Near linear time perfect simulation of corrugated surfaces

Mark Huber

9 Oct, 2010

Corrugated

corrugated *(adjective) shaped into alternating parallel grooves and ridges.*

Corrugated Surface

corrugated surface *(adjective) labeled bipartite graph where one partition consists only of local maxima.*

20	6	16	8	10
5	18	12	23	2
17	11	15	14	19
3	21	13	22	4
24	9	25	1	7

Today's talk

Generating Corrugated Surface

- Near linear time for all bounded degree bipartite graphs
- $O(\Delta^2 \cdot n \ln n)$, $\Delta =$ max degree of graph

Counting Corrugated Surfaces

- Turning problem into exponential family
- Randomized adaptive cooling schedules

Previous work

Caracciolo, Rinaldi, Sportiello (2009)

- First perfect simulation method for corrugated surfaces
- Only tested method on two dimensional square lattices
- No analysis of running time
- Experimental evidence $O(n \ln n)$
- Used Metropolis chain with continuous embedding

Special case of problem of linear extensions

- Linear extension = ranking consistent with partial order
- Current fastest lin. ext. alg: $O(n^3 \ln n)$ (Huber 2006)
- General problem #P complete (Brightwell & Winkler 1991)
- Corrugated surface: complexity unknown

Generating Corrugated Surfaces

First embed problem in continuous space

State spaces

$$\Omega_{\text{discrete}} = S_n,$$
$$\Omega_{\text{continuous}} = [0,1]^n$$

Continuous to discrete

- Each $\vec{x} \in \Omega_{\text{continuous}}$ gives a ranking in Ω_{discrete}:

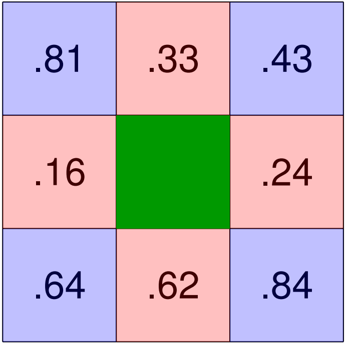
$$\vec{x} = (.32, .48, .14, .99) \Rightarrow (2, 3, 1, 4)$$

- Each permutation = volume $1/n!$

Gibbs sampler in continuous space

.81	.33	.43
.16	.91	.24
.64	.62	.84

Gibbs sampler in continuous space



Choose vertex uniformly

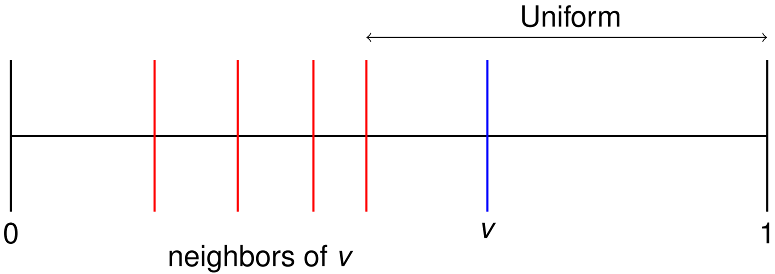
Gibbs sampler in continuous space

.81	.33	.43
.16	.72	.24
.64	.62	.84

Choose vertex uniformly

Choose new value Unif($[.62, 1]$)

Another picture



Taking a step in continuous space

Gibbs sampler Markov chain step

Input: Current state of chain x, *Output:* Next state x

1) Choose a vertex v uniformly at random
2) **if** v is a local maxima
3) **let** b be the largest value of a neighbor of v
4) **draw** $x(v)$ uniformly over $[b, 1]$
5) **else** (v is a local minima)
6) **let** b be the smallest value of a neighbor of v
7) **draw** $x(v)$ uniformly over $[0, b]$

Taking a step in continuous space

Standard implementation

Input: Current state of chain x, $v \in V$, $U \in [0, 1]$
Output: Next state x

1) **if** v is a local maxima
2) **let** b be the largest value of a neighbor of v
3) **let** $x(v) \leftarrow b + (1 - b)U$
4) **else** (v is a local minima)
5) **let** b be the smallest value of a neighbor of v
6) **let** $x(v) \leftarrow bU$

Say $v \sim \text{Unif}(V)$ and $U \sim \text{Unif}([0, 1])$
Takes one step in Markov chain

Perfect simulation

Definition (Perfect simulation algorithm)

A method that draws examples exactly from the target distribution in a random amount of time

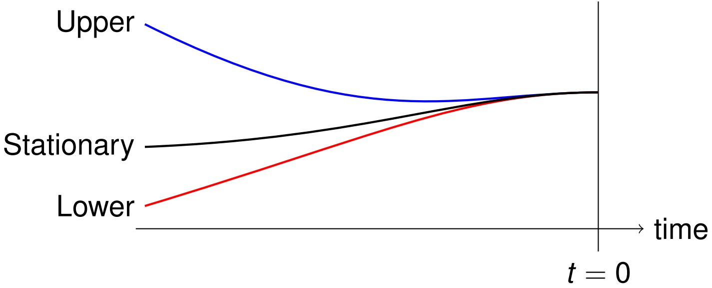
Example (Perfect simulation methods)

Acceptance/Rejection, Monotonic Coupling from the Past, Randomness Recycler, Partially recursive acceptance/rejection

Monotonic CFTP (Propp & Wilson 1996)

