_t, g)
\]

Iterated transition  \( F_t = f_{-1} \circ \cdots \circ f_t \)

Propp-Wilson: if you know \( U_t \) for all \( t \geq \tau(n) \), where

\[
\tau(n) = \sup\{ t < 0 : F_t \text{ is constant} \},
\]

then you know \( X_0 \).

Prop: \( \tau(n) \) is of the same order of magnitude as the mixing time of the chain!
The Nummelin update rule

Nummelin coefficient:

\[ A_1 = \sum_{j=1}^{K} \min_{1 \leq i \leq K} P(i, j) \]

Update rule \( \phi : [0, 1[ \times G \to G \) such that

\[ u \leq A_1 \implies \forall i, i' \in G, \ \phi(u, i) = \phi(u, i') \]

Regeneration if \( U_t \leq A_1 \), then \( X_{t+1}, X_{t+2} \ldots \), is independent from \( X_t, X_{t-1}, \ldots \).

\[ \implies \text{alternative coupling from the past: wait for a regeneration!} \]
Outline

1. Coupling From the Past: Propp and Wilson’s algorithm
2. Chains of Infinite Order
3. Perfect Simulation for Chains of Infinite Order
4. Implementing the Algorithm
Context Tree Sources

Variable Length Markov Chains: the order of the chain is allowed to depend on the past according to some tree structure.

Example: $T = \{1, 10, 100, 000\}$

$$\mathbb{P}(X_4 = 00110 | X_{-1}^0 = 10)$$

$$= \mathbb{P}(X_1 = 0 | X_{-1}^0 = 10) \times \mathbb{P}(X_2 = 0 | X_{-1}^1 = 100) \times \mathbb{P}(X_3 = 1 | X_{-1}^2 = 1000) \times \mathbb{P}(X_4 = 1 | X_{-1}^3 = 10001) \times \mathbb{P}(X_5 = 0 | X_{-1}^4 = 100011)$$
Context Tree Sources

Variable Length Markov Chains: the order of the chain is allowed to depend on the past according to some tree structure.

Example: $T = \{1, 10, 100, 000\}$

\[
\mathbb{P}(X_4^4 = 00110|X_0^{-1} = 10) = \mathbb{P}(X_1 = 0|X_0^{-1} = 10) \times \mathbb{P}(X_2 = 0|X_1^{-1} = 100) \times \mathbb{P}(X_3 = 1|X_2^{-1} = 1000) \times \mathbb{P}(X_4 = 1|X_3^{-1} = 10001) \times \mathbb{P}(X_5 = 0|X_4^{-1} = 100011)
\]
Context Tree Sources

Variable Length Markov Chains: the order of the chain is allowed to depend on the past according to some tree structure

Example: $T = \{1, 10, 100, 000\}$

\[
P(X_1^4 = 00110 | X_0^{−1} = 10) = P(X_1 = 0 | X_0^{−1} = 10) \times P(X_2 = 0 | X_1^{−1} = 100) \times P(X_3 = 1 | X_2^{−1} = 1000) \times P(X_4 = 1 | X_3^{−1} = 10001) \times P(X_5 = 0 | X_4^{−1} = 100011)
\]
Context Tree Sources

Variable Length Markov Chains: the order of the chain is allowed to depend on the past according to some tree structure

Example: $T = \{1, 10, 100, 000\}$

$$\mathbb{P}(X_1^4 = 00110 | X_{-1}^0 = 10) = \mathbb{P}(X_1 = 0 | X_{-1}^0 = 10) \times \mathbb{P}(X_2 = 0 | X_{-1}^1 = 100) \times \mathbb{P}(X_3 = 1 | X_{-1}^2 = 1000) \times \mathbb{P}(X_4 = 1 | X_{-1}^3 = 10001) \times \mathbb{P}(X_5 = 0 | X_{-1}^4 = 100011)$$
Chains of Infinite Order

Context Tree Sources

Variable Length Markov Chains: the order of the chain is allowed to depend on the past according to some tree structure

Example: \( T = \{1, 10, 100, 000\} \)

\[
P(X_1^4 = 00110|X_{-1}^0 = 10) = P(X_1 = 0|X_{-1}^0 = 10) \times P(X_2 = 0|X_{-1}^1 = 100) \times P(X_3 = 1|X_{-1}^2 = 1000) \times P(X_4 = 1|X_{-1}^3 = 10001) \times P(X_5 = 0|X_{-1}^4 = 100011)
\]
Context Tree Sources

Variable Length Markov Chains: the order of the chain is allowed to depend on the past according to some *tree structure*

