Perfect simulation of repulsive point processes

Mark Huber

Department of Mathematical Sciences, Claremont McKenna College

29 November, 2011

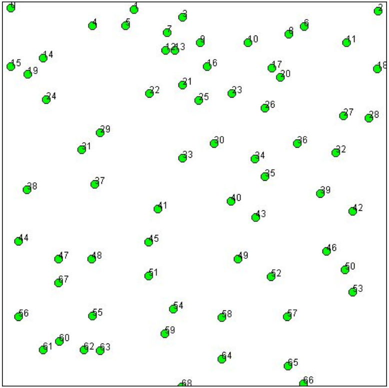
Some repulsive things

Spanish towns

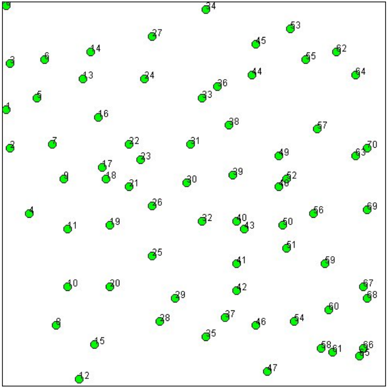
Pine trees

Two data sets

Locations: Spanish towns

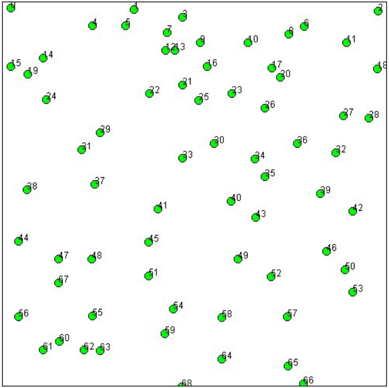


Locations: Swedish pines

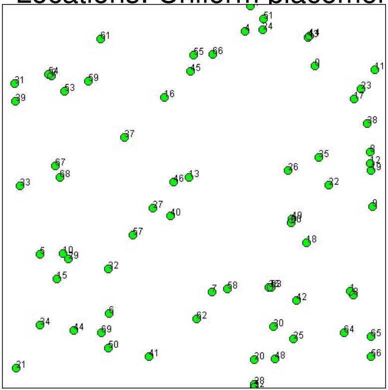


Points farther apart than under uniformity

Locations: Spanish towns



Locations: Uniform placement



Modeling repulsion

Modeling repulsion

Two common modeling approaches

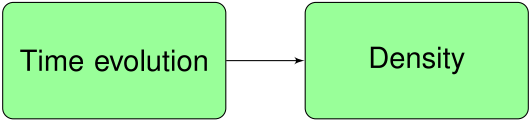
1. Give story about how process developed (time evolution)
2. Family of densities with respect to Poisson point process (density)

Both based on Poisson point process

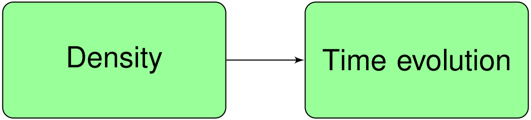
- # of points has Poisson distribution
- Given the # of points, place points uniformly independently
- Mean of Poisson is λ times size of region

Today's talk

Matérn type III process



Strauss process



Strauss process uses density

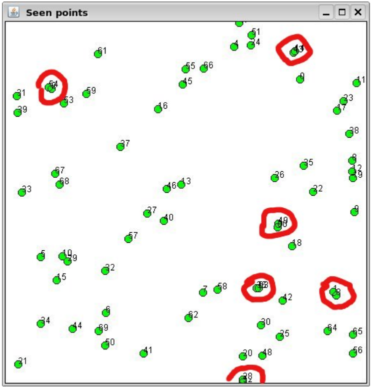
Penalize PPP as follows

- Distance parameter R
- Penalty parameter $\gamma \in (0, 1)$
- Intensity parameter λ
- For points x, let $r(x) =$ # of pairs less than R apart
- For points x, density $f(x) = \gamma^{r(x)} \lambda^{\#x} / Z(\gamma, \lambda)$

Parameter effects

- Big R spaces points farther apart
- Small γ means fewer points violate R distance
- Higher λ means more points

Example of Strauss



$$R = .02$$

$$f(x) = \gamma^6 \lambda^{55} / Z$$

Basic perfect simulation for Strauss

Acceptance rejection

- Draw a PPP X with intensity λ
- Find $r(X)$
- Accept X as Strauss draw with probability $\gamma^{r(X)}$
- Otherwise, reject and start over

