Exact Sampling: The Propp-Wilson Algorithm

Matti Järvisalo

10th November 2003

Based on
Sections 10 and 11 of *O. Häggström. Finite Markov Chains and Algorithmic Applications. Cambridge University Press, 2002.*

and on

J.G. Propp, D.B. Wilson. Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics. Random Structures and Algorithms 9, pp. 223-252, 1996.

Motivation (1/2)

Recall: the objective is to produce random samples according to some given distribution on a finite set

Coupling:

- Produced output "*near enough*" the equilibrium distribution (measured with a suitable metric)
- *Analytic methods* for deriving *upper bounds* on the time of convergence

Problems:

- What's close enough?
- Deriving upper bounds can be tedious

Motivation (2/2)

Exact sampling (Propp-Wilson):

- An *algorithmic* idea
- Produced output distributed *exactly* according to the equilibrium distribution
- *No guaranteed bounds* for time of convergence

Contents

- The Propp–Wilson algorithm
- Sandwiching
  - Attempts to make Propp–Wilson more feasible computationally *for certain cases*
- An example application: the Ising model

Propp-Wilson vs Ordinary MCMC

- Run *multiple copies* of a Markov chain instead of just one
  - The copies will have different initial values
- Run *from the past to present* instead of from the present onwards
  - As we'll see, this is critical
  - "*coupling–from–the–past*"

Propp–Wilson: Preliminaries

Task: Sample from a given probability distribution π an a finite set $S = \{s_1, \ldots, s_k\}$.

- Construct a reversible, irreducible and aperiodic Markov chain with state–space S and stationary distribution π

Let

- P be the transition matrix of the chain
- $U_0, U_{-1}, U_{-2}, \ldots$ be a sequence of i.i.d. random numbers distributed uniformly on $[0, 1]$
- $\phi : S \times [0, 1] \to S$ a valid *update function*
- $N_1 < N_2 < \ldots$, where $N_i \in \mathbb{N}$, e.g. $(N_1, N_2, \ldots) = (1, 2, 4, 8, \ldots)$

The Propp–Wilson Algorithm

1. $m := 1$
2. **For each** $s \in S$
   Starting in s, simulate the Markov chain from time $-N_m$ to time 0 using ϕ with $U_{-N_m+1}, U_{-N_m+2}, \ldots, U_0$
3. **If** all k chains end up in *the same state* s' at time 0
   return s'
   else
   $m := m + 1$
   goto 2

The Propp–Wilson Algorithm Illustrated (1/2)

We have

- $(N_1, N_2, \ldots) = (1, 2, 4, 8, \ldots)$
- $S = \{s_1, s_2, s_3\}$

Here's how it goes:

- $N_1 = 1 \Rightarrow$ start by running the chain from time -1 to 0
- Let's assume that we end up with

$$\begin{cases} \phi(s_1, U_0) = s_1 \\ \phi(s_2, U_0) = s_2 \\ \phi(s_3, U_0) = s_1. \end{cases}$$

- $\phi(s_1, U_0) \neq \phi(s_2, U_0) \Rightarrow$ we back up in time

The Propp–Wilson Algorithm Illustrated (2/2)

- $N_2 = 2 \Rightarrow$ run the chain from time -2 to 0
- Now we end up with

$$\begin{cases} \phi(\phi(s_1, U_{-1}), U_0) = \phi(s_2, U_0) = s_2 \\ \phi(\phi(s_2, U_{-1}), U_0) = \phi(s_3, U_0) = s_1 \\ \phi(\phi(s_3, U_{-1}), U_0) = \phi(s_2, U_0) = s_2. \end{cases}$$

- Backing up once again gives

$$\begin{cases} \phi(\phi(\phi(\phi(s_1, U_{-3}), U_{-2}), U_{-1}), U_0) = \ldots = s_2 \\ \phi(\phi(\phi(\phi(s_2, U_{-3}), U_{-2}), U_{-1}), U_0) = \ldots = s_2 \\ \phi(\phi(\phi(\phi(s_3, U_{-3}), U_{-2}), U_{-1}), U_0) = \ldots = s_2 \end{cases}$$

Victory!