Example: \( T = \{1, 10, 100, 000\} \)

\[
\begin{align*}
\mathbb{P}(X_4 = 00110 | X_{-1} = 10) &= \mathbb{P}(X_1 = 0 | X_{-1} = 10) \times \mathbb{P}(X_2 = 0 | X_{-1} = 100) \times \mathbb{P}(X_3 = 1 | X_{-1} = 1000) \times \mathbb{P}(X_4 = 1 | X_{-1} = 10001) \times \mathbb{P}(X_5 = 0 | X_{-1} = 100011) \\
&= 3/4 \times 1/3 \times 4/5 \times 1/3 \times 2/3
\end{align*}
\]
Histories

History \( w = w_{-\infty: -1} \in G^{-\mathbb{N}^*} \)

Ultrametric distance \( \delta(w, z) = 2^{\sup\{k < 0 : w_k \neq z_k\}} \)
\( \implies (G^{-\mathbb{N}^*}, \delta) \) is a complete and compact set.

Ball \( B \subset G^{-\mathbb{N}^*} \) is a (closed or open) ball if
\[
B = \left\{ zs : z \in G^{-\mathbb{N}^*} \right\} \text{ for some } s \in G^*
\]

Trees and roots \( B = \mathcal{T}(s), s = \mathcal{R}(B) \)

Ex: \( \mathcal{T}(\varepsilon) = G^{-\mathbb{N}^*} \), the radius of \( \mathcal{T}(s) \) is \( 2^{-|s|-1} \)

Piecewise constant A mapping \( f \) defined on \( G^{-\mathbb{N}^*} \) is piecewise constant if the exists a family \( \{s_j\}_{j \in \mathbb{N}} \) of elements of \( G^{-\mathbb{N}^*} \) such that \( f \) is constant on each ball \( \mathcal{T}(s_j) \).

Projection \( \Pi^n : G^{-\mathbb{N}^*} \rightarrow G^n \) be defined by \( \Pi^n(w) = w_{n:-1} \).
Kernels

Kernel \( P : G^{-\mathbb{N}^*} \to \mathcal{M}_1(G) \)

Total Variation distance: for \( p, q \in \mathcal{M}_1(G) \),

\[
|p - q|_{TV} = \frac{1}{2} \sum_{a \in G} |p(a) - q(a)| = 1 - \sum_{a \in G} p(a) \wedge q(a)
\]

Process \( (X_t)_{t \in \mathbb{Z}} \) with distribution \( \nu \) on \( G^\mathbb{Z} \) is compatible with kernel \( P \) if the latter is a version of the one-sided conditional probabilities of the former:

\[
\nu (X_i = g | X_{i+j} = w_j, j \in -\mathbb{N}^*) = P(g|w)
\]

for all \( i \in \mathbb{Z}, g \in G \) and \( \nu \)-almost every \( w \).
Kernel continuity

continuity \( P : (G^{-\mathbb{N}^*}, \delta) \rightarrow (\mathcal{M}_1(G), | \cdot |_{TV}) \)

oscillation of \( P \) on the ball \( \mathcal{T}(s) \)

\[ \eta(s) = \sup \left\{ |P(\cdot |w) - P(\cdot |z)|_{TV} : w, z \in \mathcal{T}(s) \right\}. \]

P1: \( P \) is continuous if and only if
\[ \forall w \in G^{-\mathbb{N}^*}, \eta(w_{-k::-1}) \rightarrow 0 \] as \( k \) goes to infinity.

P2: \( P \) is continuous if and only if
\[ \sup \{ \eta(s) : s \in G^{-k} \} \rightarrow 0 \] as \( k \) goes to infinity.

P3: \( P \) is uniformly continuous if and only if it is continuous.
Outline

1. Coupling From the Past: Propp and Wilson’s algorithm
2. Chains of Infinite Order
3. Perfect Simulation for Chains of Infinite Order
4. Implementing the Algorithm
Existing CFP algorithms

Comets, Fernandez, Ferrari 2002 simulation algorithm using a Kalikow-type decomposition of the kernel as a mixture of Markov Chains of all orders. Require strong continuity conditions.

De Santis, Piccioni mix the ideas of CFF and the algorithm of PW: they propose an hybrid simulation scheme working with a Markov regime and a long-memory regime.

Gallo 2010 Relaxes the continuity condition, replaced by conditions on the shape of the memory tree.

Our goal: describe a single procedure that generalizes the sampling schemes of CFF and PW in an unified framework.
Update rules

**Def:** $\phi : [0, 1] \times G^{-N^*} \rightarrow G$ is called an *update rule* of $P$ if

$$U \sim \mathcal{U}([0, 1]) \implies \phi(U, w) \sim P(\cdot | w)$$

for all $w \in G^{-N^*}$.