Takes too long unless λ small or γ big

Matérn Type I

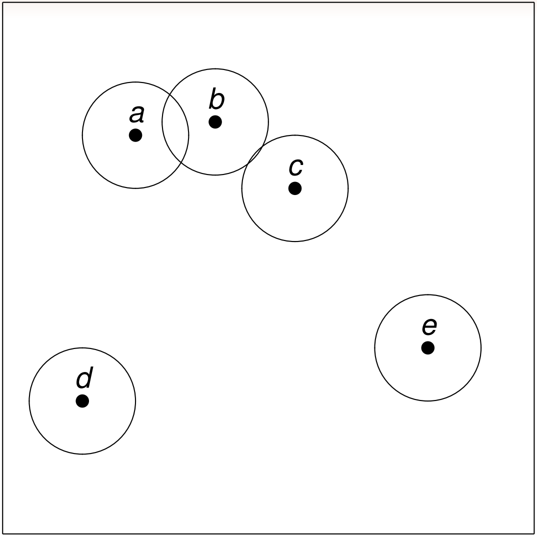
Matérn is a storyteller

- Introduced three ways of explaining the repulsion
- Type I, II, and III

Simulate a Matérn Type I as follows

- First simulate a Poisson point process (PPP)
- Remove any point within R distance of another point

Matérn Type I example

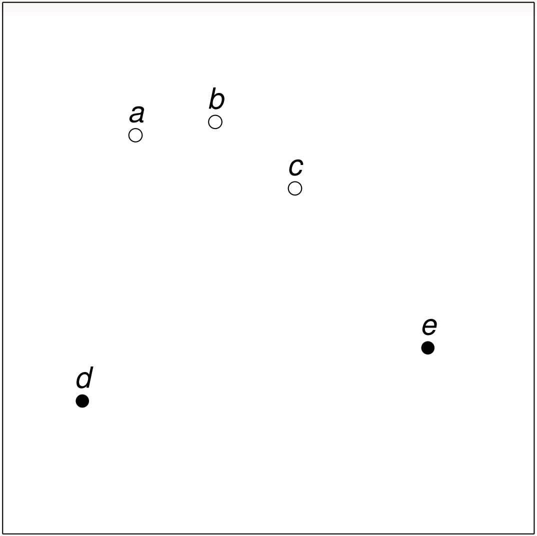


Circles of radius $R/2$

Circles touch =
points eliminated

Points a, b, c eliminated

Type I: After removal



Call *a*, *b*, *c* *ghost points*

Call *d*, *e* *seen points*

Ghost points exert
invisible pressure

Type I: Comments

Problems

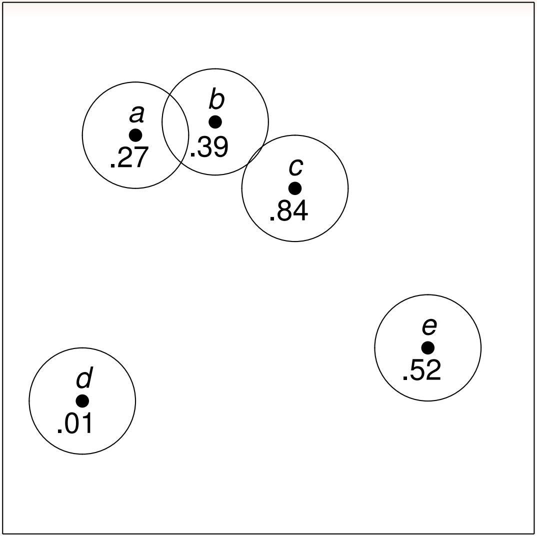
- Too many eliminations
- As $\lambda \to \infty$, # of points $\to 0$
- Need method that preserves some points

Matérn Type II: Older points rule!

Simulate a Matérn Type II as follows

- First simulate a Poisson point process (PPP)
- Assign each point a birthday in $[0, \infty)$
- Remove any point within R distance of an *older* point