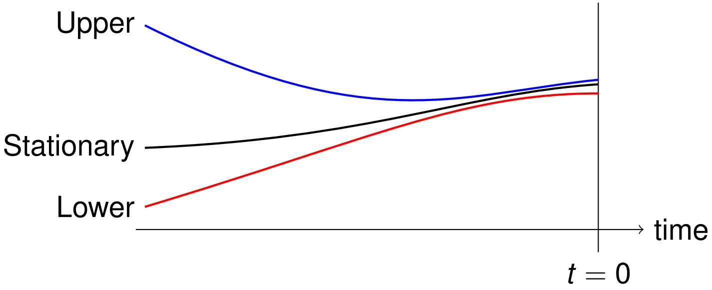
- Need monotonic chain with upper and lower states
- Needs upper and lower states to come together
- (Sandwiches stationary state in between)

Picture of monotonic CFTP



Standard implementation: monotonic, doesn't couple

Say $X \leq Y$ if $X(v) \leq Y(v)$ for all v



Fortunately coupling unnecessary for permuations

Definition (Interleaving)

Upper bound x and lower bound y *interleave* if the sorted values of x and y alternate between x and y

Fact

All points trapped between interleaving upper and lower bounds map to a single permutation

Example

$$x = (.32, .48, .14, .99), \quad y = (.28, .44, .08, .91)$$

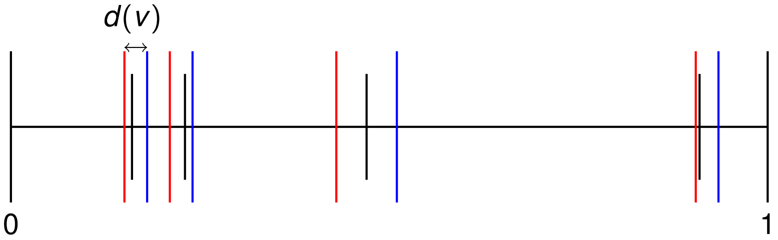
$$.08 < .14 < .28 < .32 < .44 < .48 < .91 < .99.$$

Any z with $y \leq z \leq x$ encodes: $(2, 3, 1, 4)$

Is interleaving likely

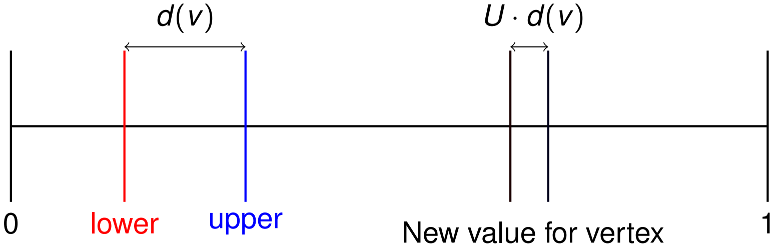
Yes: here's why

- Upper and lower processes contain stationary process
- Smallest distance in stationary process $\Omega(1/n^2)$
- Distance from upper and lower $O(1/n^2)$ gives interleave



Want $d(v) <$ half smallest distance in stationary state

Shrinking $d(v)$



Bounded degree Δ

Each vertex can affect Δ nodes so

$$d(v) < \Delta \mathbb{E}[U \cdot d(v)]$$

So on average $d(v)$ can grow. Solution:

$$d(v)^{2\Delta - 1} \geq \Delta \mathbb{E}[(U \cdot d(v))^{2\Delta - 1}] = (1/2)d \cdot d(v)^{\Delta - 1}$$

On average $d(v)^{2\Delta - 1}$ shrinks by $(1 - (1/2)/n)$ each step

Counting Corrugated Surfaces

Counting for exponential families

Sampling to counting

- Want to use samples to estimate measure of Ω
- Best algorithms are for counting discrete exponential families:

$$H(x) : \Omega \to \{0, 1, 2, \ldots, n\}, \ \pi(\{x\}) \propto \exp(\beta H(x)).$$

- Example: Ising model
- Štefankovič, Vempala, Vigoda (2009) have a method...
- ...requires $O((\ln \#\Omega)\sqrt{\ln n})$ samples

Far from home strategy

Create "home" for each vertex

- For blue vertices (local maxima), home is $[1/2, 1]$
- For red vertices (local minima), home is $[0, 1/2)$
- Let $H(x)$ be number of vertices away from home
- Draw x using $\pi(x) \propto \exp(-\beta H(x))$

How this helps

- Volume of home is known: $(1/2)^n$
- Makes problem in right form for SVV (2009)
- Generation algorithm can also draw from this π

Conclusions

Ideas

1. Gibbs instead of Metropolis
2. Interleave instead of exact coupling
3. Difference upper and lower process $d(v) \to d(v)^{2\Delta-1}$
4. Far from home method for penalizing configurations

Results

- Faster generation & counting corrugated surfaces
- Generation: $O(\Delta^2 n \ln n)$ in graphs with max degree Δ
- Counting: $O(\Delta^2 n^2 (\ln n) \cdot \text{Monte Carlo error factor})$ in graphs with max degree Δ

References

Sergio Caracciolo, Enrico Rinaldi, and Andrea Sportiello.
Exact sampling of corrugated surfaces.
J. Stat. Mech. Theory Exp., Feb 2009.

M. Huber.
Near linear sampling of corrugated surfaces.
preprint, 2010.

D. Štefankovič, S. Vempala, and E. Vigoda.
Adaptive simulated annealing: A near-optimal connection between sampling and counting.
J. of the ACM, 56(3):1–36, 2009.