Victory, you say?

That's right:

- Imagine that we would continue by running the chains from times $-8, -16, \ldots$
- We reused the random numbers
  $\Rightarrow$ no matter what state we hit at time -4, we will still end up in state s_2 at time 0
  $\Rightarrow$ running from time -4 equals to running from time ∞ to the present (wow)

The Exact Sampling Property

Let the preliminary assumptions hold.

Theorem. Suppose that the Propp–Wilson algorithm terminates with probability 1, and write Y for it's output. Then, for any $i \in \{1, \ldots, k\}$, we have

$$\mathbf{Pr}(Y = s_i) = \pi_i.$$

Variations that are Intuitive, but Don't Work

"Coupling–to–the–future." Why not just start from time 0 and start chains from all states, and stop when they coalesce?

"Recycle, not". Why bother with storing and reusing the random variable?

Things to Notice about Propp–Wilson

- Possibly an infinite loop
  $\Rightarrow$ might not terminate
- The update function ϕ plays a crucial role
- Even a valid but badly chosen ϕ can lead to termination probability 0
- The choice of $(N_1, N_2, \ldots)$ makes a difference
- The random numbers need to be stored

Sandwiching

- A true set–back with Propp–Wilson:
  Simulating k Markov chains when the state–space is large is infeasible.
- For cases with certain properties, sandwiching is one answer
- Instead of running k chains, we only need to run 2!

Sandwiching Requires Monotonicity

- For sandwiching to work, we need *monotonicity*: to have a partial ordering on the states that is not broken by the update function.
- Intuitively, path starting from a “higher” state never dips below a path starting from a “lower” state.
  $\Rightarrow$ The invention here is that running two chains is enough: one starting with the “top” state and the other from the “bottom” state.
- Ladder walk with a valid update function (a toy example).
- Critical: without a proper ordering–update function –pair sandwiching doesn’t work!

Applying Propp–Wilson: the Ising Model (1/2)

Let $G = (V, E)$ be a graph.

The Ising Model: a way of picking a random element from $\{-1, +1\}^{|V|}$ (the set of *configurations*)

- Main quantities:
  - *temperature*: $T \geq 0$
  - *energy* of a configuration $c \in \{-1, +1\}^{|V|}$:

$$H(c) = -\sum_{\{x,y\} \in E} c(x)c(y)$$

Applying Propp–Wilson: the Ising Model (2/2)

We pick a random configuration a temperature T according to the probability measure

$$\pi(c) = \frac{e^{-H(c)/T}}{\sum_{c' \in \{-1,+1\}^{|V|}} e^{-H(c')/T}}$$

- At $T = 0$ the probability is divided evenly between “all plus” or “all minus” configurations
- With high temperatures “low energy” configurations are favoured
- Physical interpretation and phase transition

Simulation Algorithms for the Ising Model (1/2)

We need

- a sampler (a Markov Chain) for the configuration space; and
- an suitable ordering to enable sandwiching.

The (Gibbs) sampler:

- Given X_n, obtain X_{n+1} by
  - picking a vertex $v \in V$ at random, and
  - updating by

$$X_{n+1}(v) = \begin{cases} +1 & \text{if } U_{n+1} < \pi(X_n(v) = +1 \mid X_n(V \setminus \{v\}) = c_{-v}) \\ -1 & \text{otherwise} \end{cases}$$

Simulation Algorithms for the Ising Model (2/2)

A Propp–Wilson algorithm based on this sampler:

- $2^{|V|}$ chains
- At each time, pick the same vertex to update in all the chains

A partial ordering is intuitive for sandwiching

- For any $c_1, c_2 \in \{-1,+1\}^{|V|}$ we define $c_1 \preceq c_2$ if $c_1(v) \leq c_2(v)$ for all $v \in V$

Conclusion

- The Propp–Wilson algorithm
  - *Algorithmic* idea, *exact* sampling
  - No convergence bounds
  - Practically infeasible for large state–spaces without e.g. *sampling*
  - Care is needed in selecting *the update function* and (if one exists) the *ordering* on the states