Type II: Picture

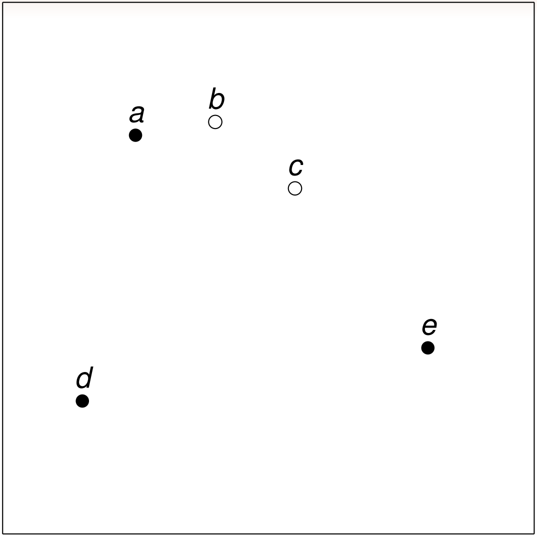


Circles of radius $R/2$

Point *a* eliminates *b*

Point *b* eliminates *c*

Type II: After thinning



Call *b*, *c* *ghost points*

Call *a*, *d*, *e* *seen points*

Ghost points exert
invisible pressure

Type II: Comments

Better, but not perfect

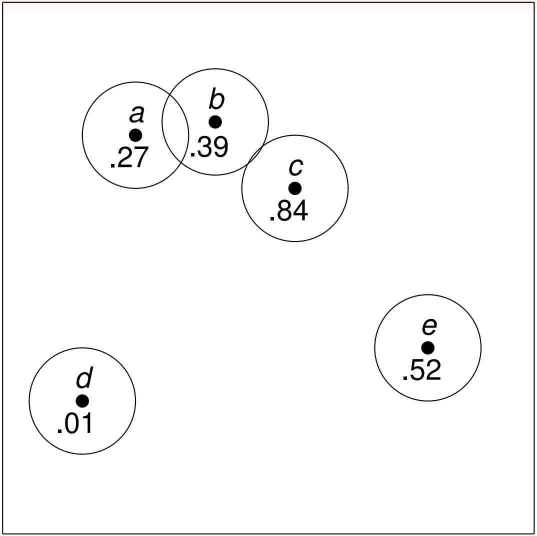
- First points survive
- Higher number of points than Type I
- Why should b take out c if already killed by a?

Matérn Type III

Simulate a Matérn Type III as follows

- First simulate a Poisson point process (PPP)
- Assign each point a birthday in $[0, \infty)$
- Run time forward
- Only allow point birth if not within R of older born point

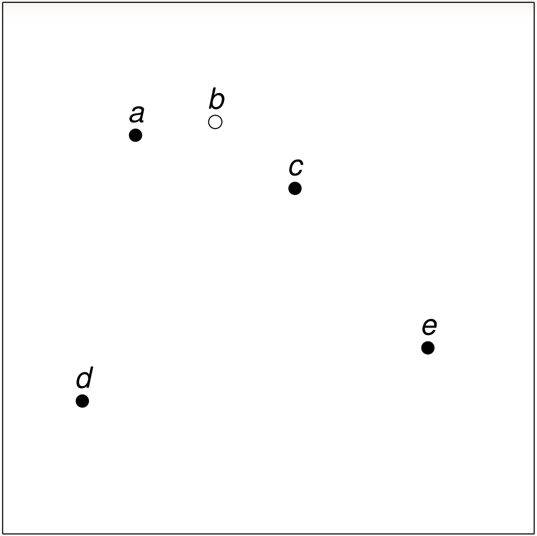
Type III: Picture



Circles of radius $R/2$

Point a eliminates b

Type III: After thinning



Call *c ghost point*

Call *a*, *c*, *d*, *e seen points*

Ghost points exert invisible pressure

Type III: Comments

The good

- Very natural story
- Try to add towns/trees
- If too close to existing town/tree, dies off

The bad

- Density not in closed form–nasty high dimensional integral
- Makes it difficult to do maximum likelihood estimate/posterior

Using Matérn Type III for inference

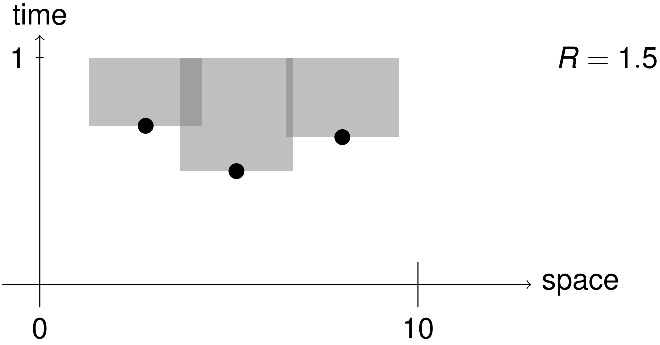
The plan

- First turn story of Matérn into density (as with Strauss)
- Build Markov chain for density
- Build perfect sampler around Markov chain
- Build product estimator to utilize samples effectively
- (Those last two are my area of research)