**Prop:** There exists an update rule $\phi$ of $P$ such that:

$$\forall s \in G^*, 0 \leq u < 1 - |G|\eta(s) \implies \phi(u, \cdot) \text{ cst on } T(s).$$

**Prop:** If $P$ is continuous, then for all $u \in [0, 1]$ the mapping

$$w \rightarrow \phi(u, w)$$

is continuous, i.e, piecewise constant.
Perfect Simulation Scheme

Goal: draw \( (X_n, \ldots, X_{-1}) \) from a stationary distribution compatible with \( P \)

Tool: semi-infinite sequence of i.i.d. random variables \( U_t \sim U([0, 1[) \)

Idea: \( S_t = (\ldots, X_{t-1}, X_t), t \in \mathbb{Z} \) is a Markov Chain on \( G^{-\mathbb{N}^*} \), with kernel \( Q \) given by:

\[
\forall w, z \in G^{-\mathbb{N}^*}, \quad Q(w|z) = P(w_{-1}|z) \mathbb{1}_{w_{i-1}=z_i: i<0}.
\]
A Propp-Wilson Scheme

Local transition \( f_t : G^{-\mathbb{N^*}} \rightarrow G^{-\mathbb{N^*}} \) be defined by
\[
f_t(w) = w\phi(U_t, w);
\]

Iterated transition \( F_t = f_{-1} \circ \cdots \circ f_t \)

Projection \( H^n_t = \Pi^n \circ F_t \)

Continuity: \( H^n_t \) is a piecewise constant mapping

Propp-Wilson: if you wait for
\[
\tau(n) = \sup\{ t < n : H^n_t \text{ is constant} \},
\]
you will know \((X_n, \ldots, X_{-1})\)
Local Continuity Coefficients

For every $\underline{w} \in G^{-\mathbb{N}^*}$ the continuity of kernel $P$ is locally characterized by the coefficients

$$a_k(g|w_{-k:-1}) = \inf \{ P(g|z) : z \in \mathcal{T}(w_{-k:-1}) \}$$

$$A_k(w_{-k:-1}) = \sum_{g \in G} a_k(g|w_{-k:-1})$$

$$A_k^- = \inf_{s \in G^{-k}} A_k(s)$$

$$\alpha_k(g|w_{-k:-1}) = A_{k-1}(w_{-k+1:-1}) + \sum_{h < g} \{a_k(h|w_{-k:-1}) - a_{k-1}(h|w_{-k+1:-1})\}$$

$$\beta_k(g|w_{-k:-1}) = A_{k-1}(w_{-k+1:-1}) + \sum_{h \leq g} \{a_k(h|w_{-k:-1}) - a_{k-1}(h|w_{-k+1:-1})\}$$
Local characterization of the kernel continuity

Let $P$ be a fixed kernel on $G$.

**Prop:** For all $s \in G^*$,

$$1 - |G| \eta(s) \leq A_{|s|}(s) \leq 1 - \eta(s).$$

**Prop:** The three assertions are equivalent:

(i) the kernel $P$ is continuous;
(ii) $\forall w \in G^{-\mathbb{N}^*}, A_k(w_{-k};-1) \to 1$ as $k \to \infty$;
(iii) $A_k^{-} \to 1$ as $k$ goes to infinity.
Construction of the update rule

Prop: For every \( w \in G^{-\mathbb{N}^*} \),

\[
[0, 1[ = \bigsqcup_{g \in G, k \in \mathbb{N}} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})].
\]

Def: The mapping \( \phi : [0, 1[ \times G^{-\mathbb{N}^*} \to G \) is defined as follows:

\[
\phi(u, w) = \sum_{g \in G, k \in \mathbb{N}} g [\alpha_k(g), \beta_k(g)](u).
\]

Prop: \( \phi \) is an update rule such that \( \forall s \in G^*, \forall u \in [0, 1] : \)

\[
\forall w, z \in T(s), \quad u < A_{|s|}(s) \implies \phi(u, w) = \phi(u, z).
\]
Figure: Graphical representation of an update rule $\phi$ on alphabet \{0, 1, 2\}: for each $w_{-k:-1}$, the intervals $[\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})]$ are represented in blue ($g = 0$), red ($g = 1$) and green ($g = 2$). For example, $P(1|1) = \alpha_0(1|\varepsilon) + \alpha_1(1|1) = 1/8 + 1/4$, and $P(0|00) = \alpha_0(0|\varepsilon) + \alpha_1(0|0) + \alpha_2(0|00) = 1/4 + 1/8 + 0$. 
Outline

1. Coupling From the Past: Propp and Wilson’s algorithm
2. Chains of Infinite Order
3. Perfect Simulation for Chains of Infinite Order
4. Implementing the Algorithm
Complete suffix Dictionaries

Def: a (finite or infinite) set of words $D \subset \mathcal{P}(G^*)$ is a CSD if one of the following equivalent properties is satisfied:

- every $w \in G^{-\mathbb{N}^*}$ has a unique suffix in $D$:

$$\forall w \in G^{-\mathbb{N}^*}, \exists! s \in D : w \geq s ;$$

- $\{ \mathcal{T}(s) : s \in D \}$ is a partition of $G^{-\mathbb{N}^*}$:

$$G^{-\mathbb{N}^*} = \bigcup_{s \in D} \mathcal{T}(s).$$

The depth of $D$ is

$$d(D) = \sup\{|s| : s \in D\}$$

The smallest possible CSD is $\{\epsilon\}$: it has depth 0 and size 1. The second smallest is $G$, it has depth 1.
Representation as a trie

A CSD $D$ can be represented by a trie, that is, a tree with edges labelled by elements of $G$ such that the path from the root to any leaf is labelled by an element of $D$.

Figure: Left: the trie representing the Complete Suffix Dictionary $D = \{0, 01, 11\}$. Right: $\{00, 10, 001, 101, 11\} \succeq \{0, 01, 11\}$. Both examples concern the binary alphabet.

If $D$ and $D'$ are such that $\forall s \in D', s \succeq D$, then we note $D' \succeq D$. 
Implementing the Algorithm

**Piecewise constant functions**

**Def:** For a CSD $D$, we say that a function $f$ defined on $G^{-N^*}$ is $D$-constant if

$$\forall s \in D, \forall w \in T(s), f(w) = f(\emptyset s).$$

**Def:** For every $h \in G^{-N^*} \cup G^*$ we define

$$f(h) = f(T(h)) = f(D'(h))$$

and note that if $h \succeq D$, $f(h)$ is a singleton.

**Minimal CSD** $D^f = \text{CSD with minimal cardinality such that } f \text{ is constant on each of its elements.}$

**Pruning** if $f$ is $D$-constant, then $D^f$ can be obtained by recursive pruning of $D$. 
Implementing the Algorithm

Recursive construction of $H_t^n$

The mapping $H_t^n$ being piecewise constant, we define $D_t^n = D^{H_t^n}$.

- **Initialization:** $D_{-1}^{-1} = G, \ \forall g \in G, \forall w \in T(s), H_{-1}(w) = g$.
- For $t < -1, s \in D(U_t)$ denote $\{g_t(s)\} = \phi(U_t, s)$ and define $E_t^n(s)$ as follows:
  - if $s g_t(s) \geq D_{t+1}^n$, let $E_t^n(s) = \{s\}$;
  - otherwise, let $E_t^n(s) = \bigcup_{h g_t(s) \in D_{t+1}^n(s g_t(s))} \{h\}$.

- Let $E_t^n = \bigcup_{s \in D(U_t)} E_t^n(s)$. $E_t^n$ is a CSD, and $H_t^n$ is $E_t^n$-constant.
- $D_t^n$ is obtained by pruning $E_t^n$.
- for $t = n$, $D_t^n$ is equal to $D_t^{t+1}$ unless $D_t^{t+1} = \{\epsilon\}$, in which case $D_t^n = G$. 
Implementing the Algorithm

How it works

\( \partial_k \) and \( \phi(U_k, \cdot) \)

\( \partial_k \) and \( \phi(U_k, \cdot) \)

1 and \( \text{pruning} \)

1

1

Figure: Obtaining \( D_t^n \) from \( D_t \) and \( D_{t+1}^n \). For each function \( \phi(U_t, \cdot) \), \( D_{t+1}^n \) and \( D_t^n \), we represent a CSD on which it is constant, and the values taken in each leaf; here, \( G = \{0, 1\} \).
Example

Renewal process For all $k \geq 0$, let

$$P(0|01^k) = 1 - 1/\sqrt{k}$$

Not Harris Observe that $P(1|0) = \lim_{k \to \infty} P(0|01^k) = 1$, so that $a_0 = 0$.

Slow continuity for $k \geq 0$, $A_{k+1} = A_k(01^k) = 1 - 1/\sqrt{k}$, so that

$$\sum_{n} \prod_{k=2}^{n} A_k^* < \infty$$

$\implies$ the continuity conditions of [Comets, Fernandez, Ferrari] and [De Santis, Piccioni] do not apply.

yet the algorithm works well
Example: the coupling illustrated

Figure: Graphical representation of the of $P$ - blue stands for 0, red stands for 1
Conclusion

The perfect simulation scheme described in this presentation is

Versatile: works as well for Markov Chains and for (mixing) infinite memory processes

Powerful: needs weak continuity assumptions to converge

Fast: for (large order) Markov chains, much faster than Propp-Wilson’s algorithm on the extended chain: all the tries encountered in the algorithm are of size at most $|D| \times d(D) \ll 2^{|D|}$.

but a little hard to implement...