Matérn Type III: Time evolution to density

Casting shadows

The seen points cast a shadow across time



Any ghost points must lie in shaded region
More shadow = more space for ghost points

From shadows to density

Poisson process in spacetime

- Looks like PPP conditioned so that no points lie in shadow
- Parameters $\theta = (\lambda, R)$, region S
- Let $A_\theta(x, t)$ be the area of the shadow in spacetime
- For seen points x with no shadow violations:

$$f_{\text{seen points}}(x, t|\theta) = C\lambda^{\#x} \exp(\lambda A_\theta(x, t))$$

Larger shadow means more likely

Alternate way of drawing Matérn Type III process

- Fix points in x
- Draw new PPP y to add to x
- If all of y lies in shadow of x, accept x as Matérn Type III
- Otherwise draw new y and repeat

Acceptance/rejection leads to density

What is chance of accepting a given y?

- Let $|S|$ be size of region, $A_\theta(x,t)$ size of shadow
- Probability no points outside of shadow

$$\exp(-\lambda(|S| - A_\theta(x,t))) = \exp(\lambda A_\theta(x,t))\exp(-\lambda|S|)$$

The problem of unknown time of birth

Big problem

- We do not see the t values!
- To get the density for just x integrate out t:

$$g(x|\theta) = C\lambda^{\#x} \int_{t \in [0,1]^{\#x}} \exp(\lambda A_\theta(x,t))$$

- Doing this integral directly extremely nasty
- Even calculating $A_\theta(x,t)$ hard

Monte Carlo methods to the rescue!

Given seen points x, want to approximate $g(x|\theta)$

- Monte Carlo approach: treat t values as auxiliary variables
- Given x: randomly choose t from $f(x, t|\theta, x)$
- Allows use to estimate $g(x|\theta)$

How to draw time stamps given locations?

Markov chain Monte Carlo (MCMC)

1. Build chain so stationary distribution = target distribution
2. Under mild conditions (ϕ-irreducibility, aperiodicity) limiting distribution will equal stationary distribution
3. Run chain “for a long time” to mix, then get samples

Metropolis protocol for building Markov chain

One step of Metropolis

1. Propose moving from (x, t) to (x, t')
2. Accept move with probability
$$\min\left\{\frac{f(x, t'|\theta)}{f(x, t|\theta)}, 1\right\}$$
3. Otherwise stay where you are

Even better...

Perfect simulation techniques

- For some Markov chains, possible to do better
- Can draw exactly from stationary distribution
- Without worrying about mixing time

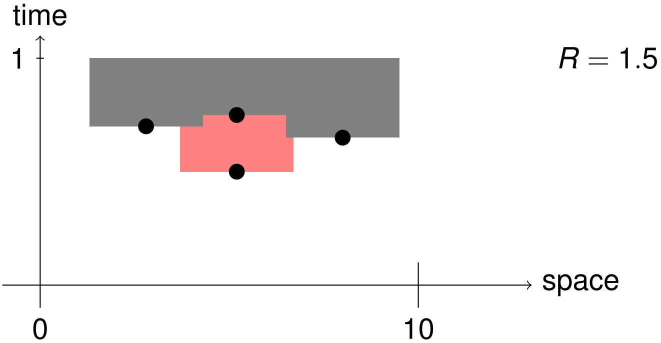
Perfect sampling protocol

- Coupling from the past (Propp & Wilson 1994)
- CFTP converts Markov chains to perfect samplers
- A good property of Markov chain: monotonicity
- Monotonicity not necessary for CFTP, is sufficient

Is Metropolis for Matérn type III monotonic?

Propose changing time stamp for one point in *x*

- How does the shadow change?



- Probability accept move = exp(-area of shadow change)

How to flip an $\exp(-\mu)$ coin

Fun facts about Poisson point processes

- For $X \sim \text{PPP}(B)$, $\mathbb{P}(\#X = 0) = \exp(-\mu(B))$
- For regions $A \subset B$, $X \sim \text{PPP}(B)$ then $X \cap A \sim \text{PPP}(A)$

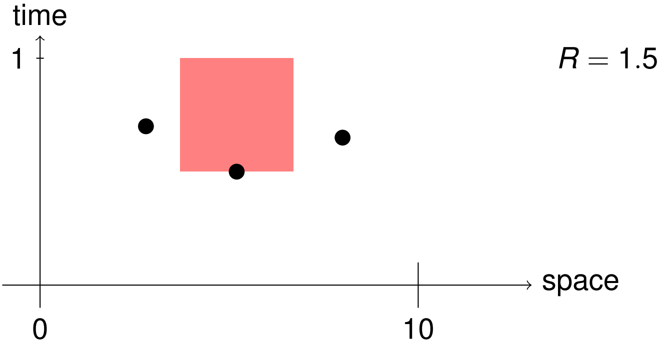
To check if move in Metropolis

- Draw PPP(largest possible change in shadow)
- If PPP restricted to actual change in shadow empty, move
- Otherwise, stay at current time stamp

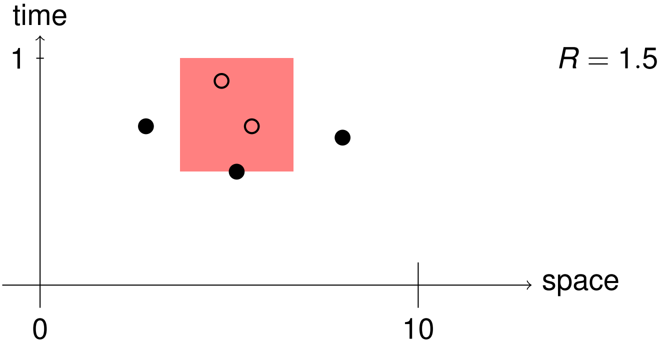
Using PPP to flip exponential coins:

- First appears in Beskos, Papaspiliopoulos, Roberts (2006)
- Perfect simulation of diffusions

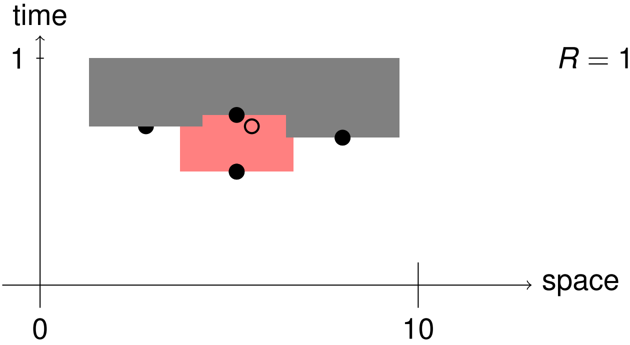
Example



Example



Example



$R = 1.5$

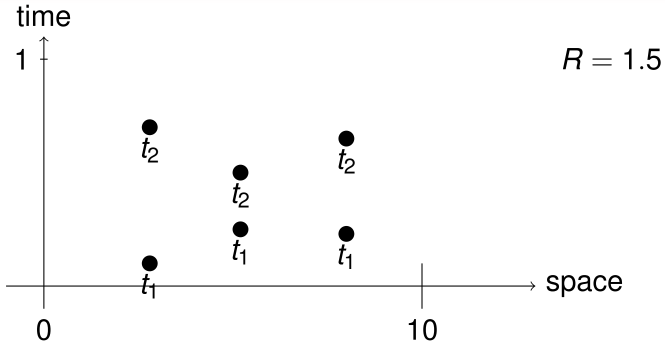
Montonicity of Markov chain step

For $t_1 \leq t_2$:

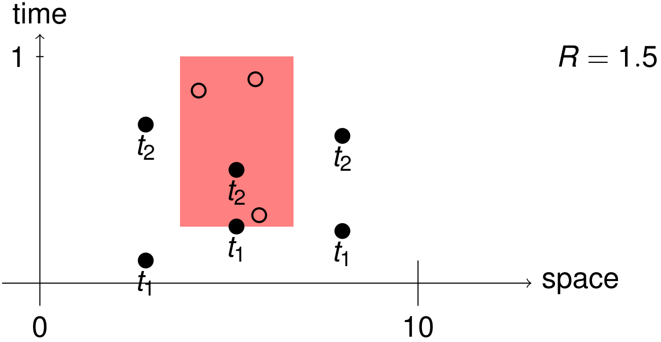
- Run one step of Markov chain for t_1 and t_2
- Use same auxiliary PPP in change of shadow
- Then after step, still have $t_1 \leq t_2$

Immediately gives us perfect sampling!

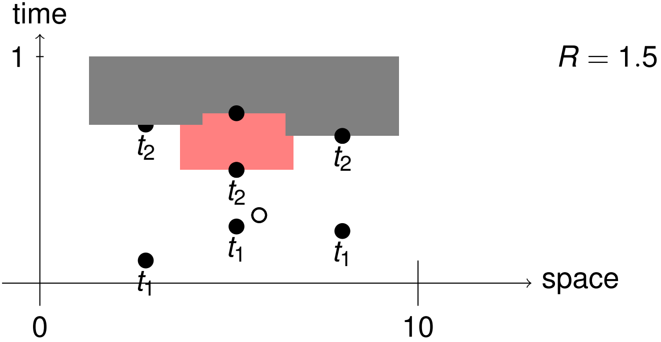
Example of Monotonicity



Example of Monotonicity

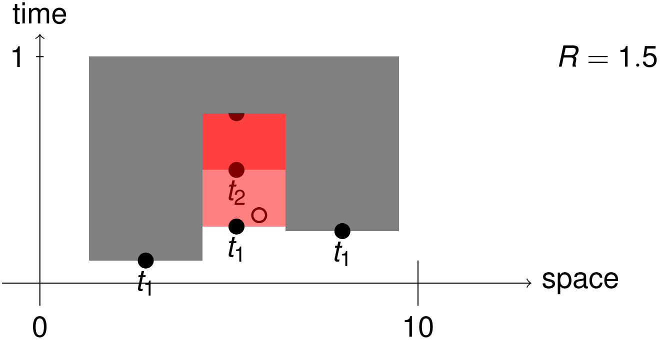


Example of Monotonicity



t_2 accepts move

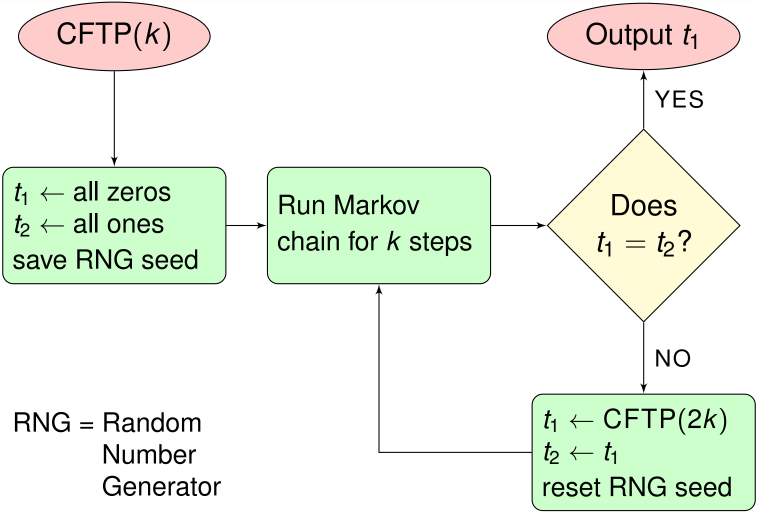
Example of Monotonicity



t_2 accepts move

t_1 rejects move

Monotonic CFTP flowchart



RNG = Random Number Generator

Monotonic CFTP details

Running CFTP(k)

1. Set t_1 to all 0's, t_2 to all 1's
2. Save seed to random number generator
3. Take k steps in the Markov chain
4. If $t_1 = t_2$, then set T_k to be this common value
5. Else
   1. Get T_0 by calling CFTP$(2k)$ recursively
   2. Reset seed to random number generator to what it was in step 2
   3. Get T_k by taking k steps in the Markov chain
6. Output T_k

What's the point?

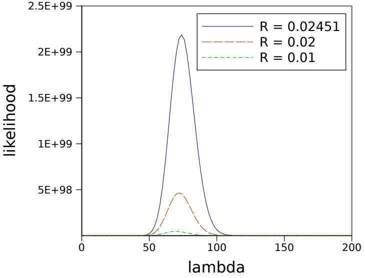
Why do we need samples?

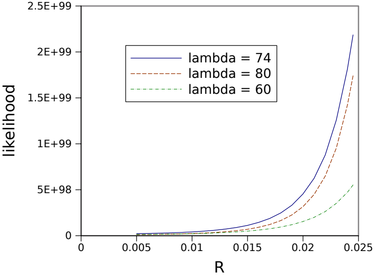
- Ability to sample gives approximation of nasty integral
- Use TPA or IS+TPA to go from samples to integral

Once you have that integral

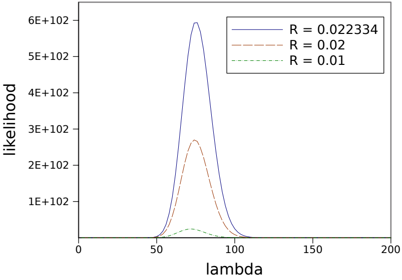
- Gives density of data under Matérn model
- Basis for maximum likelihood
- ...or posterior analysis

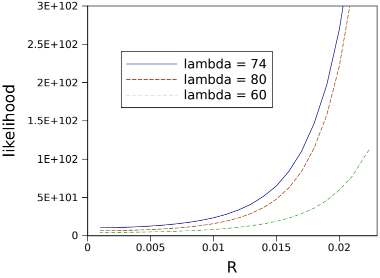
Results: towns





Results: trees

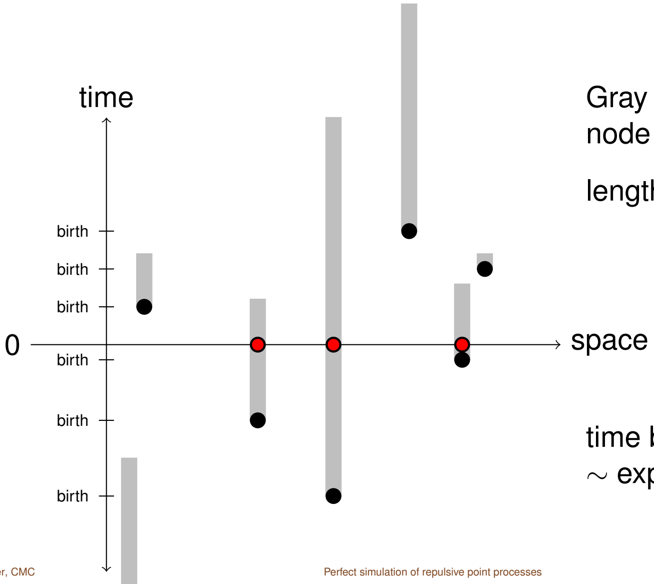




Strauss: Density to time evolution

Preston (1977) Birth-Death Chains

Adding time dimension to Poisson point process



Gray bar =
node lifespan

length bar $\sim$ exp(1)

time between births
$\sim \exp(\lambda\mu(S))$

Preston Birth-Death Chains

Points are born, and later die

- Rate of births is λ times area of region
- Each point dies at rate 1
- (Rate is parameter of exponential random variable)

Points alive at time 0 form PPP

Preston for Strauss

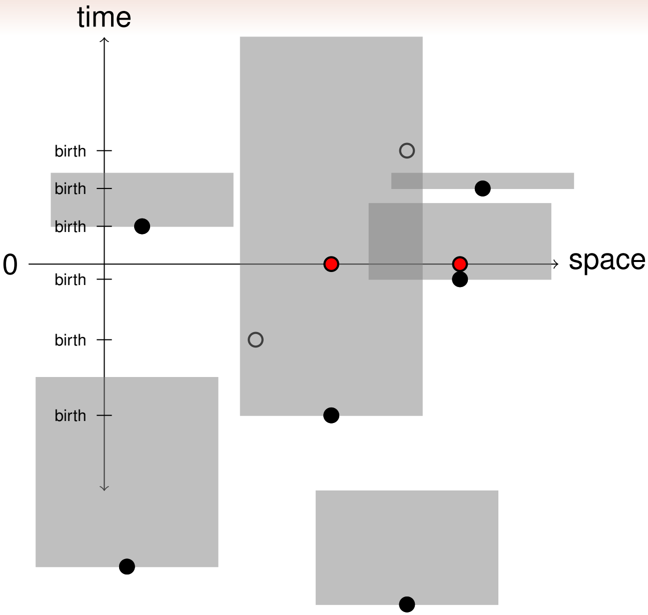
Recall Strauss density

- Penalty γ for pair of points within distance R

When point is born...

- If point within distance R of point already born...
- ...point only born with probability γ

Picture for Strauss $\gamma = 0$



Observations

Red points

- Strauss process subset of earlier points
- Kendall and J. Møller (2000): can find red points by looking backwards in time

Adding a Swap move

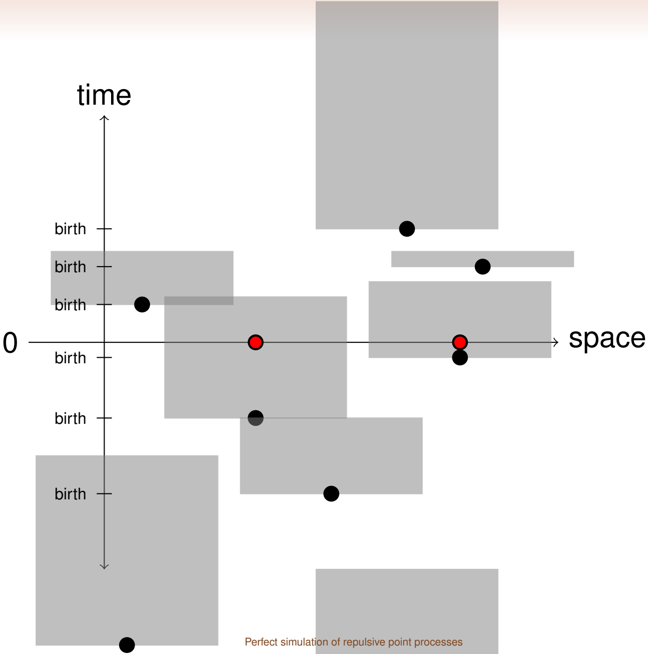
Swap move

- Broder (1986): swap move for perfect matchings
- Dyer and Greenhill (2000): swap move for independent sets of graphs

Adding Swaps to Preston

- Huber (2011)
- When point not born, give chance to swap
- If only blocked by one point...
- ...remove blocker, allow birth

Picture with swap



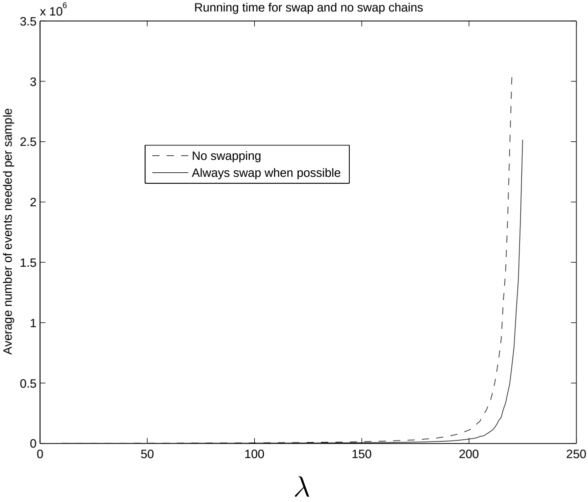
Swap move helps chain mix better

Easier to find red points

- Verified experimentally on plane
- Points affect fewer points in future
- Need to be blocked by at least two points to affect future
- So effect of point on later points cut in half

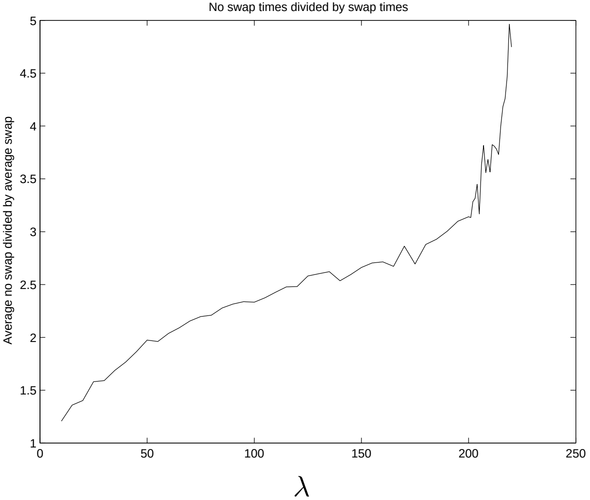
Running times with swap

Strauss model on $S = [0, 1]^2$, $\gamma = 0.5$, $R = .05$



Running times with swap

Time for no swap divided by time for swap



Conclusions

For Matérn type III models

- Built a density (not in closed form)
- Can approximate density using perfect MCMC methods
- Allows MLE or posterior analysis

For Strauss process

- Already had a density (but not in closed form)
- Added new type of Markov chain move
- Seems to speed up chain in practice

References

HUBER, M.L. (2011).
Spatial Birth-Death-Swap Chains.
Bernoulli (forthcoming paper, available online)

HUBER, M.L. AND R.L. WOLPERT (2009).
Likelihood based inference for Matérn type III repulsive point processes.
Advances in Applied Probability **41**, 958–977.

MATÉRN, B. (1986).
Spatial Variation, vol. 36 of *Lecture Notes in Statistics*.
New York, NY: Springer-Verlag, 2nd ed.
(first edition published 1960 by Statens Skogsforsningsinstitut, Stockholm).